Integrated Framework for Hands-on Cybersecurity Training: CyTrONE

Razvan Beuran, Dat Tang, Cuong Pham, Ken-ichi Chinen, Yasuo Tan, Yoichi Shinoda

Japan Advanced Institute of Science and Technology

Abstract

Hands-on cybersecurity education and training activities are critical given that cyberattacks occur nowadays on an ever-increasing scale. Only such practical activities can ensure that trainees will acquire the actual skills necessary to promptly deal with security incidents in real-life situations. However, current programs rely significantly on the manual setup and configuration of the learning and/or training environments used, which is a tedious, inefficient and error-prone approach.

In this paper, we present an integrated cybersecurity training framework, named CyTrONE, that we designed and implemented to address such shortcomings by automating the training content generation and environment setup tasks. After we discuss the architecture and implementation of the framework, we demonstrate the framework effectiveness by evaluating it from both functionality and performance perspectives. Our results show that CyTrONE is well-suited for actual training activities in terms of features, usability and execution performance.

Keywords: cybersecurity education, hands-on training, cyber range, information security testing and assessment, learning management system

1. Introduction

In recent years, large-scale cyberattacks have occurred worldwide more and more frequently, and with ever greater consequences. A security breach at Yahoo in 2014, that was only disclosed in 2016, compromised the accounts of almost 1 billion users—the largest discovered breach in the history of Internet. In October 2016, a DDoS attack with traffic exceeding 1.2 Tbps was conducted on the Dyn DNS provider in the U.S. using the Mirai IoT botnet—the largest DDoS attack to date, that resulted in the inaccessibility of several high-profile websites. The WannaCry ransomware campaign in May 2017 infected over 400,000 computers in 150 countries—the largest ransomware attack to date.

Hands-on cybersecurity education and training are becoming more and more relevant in these circumstances, as the only manner in which such security incidents can be prevented and handled adequately (see [1] for a study on this topic). The governments of many countries understood the gravity of the situation, and actively support such activities, including through wide-scale training programs to which participation is free.

In Japan, for instance, the National Institute of Information and Communications Technology coordinates a program named CYDER (Cyber Defense Exercise with Recurrence) [2] that provides regular hands-on training to IT personnel of national and local government organizations and large companies. We also note the Hardening Project [3], a security contest organized by the Web Application Security Forum in Japan, in which teams of security experts and IT professionals compete with each other in terms of the service level they can provide for a virtual but realistic e-commerce company. Amongst the paid cybersecurity education and training programs available internationally, we mention the training provided by the SANS Institute, both as live courses and online training (NetWars) [4].

Most current cybersecurity education and training programs employ hands-on activities aimed at improving the functional skills and abilities of the participants. In many cases, the required practice environments are prepared via manual setup and configuration, an approach that is tedious, ineffective, and error-prone. This is the main reason that impedes the wider-scale generalization of such programs.

Although training programs may employ various tools behind the scenes to facilitate setup tasks, these tools are not disclosed, hence they do not benefit the public at large. For reference, our analysis of training programs in Japan [5] has shown that only one of the

surveyed programs, namely the Hardening Project, is consistently automating setup tasks. Nevertheless, the said automation refers only to the execution environment itself, and not to security content creation, which is still done manually by experts.

The Cyber Range Organization and Design (CROND) NEC-endowed chair was created at the Japan Advanced Institute of Science and Technology (JAIST) with the goal of supporting cybersecurity training activities. The focus of the JAIST effort is to make it possible to conduct wide-scale education and training programs that reach young people in universities, colleges and even high schools. This is in contrast to current training programs, which typically only target security professionals from various organizations, companies and in the military.

In this context, we have designed and implemented CyTrONE (Cybersecurity Training and Operation Network Environment), a cybersecurity training framework that facilitates training activities by providing an open-source set of tools that automates the training content generation and environment setup tasks. The advantages of this approach are threefold: (i) improve the reliability of the training setup; (ii) decrease the setup time and cost; (iii) make training possible repeatedly and for many participants. CyTrONE and its components, including sample training content, are freely available for download via our GitHub page [6].

The main contributions of the present paper are:

- Present the design and implementation of the CyTrONE cybersecurity training framework;
- Analyze the functionality of the CyTrONE framework to demonstrate its suitability for cybersecurity training;
- Evaluate the framework execution performance to prove its usability in practical training scenarios.

A preliminary version of this work was presented in [7]. The improvements in this paper compared to our previous publication are as follows: (i) provide more insight into the design process, including an overview of cybersecurity training characteristics and requirements (Section 2.1); (ii) discuss in more detail the system implementation, such as the extensions related to the user interface (Section 3.1); (iii) present more thoroughly the two subcomponents of CyTrONE, namely `cnt2lms` and CyRIS (Sections 3.4 and 3.5); (iv) introduce a novel use of the Moodle e-learning system for interactive training, achieved via the deployment of a dynamic cyberattack module (Section 3.6); (v) a more thorough evaluation of system capabilities and usability, including through user trials and an analysis of the content creation process (Sections 4.2 and 4.3).

The remainder of the paper is organized as follows. In Section 2 we discuss the overall framework design, and in Section 3 we provide more technical details about the actual framework implementation, and its components. This is followed in Sections 4 and 5 by functionality and performance evaluations of the framework that demonstrate it meets the design requirements. We continue with a related work section, and the paper ends with conclusions, acknowledgments, and references.

2. CyTrONE Design

In this section we present the design requirements that have driven the development of CyTrONE, and the actual design of the framework.

2.1. Requirements

The main goal of cybersecurity training is to improve the trainee readiness to deal with real-life incidents. We have identified three main categories of training activities. Only through the combination and repetition of such activities it becomes possible to improve the readiness of the training participants, as shown in Figure 1:

- *Attack-oriented training*, focusing on how to reproduce exploits on known vulnerabilities;
- *Forensic analysis-oriented training*, which provides a deeper understanding of the attack and of its consequences;
- *Defense-oriented training*, that builds skills related to vulnerability patching, and how to protect ICT systems in general.

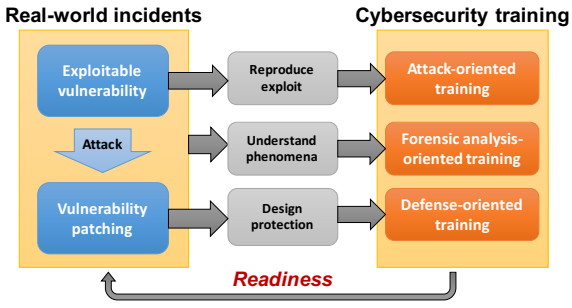

Figure 1: Relationship between real-world incidents and cybersecurity training activities.

