CRaaS: Cyber Range as a Service

K. S. Srinivas(✉), M. Suhas, P. Srinath, K. C. Sneha, D. G. Narayan, and P. Somashekhar

KLE Technological University, Hubballi 580031, India
srinivaskshegde44@gmail.com
https://www.kletech.ac.in/

Abstract. The Internet and networking have evolved substantially into new forms and types. These technologies are used by everyone in the 21^{st} century. After the COVID pandemic, everything has become more virtual, and most transactions, including studies, were accomplished online. With the growth in technology and services, the risk of them being exploited increases. Cybersecurity is a collection of technologies, procedures, and practices aimed at preventing attacks, damage, and illegal access to networks, devices, programs, and data. Overcoming the cybersecurity challenges is even more complicated due to the lack of training and unavailable cybersecurity environments. Cybersecurity experiments must be run in a realistic and controllable environment. In this work, we have set up a virtual environment to provide the required resources for the user to learn cybersecurity skills. We have demonstrated eight cyber attacks. Each attack is provided with a demo video and an automated version of the attack. The user is provided with steps to practice how the attack works and instructions to mitigate these attacks. These resources are made available on-demand to the user through a web portal. The performance analysis has been done for every attack. The launch time of the automated attacks is also recorded and analyzed.

Keywords: CRaaS · Cybersecurity · Virtualization · Automation

1 Introduction

The technique of protecting computers, servers, mobile devices, electronic systems, networks, and data from hostile and vicious attacks is known as cybersecurity. This field is becoming increasingly significant due to the increased reliance on computer systems, the internet, and wireless network standards such as Bluetooth and Wi-Fi, and due to the growth of "smart" devices, including smartphones, televisions, and the various devices that constitute the IoT (Internet of things). Due to a large user base, varied services supplied, architectural complexity, availability, and other variables, these systems are exceedingly difficult to defend. These problems have resulted in insecure systems, owing to a scarcity of experienced cybersecurity professionals and the complexity of putting up experimental cybersecurity settings. A cyber range is a platform that allows teams of

S. Mekhilef et al. (Eds.): ICEEE 2022, LNEE 894, pp. 565–576, 2022.
https://doi.org/10.1007/978-981-19-1677-9_51

professionals to exercise cybersecurity. Cyber ranges offer a safe and legal place for cybersecurity training, practice, and warfare. Threat isolation is achieved by teaching trainees how to perceive and respond to real-world difficulties in a controlled setting. This method ensures that customer data and infrastructure are never jeopardized as a result of cybersecurity training. Few companies like Cyberbit Range, CYRIN, and U.S. Cyber Range use cyber range platforms to provide training of cybersecurity skills.

We have set up a cyber range that provides the user with a variety of methods to learn cybersecurity skills. The key feature of our cyber range is that it gives the user a set of commands that focus on the attacks and also how they are delivered. It implies that the user has a wide range of alternatives for learning cybersecurity. We lay down the fundamentals of each attack that we demonstrate on our learning portal. The user is provided with virtual computers to utilize in order to gain hands-on experience with cybersecurity skills without having to own a virtual machine. The portal includes all of the procedures necessary to perform manual and automated attacks. Furthermore, we have included recorded videos that explain the cyberattacks.

The paper is organized as follows: Related work section includes the works on creating virtual platforms for users to learn and practice cyber attacks using different methods. Proposed Methodology includes the implementation of the system. The results section provides the results and performance analysis of the system. The conclusions and future work are mentioned in the Conclusions of the paper.

2 Related Work

Authors in [1] have discussed the deployment of six cybersecurity attacks namely packet sniffing, brute-force password cracking, DDoS experiment, DNS attack, buffer overflow experiment, and Cross-Site Scripting(XSS) attack. Authors have also discussed the performance evaluation of CLaaS in terms of time required to create a virtual cybersecurity experiment concerning the number of VMs, CPU utilization, and Disk latency during large deployment, and network and disk utilization during attack scenarios. Authors in [2] have discussed HTTP DDoS attacks on virtual machines in OpenStack. The authors further elaborated on the webserver performance testing tools that were used to observe the results. Authors used the DDoS attacking tools, real-time HTTP flooding datasets available on the internet, and virtual machines called “bots” to bring down the webserver running in the cloud environment.

