On-demand deployment and orchestration of Cyber Ranges in the Cloud

Alessandro Placido Luise[a], Gaetano Perrone[a], Claudio Perrotta[b] and Simon Pietro Romano[a]

[a]*University of Napoli Federico II, Department of Electrical Engineering and Information Technology, Via Claudio 21, 80125 Napoli, Italy.*

[b]*Epsilon SRL, Napoli, Italy*

Abstract

In this paper we present a framework for the dynamic deployment, configuration and orchestration of cyber ranges in a cloud-based environment. We propose a distributed architecture that is composed of a number of interacting components, each looking after a specific facet of the integrated set of requirements coming out of the design phase. The architecture in question is indeed capable of offering environment isolation, remote access management and control, procedures automation, secure operation and accountability. A formal description of the concept of a cyber range is provided in the paper, together with a taxonomy associated with the different kinds of resources it can involve. A complete implementation of the proposed framework through Amazon Web Services is also illustrated, so to help the reader figure out how the overall design can be easily mapped onto a specific provider of cloud resources.

Keywords

Cyber Ranges, Cloud architectures, Dynamic deployment, Orchestration, Security

1. Introduction

The aim of cybersecurity is to provide defensive measures against cyber attacks that could disrupt the proper functioning of a complex Information System. The consequences of a clever plan of attack depend on the severity of the vulnerabilities that a criminal can exploit in order to perform unlawful operations. Security experts are responsible for identifying such vulnerabilities and implementing defensive mechanisms to avoid, or in severe cases minimize, damages.

The best way to defend against threats is to acquire hacking skills and be able to launch effective attacks with the aim of learning from the attacker's perspective and consequently strengthening the overall architecture. Doing so in an uncontrolled environment is obviously illegal and can lead in some cases to falling in trouble with the law. Future professionals require training platforms that provide deliberately vulnerable simulation and emulation environments in order to acquire penetration testing skills. A "Cyber Range" is a virtual environment for

ITASEC 2021: Italian Conference on CyberSecurity April 07–09, 2021, Online

alessandropluise@gmail.com (A. P. Luise); gaetano.perrone@unina.it (G. Perrone); claudio.perrotta@epsilonline.com (C. Perrotta); spromano@unina.it (S. P. Romano)

cybersecurity testing designed for learning and education, whose architecture relies on tools like, e.g., virtual machines, docker containers and virtual networks.

Over the years, cyber ranges have started to play a central role in education and training of security experts. State organizations made substantial economic investments to create cyber ranges with the aim of military training to cyber warfare. Cyber ranges are evolving in more complex environments and simulations are getting closer and closer to real world scenarios. These innovations bring with them major technical challenges and complex problems to solve. To date, little has been done to automate the process of creation of cyber security training environments. Moreover, cyber ranges are recently evolving into service-based solutions. Remote Access Control, Credential Management, Automation and Security are the aspects that nowadays require innovation in this field and this is where our work focuses on. We propose a design of a services-based platform for the dynamic deployment of cyber ranges, based on containers and virtual machines. The goal is to manage the creation and termination of training environments in a dynamic, scalable and secure way. Moreover, the architecture satisfies the property of portability, which allows to consider its implementation with different cloud providers. In our case it was decided to use the cloud-computing platform offered by Amazon (Amazon Web Services — AWS), which is cheap, highly reliable and configurable. EC2 (*Elastic Computing Cloud*) instances are Virtual Machines configured with Amazon Machine Images (AMI), a type of virtual appliance that runs on top of a hypervisor. They play a central role in our work, as they are responsible for hosting Docker Security Playground (DSP) [1], a microservices-based framework whose architecture is based on docker[1] and docker-compose[2].

Currently, DSP can be used only by a single user. The proposed solution allows to implement DSP collaborative laboratories that involve interaction between *red* (i.e., the attackers) and *blue* (i.e., the defenders) teams through a shared environment. However, EC2 Instances do not only act as a mere "host" for DSP. They are also an active entity in the reproduction of cyber security scenarios, since they can run several different Operating Systems, hence allowing for the creation of complex and diverse virtual and hybrid environments. In such a scenario, environments separation and access control become key to the success of the cyber range. A user shall be able to access only resources that were assigned to him/her by the system. Moreover, these resources must be isolated from the outside, so to prevent internet access and protect the environment from known attacks to cloud infrastructures such as cryptojacking. Scenarios range among hacking web applications, cracking telnet access, buffer overflow, WiFi hacking, and many others. Users have access to a collection of multidisciplinary laboratories, which brings many benefits from an educational point of view. Much has been done also to automate creation and management of resources, ensuring a dynamic and reliable environment.

