AWS EC2 Public Cloud Cyber Range Deployment

Razvan Beuran, Zhe Zhang, Yasuo Tan
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

***Abstract*—Cybersecurity training activities require specialized environments, typically called cyber ranges, to make it possible for trainees to acquire not only security knowledge, but also practical security skills. However, the setup of these training environments is a tedious task, which hinders the wider use of cyber ranges for security training. In its turn, this has a negative impact on the development of the cybersecurity workforce that is exceedingly necessary in our network-centric society.**

In this paper we introduce our approach of using the Amazon Web Services (AWS) Elastic Compute Cloud (EC2) public cloud for cyber range deployment, thus making it possible to conduct cybersecurity training activities at scale and at a relatively low cost. Our system was implemented by extending the functionality of the cyber range instantiation system CyRIS that is available as open source on GitHub. We evaluated our implementation from several perspectives, demonstrating that public cloud deployment can provide similar functionality and performance compared to local server or private cloud deployment, while avoiding the high purchase and management costs associated to those.

***Index Terms*—cybersecurity training, hands-on training, cyber range, public cloud, AWS EC2**

1. Introduction

The lack of cybersecurity professionals with adequate skills for fighting the security threats our modern society is faced with lead to an increase in recent years of the number of cybersecurity training programs that are being offered by academic institutions, commercial companies, government organizations, etc. Simultaneously, software tools that support the training activities, such as Capture The Flag (CTF) or cyber range creation platforms, have also been released publicly, often as open-source software published on GitHub.

The key component of many of these training programs is the use of specialized environments created for security training purposes, also known as cyber ranges. Note that in this paper we use the term "cyber range" in a broad sense to refer to all cybersecurity training environments, ranging from simple training environments made of a single host or virtual machine (VM), which are often used in CTF-type training, to complex and realistic environments with tens of VMs, as employed in more advanced forms of cybersecurity training.

The use of virtualization technology, typically in the form of VMs and sometime containers, is a widely-used solution for cyber range deployment, as it offers great flexibility regarding the number of cyber range components and their settings for each cyber range instance. While individual learners can conduct training activities using VMs on their own computers, for large-scale deployments, such as in academic institutions, dedicated servers must be used for running the VMs, especially for complex cyber ranges. The need to purchase and manage such a hardware infrastructure places a cost burden that may discourage certain institutions from organizing their own cybersecurity training activities.

In this paper we present an approach for cyber range deployment using public cloud infrastructure that makes it possible for training organizers to minimize deployment costs as a result of the pay-per-use model used by public cloud service providers. In particular, our implementation leverages the functionality of the Amazon Web Services (AWS) Elastic Compute Cloud (EC2) infrastructure, which is one of the most ubiquitous and easy-to-use public cloud services [1].

In terms of cyber range creation, our implementation is based on the cyber range instantiation system CyRIS developed by the Cyber Range Organization and Design (CROND) NEC-endowed chair at Japan Advanced Institute of Science and Technology (JAIST), in Ishikawa, Japan, which is publicly available as open source on GitHub [2], [3]. CyRIS has rich cyber range creation features and is well documented, but originally only had support for Kernel-based Virtual Machine (KVM) virtualization technology; hence, we selected CyRIS for extending its functionality to also make possible public cloud cyber range deployment.

The three main contributions of this paper are summarized below:

- Present the extended CyRIS architecture that adds AWS EC2 cyber range deployment support to the original CyRIS, and discuss its implementation
- Evaluate the extended CyRIS implementation from functionality and performance perspectives, demonstrating the feasibility of our approach
- Discuss the advantages and disadvantages of public cloud versus local cyber range deployments based on our implementation and usage experience

The remainder of this paper is organized as follows. In Section 2 we present several works related to our research. Then, in Section 3, we introduce the original CyRIS and the extended architecture that we designed, followed by implementation details in Section 4. The extended CyRIS implementation is thoroughly evaluated from several perspectives in Section 5. We then discuss in detail the advantages and disadvantages of our solution in Section 6. The paper ends with conclusions and a list of references.

2. Related Work

Research related to our paper can be split into three main categories, plus some additional works, as we will discuss next. We also suggest reference [4] for a comprehensive survey on cyber ranges and security testbeds.

2.1. CTF-Style Training

CTF-style training relies on providing binary files or simple training environments made of a single host or VM to trainees, with the goal of developing technical skills related to cybersecurity. Each CTF challenge focuses on a single issue, such as a certain type of web vulnerability (e.g., SQL injection, path traversal), a binary analysis technique (e.g., disassembly), etc. Gamification techniques, such as rankings and badges, are used to motivate the participants. We introduce below some of the representative tools and programs in this area.

Facebook CTF is an open-source platform made public by Facebook that supports quiz, flag and king-of-the-hill types of CTF training [5]. This platform is intended as an easy interface for trainees to be able to access the challenges and keep track of their progress. However, Facebook CTF does not provide any assistance with cyber range setup or deployment.

Raj et al. proposed the use of application containers instead of virtual machines as a solution for improving the scalability of CTF competitions [6]. Their work emphasizes the lower amount of resources and engineering effort of their solution, and while their experiments were conducted on a single server, tools such as Docker Swarm [7] are mentioned as a way for deployment to multiple physical machines, including the use of public cloud servers such as AWS EC2.

CTF-style training is often organized in the forms of public competitions that use heavily-customized or proprietary CTF tools. Examples of such free-to-attend competitions are the DEFCON convention [8] in the US, and SECCON (SECurity CONtest) [9] in Japan, which is a qualifying stage for DEFCON.