In [3], authors have presented a life cycle for the design of testbeds in education. The authors demonstrated the process of SQL injection. This paper presents a life cycle for the designing of testbeds in education. The life cycle consists of 7 steps: design, configure the environment, deploy environment, define challenges, deploy challenges, conduct challenges, maintain environment and maintain challenges. Authors in [4] describe the design decisions made during development. The authors demonstrated the key features of the KYPO cyber range. It is for

researching and developing new security methods, tools and for training security teams and students. Any web browser can access it. KYPO is highly scalable, cost-effective, and flexible. It can run on platforms like OpenStack or OpenNebula.

Authors in [5] have discussed few DDoS attacks like IP spoofing, SYN flooding, smurf attack, buffer overflow attack, ping of death, and teardrop attack that have been experimented on a cloud environment. Further, the authors have discussed the defensive mechanisms and preventive measures for all the attacks in the cloud-based system. Authors in [6] have presented the Tele-Lab project, online e-learning IT security lab. The portal and tutoring environment, the virtual machine pool, the database, NX server, and central Tele-Lab control server are the key components of this architecture. It offers three kinds of content namely, information chapters, introductions to security and hacker tools, and practical exercises.

In [7], authors have proposed a cybersecurity exercises system(CyExec) in a virtual computer environment. The training content on CyExec is divided into two parts: basic and applied. Regarding the basic part, they installed WebGoat, an open-source vulnerability scanner, on CyExec and created a curriculum and training manual for the WebGoat activities. On the applied part, they created and deployed their own cyberattack and defense training materials on CyExec. Authors in [8] have designed a system where students can work with network devices. The user has to use a link provided within the learning platform, also a booking slot is made available so that only that student can access the system with an active role.

Authors in [9] discussed the implementation of the cybersecurity research testbed in a controlled environment using OpenStack. The authors have shed light on analyzing the effect on CPU utilization over several virtual machines running simultaneously. In [10], authors have discussed building a remote virtual security laboratory called Tele-lab. The authors explain about setting up the virtual laboratory in a container. The authors also presented the uses of containers designed in training the environment of man in the middle (MITM) attack and firewall learning units.

3 Proposed Methodology

The global system allows the user being able to login or register to the web portal through any web browser, and be able to access the web portal and start improving his cybersecurity skills by indulging in the virtual and automated environment. Figure 1 shows the architectural design of the system.

The user logs in to the system by the login page provided by the web portal through any web browser. A new user can register to the portal. After registering, the user's credentials are stored in a database. When a user attempts to log in, the credentials they provided during registration are used to authenticate them. After logging in, the user is greeted with a user interface. Users may view video sessions of cybersecurity assaults and acquire new cybersecurity skills, or they

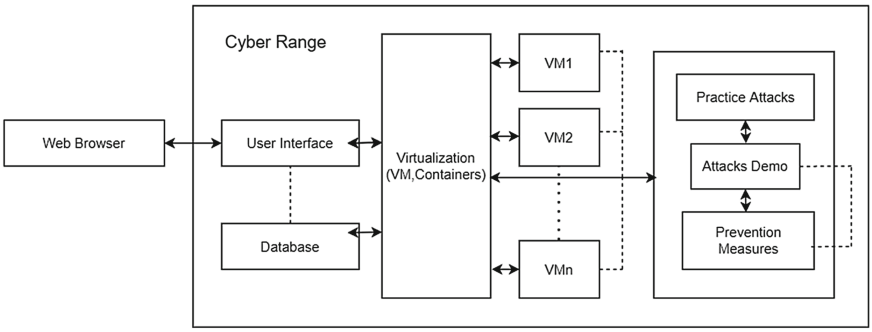

Fig. 1. System model

can observe an automated attack that happens on demand when they click the "Launch Attack" button. With these demonstrations, the user can understand the working of the cyberattacks.

