Interactive Cybersecurity Defense Training Inspired by Web-based Learning Theory*

Dat Tang, Cuong Pham†, Ken-ichi Chinen, Razvan Beuran
Japan Advanced Institute of Science and Technology
Email: dat.tt@jaist.ac.jp

***Abstract*—As cybersecurity has become a huge problem of today's society, the demand for cybersecurity experts is increasing. Therefore, many researchers and organizations create training environments for professionals to have hands-on activities. However, in most cases, they are created manually and only focus on features, without a pedagogic point of view. This research proposes an interactive training interface aiming to bring a better interaction inspired by the web-based learning theory. Moreover, it runs on top of a system that can setup the training environment automatically and perform unmanned cyber attacks.**

In this paper, we first 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. After that, a comparison of the implemented interface with the theory of interaction in web-based learning is presented. Based on this theory, a survey is conducted on participants with various cybersecurity background levels to evaluate the effectiveness of the training, and what participants expect from hands-on cybersecurity defense training programs. The time for environment creation is also evaluated.

***Index Terms*—Cybersecurity training; Interaction; Web-based learning; Cyber range**

I. Introduction

With the raise of the digital age, our global society is going into the era of Internet of Things (IoT), Big Data, Cloud Computing and Machine Learning, which integrate deeply into the human life. Along with that, cybersecurity is becoming a real concern for people. Cyber attacks are now not done just for fun, to get fame, nor are they coming from an individual. They can be prepared by organizations or states, and anyone can be a victim. The most recent and famous attack is WannaCry, which infected more than 400.000 machines [1], and victims had to pay money to restore their data. Various others can be named, and they happen everyday, and are aimed at everyone.

In order to prevent or detect cyber attacks, people soon realized that the best way is practicing hands-on training, where trainees work in a testing environment that mimics real-life situations. This testing environment is called *cyber range*. In a cyber range, many vulnerabilities are reproduced, many attacks are performed, so 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 the attack.

Realizing the limitation of other training programs, Beuran et al. [2] proposed a cybersecurity training framework called CyTrONE (Cybersecurity Training and Operation Network Environment) [2]. This framework is powered by an open-source tool called CyRIS (Cyber Range Instantiation System) [3]. CyRIS provides a flexible and scalable mechanism to setup cyber ranges from text-based representations called cyber range descriptions.

The framework achieved good results in term of setting up the training environment [2]. However, both CyTrONE and many other cybersecurity training systems face common drawbacks: (1) Attacks performed for defense training purposes are still done manually by experts; and (2) Many features are added to systems without any pedagogic point of view - the interaction quality between learners and the training system is not evaluated. For these reasons, we designed and developed an improvement of the CyTrONE framework. This paper has two main contributions:

- Develop a framework to perform automatic cyber attacks for training purposes, so that this work is unmanned;
- Propose an interactive web-based interface to have a better communication between learners and the training system, whose development follows pedagogy theory.

The remainder of this paper is organized as follows. Section 2 is a review of background knowledge related to this research. Requirements for an advanced defense training system are defined in Section 3. Then, in Section 4, we describe how the system is implemented to satisfy these requirements. The evaluation comes in Section 5, where we compare the implemented interface with a theory about authentic learning, which is believed to lead to better training outcomes. We also explain the result of a user experience survey and an environment creation time evaluation. The paper ends with conclusions and references.

II. Related work

A. Automation in cybersecurity training

In the specific case of cybersecurity training, normal automation 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.

From surveying researches and projects which also reproduce cyberattacks, various ones were found. From the military

*This work was supported by JSPS KAKENHI Grant Number 17K00478.
†The author is now with Institut Mines-Télécom SudParis.

area, SIMTEX and SAST can be named. However, SAST only uses a machine with installed tools as an attacker for the whole system [10], while SIMTEX is customized for the computer network of the U.S. Air Force [14]. More importantly, they are military projects, so there is no way to clearly understand what they can do or contribute to improve them. Other researches from Michael E. Kuhl et al. [12] and Michael Liljenstam et al. [13] 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 for hands-on activities. Ariel et al. [11] 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.