The paper is divided in five sections. We start by analyzing related work in Section II, to better study the recent advances on cyber range technologies, as well as to identify areas for improvement. Section III describes how we designed an architecture for the dynamic deployment of training environments. In the same section, we also propose solutions for overcoming current cyber range limitations. An implementation of the proposed architecture based on Amazon Web Services (AWS) is discussed in section IV. Section V summarizes the main contributions of

[1] https://www.docker.com/

[2] https://docs.docker.com/compose/

the work and describes directions of future work.

2. Related work

The authors of [2] describe the current limitations of cyber ranges and make remarks on those that can be future improvement areas. Credential management is considered one of the aspects to focus on. They also observe that a cyber range must take advantage of automation mechanisms for dynamic configuration of resources. Another important observation is made on remote access management, which means ensuring adequate and available remote access to various subsystems the end users.

In [3] the concept of *cyber arena* is introduced with its high level requirements. The authors remark that, to obtain better results from an educational point of view, cyber ranges must evolve in order to represent systems and technologies on a wider scale. A cyber arena must simulate realistic scenarios, reflecting real cyber domain complexity, along with Internet and internal network traffic. The authors' future goal is to provide more specific technical requirements for this cyber range evolution and implement state of the art training exercises.

The work presented in [4] makes uses of cyber ranges to improve what they call *cyber defense situational awareness*, both at the individual and the community level. First of all, the authors describe mental models and situation awareness levels, namely perception, comprehension and projection. Then, they propose a cyber range architecture composed of a remote desktop gateway, to enable remote access to users, a hypervisor, responsible for running the virtual machines, and an orchestrator responsible for deploying and customizing virtual machines. The authors plan to improve their architecture to support VMware and/or Hyper-V, as well as connect and interoperate with other cyber ranges.

Cyber ranges can be used in many different contexts. As an example, the framework presented in [5] proposes a three phases process aimed at preparing roles for EXCON (EXercise CONtrol) teams. The authors' idea is to enable full scaled cyber-incident exercises.

The authors of [6] propose a cyber range for power industry, since power grids (or, more generally, energy distribution systems) are potential targets in cyber warfare. They implement a service-oriented resource management framework using OpenStack[3]. The physical architecture is divided into a Management Network and a Business Network. The business network makes use of a firewall to manage user access to the service. They also implement an evaluation component which performs operations such as load balancing, information logging and health monitoring. They focus their attention also on the security challenges to overcome, one of which is separation between virtual resources and the Internet.

Development methodologies are instead the main focus of [7]. The authors describe a testbed design life cycle and propose a running example whose implementation is based on OpenStack.

Besides the research literature, many examples of security training tools can be cited. OS-level virtualization products, like Docker, deliver software in packages called containers. A lot of vulnerable web applications and network security tools running on containers are available on the Internet. Thanks to tools like *docker-compose*, it is possible to bring these containers together and create virtual cyber ranges of different complexity levels. Docker Security Playground [1] is

[3] https://www.openstack.org/

a microservices-based framework that makes extensive use of both docker and docker-compose to experiment with security scenarios (virtual cyber ranges). It also lets users create their own training laboratories. The possibility of customizing docker containers in many different ways allows to emulate very heterogeneous environments without the cost of a complete virtualization. This, as we will further discuss in the next section, does represent an advantage for the design of the architecture we propose in this paper.

3. System architecture

In this section we show how the overall system is designed to meet the following requirements :

- **Environment isolation and Remote Access Management**. Every user must be able to practice hacking within an isolated environment, that does not allow intrusion of unauthorized entities whose action could undermine the proper functioning of the scenario. To achieve environment isolation, scenarios are divided into smaller entities separated from each other. All these entities will reside in the same virtual environment called *Cyber Range Environment*. A component called Remote Access Controller directs user's traffic from a given source to the correct destination.
- **Automation**. The system must make use of automation mechanisms for the configuration of cybersecurity scenarios. A scenario that has been requested by a user must be available in a few minutes and must not involve manual configuration from a system administrator. Virtual Machines, networks and other virtual resources are pre-built and instantiated by the Back-end Resource Manager.
- **Security**. The system shall be able to perform real-time inspections of all the activities within the Cyber Range Environment. The goal is to detect anomalous situations like the creation of more resources than allowed or unauthorized access to a laboratory belonging to another user. The Cluster Security Controller is responsible for processing events occurring in real-time in the Cyber Range Environment and takes automated defense and mitigation actions in case of anomalies.
- **Accountability**. A component called Credential Manager is responsible for identifying a single user and determining actions within the system. All users' actions need to be tracked down, both to allow the Cluster Security Controller to apply the right defence measures and to update the current state of the Cyber Range Environment.