Furthermore, the user is provided with a practice session for a hands-on experience. The practice option provides the users with virtual machines necessary to perform the attacks. The user need not have any virtual machine of his own. The user can directly connect to the virtual machine using a remote desktop connection. The portal also contains the steps that have to be followed to perform an attack when the user is practicing. The steps to prevent these attacks are available to the user on the portal.

We use a modular approach to design and implement the system as it helps to divide the different aspects of the entire system into specific objectives for a module. It helps to maintain each module effectively rather than designing the complete system.

3.1 Modules

Our system is mainly classified into three major modules. They are the UI module, the VM provisioning module, and the automation module. The significance of each module is discussed in this section. Figure 2 depicts the modules designed in our proposed system.

1. UI Module
   This module is developed using a python framework, Django. It consists of two components, authentication and a learning portal.
   - Authentication: Authentication provides login and register options. A user can log in to his previously existing account or register for the portal. A successful registration leads to the credentials stored in a database. We have used PostgreSQL as our database management system. When users try to log in, they are authenticated by checking if they have already registered on to the portal or not. And also by checking if the password entered matches with the password entered while registering. If all the conditions are satisfied, the user is logged in to the platform successfully.

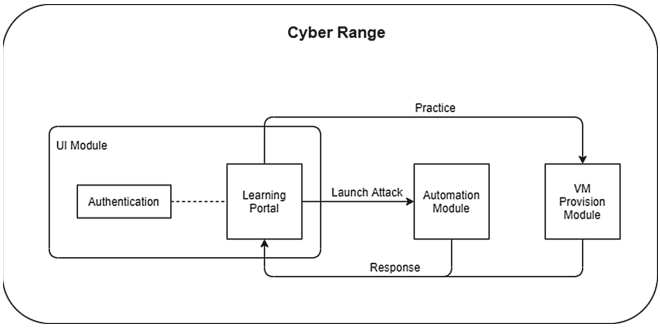

Fig. 2. Cyber Range

– Learning Portal: The learning portal has a set of cyberattacks that the user can learn. On clicking the "Learn" button, the user gets a brief description of the attack, a demo video of the attack, practice, and the launch attack option. The "Launch attack" button, when clicked, launches an automated attack for the user to visualize the attack. The "Practice" option provides the user with a virtual machine for a hands-on experience. Users can imply their knowledge in this environment to develop cybersecurity skills. Users must follow the steps provided on the portal to execute a cyberattack.

2. VM Provosioning module
   VM provisioning module is responsible for the provisioning of virtual machines. Provision is achieved by cloning an existing base machine using an automation script. The base VM contains the necessary tools and software required to perform an attack. Two scenarios make use of this module. The first scenario is the provision of the VM to a user when he clicks the "Practice" button in the web portal so that they can practice cybersecurity skills. The second scenario is in the automation module. The algorithm 1 shows the provisioning of a VM. A primary requirement of this algorithm is an SSH connection. There must be an SSH connection between the user computer and the host machine before the provisioning procedure can begin. SSH(Secure Shell) is a network protocol that helps in the establishment of a secure connection between the user's system and the host machine. The specified base operating system is executing as a guest on this host machine. A file called "IP pool" contains a list of IP addresses. IP addresses are assigned to freshly created VMs using the IP pool.
3. Automation Module
   This module comprises the on-demand execution of cyberattacks. If a user wants to see how the attack works in real-time, they may look at the real-time automated attack, which provides them a better picture. This module encloses two main components. The first component is the automation of the attacks. This component consists of all the attacks that are automated. The second component is the VM provisioning module. As mentioned in the second

scenario in Sect. 2, we realize the automation module by integrating the VM provision module with the automation of the attacks. The base system stores the automation scripts for the attacks. These scripts are inherited whenever a VM is provisioned using the portal. A machine that runs metasploitable2 OS is used as a victim machine for most of the attacks. The Algorithm 2 defines the generalized algorithm of an automation module.