To accomplish these tasks, cybersecurity education and training programs employ two key elements: (i) a method for conveying to trainees the information to be learned, and (ii) a mechanism for allowing them to practice the knowledge they acquire. Traditionally, information is provided via printed materials, but the digital era made it possible to use electronic formats, including interactive mechanisms such as e-learning, via Learning Management Systems (LMS). Hands-on practice is conducted in training environments created specifically for such purposes, also known as *cyber ranges*.

The issues of current cybersecurity education and training programs need to be considered in order to find ways to improve their effectiveness. A deep analysis that we conducted in [5] led to the establishment of a set of requirements for any effective security training program which refers to both aspects identified above, namely the training content, and hands-on activities. Therefore, in terms of practical implementation, those requirements can be mapped into two features that are necessary for creating an effective cybersecurity training framework:

1. Ability to modify and add new training content in an easy manner;
2. Ability to automatically create and manage the training environment.

2.2. Framework Design

Figure 2 outlines the overall framework design. CyTrONE uses input from a training coordinator to generate the training content for a particular training session, and uploads it to an e-learning system. CyTrONE also creates the cyber range training environment corresponding to that training content. This automatic generation is made possible through the use of an easily updatable training database, which contains all the necessary information, both regarding the training content that is shown to the trainees, and the properties of the associated training environment.

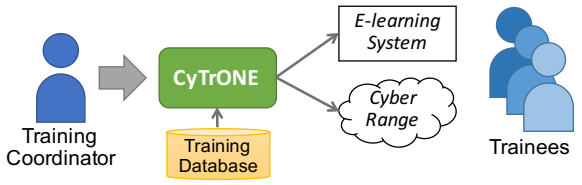

Figure 2: Overview of the cybersecurity training framework design.

Next we discuss the manner in which this design addresses the requirements discussed in Section 2.1.

Training Content Management. To satisfy the first requirement regarding the easy modification and addition of training content, we have decided to use the YAML text-based format [8] for representing the training scenarios (overview, questions, answers, hints, and so on). Thus, this information can be easily updated by the training organizers as desired, without the need for significant technical knowledge.

Within the CyTrONE workflow, the text-based training content description is converted to an appropriate format, and uploaded to the LMS without any user intervention. Moreover, we envisage that in the future questions could also be automatically generated based on meta-level descriptions, by using information from a richer training database (see Section 6).

Cyber Range Creation. Cyber ranges are often highly customized environments, and their setup requires advanced cybersecurity expertise, which leads to high costs for the training activities. The reuse of cyber ranges for subsequent training sessions is often considered as an acceptable solution, but this limits the quality of the training, since the environments cannot be updated if the need arises. Furthermore, this engenders the possibility of information leakage, hence it decreases the effectiveness of the cyber range for skill evaluation.

The cyber range instantiation functionality in our framework addresses this issue, and thus meets the second requirement for an effective cybersecurity training: the automatic creation and management of the practice environment.

2.3. Applications

The text-based training description in the approach outlined above is well suited for training related to technical skills and knowledge of individuals and/or teams, through questions that focus on distinct tasks and that require trainees to investigate and provide solutions for those tasks. The assumptions regarding the types of training that can be conducted via our framework are:

- Each training activity objective can be formulated as a problem, which may depend or not on other problems in the same lesson (e.g., determining the IP address of a server, or finding a flag in a protected location, such as the root directory, etc.);
- The system can decide automatically whether a participant was able to solve the problem by comparing the trainee answer to a known value (e.g., the IP address of the server in the cyber range, the flag stored at the protected location, etc.).

Although our assumptions may seem restrictive, they fully cover the quiz-based Jeopardy type of training often encountered in Capture The Flag (CTF) competitions, as well as any other type of training content that can be formulated within the limits of the restrictions above. Currently, team skills such as communication abilities, or complex skills such as those necessary in attack-defense training, cannot be evaluated via the e-learning part of our framework. Nevertheless, the cyber range instantiation component could be used to create the corresponding training environment even in those circumstances. Moreover, we plan to extend the system to team training via meta-level descriptions of the training content, and through an improved support for team skill evaluation in the training platform.

Given this context, we outline below some of the advantages and possible uses of our approach:

- Bridge the gap between descriptions of training content, such as the U.S. NIST Technical Guide to Information Security Testing and Assessment [9], and the environment in which the corresponding training activities are to occur;
- Provide flexibility in creating cyber ranges and updating their content based on information regarding recent security incidents, the skill level of the participants, and so on;
- Decrease the cost of setting up complex training environments, and thus improve the scalability of cybersecurity training, by allowing for a large number of training sessions and participants.

3. CyTrONE Implementation

The architecture of the CyTrONE cybersecurity training framework includes the following components, as shown in Figure 3:

- A user interface to allow training coordinators to select the properties of the training activity;
- A training database containing all the resources and data needed for content generation and environment creation;
- A management module for coordinating the entire framework;
- Additional modules for processing the training content description, and for performing cyber range instantiation;
- A learning management system (Moodle) for displaying content to trainees;
- A set of servers and network equipment infrastructure for the cyber range deployment.

3.1. User Interface

The user interface (UI) of CyTrONE makes it possible for the training coordinators to interact with the framework, mainly in order to decide the content of a particular training session. We currently have two types of UI, one intended for mobile devices, which uses a wizard design pattern, and a web interface which allows for richer content. The UIs use the standard HTTP protocol and JSON format [10] to communicate with the training management server, so alternative implementations are also possible.

Wizard UI. The wizard interface is implemented in the Swift programming language, and can run on iOS devices. The UI guides the organizer through selecting the type of training (“Scenario Based”, “Topic Based”), the class of training (e.g., “Security Testing and Assessment”, “Incident Detection and Response”), and finally the training difficulty level (e.g., “Easy”, “Moderate”, “Hard”). The last screen summarizes the training parameters; the organizer can create the actual training session, or go back and modify some of the choices.

Several screenshots of the UI are shown in Figure 4. From left to right we present: (i) the initial screen providing several choices to the user; (ii) the confirmation screen displaying a summary of the settings for the cyber range to be created; (iii) the cyber range creation notification shown when the setup procedure is completed.

The UI also has a settings screen for configuring aspects such as the hostname or IP address of the training server, the number of cyber range instances to be created, output file formats, etc.

Web UI. The second kind of UI is implemented as a web interface using JavaScript and PHP. This UI provides full freedom to the training coordinators, as they can upload content and cyber range description files directly to the training server. Naturally, the necessary resources for performing the content generation and environment creation need to be already in the training database.

