CyTrONE: An Integrated Cybersecurity Training Framework

Razvan Beuran, Cuong Pham, Dat Tang, Ken-ichi Chinen, Yasuo Tan and Yoichi Shinoda
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

Keywords: Cybersecurity Training, Cybersecurity Education, Cyber Range, Information Security Testing and Assessment.

Abstract: In a world in which cyber-attacks occur on a daily basis, cybersecurity education and training are indispensable. Current training programs rely on manual setup and configuration for hands-on activities, which is a tedious and error-prone task. In this paper we present CyTrONE, an integrated cybersecurity training framework that we designed and implemented to address such shortcomings. The key insight is automating the training content generation and environment setup tasks. The advantages of this approach are: (i) improve the accuracy of the training setup; (ii) decrease the setup time and cost; (iii) make training possible repeatedly, and for a large number of participants. In the paper we thoroughly discuss the architecture and implementation of the framework, and we evaluate it from several perspectives in order to demonstrate that CyTrONE meets the aforementioned objectives.

1 INTRODUCTION

Cyber-attacks occur worldwide on a daily basis. The cyber breach at the Target stores in the U.S. in 2013 compromised the credit and debit card information of 40 million shoppers. In 2015 the Japan Pension Service was hacked, which led to the exposure of personal data of 1.25 million of its users. A recently disclosed breach at Yahoo, that actually occurred in 2014, compromised the accounts of more than 500 million users—and it is considered to be the largest discovered breach in the history of Internet so far.

In this context cybersecurity education and training are becoming more and more relevant, as the only way in which such cyber breaches can be prevented and handled adequately. There is a plethora of available training programs, most of them paid, but also several to which participation is free. In Japan, for instance, CYDER (MIC, 2016) is a program coordinated by the Ministry of Internal Affairs and Communications for providing hands-on training to IT personnel of government organizations and large companies. Hardening Project (WASForum, 2016) is 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 security improvements they can provide in a realistic e-commerce company network. Also in Japan, enPiT-Security (SecCap) (enPiT University Consortium, 2016) is an education program targeting students that is supported by the Ministry of Education, Culture, Sports, Science and Technology, and organized by a consortium of five universities. 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) (SANS Institute, 2016).

Such education and training programs make use of specifically set-up training environments for hands-on activities meant to improve the practical skills of the participants. However, many of the current training programs rely on the manual setup and configuration of these environments, which is a tedious and error-prone task. Although some training programs may employ various tools behind the scenes to facilitate setup tasks, such tools are not disclosed, therefore they do not benefit the public at large. Thus, our analysis of training programs in Japan (Beuran et al., 2016) has shown that only one of the surveyed programs, namely Hardening Project, is consistently automating setup tasks, even though the said automation refers only to the execution environment itself, and not to security content creation, which is still done manually.

We believe in the *democratization* of cybersecurity training. We believe that, while the improvement of the skills of current security professionals in various organizations, companies and in the military is of course important, the only solution for making it possible to cope with the ever increasing cybersecurity threats is to have large-scale education and training programs that reach young people in universities, colleges and even high schools.

Beuran, R., Pham, C., Tang, D., Chinen, K-i., Tan, Y. and Shinoda, Y.
CyTrONE: An Integrated Cybersecurity Training Framework.
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 157-166
ISBN: 978-989-758-209-7

For this reason we have proceeded to design and implement CyTrONE (Cybersecurity Training and Operation Network Environment), a cybersecurity training framework that aims to facilitate training activities by providing an open-source framework that automates the training content generation and environment setup tasks. The advantages of this approach are threefold: (i) improve the accuracy of the training setup; (ii) decrease the setup time and cost; (iii) make training possible repeatedly, for many participants.

We stress that CyTrONE is not a cloud controller, such as OpenStack (The OpenStack Foundation, 2016), nor simply a management tool, such as Ansible (Red Hat, Inc., 2016) or Chef (Chef, 2016), although it shares some features with such systems. CyTrONE is indeed an *integrated framework* for cybersecurity training that covers all the necessary functionality, from user interfaces for both organizers and trainees, continuing with training content generation and security content creation, and finishing with training environment setup and cleanup.

The main contributions of the present paper are:

- Present the design and implementation of the CyTrONE cybersecurity training framework;
- Evaluate the CyTrONE framework from several perspectives to demonstrate that it meets the aforementioned goals;
- Discuss future directions for improving the present framework in particular, and cybersecurity training in general.