Algorithm 1. VM provisioning

Require: SSH connection, an IP pool
Ensure: User receives an IP address of the VM
Let b_{vm} be the base VM and *ipool* be the IP pool
if running(b_{vm})!=true **then**
    Let c_{vm} be the clone VM
    Assign name to c_{vm}
    Change default c_{vm} IP to static IP address from *ipool*
    c_{vm} IP = POP(*ipool*)
    Restart c_{vm}
    Restart XRDP Server
    Provision the VM to the seeker
end if

Algorithm 2. Generalized Automation of attacks

Require: Provisioned VM, metasploitable
Ensure: Automation of the attack
Let a be the attacker machine and v be the victim machine
if running(a)=true and running(v)=true **then**
    Search for IP addresses connected to the network
    targetAddress = IP(v)
    Perform attack using commands
    Press Ctrl+C to stop
end if

4 Results and Discussion

This section provides the results of the proposed system and the related discussions. The Sect. 4.1 represents experimental setup of CRaaS. In Sect. 4.2 performance analysis of the attacks is discussed.

4.1 Experimental Setup

The experimental setup includes a real-time cloud environment. The following tables represent the operating systems, software, and tools used to perform and analyze the attacks (Table 1).

Table 1. Configuration of machines

Type of machine	Operating system	vCPU	RAM (GB)	HDD (GB)	Version
Host	Ubuntu	8	32	1024	16.04.1 LTS
Base, attacker	Kali linux	4	2	20	2020.3
Victim	Metasploitable 2	2	1	10	8.04

Table 2. Software/Tools used

Software/Tools	Version
Wireshark	3.2.1
EtherApe	0.9.19
hping3	3.0.0
Metasploit console(msfconsole)	5.0.71
Yersinia	0.8.2
BeEf	0.5.0.0
Burp Suite	2020.12.1
xrdp	0.9.12
Python	3.7.2
Paramiko	2.7.2
pgAdmin	5.2
Django	2.2.3

4.2 Performance Analysis

In this section, we analyze VM launch time, experiment launch time, and CPU utilization of attacks.

VM Launch Time. Analysis of VM launch time as represented in Fig. 3 is performed by taking time(in seconds) on the y-axis and trials on the x-axis. We observe that each trial produces a different time value, this is due to the system's background load. We estimate that launching a virtual machine takes 1.3 min (76.51 s) on average.

Experiment Launch Time. As shown in Fig. 4, analysis of launch time of automated attacks is carried out by taking time on the y-axis and different types of attacks on the x-axis. Before actually starting the attack, the search for IP addresses and the other operations outlined previously are included in the launch time. We observe that DHCP Starvation takes the highest time while UDP flooding takes the least time to launch the attack. In the UDP flooding attack, UDP packets are sent following a request-response mechanism that takes less time. While in a TCP flooding attack, a client-server TCP connection is a necessity and it is established through a 3-way handshake resulting in consuming

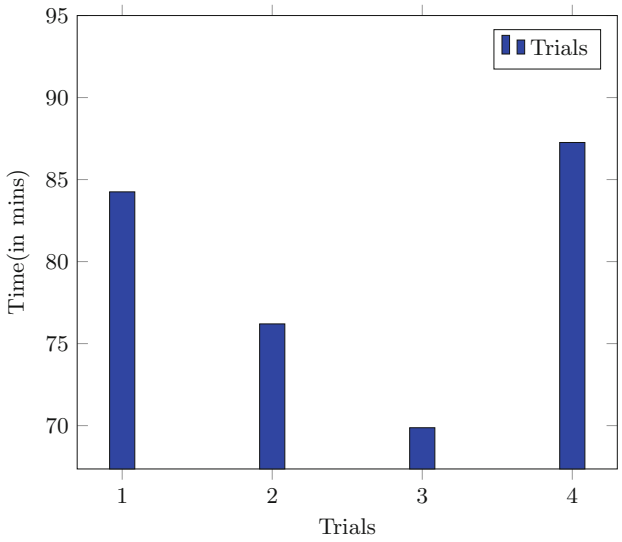

Fig. 3. VM launch time analysis

more time than UDP flooding. In IP spoofing, the attacker makes 3rd step of the 3-way handshake unachievable. Hence, the time taken to launch the attack is lesser than TCP flooding and higher than UDP flooding.