The web UI also provides a visualizer for the training environment, which displays all the cyber range instances that are active at a certain moment, and their elements. The status of each cyber range component

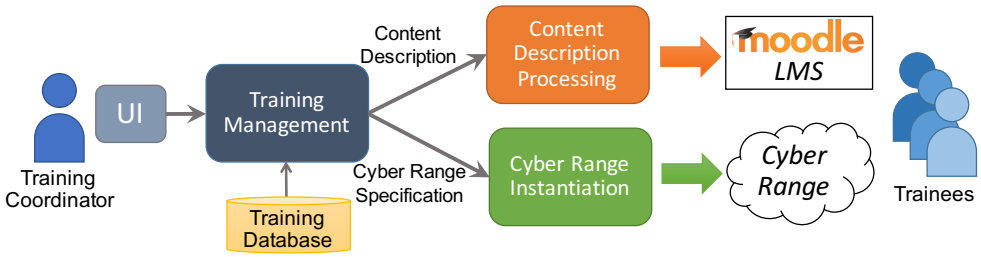

Figure 3: Architecture of the CyTrONE cybersecurity training framework.

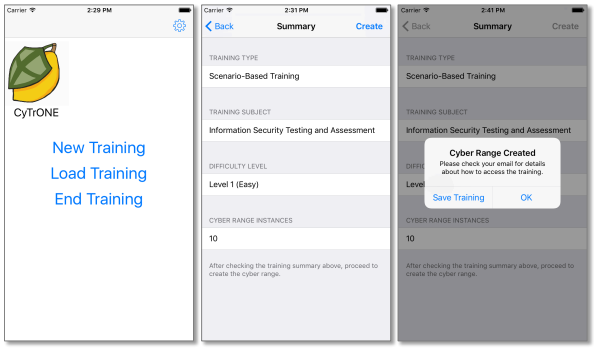

Figure 4: Screenshots of the wizard UI; from left to right: the initial screen, the confirmation screen, and the range creation notification.

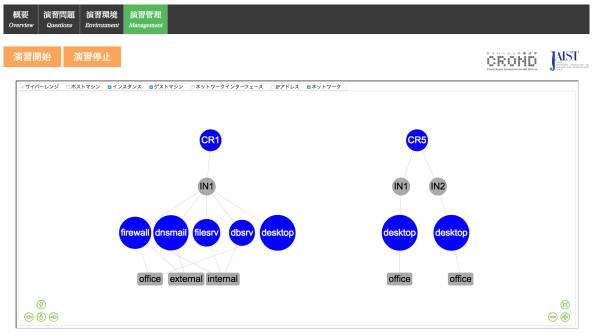

Figure 5: Screenshot of the web-based UI, namely the cyber range visualization tab.

is updated and displayed in real time—whether the virtual machine (VM) guests are running or not, whether the virtual network interfaces are active or not, etc. Detailed information about these elements can be obtained by hovering the mouse over them (e.g., the IP address of a network interface, and so on).

Figure 5 provides a screenshot of the cyber range visualization tab of the web UI for the case when two cyber ranges are present: (i) CR1 with one training environment instance containing 5 VMs (firewall, DNS and mail server, file server, DB server, desktop) and 3 subnetworks (office, external, internal); (ii) CR5 with two instances, each containing 1 VM (a desktop) and 1 network (office). Once the VMs in a cyber range become accessible, the color blue is used to illustrate their active status (inaccessible VMs are shown in red).

3.2. Training Database

The training database contains descriptions of the training content and the associated training environment, as well as all the resources necessary to create the training environment, such as custom files or programs, VM base images, etc.

As mentioned already, we use the text-based YAML format to describe the training content. Thus, the training content can be easily altered via a text editor should one wish to make any modifications or additions. Figure 6 contains a brief example that includes the overall description of a training level, and one question. A training activity has an id, a title, an overview, an optional level number, and includes a list of questions. For each question one must specify the id, the question body, the correct answer, and can also include a list of hints.

The associated cyber range is also described using YAML, in the form of a template. In Figure 7 we include an example cyber range specification template that includes:

- *Host settings* regarding the physical host(s) on which the cyber range is to be instantiated: host id and management and virtual bridge IP addresses, as well as the account used to access the host;
- *Guest settings* regarding the content that is to be prepared on the cyber range virtual machines: guest id and base VM information, and tasks to be executed to prepare the guest (adding accounts, installing packages, emulating attacks and malware);
- *Clone settings* concerning the replication of VMs to create multiple cyber range instances for trainees: a range id template, id(s) for the host(s)

```
---
- training:
  - id: L1-EN
    title: |
      Investigate the security of a desktop
      computer
    overview: |
      Today is your first day on the job as
      a sysadmin. Your boss tells you that he
      suspects somebody tried to hack into
      your company's network, and asks you to
      investigate a possible cyber attack
      that may have happened when the system
      administrator was a guy called Daniel
      Craig. The boss sits you in front of
      the previous sysadmin's computer, and
      wishes you good luck.

      You glance at the machine and
      reluctantly get to work.
    level: 1

    questions:
    - id: L1-EN-001
      body: |
        The operating system and kernel
        release number can tell you about
        the possible vulnerabilities of a
        computer. Find out the full kernel
        release number of the machine (e.g.,
        3.4.5-6.7.8.abc.x86_64).
      answer: 3.10.0-514.21.1.el7.x86_64
      hints:
      - You can use the command uname to
        find out OS details.
      - $ uname -r
      - An alternative solution is to check
        the /proc/version file.
```

Figure 6: Excerpt of a training content description in YAML format that contains the activity overview and one sample question.

on which the cyber range is to be created, types and number of guests, and network topology.

Note that the fields that depend on practical aspects such as training location, identity of the coordinator, and so on, make use of *variables* that will be replaced by the Training Management module based on user settings, in a manner similar to Ansible [11] variables. For instance, the value of the management IP address of the host on which the instantiation is to be done (variable `host_mgmt_addr` on line 4 in Figure 7) is only decided and allocated at cyber range creation time, using information on the hosts allocated to a particular coordinator.

```
---
- host_settings:
  - id: host_1
    mgmt_addr: {{ host_mgmt_addr }}
    virbr_addr: {{ host_virbr_addr }}
    account: {{ host_account }}

- guest_settings:
  - id: desktop
    basevm_host: host_1
    basevm_config_file: /imgs/basevm_desktop.xml
    basevm_type: kvm
    basevm_name: basevm_desktop
    tasks:
    - add_account:
      - account: daniel
        passwd: daniel_passwd
        full_name: Daniel Craig
    - install_package:
      - package_manager: yum
        name: wireshark
    - emulate_attack:
      - attack_type: ssh_attack
        target_account: daniel
        attempt_number: 123
        attack_time: 20170123
    - emulate_malware:
      - name: DAEMON
        cpu_utilization: 40
        mode: dummy_calculation

- clone_settings:
  - range_id: {{ clone_range_id }}
    hosts:
    - host_id: host_1
      instance_number: {{ clone_instance_number }}
      guests:
      - guest_id: desktop
        number 1
        entry_point: yes
      topology:
      - type: custom
        networks:
        - name: office
          members: desktop.eth0
```

Figure 7: Sample of a cyber range specification template in YAML format that specifies a basic training setup with one desktop.

