CyRIS: A Cyber Range Instantiation System for Facilitating Security Training

Cuong Pham
Japan Advanced Institute of Science and Technology

Dat Tang
Japan Advanced Institute of Science and Technology

Ken-ichi Chinen
Japan Advanced Institute of Science and Technology

Razvan Beuran
Japan Advanced Institute of Science and Technology

ABSTRACT

Cyber ranges are well-defined controlled virtual environments used in cybersecurity training as an efficient way for trainees to gain practical knowledge through hands-on activities. However, creating an environment that contains all the necessary features and settings, such as virtual machines, network topology and security-related content, is not an easy task, especially for a large number of participants. Therefore, we propose CyRIS (Cyber Range Instantiation System) as a solution towards this problem. CyRIS provides a mechanism to automatically prepare and manage cyber ranges for cybersecurity education and training based on specifications defined by the instructors. In this paper, we first describe the design and implementation of CyRIS, as well as its utilization. We then present an evaluation of CyRIS in terms of feature coverage compared to the Technical Guide to Information Security Testing and Assessment of the U.S National Institute of Standards and Technology, and in terms of functionality compared to other similar tools. We also discuss the execution performance of CyRIS for several representative scenarios.

Keywords

Cybersecurity; cyber range; cybersecurity education; cybersecurity practice

1. INTRODUCTION

Cybersecurity, also known as IT security, focuses on protecting networks, computers, programs and data from attack, damage, unintended or unauthorized access. As the world becomes increasingly interconnected, governments, corporations, financial institutions and other businesses store a great deal of confidential information on computers and transmit it across network everyday. With the growing volume and sophistication of cyber attacks, ongoing attention to cybersecurity is required to protect sensitive business and personal information.

A lot of courses about cybersecurity are available for both beginners and advanced learners nowadays. For beginners, the main purpose of the courses is to raise their awareness about cyber attacks and help improve their skills in protecting themselves and their business. In contrast, advanced courses focus learners on deep concepts of security. Most of the content requires trainees to deal with challenging and hands-on practice with real world scenarios. A cyber range is a virtual environment that participants can access and investigate to find answers to questions, and acquire practical skills during the training. It is specifically designed for a training session, in that it contains all the infrastructure (machines, networks, tools, etc.) and security settings which are related to the content of the course. It also needs to be well controlled in the networking perspective, which has to be isolated from the outside to avoid traffic leakage and separated among trainees to prevent accessing each other's environment. Currently, the task of preparing cyber ranges for cybersecurity training is done manually, which is time consuming, with lots of effort and skills required, and error-prone.

To address this difficulty, we propose a cyber range instantiation system called CyRIS as a solution of reliably and repeatably creating and managing environments for cybersecurity training courses. It is a component of a training support framework called CyTRONE, which is being developed at JAIST (Japan Advanced Institute of Science and Technology), that automatically creates realistic training environment based on instructors' requirements [1].

In CyRIS, a cyber range definition description is created by instructors to describe in detail the content of the training. This description provides instructors with a way to thoroughly review the content that they are going to teach participants in their course. CyRIS then takes the file as an input and automatically creates the desired environment without human interaction. CyRIS offers both basic functions that are commonly seen in other automated configuration management tools, and security functions, which are able to reproduce actual security-related incidents to create corresponding settings. This group of features is unique in CyRIS compared to other tools, and they play a key role in creating realistic cyber ranges. In this paper, we describe the design and implementation of CyRIS. In addition, we present an evaluation of CyRIS in terms of security feature coverage along with functionality coverage, and execution performance.

SoICT '16, December 08-09, 2016, Ho Chi Minh City, Viet Nam

DOI: http://dx.doi.org/10.1145/3011077.3011087

The remainder of the paper discusses related work (Section 2), presents our work of CyRIS in detail (Section 3), evaluates our system in several perspectives (Section 4), and ends with conclusions and future work (Section 5).

2. RELATED WORK

Various tools have been published that provide users an automated way to configure both physical and virtual environments. These configurations include installing content, setting up system and applications, and so on. Ansible [2], Vagrant [4], and Chef [5] are well-known tools in this area. They allow users to specify settings and content in a "recipe" and create a desired environment from a clean node. However, none of these tools has the function of configuring network service and topology among nodes, and it also gets harder when users have to create multiple recipes for circumstances that involve setting up a large environment with many nodes.