DHCP starvation takes more time mainly because of two aspects, the attacker floods the DHCP server and the attacker forwards the traffic of the victim machine.

CPU Utilization. CPU Utilization is considered as the primary analytical metric to analyze the performance of the attacks. Performance study is carried out by producing CPU consumption graphs of the attacks. All the readings are taken from the attacker machine.

- TCP Flooding
  CPU utilization before and after performing TCP Flooding attack has been recorded and a bar graph has been plotted for the same by taking CPU utilization(in %) on the y-axis and each trial performed on the x-axis as shown in Fig. 5. We observe that each trial produces a different CPU utilization value, this is due to the system's background load. From the graph, it is inferred that up to 70% of the CPU is utilized during the attack.
- UDP Flooding
  CPU utilization before and after conducting the UDP Flooding attack was recorded, and a bar graph was generated by plotting CPU utilization(in %) on the y-axis and each trial done on the x-axis, as shown in Fig. 6. We observe that each trial produces a different CPU utilization value, this is due to the inconsistency in the number of packets sent and the background load on the

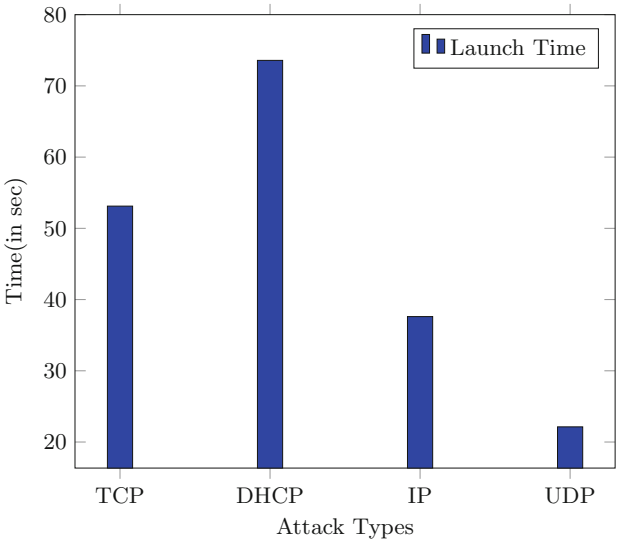

Fig. 4. Experiment launch time analysis

CPU. The attack proves to be CPU intensive by reaching a peak of almost 97% of CPU utilization.

– DHCP Starvation
CPU utilization before and after carrying out DHCP Starvation, Spoofing, and Sniffing attack has been recorded and bar graph has been plotted for the same by taking CPU utilization (in %) on the y-axis and trials performed on the x-axis as the graph in Fig. 7 represents. We observe that each trial produces a different CPU utilization value, it is due to the alterations in kill time of the process and background load on the CPU. This attack also depends on high CPU utilization. The factors affecting this are the IP resource exhaustion from the real-time DHCP server caused by the attacker machine and the attacker machine acting as a router and forwarding traffic of the victim machine.
– IP Spoofing
CPU utilization before and after performing IP Spoofing attack was recorded and bar graph was generated for the same by taking CPU utilization percent on the y-axis and each trial performed on the x-axis as shown in Fig. 8. We observe that each trial produces a different CPU utilization value, it is caused by inconsistency in the number of packets sent and the system's background load. We estimate that IP spoofing utilizes 56% CPU on average.