The examples shown so far are part of sample content released with CyTrONE that follows the U.S. NIST technical guide mentioned previously [9]. We envisage that training organizers could easily add more content as they see suited, and that eventually such content will be released publicly for the benefit of the entire commu-

nity, or at least licensed to other training programs.

3.3. Training Management

The Training Management module is a key component of CyTrONE. It handles all the communication between the UIs and the framework, as well as the internal communication with other framework modules, including result and error reporting, etc. The implementation is done in Python and is multi-threading, thus allowing for multiple simultaneous users.

Training coordinator input is used by the Training Management module to select from the training database that was already described in Section 3.2 the relevant files for a given training activity. Then the module employs those files to generate a detailed specification of that activity, both from the point of view of training content and the corresponding training environment.

Regarding training content, a Content Description similar to what was shown in Figure 6 will be created and sent to the Content Description Processing module (see Section 3.4 below). Communication takes place using the HTTP protocol and JSON format.

As for cyber range instantiation, a template such as the one shown in Figure 7 will be combined with actual user settings (IP address of the hosts, etc.) to create an actual Cyber Range Specification that will be sent to the Cyber Range Instantiation module (see Section 3.5). In this case too communication takes place using the HTTP protocol and JSON format.

3.4. Content Description Processing

Our framework relies on the use of e-learning systems as a simple, flexible and scalable manner to provide an interface for the trainees to refer to the training questions, request hints, etc. The e-learning system also verifies trainees answers, computes results, manages score statistics, and so on.

The function of the Content Description Processing module, also implemented in Python, is mainly to convert the training content description that is generated by the Training Management module to a format that is suitable for e-learning systems. For this purpose we have selected the SCORM format, which is widely used for representing e-learning content [12].

The SCORM file resulting from the above conversion can be imported into most e-learning systems. We are currently using Moodle [13] as e-learning system, since it appears to be popular in many communities and well supported. For Moodle, our system also provides content upload capabilities, so that the whole training flow can be automated.

The entire functionality described above is provided via a tool named `cnt2lms`, which was also released on our GitHub page [6]. Figure 8 presents an overview of the content description processing mechanism by which the description file is provided to the `cntlms` tool, which will then upload the content to Moodle.

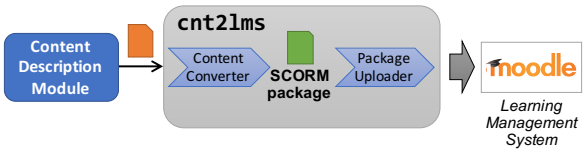

Figure 8: Overview of the content description processing mechanism.

There are two main modules in `cnt2lms`. First is the Content Converter module, which takes the YAML-based training content description file as input. The file is parsed, and the included information, such as the body and answer of questions, hints, etc. is used to generate a SCORM package. For this purpose a package template that includes content placeholders is used, and the actual question information is inserted into these placeholders.

The second module, named Package Uploader, subsequently uploads the generated SCORM package to the appropriate location on the Moodle LMS server. Thus, the new activity becomes available for trainees, who can then consult the questions and proceed with the training.

Although the conversion to a SCORM package is a generic function of our system, other aspects such as enabling automatic package upload, or allowing trainee control over the cyber range (see Section 3.6), required a deeper integration with Moodle. This lead to a certain loss of generality, which we nevertheless see as an optional added value that training organizers can choose to ignore if they prefer other LMSs.

3.5. Cyber Range Instantiation

A core component of the CyTrONE framework, named CyRIS (Cyber Range Instantiation System), has already been developed for automatically creating cyber ranges based on specific descriptions [14]. The Cyber Range Instantiation module in CyTrONE, which is implemented in Python as well, manages CyRIS in order to create and destroy cyber ranges on demand based on the range specification file generated by the Training Management module. The overview of this process (see Figure 9) indicates how the cyber range specification is provided to CyRIS, which will then instantiate the cyber range accordingly.

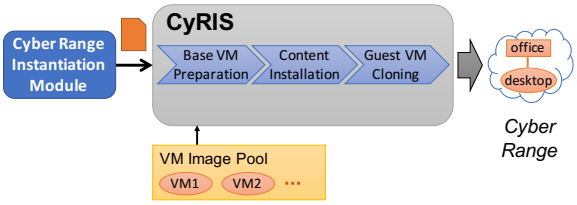

Figure 9: Overview of the cyber range instantiation process.

CyRIS consists of three main modules, as follows. Base VM Preparation module is in charge of preparing the VM images made available in the VM Image Pool for instantiation. For this purpose the disk images are copied to the working directory, and the corresponding VMs are started. Then, the basic setup of the VMs is conducted, such as configuring ssh access, hostname, network connectivity, etc.

The Content Installation module includes the core functionality of CyRIS, as it sets up all the training content in the VMs according to the cyber range specification. This task is composed of: (i) environment setup operations (managing accounts, installing software, copying required files, executing programs as needed, and configuring the network); (ii) security-related operations (configuring the firewall, emulating malware and cyberattacks, capturing traffic).

The Guest VM Cloning module contains the mechanisms related to creating clones of the prepared VMs in order to produce multiple instances of the cyber range, as needed for multi-user training. For this purpose, the configured base images are first copied to all the hosts on which the cyber range is to be instantiated. Once the cloned VM instances are started, the user accounts and passwords for accessing the cyber range are randomly generated. In addition, the network topology between cloned VM instances is configured according to the cyber range specification.

Note that the VM Image Pool mentioned above contains all the base VM disk images for the guests of the instantiated cyber ranges. These base VM images, which must be created in advance, use different operating systems, and contain various initial settings, as needed for the planned training activities. In addition, the Training Database discussed in Section 3.2 is used by CyRIS for all the custom software and scripts that may be necessary to instantiate a certain cyber range.

Our framework is fully compatible with CyRIS thus it can use all the functionality of the instantiation system, which includes:

- *Training environment setup*, with functions such as account management, tool installation, network configuration, etc.;
- *Security content generation*, with functions such as log generation, firewall configuration, malware and cyberattack emulation, and so on (cf. Table 1).

One important characteristic of the Cyber Range Instantiation module in the context of this paper is that the messages associated with all the setup steps involved in the cyber range instantiation are logged. At the end, the logs are automatically checked for errors, so that the correctness of the entire setup can be validated.