For network management configuration tools, we have cloud controllers such as OpenStack [3] and VMWare vSphere [6]. These tools are able to create and manage a cloud system with multiple virtual machines and network service among them. Another tool is SpringOS [7] that is currently in use at StarBED facility [8] for managing physical nodes and network experiments. It has the functionality of controlling network topology among nodes, which is missing on the other tools. Nevertheless, these tools lack the ability of setting content and configuration on individual nodes. In addition, SpringOS is designed only for StarBED use purpose.

Shingo Yasuda et al. introduce Alfons [9] as a recent tool that can be used to create environments for cybersecurity training and malware analysis. They consider an original node is a clean operation system with unique data files inserted. Therefore, copying files and executing scripts are two main mechanisms Alfons uses to create the required environments. This implementation is simple while highly efficient. Nonetheless, often content in cybersecurity that requires launching actual incidents and it is out of Alfons's coverage. Moreover, its source code is closed.

Recently, Facebook has introduced its own CTFs platform to open source [10]. Facebook's CTF provides users a free platform that takes care of the maps, team registration, and scoring. It offers a way to make security education easier and more accessible for people who are interested in information security and technical skills. It also brings schools, organizations, and others who lack resources a chance to host their competition and teach students and employees about hacking skills. On the contrary, it is limited in the range of security knowledge, when only a small set of challenges is publicly available in the competition and mainly focuses on hacking ability.

The SANS Information Security Institute [15] and CERT of the Software Engineering Institute [16] are two of the most famous and trustworthy organizations in the world regarding cybersecurity training. They offer hands-on, interactive and practical courses for a large number of trainees in different important topics about cybersecurity. However, their cyber range creation processes are not made public.

Compared to these tools, CyRIS is an open-source tool for creating cyber range environments in a flexible and efficient manner. It is implemented in the Python language. CyRIS has both functions for installing content on a a large environment with many nodes, and for configuring the network service among them. Moreover, CyRIS offers built-in features for creating specific security-related content by launching real incidents, which greatly facilitates the preparation work for the course instructors.

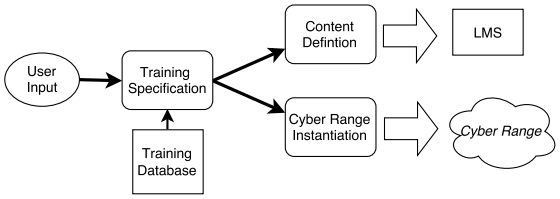

Figure 1: The system architecture of the cybersecurity support framework CyTRONE framework.

3. CYBER RANGE INSTANTIATION SYSTEM (CyRIS)

CyRIS is the core component of the cybersecurity training framework CyTrONE [1], which is being developed at JAIST. The system architecture of CyTrONE is described in Figure 1. It consists of three main parts, which are:

- **Training Specification** Based on user inputs and the training database (including training scenarios, well-known security incidents and vulnerability information, etc.), this module creates a *content description* and a *cyber range description* that define the content and activity of the training.
- **Content Definition** This module takes the content description and generates the corresponding training content for an LMS (Learning Management System), currently using Moodle [11].
- **Cyber Range Instantiation** This module is using CyRIS, which takes the cyber range description and automatically creates the corresponding cyber range environment.

The traditional approach for practical cybersecurity training is to use a dedicated and isolated physical computer infrastructure as the training environment. Such infrastructure are expensive in terms of creation and maintenance, and inefficient in terms of scalability to serve a large number of trainees. Because of these significant drawbacks, using virtualization technology is a solution towards this problem. This solution provides a way to create a comprehensive and realistic security training environments at a low-cost and in a scalable manner. Being aware of this, CyRIS is dedicated for constructing virtual cyber range environments using KVM virtualization platform [13], [14].

In this section, we present the design and implementation of CyRIS, which is a tool for automatic preparing and managing cyber ranges for cybersecurity training based on a cyber range description provided by the organizers. We first introduce the CyRIS working flow and the format of the description file, then go into detail about its security features.