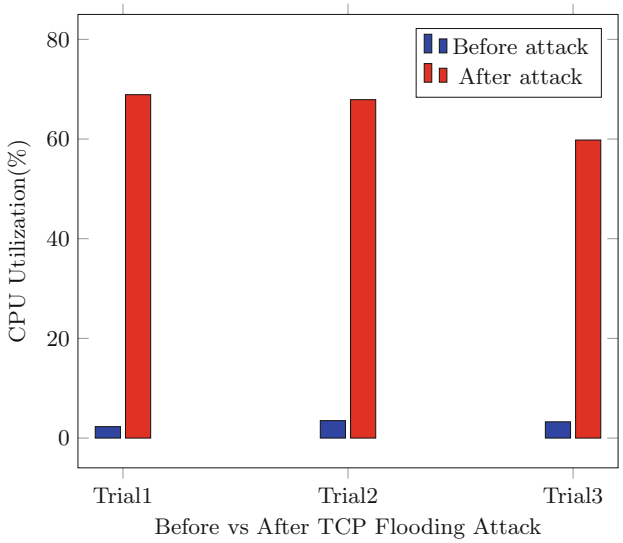

Fig. 5. CPU Utilization bargraph of TCP flooding attack

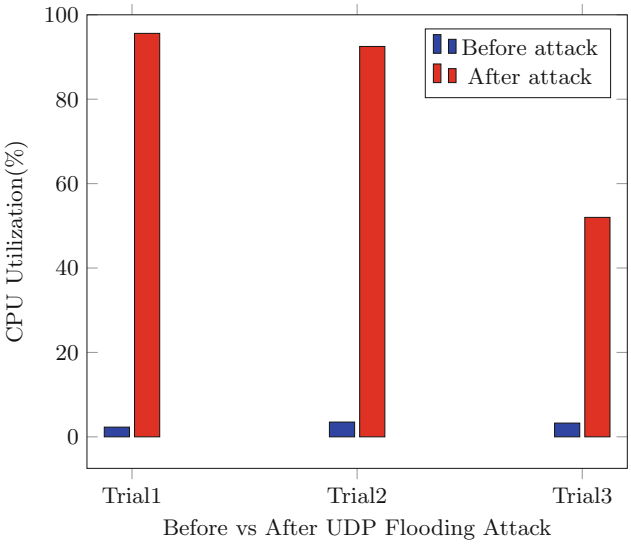

Fig. 6. CPU Utilization bargraph of UDP flooding attack

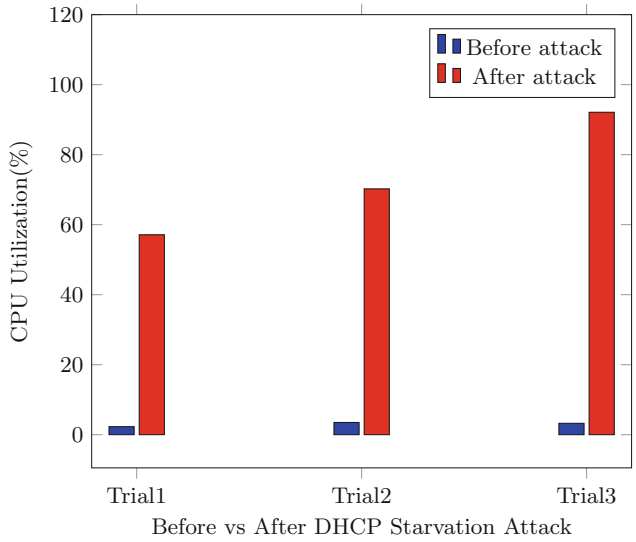

Fig. 7. CPU Utilization bargraph of DHCP starvation attack

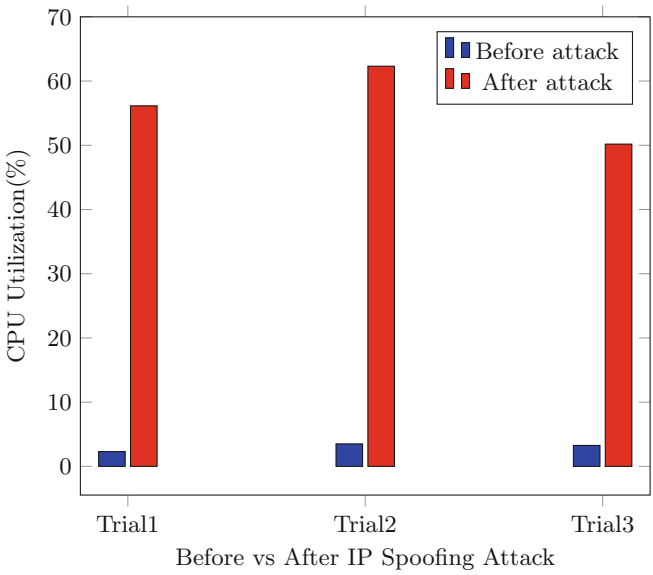

Fig. 8. CPU Utilization bargraph of IP spoofing attack

5 Conclusion

Cybersecurity concerns are increasing as there is growth in the complexity of cyberattacks and the sheer count in which they occur is prevailing. People lack cybersecurity understanding, which is one of the primary causes of the upright

cyber breaches that happen every single day. With our portal, we provide the end-user with a smooth experience where they can not only learn the skills but also play with those skills in our controlled environment. The existing portals are for penetration testing and conduction of cyber challenges. Some portals offer the information of the attacks, but they are either more complicated to understand or the conveying of the overall knowledge is not suitable for beginners. Therefore, we built a portal that provides information on the cyberattacks to the end-user, making the user understand how these attacks are conducted and how they can be prevented.

Cyberattacks are not limited to a few ten attacks as a whole. There are new variations of a similar attack every year. As part of our future work, we plan to create a portal that includes several additional cyberattacks as well as measures to mitigate them.

References

1. Tunc, C., Hariri, S.: CLaaS: cybersecurity lab as a service. J. Internet Serv. Inf. Secur. **5**(4), 41–59 (2015)
2. Dhanapal, A., Nithyanandam, P.: An openstack based cloud testbed framework for evaluating http flooding attacks. Wirel. Netw. **27**, 5491–5501 (2019). https://doi.org/10.1007/s11276-019-01937-4