2.2. Advanced Cybersecurity Training

Cybersecurity professionals require more advanced skills than the basic ones exercised via CTFs, and they should have a high-level of expertise in one or more of the three areas of cybersecurity training: attack, forensics and defense. Consequently, more advanced programs and tools have been developed that target such professionals.

SANS Institute in the US organizes a multitude of paid training programs that cover a wide range of topics and include various certifications, one of the most relevant being the SANS NetWars training courses [10]. While some of these courses are provided on-site, an online course named "NetWars Continuous Online Skill-Sharpening Range" is also available [11]. No official information is available about the manner in which SANS cyber ranges are deployed, but we presume that custom tools are used for deployment on dedicated servers.

Hardening Project is a two-day training contest organized by the Web Application Security (WAS) Forum in Japan starting from 2012 [12]. Participating teams compete in terms of the security hardening they can provide to a virtual e-commerce web site created for the purpose of the event. Thus, the focus of the event is on maximizing the strength of the defensive skills of the participants in realistic settings, with attacks being conducted live by security experts. The cyber range is deployed using custom tools on dedicated servers managed by the National Institute of Information and Communications Technology (NICT), Japan.

A spin-off program, named Micro Hardening [13], is offered on a commercial basis, in which case the trainees' environments are attacked automatically via scripts for several times during the training activity. In this case again custom tools are used for cyber range deployment, but the deployment takes place on a cloud infrastructure provided by the Sakura Internet ISP in Japan.

Various stand-alone commercial systems for security training are also available for purchase, such as the Boeing Cyber-Range-in-a-Box (CRIAB) [14]. These systems use dedicated hardware and proprietary software, thus lacking scalability and creating a vendor lock-in; moreover, they cannot be extended from a research perspective.

2.3. Open-Source Cyber Range Platforms

While many of the projects and tools described so far use custom tools that are not made public, or cover only specific aspects of cybersecurity training, to the best of our knowledge there are only few open-source cyber range platforms that are generic enough to cover all aspects of cybersecurity training, as we describe next.

KYPO is such as platform, designed to run multiple cybersecurity exercises in parallel, including exercises that require sandboxes with many hosts, via private cloud deployment [15]. In particular, KYPO leverages the functionality of the OpenStack cloud controller [16] to deploy the training environment. A training portal is used to interface with the trainees, which may pose learning issues for first-time users. We note that KYPO developers have also created Cyber Sandbox Creator (CSC), which is a lightweight distributed lab environment that can be run directly on students' computers, however at the cost of environment scale restrictions [15].

EDURange is a platform for hosting on-demand interactive cybersecurity exercises using the AWS EC2 cloud [17]. The focus of the developers seems to be the creation of exercises (*File Wrangler*, *Ssh Inception*, *Total Recon*, etc.) revolving around a particular tool or skill, and the platform development also appears driven by the creation of such exercises. While the platform has a relatively large number of features, we found the documentation to be lacking, especially with regard to how instructors can create new exercises. Moreover, EDURange uses a custom interface for managing the training, which may be difficult to learn and operate proficiently.

CyTrONE is the integrated cybersecurity training framework released by CROND as open source on GitHub in 2017 [18]. The two main functions of CyTrONE are: (i) to facilitate the creation and management of security training content; (ii) to support the deployment of the corresponding cyber ranges used for training. A key aspect of CyTrONE is that it leverages the functionality of an

existing Learning Management System (LMS), Moodle [19], to present the scenario to trainees and manage their progress, hence it uses an interface that is probably already familiar to students. CyRIS (Cyber Range Instantiation System) is a core component of CyTrONE that is in charge of its second function, as it automates the creation and management of the cyber range training environments [2]. The fact that CyTrONE and its modules are well documented via detailed user guides, the use of Moodle as interface, the straightforward modular architecture of CyRIS and the fact that it only supports cyber range deployment via KVM made it a strong candidate for our research, and CyRIS became the basis of our extension work to add AWS support for cyber range deployment.

2.4. Other Works

In [20] a general architecture for cyber defense education is presented, which can potentially serve as model for future cyber range systems, especially when considered from an education perspective.

Alfons is a system for the construction of "mimetic" network environments intended for realistic cybersecurity training [21]. This system has useful features, such as a language for describing the target environment that makes possible environment construction automation, but it was not released publicly, and is used exclusively for training activities organized or sponsored by NICT in Japan.

Security experiment testbeds are sometimes used for cybersecurity education purposes, and perhaps the most well-known example in this category is DETER, which is one of the first open-access cybersecurity experimentation testbeds that was used already in many projects worldwide [22]. Another security experiment testbed is Testbed@TWISC, a large-scale network emulation testbed part of the Taiwan research and education network that has been used to support research and cybersecurity education in Taiwan for more than 10 years [23].

3. Approach Overview

In this section we provide an overview of our methodology, and introduce the cyber range instantiation system CyRIS on top of which our implementation is built. We then present the architecture that we designed in order to enable AWS cyber range deployment for CyRIS.

3.1. Methodology Outline

Our approach was guided by two main principles:

1) Preserve the cyber range creation functionality of CyRIS, so that the same types of cyber ranges and cyber range content can be created
2) Extend as needed the support in terms of VM creation and deployment, so that the cyber ranges created by CyRIS can be deployed via AWS

Regarding the first item, we took great care to preserve the critical CyRIS processing stages, and integrated them seamlessly into the new processing flow, as it will be described in Section 3.3.