3.1 CyRIS Overview

The working flow of CyRIS is described in Figure 2. Two inputs that the system takes to create a desired cyber range

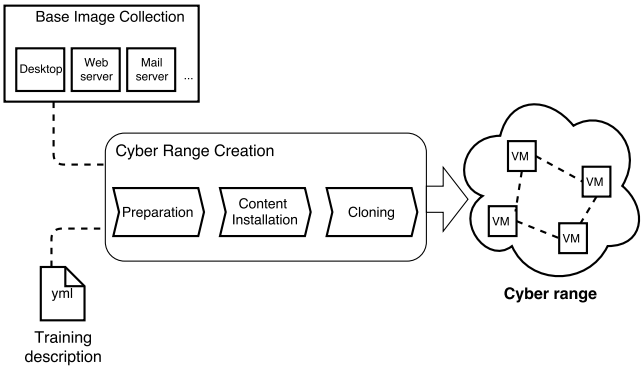

Figure 2: The working flow of CyRIS.

are a set of base images and a cyber range definition description. The base images are under the raw format that is used for KVM virtualization. They contain a pre-installed operating system and several basic system configuration (e.g. host name, ssh keys, IP address, etc.). A program is created for automating the task of setting these configurations. Large companies and organizations nowadays often choose RedHat Enterprise Linux as the operating system for their servers because of the platform's performance, stability and security, which let them build their IT infrastructure across the enterprise. For this reason, CentOS 7, which is the latest community version of RedHat Linux, is our choice as the main operating system for cyber range environments that CyRIS will prepare security-related content on.

The cyber range description file is for describing the composition and content of the cyber range. It can be created manually or by automated tools. This description is written under YAML, a text-file format (the reason we choose this format but not the well-known XML is that it is similar in terms of functionality, but much better in terms of readability). It defines all the necessary information needed for creating a cyber range. Figure 3 gives an example of how a description file looks like. It is divided into three parts as follows:

- `host_settings` contains information about the hosts that the cyber range is deployed on, including an id, a management address, and a management account.
- `guest_settings` provides information about the base images. The tag `tasks` defines all the content of the cyber range that CyRIS needs to prepare. In this example, the cyber range consists of several settings, including installing the tool wireshark, emulating a DDoS attack, capturing traffic and deploying an emulated malware in the calculation mode.
- `clone_settings` gives details about the cloning phase. It has a unique range id, a management network, and a list of virtual machine addresses.

There are three main steps in the process of creating a cyber range environment, which are (i) preparation, (ii) content installation and (iii) clone phases. In the first phase, based on the global information of `guest_settings`, CyRIS starts up the corresponding base images to be ready for the next step. It then installs security content which is specified by the tag `tasks`. After finishing this, it launches the last

```
- host_settings:
  - id: host_1
    mgmt_addr: 172.16.1.2
    account: crond

- guest_settings:
  - id: desktop
    ip_addr: 192.168.122.100
    basevm_name: basevm_desktop
    tasks:
    - install_package:
      - pacakge_manager: yum
        name: wireshark

    - emulate_attack:
      - attack_type: ddos_attack
        target_account: daniel

    - emulate_traffic_capture_file:
      - format: pcap
        file_name: /home/trainee/traffic.pcap
        attack_type: ssh_attack
        attack_source: 2.95.120.235
        noise_level: medium

    - emulate_malware:
      - name: spyeye
        mode: dummy_calculation

- clone_settings:
  - range_id: 123242
    mgmt_network: 10.1.1.1/16
    - guests:
      - guest_id: desktop
        host: host_1
        mgmt_addr_list: 10.1.1.2; 10.1.2.2;
```

Figure 3: An example of a cyber range definition description. It provides all necessary information for CyRIS to construct a desired cyber range.

phase with the information from `clone_settings` to create a number of virtual machines and setup the network service, then ends the cyber range creation process.

The system running CyRIS has the architecture presented in Figure 4. A collection of hosts are connected to each other via LAN network, and one host is designated as a master node. This master node has CyRIS service running, processes the content of the description file, prepares and allocates virtual machines on other hosts, etc.

3.2 Features

There are five main features that play key roles in installing content for cyber ranges, including (i) system configuration, (ii) tool installation, (iii) incident emulation, (iv) content management and (v) clone management. This section describes in detail the functionality and implementation of each of them.