B. Pedagogy background

1) Authentic activities in web-based courses: In education, there is one thing which is often discussed, namely the authenticity of study activities. Usually, in-school problem solving is formalized and standardized, having clear and enough information, with only one solving method and one answer, and is sometimes nonsense in real life. Authentic activities require situations in class to be the same as real-life environments. They should have a meaningful context, related components, be ill defined and require both investigation and problem-solving [8].

Reeves et al. [9] identified ten main characteristics of authentic activities. Based on these characteristics, our new system was evaluated if it fits with the theory, as shown in the Evaluation section.

2) Interaction in web-based learning: The research of M.G. Moore is the most popular and cited for the topic of interaction in distance learning. Using a communication-based framework, with a sender and a receiver [4], M.G. Moore [5] defines three types of interactions, which are learner-content, learner-instructor and learner-learner interaction. After that, emphasizing the effect of technology on the interaction, Hillman, Willis, and Gunawardena [6] added a fourth type, called learner-interface interaction. In 2001, Sutton [7] came up with the fifth type of interaction, vicarious interaction.

- **Learner-content:** This is the most fundamental interaction in an educational activity. Without it, there cannot be education [5], since this is the process of interacting between learner and learning materials to transfer knowledge. With the evolution of the content, the interaction between learner and content also changes. It is no longer a one-way communication. 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 type. In distance learning, it has different characteristics than in traditional learning. The main reason for these differences is the physical distance between learner-instructor, which limits the ability to communicate between two sides. Hence, first of all, instructors must maintain the student's interest in what is being taught [5]. Secondly, instructors need to check the learning progress along with the quality of the student output.
- **Learner-learner:** The third type of interaction is made by students in a class or group. Same as with learner-instructor activities, distance learning makes it more challenging, but it is even worse, since without effort no interaction is made between learner-learner, as learners usually do not know each other or meet each other in the learning process.
- **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 guidance on how to work with the interface, but also why they need to do it [6]. In the case of distance learning, the quality of interfaces affects the overall quality much more significantly. Therefore, in distance learning, especially online-learning and web-based learning, designing an interface which provides the interaction between learners and training environment, plus is easy to use and supports additional features is an important requirement. This is also a goal of our 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 learners and the instructor [7].

C. CyTrONE

CyTrONE is being developed by the Cyber Range Organization and Design (CROND) NEC-endowed chair at Japan Advanced Institute of Science and Technology for advancing cyber range creation technologies. It supports a complete training program with two sets of training questions for easy and medium levels, by using a technical guide from National Institute of Standards and Technology, U.S. as a reference [15]. The general workflow of CyTrONE is as follows: the training content is predefined in the Moodle web-based interface [16] - a Learning Management System (LMS). When an organizer of a training program gives a text-based cyber range description as an input to the framework, CyRIS is used to set up a cyber range for this session. Then trainees can access the cyber range, and work with it to answer the questions.

III. Requirements

From the pedagogy theory on interaction in web-based learning, surveying cybersecurity training systems and working for a while with CyTrONE, we realized that a modern cybersecurity training system needs to prepare training content automatically, perform automatic attacks, and bring a better interactive interface to learners. Thus, three new requirements are proposed:

- (R1) Automation in preparing the training session;

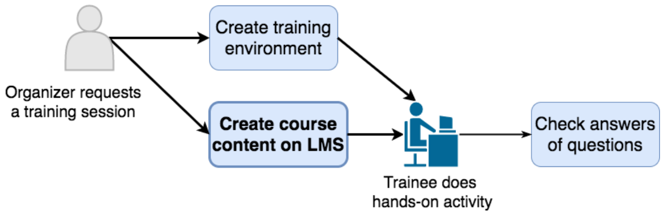


Fig. 1: Training model 1: An extension of the current model. Content on LMS is prepared automatically after a request from an organizer.

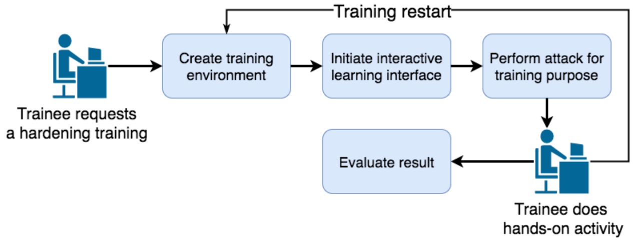


Fig. 2: Training model 2: Training with cyber vulnerability: an attack - hardening model.