As for the second item, we leveraged the Python scripting support provided by AWS EC2 to automate all the necessary steps related to VM creation and deployment in the AWS cloud; more details about the implementation will be provided in Section 4.

Note that, for simplicity reasons, in what follows we shall refer to the original CyRIS that only supported KVM virtualization as "KVM CyRIS," and to the extended CyRIS version that we implemented, which adds AWS cloud support, as "AWS CyRIS."

3.2. KVM CyRIS Overview

The key features of the CyRIS cyber range instantiation system are as follows:

- Use YAML-based text descriptions for the cyber range to be created, so that it can be easily updated and improved by instructors; content descriptions are stored in a training database together with other necessary resources for creating the training environment
- Employ KVM virtualization technology to enable the creation of multiple cyber range instances with the same content; a collection of base VM images is used for cyber range creation
- Make possible the use of off-the-shelf servers for cyber range deployment, thus eliminating the constraints of proprietary cyber range solutions
- Separate the cyber range creation process into three stages, base VM preparation, content installation in the base VM, followed by the guest VM cloning, so as to produce a set of cyber range instances with the same content

In Figure 1 we illustrate the overall workflow of CyRIS. The program starts by checking whether the input description file is syntactically and semantically correct. Then the *Preparation* stage for base VMs is initiated, with steps such as copying the disk image(s) from the base VM collection and starting them; finally, basic setup operations are conducted, such as setting up SSH access, host name, network connectivity, and so on.

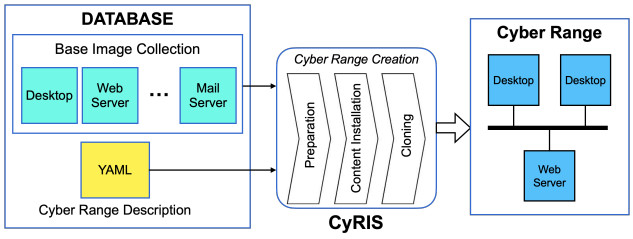

Figure 1. Overview of the KVM CyRIS workflow.

The second stage, *Content Installation*, consists first in performing all the setup tasks, such as installing content into the base VMs or emulating attacks; these steps are performed sequentially as described in the input file. CyRIS can also conduct an additional post-cloning setup step once the cloning process ends; this is essential for performing configurations that depend on the properties of each cloned VM, such as its IP address, etc.

The third stage, *Cloning*, refers to creating multiple identical cyber ranges that include guest VM clones of the prepared base VMs for several trainees to do the

same training simultaneously, and configuring their network topology. For this purpose, the configured base images are copied in parallel to all the hosts on which the cyber range instances are to be instantiated by using the `parallel-scp` command; if a cyber range instance contains multiple base images, then they are also copied in parallel. Once the cloned VMs are started, the user accounts and passwords for accessing the cyber range are randomly generated and those settings are applied.

3.3. AWS CyRIS Architecture

The functionality of the original KVM CyRIS, while rich in terms of cyber range creation features, is limited to local KVM cyber range deployment on on-premise physical hosts. Our goal is to extend its functionality to support cyber range deployment over the AWS EC2 public cloud by integrating this support into the existing CyRIS, as illustrated in Figure 2. The AWS API we implemented makes it possible to deploy cyber ranges in the AWS EC2 cloud transparently from a CyRIS user perspective.

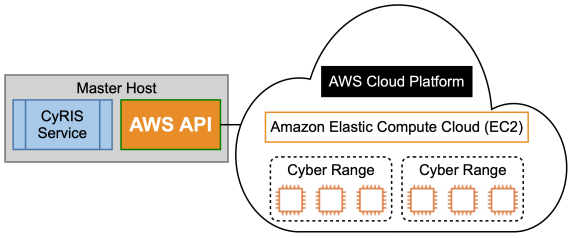

Figure 2. Overview of the extended AWS CyRIS.

Figure 3 shows the detailed processing flow of AWS CyRIS, and its operation is as follows:

1) A base EC2 instance is created first, so that security content can be installed into it before cloning; this stage also requires the creation of a dedicated security group for accessing that instance.
2) Once the base EC2 instance is created successfully, control is given to CyRIS, which will install the cyber range content according to the cyber range description file provided as input.
3) The base EC2 instance is then stopped, and a copy of it is replicated in the form of an Amazon Machine Image (AMI). Following the successful creation of the AMI, it is cloned as multiple EC2 instances that have the same content, which are the cyber range instances that will be accessed during training.
4) Finally, control is given again to CyRIS to conduct any post-cloning setup that may be required in the input cyber range description file. This includes the creation of user accounts to make subsequent access by trainees possible.

4. System Implementation

In this section we provide technical details about the AWS CyRIS implementation, with focus on the more challenging key points. Note that most of the functions below leverage the functionality of the Boto3 AWS SDK for Python provided by Amazon [24].