3.2.1 System Configuration

As being said at the beginning of Section 3.1, basic system configuration (host name, ssh keys, ip address, etc.) in base

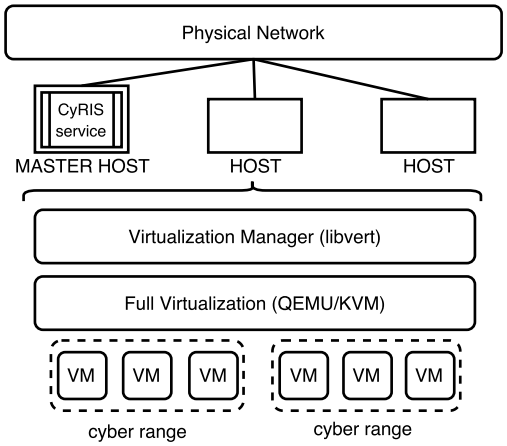

Figure 4: The architecture of the system running CyRIS.

images has been taken care in advance. Thus, CyRIS tasks about system configuration, which are about creating settings for security training, are limited to managing accounts and modifying firewall rules.

Managing accounts is used to create new accounts, edit information of or delete existing accounts. These settings can be used for practicing penetration testing techniques, when trainees identify potential weak passwords in the system and learn how to crack them.

Another functionality in this feature is modifying the firewall ruleset. In cybersecurity training, it is important to have a firewall ruleset that contains a mixture of good and bad rules. In this module, CyRIS sets up the firewall by launching a script that is provided in the cyber range description. This helps organizers to have an automated way for adding and modifying firewall rules, so that they can have a realistic training environment.

3.2.2 Tool Installation

In cybersecurity, tools are essential. Various tools are well-known and indispensable in the security world, such as wireshark and tcpdump for network sniffing related jobs, or aircrack-ng suite and john-the-ripper for penetration testing. Knowing how to use them is a must for people working in security, and it is one of the main topics of training sessions.

This feature provides organizers with a mechanism to install such tools automatically. It contains three main types of installation manners, which are package, source and custom installation. The first type installs tools from official Linux repositories using package managements (apt-get, yum, etc.). Users specify the type of package management, the name of software and its version, then the module automatically downloads and installs it on the machine. The second one builds tools directly from source. Users provide the path that leads to the source and version of compiler, then the system installs it by `make install` commands. The last one is custom installation, which gives users freedom to install any tools in any way they want. Users basically give the system a script containing all of the necessary commands. The system then runs the script and builds tools from there.

In the cybersecurity world, there are many tools that are customized and impossible to install via package managements. Moreover, in many cases that users want to have an older version of a tool that contains a security vulnerability, and it no longer exists in official repositories. Therefore, having the two latter installation methods helps users facilitate the task of installing tools in special situations.

3.2.3 Incident Emulation

This feature includes three main functions, which are (i) attack, (ii) traffic capture and (iii) malware emulation. These functions have the ability of launching actual incidents to prepare security-related content for cyber ranges.

Attack Emulation.

Recognizing attack patterns is one of the main activities during cybersecurity training. It helps trainees improve the ability of detecting whether any attack is taking place, and of defending the system against attacks. Two kinds of attacks that are usually deployed in a training environment: static attacks and dynamic attacks.

Static attacks are actions that take place before the training session starts. After performing attacks, attack traces are recorded and left in the system, so that trainees can try to find out what kind of attacks has been performed. The purpose of this activity is to improve trainee 's knowledge in forensic techniques (log and ruleset review, network sniffing and file integrity checking). On the other hand, dynamic attacks are usually deployed during the training session. For example, organizers perform DoS attack in order to reduce the performance of the system. Trainees have to know how to recognize they are being under attack and how to respond to it.

CyRIS is capable of emulating static attacks of specific types, namely SSH dictionary, DoS and DDos attacks in wired connection network. After that, CyRIS lets users capture traces of these attacks, that can prepare settings for questions related to this topic. We also plan to extend this feature to be able to perform attacks dynamically. Regarding wireless connection, because of having no way to mimic a wireless network among virtual machines in KVM platform, emulating attacks in this kind of connection is out of CyRIS's reach at the current stage.

Traffic Capture.

Alongside with attack emulation is the traffic capture feature. With this, CyRIS is able to prepare traces for various kinds of attack in both wired and wireless connection networks.

For wired connection, users define what kind of attack traces they want in the cyber range (SSH dictionary, DoS, DDoS) and CyRIS deploys the corresponding function from attack emulation feature. Simultaneously, it uses `tcpdump` to sniff the network traffic and capture traces of the attack under the `pcap` format. These traces can be given to trainees so that they can practice their forensic skills by trying to find out the pattern of the attack. Moreover, CyRIS offers instructors an option of mixing the attack traces with usual network traffic, making the capture file more realistic, and also making the problem more challenging to participants.