The diagram in Fig. 1 illustrates the main components of the system, together with their mutual interactions. We proceed in the following with a more detailed description of the components and their operations.

3.1. Cyber Range Environment

The Cyber Range Environment is a set of virtual machines that are logically divided into multiple subsets called, respectively, "Macro Ranges" and "Micro Ranges". A Macro Range *Mr* is a subset of the Cyber Range environment C that contains one and only one Remote Access Controller r. The Remote Access Controller is configured with a set of routing rules obliging users to gain

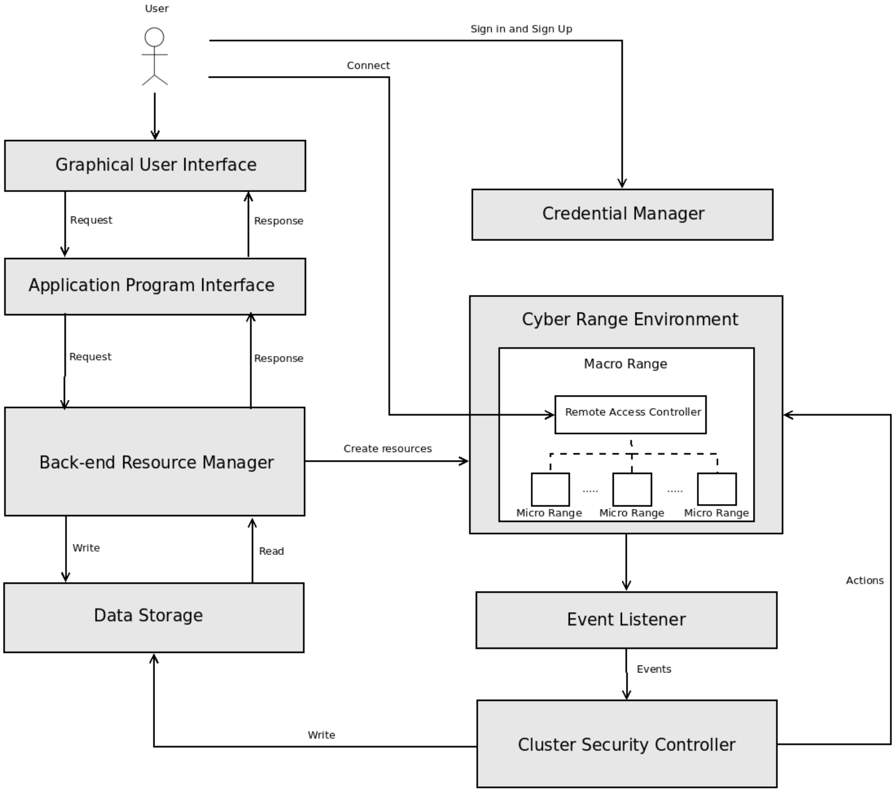


Figure 1: System Architecture

access exclusively to the assigned virtual resources. These rules are defined inside the Remote Access Controller and cannot be modified by unauthorized entities.

$$Mr \subseteq C$$
$$\exists! \quad r \in Mr$$

A Macro Range contains zero or more Micro Ranges. Micro Ranges are a non empty set of virtual machines whose usage is predisposed for either a single user or a group of users.

$$n(mr) \in \{1, 2...M\}$$

Where M is a positive integer representing the maximum number of eligible Virtual Machines per Micro Range. All the Micro Ranges in a Macro Range must be linked to the Remote Access Controller through a communication channel. A virtual machine cannot establish in any way a communication channel with a virtual machine belonging to a different Micro Range.

A virtual machine *u* is *reachable* from a virtual machine *v* if a direct communication channel can be established between them.

$$v \to u$$

At least one element of a Micro Range must be *reachable* from the Remote Access Controller.

$$\forall \quad mr \subseteq Mr \quad \exists \quad v \in mr : r \to v$$

Elements of different Micro Ranges must not be able to reach each other.