The remainder of the paper is organized as follows. In Section 2 we discuss in more detail the principles that guided the framework design, and how we addressed them in practice. Then, in Section 3, we provide more technical details about the actual framework implementation, and about its components. This is followed (Section 4) by a multi-perspective evaluation of the framework which demonstrates that it meets the design requirements. Finally, in Section 5, we discuss our vision for further improving cybersecurity education and training. The paper ends with conclusions and references.

2 CyTrONE OVERVIEW

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

2.1 Background

The training environments used in cybersecurity training are typically known as *cyber ranges*. The underlying concept of this kind of practical training is illustrated in Figure 1. Thus, in addition to theoretical lectures, the trainees have access to a network environment in which they can practice and improve their skills. Such an environment could contain traces of cyber-attacks, for use in forensics training, but could also be subjected to live attacks by white-hat hackers, for use in defense and even attack training (see (Beuran et al., 2016) for a more detailed description of security training methodologies).

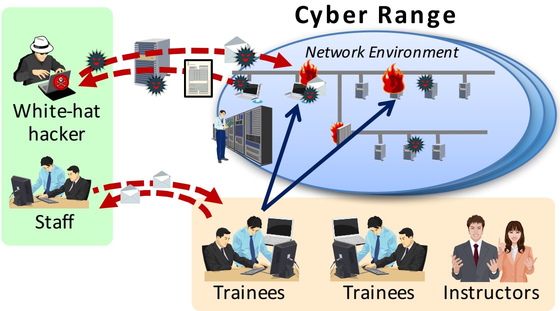


Figure 1: The concept of cybersecurity training conducted using cyber range environments.

For a realistic training, trainees could also communicate with staff members, similarly to what would be required in the case of a real cyber-attack, for instance in order to report the issue to management, to request assistance from a cybersecurity company, etc.

2.2 Design Requirements

The issues of current cybersecurity education and training programs need to be considered for improving their effectiveness. A deep analysis conducted in (Beuran et al., 2016) led to the establishment of a set of requirements for any effective security training program:

- The training content should be appropriate for the target audience in terms of knowledge and ability levels;
- The training content should be in accordance with the skills that the program aims to develop;
- The training program should use hands-on activities for developing practical abilities, so as to ensure that trainees can subsequently deal with real-life incidents;

- The training program should reach a large audience, in order to have a significant impact on the cybersecurity readiness of a country;
- The training program should have good cost/performance characteristics, so that it is sustainable on long term.

These requirements can be reflected in practice by noting that they refer to two key aspects: (i) training content, and (ii) hands-on activities. Therefore, in terms of practical implementation, the above 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.

2.3 Framework Design

We designed the cybersecurity training framework CyTrONE specifically to meet the aforementioned requirements; an overview of the framework design is shown in Figure 2.

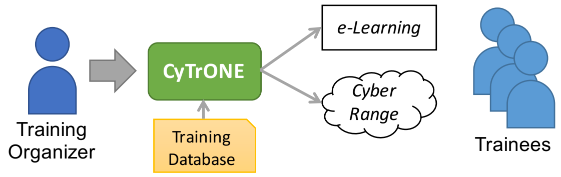


Figure 2: Design overview of the CyTrONE cybersecurity training framework.

Specifically, based on training organizer input, CyTrONE will generate the training content for a particular training session, and upload it to an e-Learning system (also known as Learning Management System, or LMS). Simultaneously, CyTrONE will create 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. This design addresses the requirements in Section 2.2 as detailed next.

2.3.1 Training Content Management

To deal with the first implementation requirement regarding the easy modification and addition of training content, we decided to use the YAML text-based format (Evans, 2016) for representing the training scenarios (descriptions, questions, and answers). Thus, this information can be easily updated by the training organizers, without a need for significant technical knowledge. Moreover, we envisage that in the future questions could also be partially generated based on meta-level descriptions, such as training topics, by using information from a richer training database (see also Section 5.2). In all cases, the corresponding training content as shown to trainees is produced automatically, without any direct intervention.