Regarding wireless connection, because of the problem that has been said in the previous paragraph, we prepare in advance traffic capture files that contain traces of attacks that are normally performed in real world, which are *replay*

attack and *DoS attack*. In addition, we prepare files related to WEP wireless network security protocol, so that trainees can investigate and learn how to crack passwords using tools like aircrack-ng. We also produce files of other types of protocol (WPA, WPA2) for reference purposes.

Malware Emulation.

Detecting malicious programs is another important technique for cybersecurity professionals. It is essential to know the methodologies to discover whether any unusual application is running under the hood.

With the malware emulation feature, CyRIS offers the ability of launching an emulated malware running in the system. This dummy malware is totally unharmful and can run under two modes at the current stage: performing a calculation or listening to a port. In the first mode, the dummy malware is able to run with a certain amount of CPU consuming, neither too low to avoid being invisible from trainees, nor too high to appear obvious. The second mode allows the malware to run as a service that listens to the network traffic via an arbitrary port. To deploy this dummy malware, users give it a name and a mode. It then appears in the list of background processes, creating a security threat in the system.

Besides, instead of using the dummy malware, there are cases when instructors want to deploy their custom malware for training participants about particular security topics. CyRIS helps them satisfy this need by having the content management feature (which will be discussed in detail in the Section 3.2.4), in that instructors simply use an executing scripts mechanism to launch their custom malicious programs in the cyber range.

3.2.4 Content Management

This feature offers three mechanisms to modify the content of an environment, including (i) copying content, (ii) executing scripts, and (iii) generating logs.

Basically an OS is a collection of files in a file system. In other words, it is possible to have every setting by inserting the correct files into the correct place [9]. CyRIS adopts this method by having the copy content function. It allows users to copy files and data from outside to the cyber range to create a setting. For example, if users want to set the host name of a machine, they simply use this function to copy a text file named `hosts` with the host name inside and copy it to the `/etc/` directory.

The second mechanism is the ability of letting users executing scripts in cyber ranges. It needs two parameters, which are `program` and `compiler`. The `program` tells CyRIS the location of the script and arguments needed, while the latter one specifies the script language, including shell script, Perl, Python and Ruby.

The last mechanism is to generate logs for recording activities that happened in the past. They can be unauthenticated log-in attempts, service startup and shutdown information, file access, security policies changes, account changes, etc. CyRIS uses two ways to create logs for these incidents, which are:

- Using other modules to actually perform the actions, so that logs will be generated consequentially. For example, managing accounts module modifies existing accounts, or attack emulation module proceeds ssh dictionary attack to create logs for failed authentication attempts.
- Using the generating logs mechanism in this feature to append textual content related to bad incidents to system log files, making it look like they have really taken place in the past. These incidents normally are hard to reproduce in real time, such as information about starting up and shutting down system.

Table 1: Current features of CyRIS

Categories	Basic features	Security features
System Configuration	Manage accounts	Modify firewall ruleset
Tool Installation	Install package Install from source Custom install	
Incident Emulation		Emulate attacks Capture traffic Emulate malware
Content Management	Copy content Execute script	Generate logs
Clone Management	Configure network Clone virtual machines	

3.2.5 Clone Management

After finishing preparing settings for cybersecurity training on base images, the clone management feature takes place to create a number of cyber range instances in order to serve multiple trainees simultaneously. Several requirements are achieved in this feature:

- *Network topology*: a cyber range consists of a set of connected virtual machines that mimics a real network environment, and its topology can be various (mesh, star, bus or ring topology). At the moment, CyRIS only support bus topology for cyber ranges, in that all of virtual machines are connected to one virtual bridge in the host, and they are accessible from there.
- *Environment separation*: cyber ranges are places for trainees to practice all kinds of security techniques, and it is possible to have traffic leakage to the outside network. To avoid this problem, it is important to isolate the training environment. In CyRIS, cyber range instances connect to the host through virtual bridges that lead to nowhere outside the host. Moreover, these virtual bridges have no connection between each other, and an account and a password are generated randomly for each trainee to access their cyber range instance via SSH connection, making sure that no one is able to access other trainees' environments.

Table 1 summarizes all the features that CyRIS offers at the current stage. We divide them into two groups that are basic and security features. The first group contains common functions that other tools also have, while the second group are to use from a security perspective. These security features are the main difference of CyRIS compared to other well-known tools.