- (R2) Automation in performing cyber attacks;
- (R3) Interaction between the system and trainees.

For implementing these requirements, our research also introduces two training models. The first one is an extension of the original model in CyTrONE, a quiz-based training, which is illustrated in Figure 1. In the original model, the content - which is a test including questions, requires manual creation in the Moodle interface. In the newly proposed model, training content is converted automatically to a SCORM [17] package, and then Moodle should be able to load it automatically without human interaction over the web-based interface.

The second model is a totally different one. In the current CyTrONE, there is only one kind of studying, via asking questions in a quiz style. This research proposes a new defense training model with cybersecurity incidents as shown in Figure 2. In this case, no learning content is required. An organizer is also not needed. A trainee can freely choose a vulnerability and start a training session; the system has an interactive interface, and cyber attacks are performed automatically. With this model, any trainee can learn independently, without any organization or event needed. Moreover, all of the three requirements above are satisfied by this model.

IV. Implementation

In Figure 3, a new system design for interactive and automatic training is described. This design has components that address the mentioned requirements. A training database is prepared to serve requests, while some functions are added to create an interactive and automatic training.

A. Automatic training session preparation

For preparing training sessions from content, a tool called `cnt2lms` was developed, which is made available on GitHub.

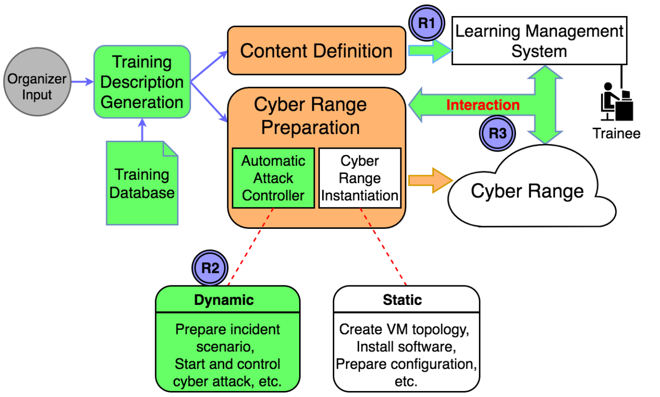


Fig. 3: New system design for interactive and automatic training.

An overview of `cnt2lms` is provided in Figure 4 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. The output SCORM package has an HTML interface, and allows trainees to interact with it in a graphical environment. It also provides necessary functions, such as connecting to the cyber range over a terminal, showing hints and submitting answers. `cnt2lms` allows organizers of training sessions to prepare the learning material on Moodle automatically and programmatically, thus saving time and effort.

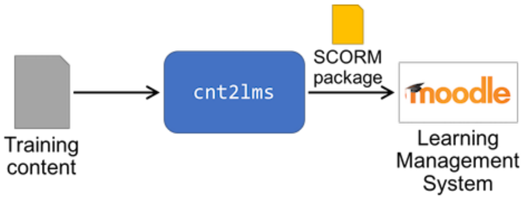
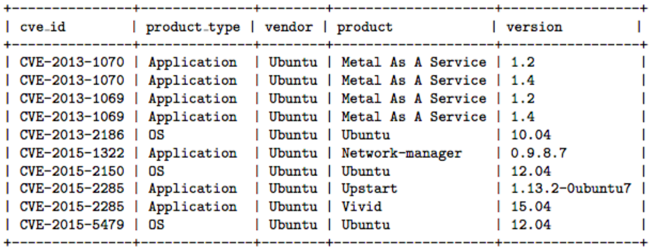


Fig. 4: Overview of cnt2lms [22].

B. Automatic cyber range preparation and attack

1) Training Database: Every cybersecurity training needs information to set up a hands-on environment and perform activities. Normally, this information is not stored and reused. It is usually based on knowledge of people in charge of setting up the machines. Our design uses CVEs [18] as input to define a training, and a database, which includes three data sets, to construct the corresponding training environment:

- *Vulnerability Database:* This is a list of every affected software version by each supported CVE. Using this database, the system can know automatically what software package and version needs to be installed in order to demonstrate an exploit. In this research, CVE Details [19] 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 of any vulnerability. A query from the database is shown in Figure 5.



```
+---------------+--------------+--------+--------------------+-----------------+
| cve_id        | product_type | vendor | product            | version         |
+---------------+--------------+--------+--------------------+-----------------+
| 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           |
| CVE-2015-1322 | Application  | Ubuntu | Network-manager    | 0.9.8.7         |
| CVE-2015-2150 | OS           | Ubuntu | Ubuntu             | 12.04           |
| 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           |
+---------------+--------------+--------+--------------------+-----------------+
```

Fig. 5: A query from vulnerability database.