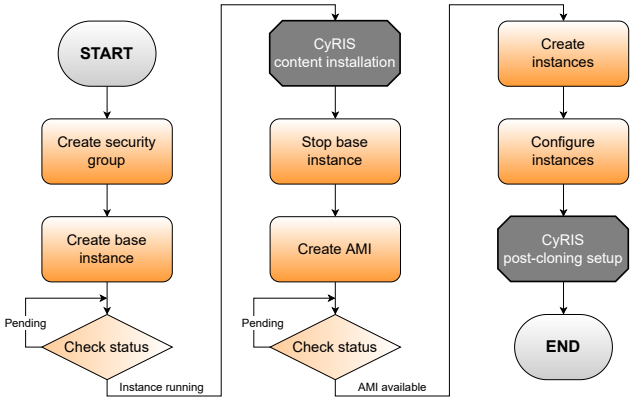

Figure 3. Detailed processing flow for AWS CyRIS.

4.1. Base EC2 Instance Creation

When using KVM CyRIS, base VM images need to be prepared in advance. However, Amazon EC2 already provides such built-in virtual computing environments, known as EC2 instances.

In order to be able to access these instances, a security group must be created first, which acts as a virtual firewall used to control incoming and outgoing traffic for a certain EC2 instance via inbound and outbound rules. In order to manage security groups, we use the two Boto3 functions named `create_security_group()` and `authorize_security_group_egress()` to create the group, then `describe_security_groups()` to check the group creation status.

Following security group creation, the Boto3 function `run_instances()` is subsequently used to create the EC2 instances, and `describe_instance_status()` to check when their creation was finalized. An important parameter for EC2 instance creation is the instance type, and AWS CyRIS makes it possible to provide this information via the cyber range description file. The current implementation supports the following instance types: Amazon Linux 1 and 2, Red Hat Enterprise Linux (RHEL) 8, and Ubuntu 16.04 LTS, 18.04 LTS and 20.04 LTS.

4.2. Content Installation

Although content installation is carried out using mostly original CyRIS code, some changes were required in order to enable AWS support, as summarized below.

In the new system users must specify in the AWS CyRIS cyber range description whether VMs should be created via KVM (default) or AWS. If AWS is specified, then the type of the AWS EC2 instance needs to be specified as well, such as `amazon_linux` designate an Amazon Linux 1 instance.

Internally, these choices also control the manner in which CyRIS connects to the created instances. For example, for KVM creation the `root` user is employed, but this is not possible for AWS, which assigns specific user names for each instance type, such as `ec2-user` for Amazon Linux and RHEL, and `ubuntu` for Ubuntu instances. Moreover, password-based login is disabled for EC2 instances, and SSH public-private key pairs need to be used instead for AWS cloud deployment.

4.3. AMI Instance Cloning

Amazon Machine Image (AMI) is an AWS EC2 cloud template for a fully-configured virtual machine, including operating system and any additional software.

To create such an AMI, AWS CyRIS first stops the base EC2 instance into which cybersecurity training content has been installed by using the Boto3 function `stop_instances()`. While AMIs can be created from running instances too, we decided to stop the instances out of an abundance of caution, to make sure the content is completely static. An AMI is then created based on the stopped fully-configured instance by calling the Boto3 function `create_image()`, and checking its creation status via the function `describe_image()`.

The key step of this stage follows, which is to clone the saved AMI into multiple fully-configured EC2 instances, one such instance for each cyber range copy that needs to be created. This is made possible via the Boto3 function `run_instances()`. Once cloning finishes, information about the created instances is retrieved by using the function `describe_instances()`. For example, the IP addresses of the cloned instances are retrieved for later use to automatically check connectivity to the cyber range before the information about the created cyber range is provided to the training organizers.

4.4. Running AWS CyRIS

The following are the prerequisites for running the extended AWS CyRIS (assuming no local cyber range deployment via KVM is required):

- A regular computer, even a low-performance laptop, on which AWS CyRIS and the required tools/libraries are installed (e.g., Python, Boto3)
- An AWS account and the necessary credentials to access the AWS EC2 public cloud
- A cyber range description file for the cyber range to be created, which uses the necessary keywords to denote the use of AWS: `aws` for the base VM type and, for example, `amazon_linux` for the base VM OS

When actually running CyRIS there is no difference between the KVM and AWS versions, as all the changes are transparent to the end user, except for setting the base VM type and OS, as mentioned above.

A specific script, named `aws_cleanup.py`, was added, which is to be run when the training ends. This script will terminate all the EC2 instances associated with a certain cyber range, as well as delete any associated security groups and AMIs.

Once a cyber range is created, access information is presented to the training organizers, who can then pass it on to trainees, as needed. The trainees can then use SSH commands to login in to the cyber range and conduct the training activity. Training organizers also have access to the AWS EC2 Dashboard management interface, which makes it possible to easily check the creation status, and manage instances manually if needed.

A screenshot of the AWS EC2 Dashboard showing four cyber ranges with two VMs each (`desktop` and `web`) is given in Figure 4. Note that the interface indicates for each instance its status (stopped or running), its public IPv4 address, the launch time, etc. Such information is useful for managing the instances, as actions such as restarting the VMs in case of troubles can be performed via the AWS EC2 dashboard.

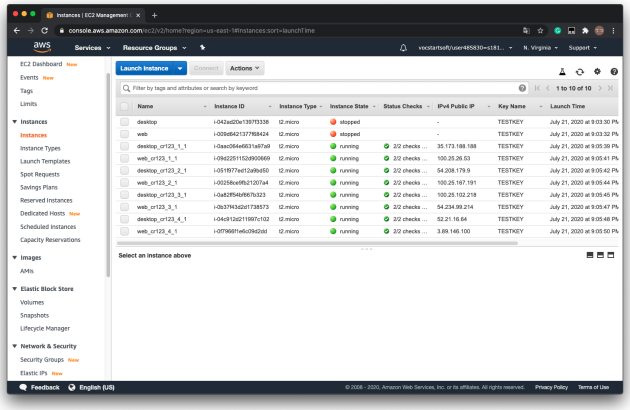

Figure 4. Screenshot of the EC2 Dashboard for AWS CyRIS.