2.3.2 Cyber Range Creation

Current cyber ranges are highly customized, and their setup requires a high level of cybersecurity expertise. This makes it prohibitive to setup complex environments, which leads to high training costs. The reuse of cyber range environments 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. Moreover, it engenders the possibility of information leakage, hence it decreases the effectiveness of the cyber range as a skill-evaluation tool. The automatic cyber range instantiation functionality in our framework addresses this issue, and thus meets the second implementation requirement for effective cybersecurity training: the automatic creation and management of the practice environment.

2.4 Advantages

We outline below some of the advantages and possible uses of our integrated training framework:

- Bridge the gap between descriptions of training content, such as the U.S. NIST Technical Guide to Information Security Testing and Assessment (Scarfone et al., 2008), and the environment in which the corresponding training activities should occur;
- Provide flexibility in creating cyber ranges and updating their content based on information regarding recent security incidents, the skill level of the participants, etc. Consequently, improve the effectiveness of the training through ensuring a higher variability of the scenarios;
- 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 detailed architecture of the CyTrONE cybersecurity training framework is shown in Figure 3. Next we shall provide more details about each component.

3.1 User Interface

The user interface (UI) in Figure 3 is intended for the training organizers. Assuming the training database is already set up as needed, the UI should make it possible for organizers to decide the content of a particular training session as easily as possible. The current implementation of the UI uses the Swift programming language, and thus the UI can run on any iOS device. Nevertheless, as the UI uses the standard HTTP protocol and JSON format (Crockford, 2006) to communicate with the training server, other implementations are possible in the future (e.g., web application, Java, etc.).

The UI employs a *wizard-like paradigm*, and it 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 actions 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.

3.2 Training Database

Currently we employ the same training paradigm with many other training programs, in that trainees are presented questions that they have to answer by carrying out an investigation in the cyber range. For our future plans regarding a more realistic training approach see Section 5.2.

We have already prepared some sample content that follows the U.S. NIST technical guide mentioned previously (Scarfone et al., 2008), and we envisage that training organizers could easily add more content as they see suited. We hope that eventually such content will be released publicly for the benefit of the entire community, or at least licensed to other training programs.

As mentioned already, the training content in our framework is described using the YAML format, therefore it is easily editable for modifications or additions. Figure 5 contains a brief example that includes the overall description of a training level, and one training question.

The associated cyber range is also described in YAML, in the form of a template. In Figure 6 we include an example cyber range specification template that includes: (i) *host settings* regarding the physical host(s) on which the cyber range is to be instantiated; (ii) *guest settings* regarding the content that is to be prepared on the cyber range virtual machines (VMs); (iii) *clone settings* concerning the replication of VMs to create multiple cyber range instances for trainees.

Note that the fields that depend on practical aspects such as training location, identity of the organizer, and so on, make use of *variables* that will be replaced by the Training Description Generation module based on user settings, in a manner similar to Ansible (Red Hat, Inc., 2016) 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 6) is only decided and allocated at cyber range creation time, based on the information regarding the hosts allocated to a particular organizer.

3.3 Training Description Generation

The Training Description Generation is a key component of CyTrONE. Its function is to generate a detailed description of the training that is to be conducted. For this purpose the organizer input is used to select the appropriate sources from the training database, which was described in Section 3.2 above.

The training description has two components. Regarding training content, a Content Description similar to what was shown in Figure 5 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 similar to the one shown in Figure 6 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.

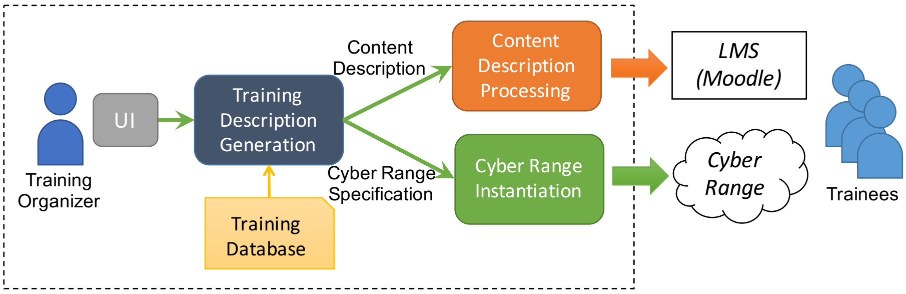
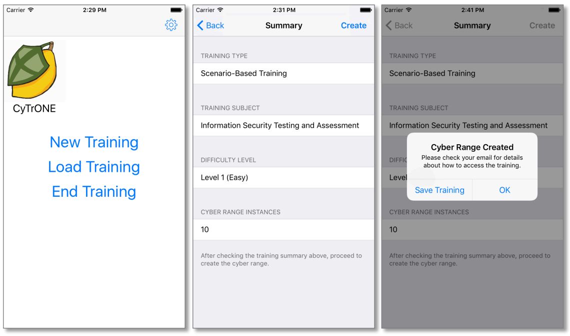