The same rule applies to elements belonging to different Macro Ranges. Fig. 2 shows an example.

Let Q and S be two Micro Ranges with v and u virtual machines belonging to Q and S respectively.

$$\nexists \quad v \in Q \quad : v \to u \quad \forall v \in Q, \forall u \in S$$

$$\nexists \quad u \in S \quad : u \to v \quad \forall v \in Q, \forall u \in S$$

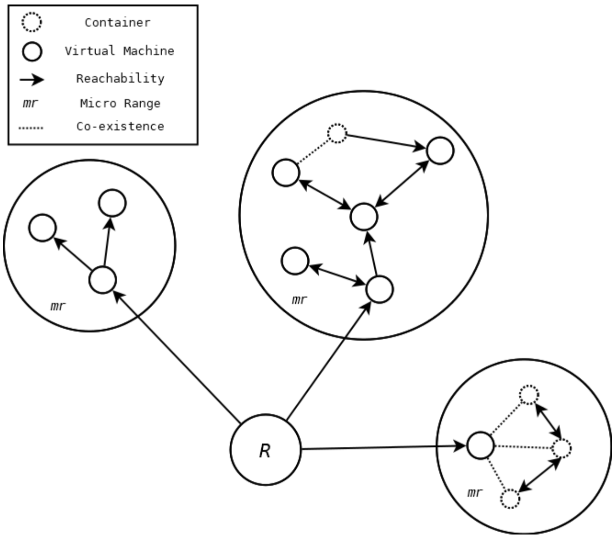


Figure 2: Representation of a Macro Range

Micro Ranges are divided into multiple categories depending on the type of resources they are composed of:

- *Virtual Micro Range*: is a Micro Range composed only of full virtualization VMs connected to each other;
- *Containerized Micro Range*: is a Micro Range composed of a single VM whose kernel serves as the base for the execution of containers (Docker, Linux Containers, Solaris);

- *Hybrid Micro Range*: is a Micro Range composed of both full virtualization VMs and one or more VM hosting containerized infrastructures. Container Networks (based, e.g., on *docker-compose*) and other VMs are connected to each other forming, de facto, a hybrid system;
- *Shared Micro Range*: this type of Micro Range can be accessed by several users at the same time. A shared Micro Range is designed to simulate red/blue team scenarios that allow interaction between attackers and defenders.

3.2. Back-end Resource Manager

The Back-end Resource Manager is responsible for:

- *Resource allocation*
  Users send resource allocation requests through the front-end application and the API. The Back-end Resource Manager receives such requests and checks if all the required preconditions are met, such as eligibility and accountability of the user who made the request. In case of laboratory creation, it checks if there are sufficient resources for allocating a new Micro Range in any Macro Range inside the Cyber Range Environment. If the maximum number of Micro Ranges per Macro Range has been reached it will proceed to create a new Macro Range, including a Remote Access Controller.
- *Routing rules declaration*
  When a new virtual machine is created, or when the ownership of a Micro Range passes to another user, the Back-end Resource Manager launches commands inside the Remote Access Controller to enable the new rules. A command is sent in order to establish a communication channel between the user who made the request and the assigned Micro Range. In case of transfer of ownership to another user, the existing rules are modified. If the system needs to revoke a user's access to a Micro Range, the Back-end Resource Manager sends a delete command related to the rule that allows communication between such two entities.
  The Back-End Resource Manager needs to be aware of the state of the Cyber Range Environment before setting up a communication channel between a user and a Micro Range. Before editing the virtual space, it reads the current state from the Data Storage and updates it every time a successful action on the Cyber Range Environment is performed. This state control mechanism prevents overlap of resource allocation or address assignment.