5. Evaluation

In this section we evaluate AWS CyRIS from the point of view of the provided functionality, and assess its execution performance for several scenarios.

5.1. Functionality Assessment

Since AWS CyRIS is based on KVM CyRIS, it inherits the existing cyber range creation functions that, as it has been shown already in [18], are sufficient to create cybersecurity training environments for realistic scenarios, such as those derived from the NIST Technical Guide to Information Security Testing and Assessment [25].

However, there are more operating systems available in AWS EC2 than those supported by KVM CyRIS, so part of our effort was dedicated to ensuring compatibility with the new operating systems of EC2 instances.

We summarize in Table 1 the KVM CyRIS compatibility in terms of guest OS, as well as the improved OS support in AWS CyRIS. Amazon Linux 1 and 2 are the standard OSs provided on AWS, and we added support for them. We also added support for RHEL 8, and also for Ubuntu 20.04 LTS, so as to we have support for the most recent versions of these OSs.

The rows in Table 1 enumerate the key features provided by CyRIS. AWS CyRIS supports all these features for all the operating systems, with the exception of two features: “Emulate attacks” and “Emulate traffic capture files”. The first one is implemented in CyRIS by using OS commands to conduct an automated and controlled attack on a VM target, such as a dictionary attack using SSH. Since AWS prohibits this type of activities, we decided not to support the function. The second feature uses the first feature internally, saving the generated traffic as a PCAP file; due to its unavailability, the second feature is also not supported. Thus, security restrictions can be an important limitation of using public clouds for cyber range deployment, as it will be discussed in Section 6.

TABLE 1. OPERATING SYSTEM SUPPORT IN AWS CYRIS VERSUS KVM CYRIS

Content Installation	KVM CyRIS			AWS CyRIS		
	CentOS 7	*Ubuntu 16.04* *Ubuntu 18.04*	*Windows 7*	*Amazon Linux 1* *Amazon Linux 2*	*RHEL 8*	*Ubuntu 16.04* *Ubuntu 18.04* *Ubuntu 20.04*
Add accounts	✓	✓	✓	✓	✓	✓
Modify accounts	✓	✓	✓	✓	✓	✓
Install packages	✓	✓	✓	✓	✓	✓
Emulate attacks	✓					
Emulate traffic capture files	✓					
Emulate malware	✓			✓	✓	✓
Copy content	✓	✓	✓	✓	✓	✓
Execute programs	✓	✓	✓	✓	✓	✓
Modify firewall rules	✓			✓	✓	✓

5.2. Performance Assessment

One main concern regarding the use of public clouds for cyber range deployment is performance, given that the deployment is done remotely over the Internet, and by using an EC2 server infrastructure upon which the training organizer has no control.

5.2.1. AWS Performance. We first looked at the manner in which the cyber range creation time changes during the day by running AWS CyRIS ten times per hour for a 24-hour period. In order to eliminate other sources of variation, the cyber range in this case contained only one instance, with no specific content configuration tasks. The average creation times for each 1-hour interval, with error bars corresponding to the standard deviation, are shown in Figure 5 (note that the reference time is Japan Standard Time, JST). The EC2 region used in these experiments was `us-east-1`, which is the only region available for the AWS Educate account we employed.

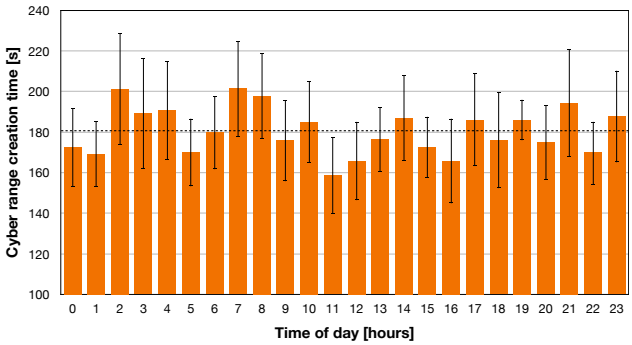

Figure 5. Cyber range creation time depending on the time of day.

While we could not distinguish any specific pattern, there is clearly a variation in the creation time depending on the time of day, as in some intervals the average creation time is lower that the global average of about 181 s (plotted with dotted line), and in some intervals higher (e.g., for time 2, 3, 4, etc.). Overall, however, the averages we measured were at most 22 s lower and 21 s higher than the global average, meaning a difference of about ±12%, which should not be of great concern for typical applications. If creation time is important, we suggest users evaluate the characteristics of their EC2 region, and try to choose as much as possible those time intervals that provide best access performance.

5.2.2. Single Range Creation. For the second set of experiments, we have compared the performance of AWS CyRIS to that of KVM CyRIS when creating a single cyber range in two scenarios:

- *Scenario #1*: The first scenario includes a single VM per trainee, playing the role of a desktop.
- *Scenario #2*: The second scenario includes 2 VMs per trainee, playing the roles of a desktop and a web server.