Training Organizer
UI
Training Description Generation
Training Database
Content Description
Cyber Range Specification
Content Description Processing
Cyber Range Instantiation
LMS (Moodle)
Cyber Range
Trainees

Figure 3: Architecture of the CyTrONE cybersecurity training framework.



TRAINING TYPE
Scenario-Based Training

TRAINING SUBJECT
Information Security Testing and Assessment

DIFFICULTY LEVEL
Level 1 (Easy)

CYBER RANGE INSTANCES
10

After checking the training summary above, proceed to create the cyber range.

Cyber Range Created
Please check your email for details about how to access the training.

Figure 4: Screenshots of the wizard-like UI aimed at training organizers; from left to right: the initial screen, the confirmation screen, and the range creation notification.

3.4 Content Description Processing

Our framework proposes the use of e-learning systems as a simple, flexible and scalable manner to provide a user interface for the trainees, so that they can refer to the training questions, request hints, etc. The e-learning system also verifies their answers, compute result statistics, and so on.

The function of the Content Description Processing module is mainly to convert the training content description (cf. Figure 5) that is generated by the Training Description Generation 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 (Advanced Distributed Learning Initiative, 2016). The resulting SCORM file can thus be imported into most e-learning systems.

We currently use Moodle as e-learning system (The Moodle Project, 2016), since it appears to be popular in many communities and well supported. We are now working on improving the user experience both for organizers and trainees through a deeper integration of our framework with Moodle. Thus, appropriate plugins will make it possible to automate tasks such as importing the SCORM package, will allow trainees more control of the cyber range, etc. Although the tight integration with Moodle leads to a loss of generality, we see it as an optional added value that training organizers can choose to ignore if they prefer other LMSs.

```
---
- training:
  - id: NIST-Level1
    description: >
      Investigate the security of a desktop
      computer.
    header: >
      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: NIST-L1-Q01
      type: fill
      content: >
        The operating system and kernel
        release (version) can tell you about
        the possible vulnerabilities of a
        computer. Find out the kernel release
        of the machine (e.g., 3.4.56-789).
      choice: "null"
      answer: 3.10.0-327
      hints:
      - hint: >
          You can use the command uname to
          find out OS details.
      - hint: $ uname -r
      - hint: >
          An alternative solution is to check
          the version file: $ cat /proc/version
```

Figure 5: Excerpt of a training content description in YAML format that contains a level header and one question.

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

- guest_settings:
  - id: desktop
    ip_addr: {{ guest_ip_addr }}
    basevm_host: host_1
    basevm_config_file: /123456/basevm_desktop.xml
    basevm_type: kvm
    basevm_name: basevm_desktop
    tasks:
    - add_account:
      - account: daniel
        passwd: daniel_passwd
      - account: trainee
        passwd: trainee_passwd
    - install_package:
      - package_manager: yum
        name: wireshark
        version: 1.8.10
      - package_manager: yum
        name: GeoIP
    - emulate_attack:
      - attack_type: ssh_attack
        target_account: daniel
        attempt_number: 123
    - emulate_malware:
      - name: spyeye
        cpu_utilization: 40
        mode: dummy_calculation