3.3. Cluster Security Controller

The Cluster Security Controller is responsible for interpreting events coming from the Event Listener and performing security checks and actions on the Cyber Range Environment. Security checks include verifying if there are anomalies in the allocation of the instances, for example if a user manages to create more resources than expected. The event record also includes information about the Micro Range owners. Lets suppose that a user is authorized to have access to only one Containerized Micro Range and, intentionally or due to a system bug, manages to get ownership

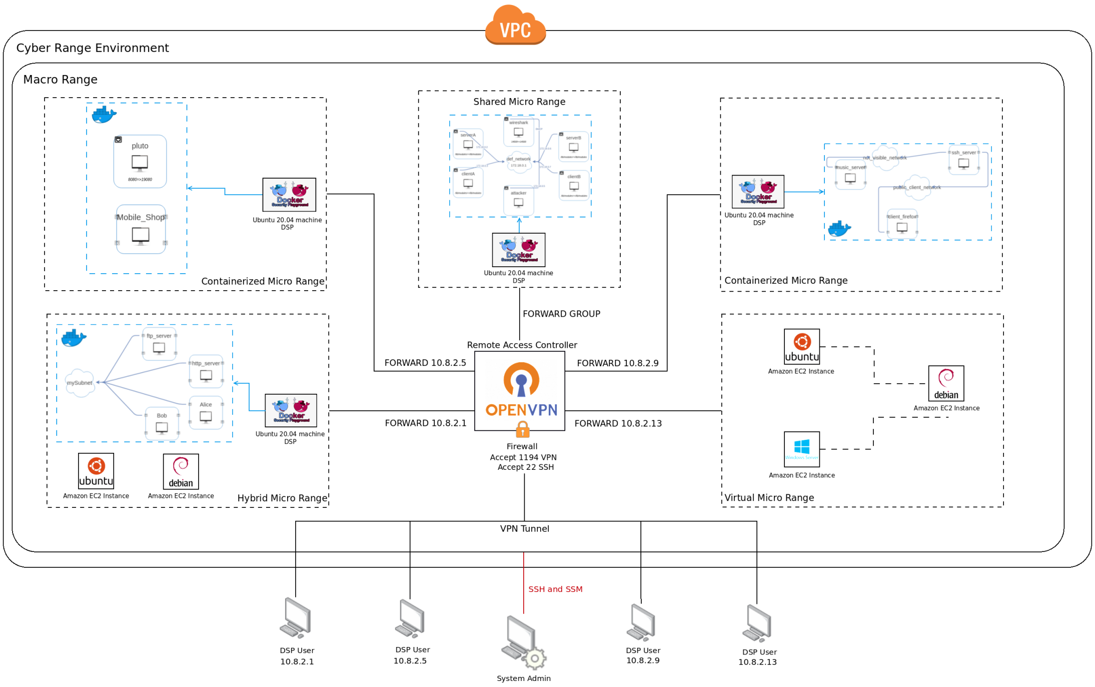


Figure 3: Cyber Range Environment implemented with AWS

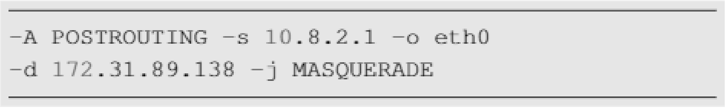
of an additional Micro Range of the same type. The Cluster Security Controller is notified about this event and, after performing security checks, detects the anomaly and destroys the resources. The Back-end resource Manager tries to prevent this type of situations by performing controls before the resource allocation phase. Though, it does not check the environment in real time, differently from the Cluster Security Controller. The Cluster Security Controller, in fact, updates the state of the overall system by updating the Data Storage if the performed actions modify resources or communication channels in the Cyber Range Environment. The automated controls performed on the active resources in the Cyber Range Environment constitute an additional security layer which is crucial for architectures of such complexity.

3.4. Credential Manager

Accountability is one of the most important aspects in this type of systems. The Credential Manager is responsible for providing authentication and access control services to the users, who are in turn identified by unique attributes like username, email or phone number. Additional security attributes are required to set limits to the resources that can be requested and keep track of payments history.

4. Implementation

We show a possible implementation of the proposed architecture using Amazon Web Services. A potential scenario is shown in Fig. 3. There are four users, each with an assigned Micro Range. All four users are assigned to one Shared Micro Range that hosts a red/blue team laboratory. The Cyber Range Environment is implemented with a Virtual Private Cloud (VPC), i.e., a virtual network that allows to dynamically activate cloud resources in a controlled fashion. A Macro Range is implemented by allocating a private subnet inside the VPC. This subnet contains virtual machines, created as Elastic Compute Cloud (EC2) Instances. EC2 Instances are virtual machines created with a special type of virtual appliance called Amazon Machine Image (AMI). There are plenty of available Operating Systems in AWS, ranging from Linux distributions (such as Ubuntu, Debian, Kali, Fedora) to different versions of Windows Server. The Remote Access Controller has been configured trough a Ubuntu 20.04 EC2 instance running an OpenVPN server. The Server is responsible for connecting users to the subnet and forwarding packets to correct destinations through the policies and NAT rules described in a firewall local to the instance. An example of NAT rule is shown in Fig. 4.



```
-A POSTROUTING -s 10.8.2.1 -o eth0
-d 172.31.89.138 -j MASQUERADE
```