Figures 6 and 7 show the average results we obtained for AWS and KVM CyRIS for each of the two scenarios, respectively. For each experiment we conducted 10 runs, and the error bars represent the standard deviation. The AWS experiments were conducted in the same conditions described above, and for the KVM CyRIS experiments we used a server with two 4-core Intel Xeon E5504 2 GHz CPUs, 72 GB memory, a 400 GB HDD, and a 1 Gbps network interface.

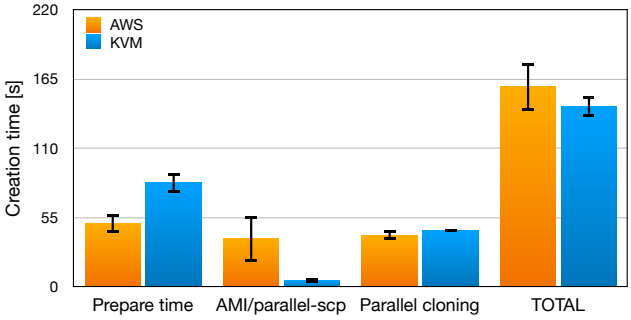

Figure 6. Single cyber range creation time for Scenario #1.

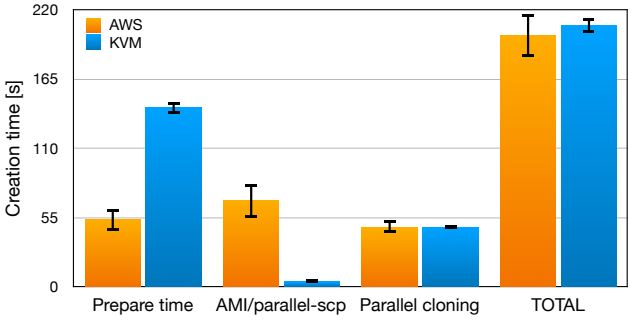

Figure 7. Single cyber range creation time for Scenario #2.

Cyber range creation time includes three main components, the time to prepare the base VM images, *Prepare time*, the time to save/copy the base VM images via AMI or `scp`, *AMI/parallel-scp*, and the time to create in parallel the VM clones, *Cloning time*, with the right-hand side bars in the two figures showing their sum, denoted by *TOTAL*.

The preparation time is shorter for AWS CyRIS due to the fact that built-in EC2 instances are available directly, without the need to copy the reference base VM image (as it is necessary for KVM CyRIS), which is an important advantage for the AWS solution.

As for the AMI creation for AWS CyRIS, and VM copying via `parallel-scp` for KVM CyRIS, we noticed that local VM copying is much faster than the AMI creation process on EC2. Another issue is that AMI creation happens sequentially when there are multiple VMs, evidenced by the almost double time shown for the 2 VM scenario in Figure 7 compared to the 1 VM scenario in Figure 6. On the other hand, `parallel-scp` effectively copies the two base VM images in parallel. AMI creation performance could be improved by parallel execution, but that requires using multi-core EC2 instances, which are more expensive.

For the final step, parallel cloning time is equivalent for both AWS and KVM CyRIS versions. We note that the total creation time for Scenario #1 is larger for AWS CyRIS compared to KVM CyRIS, and the opposite holds for Scenario #2, for which the prepare time difference is enough to offset the parallel AMI time gap. The overall difference, however, is only of about $+11\%$ and -4%, respectively, which we consider acceptable and an indication of equivalent performance for the two systems in this series of experiments.

5.2.3. Multiple Range Creation. Next we compared the performance of AWS and KVM CyRIS for the case when multiple cyber ranges of one type are created, e.g., for multiple trainees. The same two scenarios as above were used, but this time multiple cyber ranges were created for each of them. Note that the total number of VM instances that could be created on the EC2 cloud was capped to a maximum of 9 due to the settings of the AWS Educate account used in the experiments.

Figures 8 and 9 show the average results we obtained for a series of 10 runs per experiments, with error bars representing the standard deviation. All experiments were conducted in the same conditions as described already.

The first figure demonstrates that for the 1 VM scenario (Scenario #1), AWS still has lower performance than KVM even as the number of cloned VMs increases, with a creation time higher from about 11% to 9%, the smallest difference being observed for the case of 9 VMs. On the other hand, for the 2 VM scenario (Scenario #2), AWS has again higher performance than KVM, with a creation time lower from about 4% to 10%, the largest difference being observed for the case of 8 VMs (i.e., 4 cyber ranges with 2 VMs each).

While the standard deviation for AWS is clearly larger than that for KVM, we note that the difference in average performance changes favorably with the increase of the number of cloned machines, decreasing for Scenario #1, and increasing for Scenario #2 towards the right-hand side of the graph. Therefore, we conclude that overall the AWS

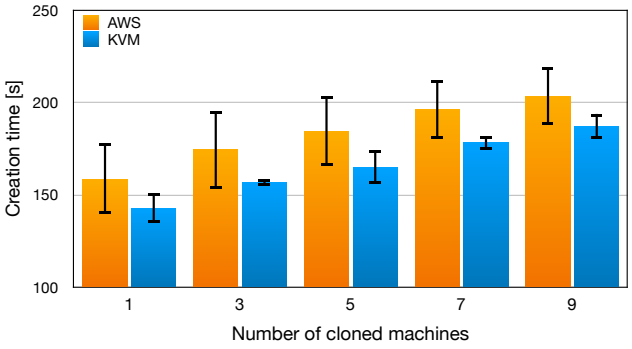

Figure 8. Multiple cyber range creation time for Scenario #1.