- clone_settings:
  - range_id: 123456
    mgmt_network: {{ clone_mgmt_network }}
    hosts:
    - host_id: host_1
      guests:
      - guest_id: desktop
        count: {{ clone_count }}
        mgmt_addr_list: {{ clone_mgmt_addr_list }}
```

Figure 6: Sample of a cyber range specification template in YAML format that represents a basic setup.

3.5 Cyber Range Instantiation

A key component of the CyTrONE framework, named CyRIS (Cyber Range Instantiation System), has already been developed for automatically creating a cyber range based on its specification (Pham et al., 2016).

The Cyber Range Specification created within CyTrONE is compatible with CyRIS, thus our framework can use all the functionality of the instantiation system. This includes: (i) training environment setup functions, such as account management, tool installation, network configuration, etc.; and (ii) security content generation functions, such as log generation, firewall configuration, malware and cyber-attack emulation, and so on (cf. Table 1).

One important characteristic of CyRIS 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 is validated. For more details about CyRIS please consult the reference cited above.

4 EVALUATION

In what follows we shall discuss the evaluation of CyTrONE from two perspectives:

Functionality. The types of cybersecurity training that can currently be conducted using CyTrONE;

Performance. The performance characteristics of CyTrONE, especially regarding cyber range instantiation at large scale.

4.1 Functionality Evaluation

The first type of evaluation we present bears on the features of the CyTrONE framework in terms of the types of cyber ranges that can be created using it, hence the kinds of training that are possible.

The U.S. NIST Technical Guide to Information Security Testing and Assessment (Scarfone et al., 2008) that we used as reference for our implementation contains three classes of technical assessment techniques:

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

In Table 1 we show that the cyber range creation features of CyTrONE can be combined in order to set up environments that cover the requirements for *all* the techniques included in the NIST guideline. Thus, we conclude that our framework provides functionality that is broad enough as to support all the training topics related to security testing and assessment, at least as they were envisaged when the said guideline was released.

Table 1: Coverage of the U.S. NIST Technical Guide to Information Security Testing and Assessment (Scarfone et al., 2008) through a combination of environment setup and 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	Log Generation	Firewall Configuration	Malware Emulation	Attack Emulation	Traffic Capture
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	○		○							

4.2 Performance Evaluation

The second kind of evaluation we conducted refers to the performance of CyTrONE, in particular related to cyber range instantiation. We have already mentioned that the democratization of large-scale training is a target of our research. 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 in the future professors in such colleges will try to set up many parallel training sessions for hundreds of students.

4.2.1 Evaluation Setup

To assess the framework performance in such conditions, we have conducted experiments on the large-scale network testbed StarBED (NICT, 2016), and used a total of up to 30 physical hosts. Note that the cyber range instantiation process is divided into three stages:

1. Preparation of the base images for VMs, currently conducted on one of the hosts, 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 the time to copy the VM base images from the master host to the other hosts, and the actual time for 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.

We conducted the performance evaluation using two training scenarios, 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 "web server", are required for this training.

4.2.2 Evaluation Results

The first performance measurements were conducted with a fixed number of virtual machines (20) that are instantiated on 1, 2, 5, and 10 hosts (with 20, 10, 4, and 2 VMs per host, respectively), so as to study the effects of distributed execution. In Figure 7 we present the average time required to instantiate cyber ranges in this experiment, for each step of the instantiation procedure and in total. 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 not necessarily doubles, since it depends on the actual content to be generated and installed for each type of VM.

The most significant effect is observed for the cloning stage. For the 1 host case (which is actually the master host), as there is no need to copy the VM base images, time is only needed to start the VMs on the master host from these images. For more hosts, the copy process, even done in parallel, becomes important. Nevertheless, although for Level 2 it is necessary to copy two VMs to each host instead of one VM for Level 1, the time is not much different until the right-most case of 10 hosts.