- *Exploit Database:* While the *Vulnerability Database* stores a list of targets, the *Exploit Database* has a list of attack modules - which are attack techniques, hacking programs or proof-of-concept scripts. These two databases give the system a general scenario for any cyber attack: an attacker does something to a victim. At present, only Metasploit [20] module names are used to call the corresponding modules.
- *Instantiation Database:* This is a complementary database for the *Vulnerability Database*. In the initial design, there was no such database. However, during the development of the system, we realized that preparing a victim environment is not as easy as installing a software program by one simple command. Some limitations occur, namely lacking old software versions in public repositories, dependencies of programs and system-specific configuration. Because of these reasons, the *Instantiation Database* is designed to cope with this limitation. Following our plan, 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 works only with CyRIS, and each CVE id links to a script to instantiate a training environment. All files and scripts for setting up a virtual machine are packed into one executable script by the tool called `makeself` [21].

2) Start and Control Attack: To satisfy *(R2) Automation in performing cyber attacks*, the system should be able to perform cyber attacks without interaction from cybersecurity experts. At the current state, Metasploit [20] is used as the main attack method. Metasploit is the most popular framework for exploiting vulnerabilities. It is usually employed for penetration testing, hence it is also suitable for training purposes. From the point of view of this research, Metasploit has three main handy features that we leverage:

- It integrates many modules for cyber attacks, which can be used for reconnaissance, exploiting vulnerabilities or privilege escalation. The attack modules are updated frequently. Therefore, using Metasploit is a good solution for automatic cyber attacks.
- Since Metasploit is a framework, attack modules have almost the same format, which means they share same options and commands. Thus it is possible to generalize attack activities.
- Metasploit supports logging and script running, so that the execution can be done programmatically, and the results of the program can be used for further actions.

In our workflow, first of all, a script is generated as an input for Metasploit. This script configures the attack options for Metasploit, which are: name of vulnerability, IP address of the victim, and log file location. After CyRIS instantiates a cyber range, the cyber range description is used to set the target of the attack. The vulnerability is searched in the Exploit Database as a match between the CVE id and a Metasploit module. The log file holds the output of Metasploit after running, and can indicate if the attack was successful or not. Then, the output is parsed and analyzed to give the attack result to the trainee, to inform him/her whether the attack was successful or failed.

C. Interactive training over Web-based LMS

In order to bring a better experience to learners, this research developed a web-based interface for defense cybersecurity training. Moodle is an e-learning platform, so it has some basic interactive features, such as user login, result and learning progress management. On top of it, a graphical interface specified for cybersecurity training was built. Following the requirement R3, the layout of the web-based interface is designed as shown in Figure 6. The main components of the interface include:

- A list of vulnerabilities to choose from;
- Buttons to start and restart a learning session, and a button for requesting attacks against the cyber range.
- An open terminal button to connect to the cyber range (victim machine);
- A text field for a short description about the selected vulnerability;
- A text field for displaying a training tutorial.