4. SYSTEM EVALUATION

In this section, we first evaluate the coverage that CyRIS can offer in terms of preparing security content for cybersecurity training. For this context, we use the U.S. NIST Technical Guide to Information Security and Testing Assessment [12] as a reference. We then discuss about the

feature comparison between CyRIS and other tools. In addition, we present results of CyRIS performance in creating representative cyber ranges.

4.1 Functionality Evaluation

This section describes our evaluation of CyRIS about feature coverage in preparing content for cybersecurity trainings, and the comparison between CyRIS and other similar tools in respect of functionality.

4.1.1 Feature Coverage

The NIST guideline [12] states a number of techniques in information security testing and assessment, which are categorized into three main groups:

- *Review techniques* relates to manual inspections and reviews to evaluate applications, architecture designs of network and systems in the purpose of discovering vulnerabilities. This group of techniques consists of documentation, log, ruleset, and system configuration review; network sniffing; and file integrity checking.
- *Target identification and analysis techniques* are testing techniques that can identify systems, ports, services, and potential vulnerabilities, and may be performed either manually or using automated tools. They include network discovery, network port and service identification, vulnerability scanning, wireless canning, and application security examination.
- *Target vulnerability validation techniques* are testing techniques that corroborate the existence of vulnerabilities, and may be performed manually or by using automatic tools, depending on the specific technique used and the skill of the test team. Target vulnerability validation techniques include password cracking, penetration testing, social engineering, and application security testing.

Table 2 shows in detail about how to combine CyRIS features to create content for different security techniques. Basically, all realistic content needed for each security technique in the NIST guideline is covered by CyRIS. Basic features like install tools and so on play the role of preparing the infrastructure for the system, and the security features prepare specific content that corresponds to each and every technique. Let's take network sniffing technique as an example. Normally to train for mastering this technique, a traffic capture file with some attack pattern is given to trainees. CyRIS first provides a way for them to investigate the traffic capture file by installing `tcpdump` or `wireshark`, depending on the specification of the instructors. It then creates the required traffic capture file by combining attack emulation and traffic capture. Another example is about vulnerability scanning technique by which trainees learn how to identify vulnerabilities in the system (e.g., malware applications, open ports, etc.). In this case, CyRIS either executes a script to start an application or deploys the dummy malware that has an unusual name and listens to an arbitrary port.

4.1.2 Feature Comparison

Table 3 shows the comparison in terms of feature between CyRIS and other recent similar tools. While basic features are common in automated environment configuration tools, we find that security features are unique to CyRIS. There

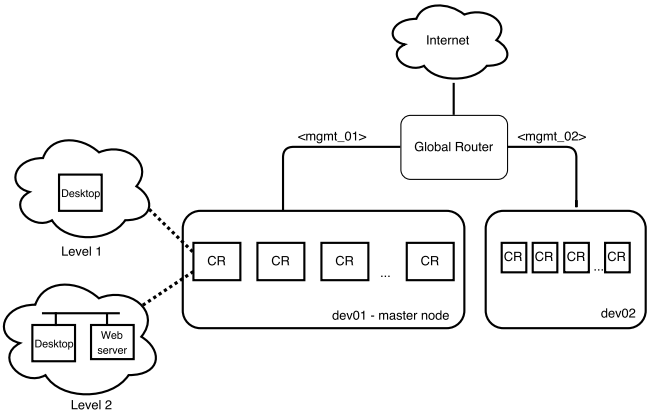

Figure 5: Network topology for evaluation.

are many security settings Alfons can prepare by executing pre-prepared scripts or copying data files from outside to the cyber range, but this process is costly and requires users to produce such files manually in advance. With the security features, CyRIS allows users to create these settings by launching real actions, in a convenient and flexible manner. This characteristic is extremely important and helpful in creating a realistic environment for cybersecurity training.

One example of this usefulness is preparing logs for unsuccessfully login attempts in CentOS 7. Alfons or Ansbile can simulate this situation by replacing the log file `/var/log/secure` with a file containing logs for this incident that users have created in advance. This method is inefficient, and it works only with people who have the habit of investigating `/var/log/secure`. For other people, they may use a command called `lastb` to check for such attempts, and none of this trace is shown at the output. In contrast, by emulating the ssh attack, CyRIS generates real logs for the incident and the information appears in both places.