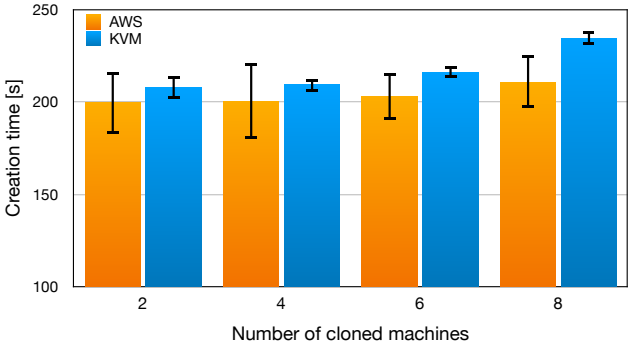

Figure 9. Multiple cyber range creation time for Scenario #2.

CyRIS performance is comparable to that of KVM CyRIS, and certain improvements are noticeable as the scale of the experiment increases.

6. Discussion

In this section we first compare local server and public cloud cyber range deployments, then discuss other aspects relevant to this research.

6.1. Local Server vs. Public Cloud Deployment

Our experience with both local server (KVM) and public cloud (AWS) cyber range deployment makes it possible to analyze the advantages and disadvantages of each solution. This analysis is summarized in Table 2.

TABLE 2. COMPARISON OF KVM VERSUS AWS CYBER RANGE DEPLOYMENT

Task/Function	KVM	AWS
Must prepare base VMs	✓	
Must copy base VMs	✓	
No base VM customization		✓
Base VM OS flexibility		✓
Fast/reliable access	✓	
Security restrictions		✓
Resource scalability		✓
Management overhead	✓	

The first part of the table refers to the way in which base VMs are managed. For KVM CyRIS, they must be prepared in advance, which represents a potential labor cost. Moreover, base VMs must be copied before content

installation so as to avoid their alteration, which introduces a time overhead. While these are obvious disadvantages, the fact that training organizers can manually customize and finely tune the prepared base VMs in advance—which is not possible when using the built-in EC2 instances—is a clear advantage to be considered.

Note that EC2 content customization is nevertheless possible, but it must be done at run-time, e.g., via scripts, which may be tedious to prepare and increase the EC2 instance preparation time. Finally, the flexibility of the large number of base VM OS types provided in EC2, which can be used without any additional effort, is an advantage for AWS.

The second part of the table refers to those aspects that are related to the public cloud nature of AWS EC2. KVM VMs deployed on local servers offer fast and reliable access for trainees, whereas the public cloud is always accessed remotely, and outages that are not under the control of the organizers are possible (while such outages are not frequent, their influence could be serious if the cyber range is used for an exam, for instance).

As for the security restrictions of public clouds, they can clearly hinder the use of certain security features and content, such as the attack emulation features of CyRIS. A related aspect is that some training activities, such as those conducted by national authorities, may require the use of confidential information and scenarios. Whether such activities can be conducted via public clouds is a case-by-case decision that must be taken by training organizers. An alternative is to use private cloud deployments that have similar advantages to public clouds, but can be controlled more strictly if such control is needed.

An advantage of public cloud deployment is that it provides a flexible and scalable solution when the number of trainees varies, both in terms of performance and the number of VMs, with the only possible limitations being related to the AWS account type and total cost restrictions. Moreover, there is no management overhead related to the administration of the AWS servers, as in the case of local server deployment.

Overall we can say that both local server and public cloud cyber range deployments have each their advantages and disadvantages, but in general the flexibility, convenience and ease of use of public cloud solutions make them strong candidates for training activities. We note that private cloud deployments are a solution that combines some of the advantages of local server and public cloud deployments, but given the purchase cost and management overhead associated with them, we see such private cloud solutions as being closer to the KVM case.

6.2. Other Aspects

So far we have discussed the advantages and disadvantages of public cloud cyber range deployment mainly from the perspective of training organizers. When considering scenario designers, the use of CyRIS is basically the same independently of the type of deployment (KVM or AWS). In some cases, however, the designers may have to find workarounds for certain AWS-related constraints, such as using real-time scripts instead of the manual configuration of base VMs, which is not possible for EC2 instances. As for the trainee perspective, the use of the two solutions is completely transparent, but the increased round-trip delay for AWS may potentially lead to a decrease in the perceived Quality of Experience (QoE).

Secondly, while it may seem that public cloud deployment has functionality limitations compared to local server deployment, our experience has shown that solutions are available for these apparent limitations. For example, after making sure that the tools employed are correctly configured and the scope of their effects is limited to the cyber range, cyber attacks can be conducted freely inside a single VM; for attacks between VMs, it is simply that an approval from the public cloud provider must be obtained in advance. Therefore, we conclude there are no major limitations in terms of the skills that the training conducted via public cloud cyber range can address, as long as the organizers have in place the right strategy and risk mitigation policies.

We return now to the issue that for any public cloud infrastructure it is impossible to predict exactly the network conditions and remote server load at any given time. However, our experiments—including the results presented in Section 5.2, and in particular in Section 5.2.1—demonstrate that despite the observed variation one can establish lower and upper expected performance bounds via repeated measurements. While our results may be considered as anecdotal evidence for the country of origin (Japan) and the target EC2 region (`us-east-1`) in our experiments, and the average values would certainly vary for a different combination of country of origin and target server region, the methodology we presented can be used by anyone to determine expected performance bounds, and to decide whether their performance requirements are met or not. We also note that, as opposed to the free AWS Educate account we used, paid AWS accounts offer the possibility to chose the target EC2 region, and in principle selecting a region that is in the country of origin (or as close as possible to it in terms of Internet access) should provide faster and more stable performance.