The total creation time plots show that, following the initial increase because of the need to copy the prepared VMs to the other hosts that was explained above—due to our parallelization— there is no significant increase in setup time afterwards. In most cases

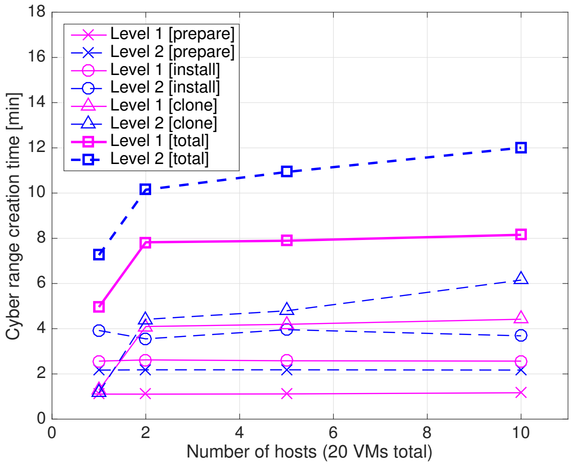


Figure 7: Cyber range creation time versus the number of hosts when using a total fixed number of 20 VMs and up to 10 instantiation hosts, for Level 1 (i.e., 20 trainees) and Level 2 (i.e., 10 trainees) scenarios.

creation finishes in a reasonable time of under 10 minutes, and only in the worst case (Level 2, 10 hosts) the creation time reaches about 12 minutes.

For the second series of measurements, we decided to keep the number of VMs per host constant (20), and assess performance for a large-scale scenario with up to a total of 600 VMs on 30 hosts (representing 600 cyber ranges for Level 1, or 300 cyber ranges for Level 2). The results shown in Figure 8 show that for preparation and installation there is basically no difference with respect to the lower-scale experiments shown before, since these operations are the same in both cases.

The cloning phase exhibits an exponential increase for the required time, with a higher exponent for Level 2, which requires copying a double number of VM base images. This is caused by the fact that, as the total amount of throughput in the interconnecting network increases, transfer times increase as well.

Nevertheless, the total creation time results show that, in a relatively 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 currently investigating the possibility to tackle the long cloning time issues by allowing each host to set up its 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 8; 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.

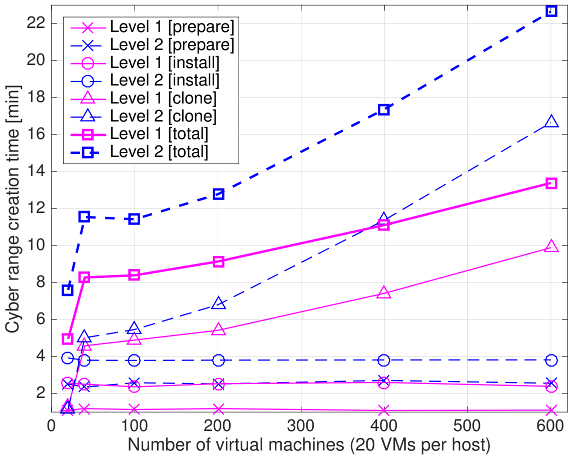


Figure 8: Cyber range creation time versus the number of virtual machines when using a fixed number of 20 VMs per host and up to 30 instantiation hosts, for Level 1 (i.e., up to 600 trainees) and Level 2 (i.e., up to 300 trainees) scenarios.

For a more detailed analysis of cyber range instantiation performance at low scale, and the differences with respect to other environment setup tools see (Pham et al., 2016).

5 DISCUSSION

In this section we compare CyTrONE to similar frameworks, and discuss two additional directions on which we plan to focus in the near future.

5.1 Comparison

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. 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 (Capture The Flag) platform, supporting quiz type, flag type and king-of-the-hill type of CTF training (Facebook, Inc., 2016). However, the Facebook CTF platform is mainly a cool UI for the training, and it does not provide assistance with a full environment setup as CyTrONE does. Moreover, there is no support for generating security content either. These tasks remain the organizer responsibility, and consequently are still tedious.

In (Raj et al., 2016), the use of application containers is proposed as a solution for improving the scalability of CTF contests. 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 planning to investigate in the future the possibility of using container technology instead of virtual machines to improve the scalability of our framework.

Closed, proprietary systems, such as the Boeing Cyber-Range-in-a-Box (CRIAB) 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, making it possible to update the content and also to expand the infrastructure depending on actual needs. 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.

By using CyTrONE, with automatic environment setup and content generation based on YAML descriptions, it becomes possible for practically anyone to conduct security training anytime and anywhere (given that host servers are available for the cyber range creation), thus leading to the democratization of cybersecurity training. 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, for instance by leveraging the advances of modern education methodologies, such as adaptive learning, etc.

We have currently reached the first concrete goal of our project: develop the fully-configurable cybersecurity training framework that based on organizer input and a training database will automatically produce the training content and training environment necessary for that training. Once its testing is finalized, we shall publicly release CyTrONE as an open-source project, so that it can be used by other organizations; the release is planned for the end of the current Japanese fiscal year (March 2017).

5.2 Training Database

Our framework currently uses a classical training paradigm of scenario-based and topic-based questions that are prepared in advance. At release we shall include samples of such training content targeting various audiences, such as technical college students, company employees, etc. However, while this approach undoubtedly serves many training purposes, especially for beginners, we are planning to also develop a *new training paradigm*, by which actual incident information is used to automatically recreate the corresponding training environment.