Besides, CyRIS is able to create virtual machines in cyber range environment and configure network among them. The network topology module is currently limited to only bus topology setup at the moment, but we plan to extend the implementation to cover custom network topologies. Moreover, CyRIS makes the environment isolated from the outside network to avoid potential traffic leakage or bad incidents. It then can generate a random account and password for each trainee so that they cannot access others' information, enhancing the security and fairness during the training.

4.2 Performance Evaluation

This section describes our experiments for evaluating CyRIS performance and presents preliminary results in representative scenarios.

4.2.1 Evaluation Environment

We constructed an environment as shown in Figure 5 for the evaluation. The physical facility consists of two servers (`dev01` and `dev02`), connected to a global router. Each server has the specifications of two 4-core Intel Xeon®E5504 2GHz CPUs, 72GB memory, 400GB HDD hard drive, and 1Gbps network interface. The first server `dev01` is designated as the master node for running CyRIS.

Cyber Range Organization and Design (CROND), an NEC

Table 2: Summary of CyRIS features required to support each security technique in the U.S. NIST Technical Guide to Information Security and Testing Assessment

NIST Security Technique	Basic Features					Security Features				
	Manage Accounts	Install Tools	Copy Files	Execute Scripts	Configure Network	Generate Logs	Modify Firewall	Emulate Malware	Emulate Attacks	Capture Traffic
Log Review		x		x		x			x	
Ruleset Review		x		x			x			
System Configuration Review	x	x		x		x	x			
Network Sniffing		x	x		x				x	x
File Integrity Checking		x	x	x						
Network Discovery		x			x					x
Network Port and Service Identification				x	x		x	x		
Vulnerability Scanning		x	x		x	x	x	x	x	x
Wireless Scanning		x	x						x	x
Password Cracking	x	x								
Penetration Testing	x	x	x	x	x	x	x	x		
Social Engineering	x		x							

Table 3: Functionality comparison between CyRIS and other similar tools

Tools	Content installation		Network configuration
	Basic features	Security features	
CyRIS	✓	✓	✓
Alfons	✓	✗	✓
Ansible	✓	✗	✗
OpenStack	✗	✗	✓

Corporation endowed chair at JAIST, has been developing a cybersecurity training for information security professionals. Its content has multiple difficulty levels, covering all the essential security techniques mentioned in the NIST guideline. Based on this training, we take two typical levels for our experiments. The first level's topic is about the security of a desktop computer. Its cyber range includes one desktop that contains content for training review and analysis techniques, such as log and system configuration review, network sniffing, vulnerability scanning, etc. The second level is designed towards advanced security people, in that it concentrates on more sophisticated security knowledge about networking. Its cyber range reflects a small company's network, which has one web server and one desktop connected to each other, and contains content for skills including network discovery, password cracking, penetration tesing, and social engineering techniques.

We use these two models of cyber range to evaluate CyRIS executing performance. We firstly create 20 instances of the cyber range Level 1 on one physical machines for a small training class. We then expand the class by installing 60 instances for the Level 1 and 30 instances for the Level 2 (60 virtual machines in total in both cases) on two physical hosts. Each virtual machine in these cyber ranges has the disk size of 8GB.

4.2.2 Experimental Results

Table 4 shows the processing time for each phase of CyRIS operation for installing cyber ranges. Each data point represents the mean of 10 experimental runs. When creating only 20 instances of Level 1 cyber range on one host, the average time is less than 5 minutes. In the second case when we create 60 instances of the same cyber range on two hosts, the time increases to roughly 11 minutes. This is mainly from (i) the time CyRIS spends on copying the base images from the master-node to the other, and (ii) the network delay when CyRIS executes clone phase on the second node via a SSH connection. The average time also increases when it comes to creating 30 instances of Level 2 cyber range in two hosts, which is about 14 minutes. Even though the number of cloned virtual machines are the same as the previous case (60 virtual machines), it is understandable that CyRIS takes longer time for Level 2 in every phase than Level 1. Level 2 has two virtual machines in each instance, while only one is in Level 1, making it longer than the other to get base images ready in the preparation phase. The security content in Level 2 is more sophisticated since it is designed for high-skilled trainees, and the network configuration task during the last phase is also needed.