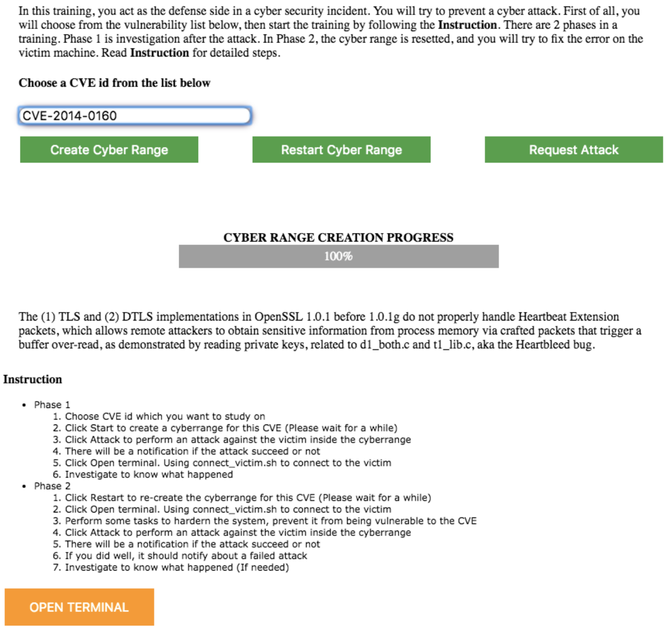
This design and its implementation follow the suggestion in [4] for a better interactive interface, such as important information on the top and in the center; buttons with a bright color to get attention; and orientation clues for every training session. Moreover, with this interface, trainees can connect to the cyber range from the web page; login and authentication processes are done automatically, thus it saves time for both trainees and organizers, while reducing the complexity of system utilization, since cyber range access credentials don't need to be distributed.

V. Evaluation

A. Authentic activity validation

Following the authentic activity theory from Reeves et al. [9], the system satisfies 9 out of 10 characteristics of an authentic learning activity. They are:

- Have real-world relevance;
- Be ill-defined, requiring students to define the tasks and sub-tasks needed to complete the activity;



In this training, you act as the defense side in a cyber security incident. You will try to prevent a cyber attack. First of all, you will choose from the vulnerability list below, then start the training by following the **Instruction**. There are 2 phases in a training. Phase 1 is investigation after the attack. In Phase 2, the cyber range is resetted, and you will try to fix the error on the victim machine. Read **Instruction** for detailed steps.

Choose a CVE id from the list below

CVE-2014-0160

CYBER RANGE CREATION PROGRESS

100%

The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

Instruction

- Phase 1
  1. Choose CVE id which you want to study on
  2. Click Start to create a cyberrange for this CVE (Please wait for a while)
  3. Click Attack to perform an attack against the victim inside the cyberrange
  4. There will be a notification if the attack succeed or not
  5. Click Open terminal. Using connect_victim.sh to connect to the victim
  6. Investigate to know what happened
- Phase 2
  1. Click Restart to re-create the cyberrange for this CVE (Please wait for a while)
  2. Click Open terminal. Using connect_victim.sh to connect to the victim
  3. Perform some tasks to hardern the system, prevent it from being vulnerable to the CVE
  4. Click Attack to perform an attack against the victim inside the cyberrange
  5. There will be a notification if the attack succeed or not
  6. If you did well, it should notify about a failed attack
  7. Investigate to know what happened (If needed)

Fig. 6: Layout of the interactive interface.

- Comprise complex tasks to be investigated by students over a sustained period of time;
- Provide the opportunity for students to examine the task from different perspectives, using a variety of resources;
- Provide the opportunity to reflect;
- Can be integrated and applied across different subject areas and lead beyond domain-specific outcomes;
- Seamlessly integrated with assessment;
- Create polished products valuable in their own right rather than as preparation for something else;
- Allow competing solutions and diversity of outcomes.

There is only one characteristic which the system does not satisfy, "*Provide the opportunity to collaborate*", since each cyber range is isolated for one learner, and the system does not currently support any special way of collaboration between students, except the forum feature of Moodle.

B. User experience

We conducted a user experience survey to check the improvement through the development of an interactive interface and automating cyber attacks for training purpose. We asked 15 people: 12 of them study or work in Information Science, with a person being a cybersecurity engineer; and 3 of them are working in non-IT related jobs. There are 7 questions, shown in Table I, with Q1 - Q4 about the quality of interaction, while Q5 - Q7 are about the result and effectiveness of the training. Participants give scores from 1 to 10, with 1 being worst and 10 being best. There is only one 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 7. From these results, some comments can be pointed out:

TABLE I: Questions for user experience survey

Topic	No	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

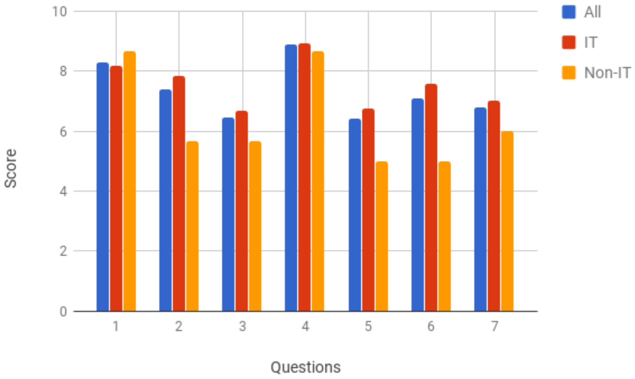


Fig. 7: Average score for each question in survey.