3.6. Learning Management System

As mentioned already, in CyTrONE we leverage the functionality of the Moodle LMS in order to present content to trainees and manage their training. The training paradigm discussed so far is shared with many other training programs: trainees are presented questions that they have to answer by carrying out an investigation in the cyber range. A screenshot of the Moodle interface for such quiz-based training is shown in Figure 10, and includes the sample content presented in Section 3.2.

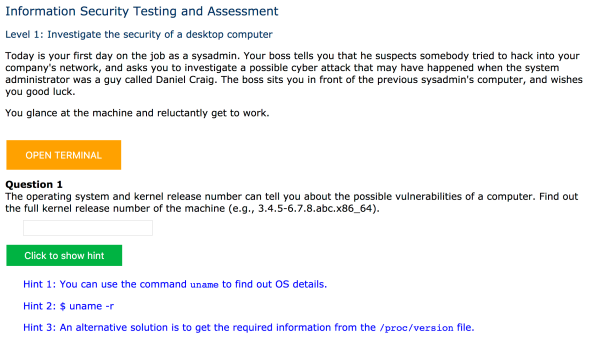

Figure 10: Screenshot of the Moodle interface for quiz-based training.

In additional to the classical quiz format, we also implemented an interactive interface in Moodle that makes possible for trainees to create cyber ranges and request attacks at their own convenience (see Figure 11). Thus, the following training model becomes possible:

1. A trainee creates a cyber range containing a vulnerability specified by a given CVE (Common Vulnerabilities and Exposures) id [15];
2. The trainee then requests an attack on the cyber range that will exploit the selected vulnerability;
3. Afterwards, the trainee performs a forensic analysis in the training environment to confirm the mechanisms and consequences of the attack;

4. Based on the analysis results, the trainee designs and implements a defense mechanism to avoid the attack (restarting first the cyber range if necessary);
5. Finally, the trainee requests again the same attack; if this fails, it means that the defense mechanism was effective; otherwise the analysis and defense cycle can be repeated as needed.

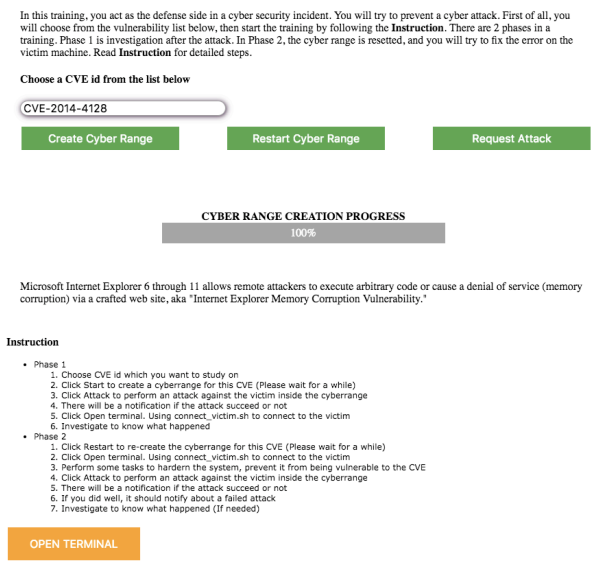

Figure 11: Screenshot of the Moodle interface for interactive training.

This training model makes it possible to address all categories of cybersecurity training shown in Figure 1, and—assuming that the necessary resources are present in the training database—this activity can be conducted by the trainees themselves in an independent manner, without requiring any external assistance (such as white-hat hackers conducting the attack).

This functionality was made possible through the implementation of a cyberattack controller that drives the Metasploit framework [16] to dynamically conduct the attack on request, and interprets its output in order to determine whether the attack was successful or not, and report this information to the trainees.

One important advantage of integrating a standard LMS such as Moodle in our workflow, as described in this section, is that it makes it possible to manage in one place the full education and training process of students (including delivering of instruction materials, tracking student progress, grading, and so on). Other solutions, such as typical CTF platforms, can only display a scoreboard for the current challenge, hence they do not provide an extensive educational support.

4. Functionality Evaluation

In what follows we shall discuss the evaluation of CyTrONE from the perspective of its features, and the types of cybersecurity training that can currently be conducted using CyTrONE, including from the point of view of some initial users.

4.1. Feature Analysis

The U.S. NIST Technical Guide to Information Security Testing and Assessment [9] that we used as reference for our implementation contains three classes of technical assessment techniques:

- *Review techniques:* Documentation review, log review, ruleset review, system configuration review, network sniffing, file integrity checking;
- *Target identification and analysis techniques:* Network discovery, network port and service identification, vulnerability scanning, wireless scanning;
- *Target vulnerability validation techniques:* Password cracking, penetration testing, social engineering.

In Table 1 we show how the cyber range creation features of CyTrONE can be combined in order to set up environments that cover *all* the techniques included in the NIST guideline. Some of them only require one or two features of CyTrONE, for instance for documentation review the file copy feature for copying the documentation to be reviewed in the cyber range. Others, such as vulnerability scanning, require the orchestration of many CyTrONE features in order to configure the training environment appropriately.

Based on this feature analysis we conclude that our framework provides functionality that is broad enough to support all the training topics related to security testing and assessment, at least as they were envisaged when the said guideline was released.

4.2. User Trials

In addition to the internal users of CyTrONE—about a dozen members of our group in JAIST who do research on related topics—we have also had the opportunity to conduct a user evaluation with external users.

In particular, four students of the Tokyo Metropolitan Technical College came to JAIST for an internship in March 2017. They have used CyTrONE in order to create security training environments inspired by AppGoat, a public web application security training tool created by the Information-technology Promotion Agency (IPA), Japan [13].

Table 1: Functionality coverage of the NIST Technical Guide to Information Security Testing and Assessment [9] through a combination of training environment setup and security content generation features.

Information Security Testing and Assessment Techniques	**Training Environment Setup**					**Security Content Generation**			
	Account Management	*Tool Installation*	*File Copy*	*Script Execution*	*Network Configuration*	*Firewall Configuration*	*Malware Emulation*	*Attack Emulation*	*Traffic Capture*
Documentation Review			○						
Log Review		○		○				○	
Ruleset Review		○		○		○			
System Configuration Review	○	○	○	○		○			
Network Sniffing		○	○		○			○	○
File Integrity Checking		○	○	○					
Network Discovery		○			○			○	
Port and Service Identification				○	○	○	○		
Vulnerability Scanning		○	○		○	○	○	○	○
Wireless Scanning		○	○					○	○
Password Cracking	○	○							
Penetration Testing	○	○	○	○	○	○	○		
Social Engineering	○		○						

Each of the four students attempted to build a cyber range containing a website with security vulnerabilities that can be accessed by trainees in order to validate their security skills, especially for penetration testing. Two of the students only had a webserver based on Apache `httpd` in their environments, but the other two students decided to also include a desktop guest, on which they could install several tools needed to conduct the penetration testing (e.g., Wireshark, John the Ripper, etc.).