3. Frank, M., Leitner, M., Pahi, T.: Design considerations for cyber security testbeds: a case study on a cyber security testbed for education. In: 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence and Computing, 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (2017)
4. Vykopal, J., Ošlejšek, R., Čeleda, P., Vizvary, M., Tovarňák, D.: Kypo cyber range: design and use cases (2017)
5. Darwish, M., Ouda, A., Capretz, L.F.: Cloud-based DDoS attacks and defenses. In: International Conference on Information Society (i-Society 2013), pp. 67–71. IEEE (2013)
6. Willems, C., Meinel, C.: Tele-lab it-security: an architecture for an online virtual it security lab. Int. J. Online Eng. **4**(2), 31–37 (2008)
7. Maki, N., Nakata, R., Toyoda, S., Kasai, Y., Shin, S., Seto, Y.: An effective cybersecurity exercises platform CyExec and its training contents. Int. J. Inf. Educ. Technol. **10**(3), 215–221 (2020)
8. Tobarra, L., Robles-Gómez, A., Pastor, R., Hernández, R., Duque, A., Cano, J.: A cybersecurity experience with cloud virtual-remote laboratories. In: Multidisciplinary Digital Publishing Institute Proceedings, vol. 31, p. 3 (2019)
9. Edgar, T.W., Rice, T.R.: Experiment as a service. In: 2017 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1–6 (2017)
10. Sianipar, J., Willems, C., Meinel, C.: A container-based virtual laboratory for internet security e-learning. Int. J. Learn. Teach. (IJLT) **2**(2), 121–128 (2016)
11. Gumaste, S., Shinde, S., et al.: Detection of DDoS attacks in openstack-based private cloud using apache spark. J. Telecommun. Inf. Technol. (2020)
12. Kachavimath, A.V., Narayan, D.G.: A deep learning-based framework for distributed denial-of-service attacks detection in cloud environment. In: Advances in Computing and Network Communications, pp. 605–618. Springer (2021)