Figure 4: NAT Rules

This rule means that all packets coming from the user whose virtual IP address is 10.8.2.1 will be forwarded to the assigned virtual machine, belonging to a Micro Range, whose private IP address is 172.31.89.138. POSTROUTING means that this rule must be applied on packets that are leaving the OpenVPN Server. In this way the user can access the services provided by the EC2 instance. The local firewall must leave ports 1194 and 22 opened, respectively to allow users to connect to the OpenVPN server and to allow administrators access to the server instance itself through a secure shell.

Containerized Micro Ranges are implemented with a Ubuntu 20.04 EC2 instance running the Docker Security Playground [1]. Some types of containers run vulnerable web applications that are accessible trough specific ports. Users have access to all available ports in their personal instance and in this way are able to launch hacking tools installed both in their local machine (the one used to connect to the OpenVPN Server) and local to the EC2 instance. In case of more complex scenarios requiring access to containers, the user can take control of the shell of the instance trough a secure SSH tunnel.

Granting shell access to users could expose the entire architecture to attacks and illicit actions. A user may perform network scanning and interfere with the correct functioning of Cyber Ranges assigned to other users. The use of these tools is only allowed inside the assigned Micro Range. AWS allows us to equip EC2 instances with a virtual firewall called *Security Group*, that ensures environment separation. A Security Group is set to accept or deny packets coming from specific private IP addresses of the subnet. In case of Containerized Micro Ranges the Security Group of the instance that hosts containers accepts incoming packets that have as source the private IP address of the Remote Access Controller (OpenVPN Server in our case), which acts as a relay for user traffic. In case of other types of Micro Range, this depends on the type of virtual scenario that has to be simulated. One might want to allow incoming and outgoing traffic to a vulnerable EC2 instance and, after the user has taken control, allow him/her to reach another instance of the Micro Range. The other VM will have a Security Group set up to allow reachability only from the hacked EC2 instance.

The use case to implement is the following: a user sends a request for a trial Containerized Micro Range of 30 minutes duration and composed of a virtual machine running the Docker Security Playground. A user can launch only one trial Micro Range per account and after 30 minutes all the allocated resources are terminated by the system.

The role of the Credential Manager is performed by *Cognito*, an AWS service providing authentication, authorization and user management. When a user registers to the service, their personal data will be saved in a Cognito user pool. Access to the API for incoming requests is regulated by a *Lambda* authorizer that fulfills the task of validating JSON Web Tokens (JWT) in the request header and checking whether a user has already sent a request for a trial Docker Security Playground Micro Range or not. If the JWT token is correctly validated, the API adds the requester's username to the body field of the request and passes it to the back-end. If the user made the request for the first time, the Lambda Authorizer sets a Cognito attribute for that user as 'false', hence indicating that he/she cannot send another request for a trial instance.

The component Back-end Resource Manager is implemented in Python3 with a Lambda Function that performs the following operations:

- **Eligibility check**: before allocating resources the Lambda function checks if the request is valid and the user exists.
- **Availability check**: every Macro Range has a maximum number of users and Micro Ranges that it can handle. The Lambda function checks if a Macro Range exists that can host the new Docker Security Playground EC2 instance. If all Macro Ranges are occupied, the Lambda function generates a new one within the VPC (Cyber Range Environment), by creating a new subnet and a new OpenVPN server instance.
- **Virtual IP and ClientID generation**: once the OpenVPN server has been selected, the Lambda function assigns a virtual static IP address to the user. The assignment operation must take into account the already assigned virtual addresses. To avoid assignment overlaps, the Lambda function reads the current Cyber Range Environment state from the Data Storage. Once the virtual static IP address has been randomly generated, it is marked as 'assigned' and the Cyber Range Environment state is updated.
- **EC2 instance allocation**: the Lambda function creates the EC2 instance using a custom Ubuntu 20.04 Amazon Machine Image with Docker Security Playground installed. The EC2 instance is tagged with an ad hoc value indicating that it is a trial instance and must be terminated after 30 minutes.
- **Commands execution on the OpenVPN Server**: the EC2 instance has a unique private IP address within the subnet that needs to be assigned to the virtual static IP address of the user with the routing rules previously discussed (Fig. 4). The Lambda function sends configuration commands to the OpenVPN Server through the AWS System Manager, so to enable the iptables routing rule.