All the four students succeeded in creating their target cyber ranges, and their feedback reassured us that our system is relatively easy to use. Students did encounter some difficulties in setting up the environment inside the cyber range, in particular the webserver, but these issues were mainly caused by OS, webserver software and browser version differences compared to those their were familiar with, or otherwise lack of a deeper technical knowledge regarding the tools their were using (e.g., how to configure `httpd` settings, etc.).

Overall, the students considered the internship very useful, as the use of CyTrONE freed them from the need to address low-level environment setup aspects, thus letting them focus more on training content creation. Moreover, one of them decided to continue using CyTrONE for his research activity, and even joined our development efforts in order to extend its functionality. In addition, some minor issues that were discovered on that occasion regarding multi-user support helped us improve CyTrONE before its public release in May 2017.

Additional user trials allowed us to determine that it takes several hours for someone with reasonable technical knowledge to set up CyTrONE and its components. This has shown that our modular approach, while flexible, resulted in a setup complexity that cannot be tackled without a certain technical background. To deal with this issue we are currently preparing an installation script that will automate most steps, thus reducing the setup time significantly. Nevertheless, alternative training approaches, such as using cloud services paired with CTF platforms, also have non-negligible setup times that cannot be easily reduced given the heterogeneous nature of such methods. Moreover, we consider that the initial setup overhead is less important than the time and effort savings during the regular system operation, and CyTrONE addresses this issue through its intrinsic characteristics regarding the unified management of both content and training environment.

We would also like to mention that CyTrONE was used in a public demonstration conducted at Interop Tokyo 2017, where it was nominated as Best of Show Award Finalist. The reception was hugely positive, with many visitors recognizing the importance of our endeavor; several of them also acknowledged the difficulties they had in setting up training events, and informed us they shall try our system in the future.

4.3. Content Creation

The public release of our framework on GitHub includes several sample questions, and we shall extend and update the set of samples as we continue the development. Nevertheless, training coordinators that decide to use our framework will face the issue of developing original content for their particular purposes. Given an existing set of problems, we distinguish two main classes of content creation:

1. Derived questions, with arbitrary changes to the problem text, but only minor changes to the cyber range content;
2. Original questions, with major changes both to the problem text, and to the cyber range content.

In the first case, modifications are easy to do without any advanced technical knowledge. Thus, we estimate that a couple of hours are enough for creating variations of existing problems. Given that educational content is often reused when teaching, once the overhead of creating the initial content is overcome, most of the CyTrONE use will fall into this category. This is particularly true if the built-in framework functionality meets all the problem requirements.

In the latter case, technical skills are of course required, mainly in order to be able to create original cyber range descriptions; hence, such modifications are more time consuming. Assuming that the training coordinators have the necessary knowledge about the tools they want to employ, our experience shows that at most one day should be enough create the content necessary for any given question. Such effort might need to be directed, for instance, towards preparing custom scripts in case the built-in range configuration mechanisms are not sufficient. Our system provides full flexibility for content creation in this sense, as the script execution mechanism, in principle, allows for arbitrary extensions.

5. Performance Evaluation

The second kind of evaluation we conducted refers to the performance of CyTrONE, in particular related to the cyber range instantiation process. We are currently in discussions with representatives of the union of technical colleges in Japan for integrating our framework with the cybersecurity program they will initiate nationwide. We expect that professors in such colleges will try to set up many parallel training sessions for hundreds of students, hence we decided to evaluate the framework performance in such conditions.

5.1. Experiment Methodology

We have conducted the performance evaluation experiments on the large-scale network testbed StarBED [17], and used a total of up to 30 physical hosts and 600 virtual machines. The physical hosts were Cisco UCS C200 M2 rack servers with two 4-core Intel Xeon E5504 2.0 GHz CPUs and 72 GB memory. In all our experiments, one of the hosts played the role of *master host*, which is in charge of performing certain initial steps (see below), and also manages the other hosts.

For the purposes of performance evaluation we divide the cyber range instantiation process into the following logical stages:

1. Preparation of the base images for VMs, currently conducted on the master host;
2. Content installation into the VMs prepared above, also conducted on the master host;
3. Cloning of the VMs on multiple hosts, which is mainly composed of copying the VM base images from the master host to the other hosts, and starting the VMs on each host from the copied base images.

For VM base image copying we use parallel copying to all the other hosts using the `parallel-scp` command; if a cyber range instance contains multiple base images, then they are also copied in parallel.

Given the above considerations, the total cyber range creation time, T_{total}, is computed as the sum of the execution times for the three logical stages, preparation, installation, and cloning:

$$T_{total} = T_{preparation} + T_{installation} + T_{cloning} \quad (1)$$

The execution times were measured for two test training scenarios that were created with reference to the NIST Technical Guide to Information Security Testing and Assessment, as follows:

- *Level 1:* A basic training which includes topics such as log and system configuration review, network sniffing, and vulnerability scanning; one VM, playing the role of a desktop PC, is needed for this training scenario;
- *Level 2:* A training of medium difficulty on topics such as network discovery, password cracking, and penetration testing; two VMs, a desktop and a webserver, are required for this training.

5.2. Preliminary Assessment

The first set of measurements were conducted with a fixed number of virtual machines, namely 20, that are instantiated on a total of 1, 2, 5, and 10 hosts (i.e., with 20, 10, 4, and 2 VMs per host, respectively), so as to study the effects of distributed execution on framework performance.

CyTrONE performance is determined by two main aspects, the duration of the content description processing task, and that of the cyber range instantiation operation (cf. Figure 3). Description processing is fundamentally a text-processing activity, therefore its duration is extremely short (e.g., under 0.5 s for all the experiments presented here). Consequently, in what follows we shall focus on the performance of the cyber range instantiation process, which is dominant.

Given that the preparation and content installation tasks are only conducted on the master host, the time it takes to perform these tasks depends mainly on the

number of VMs to be configured, and the kind of content that has to be installed, but does not depend on distributed execution aspects. The basic performance characteristics in this context for each test training scenario are summarized in Table 2; the results were obtained with 5 runs for each of the tests mentioned above.

Table 2: Basic performance characteristics for the test training scenarios in our experiments.

Scenario	**Preparation time [min]**		**Installation time [min]**	
	Average	*Std. dev.*	*Average*	*Std. dev.*
Level 1	1.12	0.03	2.58	0.03
Level 2	2.18	0.01	3.78	0.19

The results show that the preparation time doubles for Level 2 compared to Level 1, which is expected given that Level 1 uses only one VM, whereas two VMs are used in Level 2. The installation time also increases for Level 2 compared to Level 1, but it does not necessarily double, since this task's duration depends on the actual content to be generated and installed for each type of VM, which is not the same.