In this context, the framework of the ITU-T X.1500 recommendation for structured cybersecurity information exchange techniques (CYBEX) (ITU-T, 2016) is extremely relevant, and detailed information about the incidents can be obtained in standard machine-readable formats, such as Structured Threat Information eXpression (STIX) (OASIS CTI Technical Committee, 2016) or Incident Object Description Exchange Format (IODEF) (Danyliw et al., 2007). This information is the basis for reproducing the incident; furthermore, vulnerability databases such as CVE (Common Vulnerabilities and Exposures) (MITRE Corporation, 2016) will be used to recreate the target (victim) environment, and public websites such as the Exploit Database (EDB) (Offensive Security, 2016) will be used to obtain exploit code for recreating the attack in the cyber range.

The novelty of this approach is that, through the use of de-facto standards as the source of the database content included in our framework, it becomes possible to conduct training in similar conditions to a certain incident and/or vulnerability as soon as the corresponding information is made public. 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.

5.3 User Trials and Integration

Once the development of the framework is finalized, we shall proceed with several usability tasks. First of all, we'll do a series of user trials to validate the system in various training scenarios, from the point of view of: (i) content and training environment, and (ii) user interfaces both for organizers and trainees.

Secondly, we'll 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.

The usability testing we shall carry out will ensure that the framework is operating correctly in a wide range of scenarios, and the integration with existing training projects will ensure a clear and immediate contribution to society.

As more training content is added to the framework, we shall also conduct tests and surveys regarding the training quality improvement ensuing from the use of CyTrONE, which is another measure of training effectiveness.

6 CONCLUSIONS

In this paper 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.

The CyTrONE framework 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.

We have also evaluated the framework performance regarding cyber ranges instantiation, and we have 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 (Level 1), and within 15 minutes for 100 participants when using a more advanced training setup with two VMs per trainee (Level 2). Even for a total of 300 participants, the setup time is under 10 minutes for Level 1, and only a little above 20 minutes for Level 2.

Our future work includes several main directions: (i) improve the performance of the current system; (ii) introduce a new training paradigm in which machine-readable incident reports are used to automatically generate the corresponding training environments; (iii) conduct user trials for the overall framework, and integrate it with existing training programs.

REFERENCES

Advanced Distributed Learning Initiative. SCORM Official Website. http://www.adlnet.gov/adl-research/scorm/.

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

C. Evans. The Official YAML Website. http://www.yaml.org/.

Chef. Chef - Automate Your Infrastructure. https://www.chef.io/chef/.

Crockford, D. (2006). IETF RFC 4627: The application/json Media Type for JavaScript Object Notation (JSON) . In *Internet Requests for Comments, Internet Engineering Task Force*.

Danyliw, R., Meijer, J., and Demchenko, Y. (2007). Incident Object Description Exchange Format (IODEF), IETF RFC 5070.

enPiT University Consortium. enPiT-Security (SecCap) Training Program (in Japanese). https://www.seccap.jp/.

Facebook, Inc. Platform to host Capture the Flag competitions. https://github.com/facebook/fbctf.

ITU-T. Revised structured cybersecurity information exchange technique. Recommendation X.1500 (2011) Amendment 9 (03/16).

Ministry of Internal Affairs and Communications (MIC), Japan. Cyber Defense Exercise with Recurrence (CYDER) Training Program (press release). http://www.soumu.go.jp/main_sosiki/joho_tsusin/eng/Releases/Telecommunications/130925_02.html.

MITRE Corporation. Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org/.

National Institute of Information and Communications Technology (NICT), Japan. Hokuriku StarBED Technology Center. http://starbed.nict.go.jp/en/index.html.

OASIS Cyber Threat Intelligence (CTI) Technical Committee. Structured Threat Information eXpression (STIX). https://stixproject.github.io/.

Offensive Security. The Exploit Database (EDB). https://www.exploit-db.com/.

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

Raj, A. S., Alangot, B., Prabhu, S., and Achuthan, K. (2016). Scalable and Lightweight CTF Infrastructures Using Application Containers. In *Proceedings of the 2016 USENIX Workshop on Advances in Security Education (ASE '16)*.

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

SANS Institute. SANS NetWars Training Courses. https://www.sans.org/netwars.

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

The Moodle Project. Moodle – Open-source Learning Platform. https://moodle.org/.

The OpenStack Foundation. OpenStack – Open Source Cloud Computing Software. https://www.openstack.org/.

Web Application Security Forum (WASForum). Hardening Project (in Japanese). http://wasforum.jp/hardening-project/.