Table 4: Performance results of CyRIS for creating cyber range environments containing representative security-related content and network configuration

Level	Number of virtual machines	Number of hosts	Task	Average time [s]
Level 1	20	1	Prepare base images	46.1
			Install content	72.7
			Configure network and clone virtual machines	167
			Total	**285.8**
Level 1	60	2	Prepare base images	46
			Install content	71.9
			Configure network and clone virtual machines	554.2
			Total	**672.1**
Level 2	60	2	Prepare base images	47.5
			Install content	212.8
			Configure network and clone virtual machines	604.3
			Total	**846.6**

The present evaluation, however, has no comparison between CyRIS and other tools in terms of performance. The content of cybersecurity training varies from course to course, thus different environments have to be constructed accordingly. Besides, the closure of similar tools' source code makes the comparison more difficult to conduct. We plan to contact Alfons' authors [9] to ask about the content they put in the cyber range environment in their experiment, so that we can compare its performance to CyRIS.

5. CONCLUSION

In this paper we proposed CyRIS as an automated tool for creating cyber range training environments. CyRIS supports both basic features for preparing the infrastructure, and security features for configuring sophisticated security-related content. In addition, CyRIS has the ability to setup

the network service among virtual instances, and is able to isolate the environment from the outside network for safety purposes.

We evaluated CyRIS both from a security feature coverage and a functionality comprehensiveness perspectives. While the basic group of features are common among other automated configuration management tools, the security group appears unique to CyRIS. These novel security features offer instructors a new way of constructing realistic training environments in an efficient and flexible manner. We then presented a discussion about the execution performance of CyRIS for creating two realistic cyber range environments. CyRIS created a cyber range that consisting of 60 virtual machines with sophisticated security content and network configuration in about 14 minutes, which indicates that our proposed tool is efficient for representative scenarios.

Regarding future work, one important part is to enhance CyRIS ability to configure network topology. Various types of topology exist in reality, and it is important to be able to mimic them. The second aspect is to improve and extend the implementation of CyRIS so that it can deal with large-scale cyber range environments that involve hundreds (or even thousands) of virtual machines. For this, we plan to conduct further development and experiments on the StarBED testbed of National Institute of Information and Communication Technology [8].

6. ACKNOWLEDGMENTS

We would like to give special thanks to Bui Ha Duong, a master student at Japan Advanced Institute of Science and Technology, for his precious comments while implementing CyRIS and conducting the evaluation experiments.

7. REFERENCES

[1] R. Beuran, K. Chinen, Y. Tan, Y. Shinoda. Towards Effective Cybersecurity Education and Training. Research report, IS-RR-2016-003, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, October 2016.

[2] Ansible is Simple IT Automation. Retrieved March 22, 2016 from http://www.ansible.com.

[3] OpenStack - OpenStack Open Source Cloud Computing Software. Retrieved March 22, 2016 from https://www.openstack.org.

[4] Development Environments Made Easy. Retrieved August 3rd, 2016 from https://www.vagrantup.com.

[5] Chef | IT Automation for Speed and Awesomeness | Chef. Retrieved August 4th, 2016 from https://www.chef.io/chef.

[6] Server Virtualization with VMware vSphere. Retrieved August 4th, 2016 from http://www.vmware.com/pruducts/vsphere.html.

[7] Toshiyuki Miyachi, Takeshi Nakagawa, Ken-ichi Chinen, Shinsuke Miwa and Yoichi Shinoda. StarBED and SpringOS Architectures and their Performance. 7th International ICST Conference, TRIDENTCOM 2011.

[8] National Institute of Information and Communication Technology, Japan. Hokuriku StarBED Technology Center. Retrieved August 1st, 2016 from http://starbed.nict.go.jp/.

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

[10] GitHub - facebook/fbctf: Platform to host Capture the Flag competitions. Retrieved August 4th, 2016 from https://github.com/facebook/fbctf.

[11] Moodle - Open-source learning platform. Retrieved August 3rd, 2016 from https://moodle.org/.

[12] Murugiah P. Souppaya, Karen A. Scarfone. Technical Guide to Information Security Testing and Assessment. Special Publication 800-115. National Institute of Standards and Technology, 2008.

[13] Fabrice Bellard. (2011) QEMU - Open Source Processor Emulater homepage. [Online]. Available: http://www.qemu.org/, accessed: 2011-11-18.

[14] Red Hat, Inc. (2011) Kernel-based Virtual Machine (KVM) homepage. [Online]. Available: http://www.linux-kvm.org/, accessed: 2011-11-18.

[15] SANS Information Security Training | Cyber Certifications | Research. Retrieved August 4th, 2016 from https://www.sans.org/.

[16] The CERT Division | SEI | CMU. Retrieved August 4th, 2016 from www.cert.org/.