Next, in Figure 12, we present the average cloning time, as well as the total time required to instantiate cyber ranges in these experiments, which are the remaining performance characteristics that depend on the distributed execution of the framework.

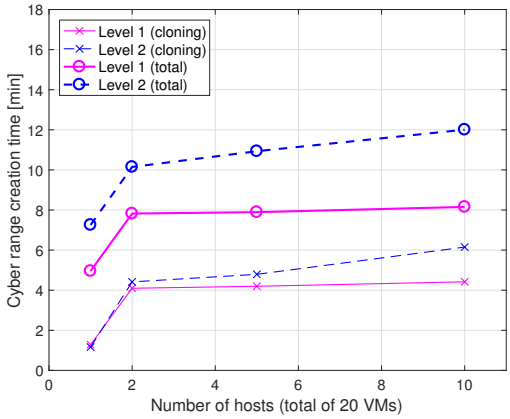

Figure 12: Cyber range creation time versus the number of hosts for a total fixed number of 20 VMs and up to 10 instantiation hosts.

Regarding the cloning stage, for the 1 host case (which is actually the master host), there is no need to copy the VM base images, hence time is only needed to start the VMs on the master host from these images. For the case of more hosts however, the copy time becomes important, even though the operation is done in parallel. Nevertheless, although for Level 2 it is necessary to copy two VMs to each host instead of one VM for Level 1, the measured time is not much different.

The total creation time results show that, following the initial increase because of the need to copy the prepared VMs to the other hosts that was explained above, thanks to our parallelization there is no significant increase in setup time afterwards. Thus, cyber range creation finishes in a reasonable time of 10 to 12 minutes.

Given that we only use standard virtualization technologies, we did not expect to achieve significantly different VM creation times compared to other virtualization approaches, such as via cloud services. As a next step, we decided to proceed with another series of experiments aimed at determining whether the overall performance characteristics of CyTrONE are within acceptable bounds even for larger scenarios.

5.3. Large-scale Experiments

For the second series of measurements, we assessed the framework performance in a large-scale scenario, with up to a total of 600 VMs on 30 hosts. We kept the number of VMs per host constant to 20; thus, this experiment set deployed up to 600 cyber range instances for Level 1, or 300 cyber range instances for Level 2, for an equal number of potential trainees. The results obtained for preparation and installation durations do not differ much from the case of the lower-scale experiments discussed above (see Table 2), hence we omit them.

The results in Figure 13 show the average cloning time and the total time required to instantiate all the cyber ranges. For these experiments, the cloning phase exhibits an exponential increase for the required time, with a higher exponent for Level 2, as this level requires copying a double number of VM base images. This effect is caused by the fact that congestion eventually occurs as the total amount of traffic sent over the interconnecting network increases, hence the transfer times increase as well.

Nevertheless, the total creation time results show that, in a large-scale setup for 100 trainees, creation can be finished in under 10 minutes for Level 1 (100 VMs), and in under 15 minutes for Level 2 (200 VMs), durations that we consider reasonable given the typical 10 minute breaks between classes. In the extreme case of using 600 VMs, for Level 1 (600 trainees) the setup is completed in under 15 minutes, and even for Level 2 (300 trainees) the setup is completed in about 22 minutes.

We believe that through further optimization we can reduce even more the total cyber range creation time. We are investigating the possibility of tackling the long cloning time issues by allowing each host to set up its

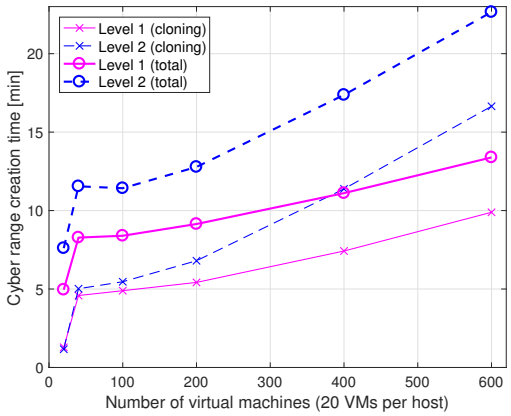

Figure 13: Cyber range creation time versus the number of virtual machines when using a fixed number of 20 VMs per host and up to 30 instantiation hosts.

own VMs (after an initial copy of raw VM images, which only needs to be done once before the very first training). This approach will not require copying the VMs from the master host during instantiation, as it is done now, and will eliminate the exponential increase seen in Figure 13; thus, we expect that the total creation time results will become relatively flat for any number of hosts. The only inconvenient would be that all the hosts need to be provided access to the repositories containing the required packages and tools to be installed, which may pose a security risk in some circumstances.

For a more detailed analysis of cyber range instantiation performance at low scale, and the differences with respect to other environment setup tools see [14].

6. Related Work

A first category of systems related to our framework are cloud controllers, such as OpenStack [18], or management tools, such as Ansible [11] and Chef [19]. Although CyTrONE shares some features with them, as an integrated framework, our system covers all the necessary functionality for cybersecurity training. This includes user interfaces for both organizers and trainees, training content generation and security content creation, as well as training environment setup and cleanup.

Realistic cybersecurity training using cyber ranges is currently mainly conducted in military environments, and the proprietary systems that are available publicly are expensive and have a low configurability level. To the best of our knowledge CyTrONE is the first open-source cybersecurity training framework that is fully configurable and flexible.

Facebook has recently released an open-source CTF platform, supporting quiz, flag and king-of-the-hill types of CTF training [20]. However, the Facebook CTF platform is mainly a cool UI for the training, and does not provide any assistance with a full environment setup in the manner of CyTrONE. Moreover, there is no support for generating security content either. These tasks remain the organizer responsibility, and consequently are still tedious. For instance, if one wishes to pair such a CTF with a training environment created via a cloud service, any required synchronization between the training content and the environment setup tasks, for instance in case of changes in content, needs to be handled independently/manually, on a case-by-case basis.

In [21], the use of application containers is proposed as a solution for improving the scalability of CTF competitions. The approach focuses only on deployment though, as content creation and management still have to be handled manually. CyTrONE has a much more thorough and general approach, and we are also currently implementing the use of container technology instead of virtual machines in order to improve scalability.

Proprietary systems, such as the Boeing Cyber-Range-in-a-Box (CRIAB) [22], create a vendor lock-in, both in terms of software and hardware. On the other hand, our open-source framework makes it possible to decouple the training content from the execution infrastructure, enabling the content update and infrastructure expansion as needed. The open-source approach also brings about perspectives for standardization of the training content format. This would create opportunities for training companies to easily produce content adapted to various levels of trainee skills, age, background, and so on, and license it without having to worry about the details of the platform on which the content is actually used.