The Cluster Security Controller is implemented through AWS Step Function, a function orchestrator that allows for sequential execution of either Lambda functions or other AWS Services. The Step Function execution is triggered by AWS Cloudwatch, a cloud resources monitoring service acting as the Event Listener. When a new instance is allocated or changes its state (for example from Stopped to Running) within the VPC, Cloudwatch sends an event description in JSON format, which is processed by the Step Function. The Step Function invokes a Lambda Function that carries out security controls, including checking the tags of the instance. If the tag 'IsTrial' of the instance is set to 'true', the next step is to terminate the instance after 30 minutes, by invoking a Lambda Function that fulfills the purpose. The JSON event passed as input of the Lambda Function includes all tags associated with the instance to be terminated. After the DSP trial instance is terminated, the Step function updates the Cyber Range Environment state within the Data Storage by marking the virtual static IP address of the user as free, so that it can be assigned to another user. Moreover, the routing rules are deleted from the OpenVPN Server, once again by launching commands through the AWS System Manager.

5. Conclusions

In this paper we have discussed how we created a platform for the dynamic deployment of Cyber Ranges. We started with an in-depth analysis of the current state of the art, to better position our work in the ongoing research field and we identified the main improvement areas. Environment isolation between Cyber Ranges has been achieved through firewalling rules and access policies. A Remote Access Controller has been developed with a OpenVPN server for remote access management. Virtual resources within the Cyber Range are deployed by the Back-end Resource Manager using automated mechanisms. The entire environment is protected by different security layers. One such layer is the Credential Manager, which keeps track of users connected to the system. The API validates incoming requests and rejects the invalid ones. All the resources and actions in the environment are controlled in real-time by the Cluster Security Controller, which performs automated defence operations in case of cyber attacks or system failures. The platform will be improved in many aspects with respect to both the general architecture and its implementation. First, we plan to evolve the architecture to dynamically deploy comprehensive Cyber Arenas other than the classical Cyber Ranges, providing Internet protocols emulation functionality. The Back-End Resource Manager will make use of orchestrators such as AWS Cloudformation to deploy Virtual and Hybrid Cyber Ranges. Moreover, The Cyber Range Environment perimeter will be hardened by taking additional security measures such as setting up Intrusion Detection and Intrusion Prevention systems.

References

[1] G. Perrone, S. P. Romano, The Docker Security Playground: A hands-on approach to the study of network security, 2017 Principles, Systems and Applications of IP Telecommunications (IPTComm) (2017) 1–8. doi:10.1109/IPTCOMM.2017.8169747.

[2] V. E. Urias, W. M. S. Stout, B. Van Leeuwen, H. Lin, Cyber Range Infrastructure Limitations

and Needs of Tomorrow: A Position Paper, 2018 International Carnahan Conference on Security Technology (ICCST) (2018) 1–5. doi:10.1109/CCST.2018.8585460.

[3] M. Karjalainen, T. Kokkonen., Comprehensive Cyber Arena; The Next Generation Cyber Range, 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW) (2020) 1–6. doi:10.1109/EuroSPW51379.2020.00011.

[4] T. Debatty, W. Mees., Building a Cyber Range for training CyberDefense Situation Awareness, 2019 International Conference on Military Communications and Information Systems (ICMCIS) (2019) 1–6. doi:10.1109/ICMCIS.2019.8842802.

[5] G. Østby, K. N. Lovell, B. Katt, Excon teams in cyber security training, 2019 International Conference on Computational Science and Computational Intelligence (CSCI) (2020) 1–6. doi:10.1109/CSCI49370.2019.00010.

[6] Y. He, L. Yan, J. Liu, D. Bai, Z. Chen, X. Yu, D. Gao, J. Zhu, Design of Information System Cyber Security Range Test System for Power Industry, 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia) (2019) 1–5. doi:10.1109/ISGT-Asia.2019.8881739.

[7] M. Frank, M. Leitner, T. Pahi, Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Educations, 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech) (2018) 1–9. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2017.23.