7. Conclusion

Public cloud deployment is a way to enable scalable cybersecurity training activities, while avoiding the fixed costs associated to the purchase and management of the server infrastructure needed for cyber ranges. By their nature, public clouds such as AWS EC2 make it possible to scale up or down the training activities as needed, for example for occasional training session, classes with a variable number of students, and so on.

In this paper we have presented an architecture and implementation that extended the functionality of the cyber range instantiation system CyRIS, which originally used only KVM technology for cyber range creation, to support deployment on the AWS EC2 public cloud infrastructure. The implementation was facilitated by the modular nature of CyRIS, and the possibility to leverage the Boto3 AWS SDK for Python from Amazon. The implementation was fully integrated with CyRIS and the CyTrONE framework, and was already released on GitHub as an upgraded version of CyRIS [3].

The functionality evaluation showed that our implementation, AWS CyRIS, successfully added support for the built-in OS types available in EC2, and the only two

features that are not currently supported for public cloud cyber range deployment are those that conflict with the default security restrictions of AWS. We also evaluated AWS CyRIS from a performance perspective, and demonstrated that it has performance characteristics equivalent to those of KVM CyRIS. Thus, in our experiments the performance was at most 11% slower for AWS compared to KVM, and at best 10% faster. In addition, our experience with both local server and public cloud deployments made it possible to discuss in detail the advantages and disadvantages of each of these two solutions.

As future work we are considering exploring the possibility to optimize AWS CyRIS performance to improve cyber range deployment time, especially for large-scale deployments. Secondly, we want to investigate solutions for performing emulated attacks that are not in conflict with the intrinsic security restrictions of AWS EC2.

References

[1] Amazon Web Services (AWS), "Amazon EC2: Secure and resizable compute capacity for virtually any workload," https://aws.amazon.com/ec2/, 2022.

[2] R. Beuran, C. Pham, D. Tang, K. Chinen, Y. Tan, and Y. Shinoda, "Cybersecurity Education and Training Support System: CyRIS," *IEICE Transactions on Information and Systems*, vol. E101-D, no. 3, pp. 740–749, 2018.

[3] Cyber Range Organisation and Design (CROND), "CyRIS GitHub page," https://github.com/crond-jaist/cyris, 2020.

[4] M. M. Yamin, B. Katt, and V. Gkioulos, "Cyber ranges and security testbeds: Scenarios, functions, tools and architecture," *Computers & Security*, vol. 88, p. 101636, 2020.

[5] Facebook, Inc., "Platform to host Capture the Flag competitions," https://github.com/facebook/fbctf/, 2017.

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

[7] Docker, Inc., "Swarm mode overview," https://docs.docker.com/engine/swarm/, 2022.

[8] "DEF CON Hacker Convention," https://www.defcon.org/.

[9] Japan Network Security Association (JNSA), "Security Contest (SECCON)," (in Japanese), https://seccon.jp/.

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

[11] ——, "NetWars Continuous Online Skill-Sharpening Range," https://www.sans.org/netwars/continuous.

[12] Web Application Security Forum, "Hardening Project," (in Japanese), https://wasforum.jp/hardening-project/.

[13] H. Kawaguchi (organizer), "Micro Hardening," (in Japanese), https://microhardening.connpass.com/.

[14] Boeing Media Room, "Boeing offers improved cybersecurity training and simulation tool," https://boeing.mediaroom.com/Boeing-Offers-Improved-Cybersecurity-Training-and-Simulation-Tool, 2013.

[15] J. Vykopal, P. Celeda, P. Seda, V. Svabensky, and D. Tovarnak, "Scalable learning environments for teaching cybersecurity hands-on," in *Proceedings of the 2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–9.

[16] The OpenStack Foundation, "OpenStack – Open Source Cloud Computing Software," https://www.openstack.org/, 2017.

[17] R. S. Weiss, S. Boesen, J. F. Sullivan, M. E. Locasto, J. Mache, and E. Nilsen, "Teaching cybersecurity analysis skills in the cloud," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE'15)*, 2015, pp. 332–337.

[18] R. Beuran, D. Tang, C. Pham, K. Chinen, Y. Tan, and Y. Shinoda, "Integrated framework for hands-on cybersecurity training: CyTrONE," *Computers & Security*, vol. 78C, pp. 43–59, 2018.

[19] The Moodle Project, "Moodle open-source learning platform," https://moodle.org/, 2021.

[20] G. Subasu, L. Rosu, and I. Badoi, "Modeling and simulation architecture for training in cyber defence education," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2017, pp. 1–4.

[21] S. Yasuda, R. Miura, S. Ohta, Y. Takano, and T. Miyachi, "Alfons: A Mimetic Network Environment Construction System," in *Proceedings of the 11th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom 2016)*, 2016.

[22] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab, "The DETER project: Advancing the science of cyber security experimentation and test," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*, 2010, pp. 1–7.

[23] P.-W. Tsai and C.-S. Yang, "Testbed@TWISC: A network security experiment platform," *International Journal of Communication Systems*, vol. 31, no. 2, p. e3446, 2018.

[24] Amazon Web Services (AWS), "Boto3 documentation," https://boto3.amazonaws.com/v1/documentation/api/latest/index.html, 2022.

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