As mentioned in Section 3.6, CyTrONE can use CVE [15] information to create an appropriate training environment and conduct on-demand an attack that exploits that vulnerability. We envisage that it will be possible to use even higher-level information to conduct more realistic training activities. In this context, the framework of the ITU-T X.1500 recommendation for structured cybersecurity information exchange techniques (CYBEX) [23] is extremely relevant. Additionally, detailed information about the incidents can be represented in standard machine-readable formats, such as Structured Threat Information eXpression (STIX) [24] or Incident Object Description Exchange Format (IODEF) [25].

If such information would be used to reproduce a

given incident, it would become possible to conduct training in similar conditions to that incident as soon as the corresponding information is made available. This would make it possible for IT professionals to immediately gain first-hand knowledge and develop response tactics, so that the said incident is avoided elsewhere.

7. Conclusions

We have presented the design and implementation of an integrated cybersecurity training framework named CyTrONE. Through the development of this framework we aim to increase the effectiveness of cybersecurity training by improving the accuracy of the training environment setup, and decreasing the setup time and costs, thus making large-scale security training possible for practically anyone, anywhere and at anytime. The flexibility of the framework, in association with the use of a Learning Management System, means that not only classical CTFs, but any other kind of training can be conducted, by leveraging the advances of modern education methodologies. An example in this sense is the innovative integration between the LMS and the cyber range that we have started; this allowed us already to conduct interactive training, as discussed in Section 3.6, and could also be used for other educational paradigms in the future, such as adaptive learning, etc.

The main advantages of our framework compared to alternative approaches are: (i) lowers the entry barrier to cybersecurity training by providing easy to manage descriptions that hide most of the technical complexity of the process; (ii) unifies the management of the training content and environment, thus reducing the overall deployment complexity, and saving operation time and effort; (iii) integrates the training activity with a standard LMS in order to provide support for the full educational process associated with cybersecurity training. Moreover, given its public release as open-source software, our framework makes possible further customization and enhancement by other parties in order to meet requirements that we have not yet considered.

CyTrONE was evaluated in terms of its functionality, and we have shown that it covers all the security testing and assessment techniques discussed in the relevant U.S. NIST guideline. As the framework was made public in May 2017, we hope that multiple parties will embrace it, and create training content for CyTrONE, thus leading to the emergence of a worldwide ecosystem for cybersecurity training.

We have also evaluated the framework performance regarding cyber range instantiation, and demonstrated that it meets reasonable target times for cyber range creation: within 10 minutes for 100 participants for a basic setup with one VM per trainee, and within 15 minutes for 100 participants when using a more advanced training setup with two VMs per trainee. Even for a total of 300 participants, the setup time is under 10 minutes for the basic level, and only a little above 20 minutes for the advanced level.

Our future work includes several improvements of the current system, both in terms of features and performance, for instance through the use of machine-readable incident reports for automatically generating the corresponding training environments. In addition, we shall proceed with the integration of the framework into the workflow of existing training programs, such as CYDER and Hardening Project that were mentioned in beginning of the paper, through our already established contacts with those program organizers.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 17K00478. The authors would like to thank Masanori Sunagawa for developing the cyber range visualizer mentioned in this paper.

References

[1] M. Bartnes, N. B. Moe, P. E. Heegaard, The future of information security incident management training: A case study of electrical power companies, Computers & Security 61 (2016) 32–45.

[2] National Institute of Information and Communications Technology, Japan, Cyber Defense Exercise with Recurrence (CYDER) (in Japanese), 2017. https://cyder.nict.go.jp/.

[3] Web Application Security Forum, Hardening Project (in Japanese), 2017. http://wasforum.jp/hardening-project/.

[4] SANS Institute, SANS NetWars Training Courses, 2017. https://www.sans.org/netwars/.

[5] R. Beuran, K. Chinen, Y. Tan, Y. Shinoda, Towards Effective Cybersecurity Education and Training, Technical Report IS-RR-2016-003, Japan Advanced Institute of Science and Technology (JAIST), 2016.

[6] Cyber Range Organization and Design (CROND), GitHub Repositories, 2017. https://github.com/crond-jaist.

[7] R. Beuran, C. Pham, D. Tang, K. Chinen, Y. Tan, Y. Shinoda, CyTrONE: An Integrated Cybersecurity Training Framework, in: Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP 2017), pp. 157–166.

[8] C. Evans, The Official YAML Website, 2017. http://www.yaml.org/.

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

[10] D. Crockford, The application/json Media Type for JavaScript Object Notation (JSON), IETF RFC 4627, 2006.

[11] Red Hat, Inc., Ansible is Simple IT Automation, 2017. https://www.ansible.com/.

[12] Advanced Distributed Learning Initiative, SCORM Official Website, 2017. http://www.adlnet.gov/adl-research/scorm/.
[13] The Moodle Project, Moodle – Open-source Learning Platform, 2017. https://moodle.org/.
[14] C. Pham, D. Tang, K. Chinen, R. Beuran, CyRIS: A Cyber Range Instantiation System for Facilitating Security Training, in: Proceedings of the International Symposium on Information and Communication Technology (SoICT).
[15] MITRE Corporation, Common Vulnerabilities and Exposures (CVE), 2017. https://cve.mitre.org/.
[16] Rapid7 LLC, Metasploit: Penetration Testing Software, 2017. https://www.metasploit.com/.
[17] National Institute of Information and Communications Technology, Japan, Hokuriku StarBED Technology Center, 2017. http://starbed.nict.go.jp/en/index.html.
[18] The OpenStack Foundation, OpenStack – Open Source Cloud Computing Software, 2017. https://www.openstack.org/.
[19] Chef, Chef – Automate Your Infrastructure, 2017. https://www.chef.io/chef/.
[20] Facebook, Inc., Platform to host Capture the Flag competitions, 2017. https://github.com/facebook/fbctf/.
[21] A. S. Raj, B. Alangot, S. Prabhu, K. Achuthan, Scalable and Lightweight CTF Infrastructures Using Application Containers, in: Proceedings of the 2016 USENIX Workshop on Advances in Security Education (ASE '16).
[22] Boeing Inc., Cybersecurity & Information Management, 2017. http://www.boeing.com/defense/cybersecurity-information-management/.
[23] ITU-T, Revised structured cybersecurity information exchange technique, Recommendation X.1500 (2011) Amendment 9 (03/16), 2016.
[24] OASIS Cyber Threat Intelligence (CTI) Technical Committee, Structured Threat Information eXpression (STIX), 2017. https://stixproject.github.io/.
[25] R. Danyliw, J. Meijer, Y. Demchenko, Incident Object Description Exchange Format (IODEF), IETF RFC 5070, 2007.