- All participants agreed that the interactive interface brings an easy way to access cyber ranges. Besides, the interface also provides enough and clear information and guidance, so trainees know what they need to do.
- There was 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 interfaces, they understand why the training happens in this way, and they believe that they can acquire knowledge after doing the training. On the other hand, non-IT people just follow the instructions without a clear understanding, and they are not sure that they can learn anything from it.
- Participants think that time spent for this training is a little long, since there is no hint to lead to the solution, and they have to study about the vulnerability by themselves.

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

TABLE II: System creation time: The overhead of the new system compared with the original one

Tasks	Average creation time	
	(s)	(%)
CyRIS with base VM	475.3	100 %
CyRIS + Prepare vulnerable machine (compile 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 %

C. Training Environment Creation

Since the vulnerable environment creation task is running on top of CyRIS, the total creation time is an important number when evaluating performance. As mentioned in the Implementation section, using the Instantiation Database helps to minimize the overhead of the new training model creation time compared to the original system. Therefore, the system creation time is compared with the original system and installing software packages by the traditional method of compiling source code, as shown in Table II. 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 possible.

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, and the environment can be prepared in advance. Note that preparing a vulnerable environment by the traditional compilation method takes almost 2.7x time, which makes it almost unusable, whereas, our system can be easily used in practice 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.

VI. Conclusion

In this paper, an improvement for cybersecurity hands-on training by applying web-based learning theory is presented. Inspired by this theory, three new requirements for a modern cybersecurity defense training are recommended regarding automatic setup training content and performing attacks, and having an interactive interface. We implemented the corresponding features by extending CyTrONE - an existing cybersecurity training framework. The new training models and web-based interactive interface satisfy all three requirements.

The implementation was evaluated by comparing with authentic learning theory, conducting a user experience survey and evaluating system performance. The new system validated positively as an authentic learning activity, while the survey proved that it creates a good interaction between the training environment and learners. The guidance is clear and easy to follow without an instructor, and people found it convenient and time-saving to connect to cyber ranges directly from the web interface. The system creation time was demonstrated to meet target times for training preparation with a small overhead on top of cyber range creation.

Our future work includes several tasks: (1) Continue to optimize environment creation time; (2) Improve the interactive interface by showing network topology, thus improving visibility; and (3) Support more vulnerabilities.

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] 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.

[3] C. Pham, D. Tang, K. Chinen, R. Beuran, *CyRIS: A Cyber Range Instantiation System for Facilitating Security Training, .* International Symposium on Information and Communication Technology (SoICT 2016).

[4] Y. Woo, T. 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.

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

[6] 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.*

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

[8] J. Herrington, T. C.Reeves, R. Oliver, Y. Woo, *Designing authentic activities in Web-based courses.* Journal of Computing in Higher Education, Fall 2004, Vol. 16(1), 3-29.

[9] T. C. Reeves, J. Herrington, & R. Oliver, *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.

[10] M. 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

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

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

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

[14] J. Davis, S. Magrath, *A Survey of Cyber Ranges and Testbeds, DSTO.* Cyber and Electronic Warfare Division, Defence Science and Technology Organisation, Department of Defence, Australia, 2013.

[15] K. Scarfone, M. Souppaya, A. Cody, A. Orebaugh, *Technical Guide to Information Security Testing and Assessment.* National Institute of Standards and Technology, 2008.

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

[17] *SCORM Explained.* Retrieved on Jul 12th, 2017 from https://scorm.com/scorm-explained/.

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

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

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

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

[22] *cnt2lms: Training Content to LMS Converter.* Retrieved on Jun. 26th, 2017 from https://github.com/crond-jaist/cnt2lms.