Learn Ethical Hacking by Solving Real Security Challenges

PHASE 1 REPORT

Submitted by

DHANVANTHRAO J (RCAS2021MCS221)

in partial fulfillment for the award of the degree of

MASTER OF SCIENCE
SPECIALIZATION IN
INFORMATION SECURITY AND CYBER FORENSICS

DEPARTMENT OF COMPUTER SCIENCE

RATHINAM COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)

COIMBATORE - 641021 (INDIA)

DECEMBER-2022

RATHINAM COLLEGE OF ARTS AND SCIENCE
(AUTONOMOUS)
COIMBATORE - 641021

BONAFIDE CERTIFICATE

This is to certify that the phase-I entitled **Learn Ethical Hacking by Solving Real Security Challenges** submitted by **DHANVANTHRAO J,**, for the award of the Degree of Master of Computer Science specialization in **"INFORMATION SECURITY AND CYBER FORENSICS"** is a bonafide record of the work carried out by him under my guidance and supervision at Rathinam College of Arts and Science, Coimbatore

Ms.Kanimozhi V,M.E., (Ph.D)
Supervisor

Mr.P.Sivaprakash, M.E., (Ph.D)
Mentor

Submitted for the University Examination held on 02.12.2022

INTERNAL EXAMINER

EXTERNAL EXAMINER

RATHINAM COLLEGE OF ARTS AND SCIENCE

(AUTONOMOUS)

COIMBATORE - 641021

DECLARATION

I, **DHANVANTHRAO J**, hereby declare that this phase-I report entitled **"Learn Ethical Hacking by Solving Real Security Challenges",** is the record of the original work done by me under the guidance of **Ms.V.Kanimozhi, M.E., (Ph.D)**, Faculty Rathinam college of arts and science, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any University.

Place: Coimbatore **Signature of the student**

Date: 02.12.2022 Dhanvanthrao J

COUNTERSIGNED

Ms Kanimozhi V

Supervisor

Contents

Acknowledgement

On successful completion for project look back to thank who made in possible. First and foremost, thank **"THE ALMIGHTY"** for this blessing on us without which I could have not successfully our project. I am extremely grateful to **Dr.Madan.A. Sendhil, M.S., Ph.D.,** Chairman, Rathinam Group of Institutions, Coimbatore and **Dr. R.Manickam MCA., M.Phil., Ph.D.,** Secretary, Rathinam Group of Institutions, Coimbatore for giving me opportunity to study in this college. I am extremely grateful to **Dr.R.Muralidharan, M.Sc., M.Phil., M.C.A., Ph.D.,** Principal Rathinam College of Arts and Science(Autonomous), Coimbatore.Extend deep sense of valuation to **Mr.A.Uthiramoorthy, M.C.A., M.Phil., (Ph.D),** Rathinam College of Arts and Science (Autonomous) who has permitted to undergo the project.

Unequally I thank **Mr.P.Sivaprakash, M.E., (Ph.D).,** Mentor and **Dr.Mohamed Mallick, M.E., Ph.D.,** Project Coordinator, and all the Faculty members of the Department - iNurture Education Solution pvt ltd for their constructive suggestions, advice during the course of study. I convey special thanks, to the supervisor **Ms.KANIMOZHI.V, M.E., (Ph.D).,** who offered their inestimable support, guidance, valuable suggestion, motivations, helps given for the completion of the project.

I dedicated sincere respect to my parents for their moral motivation in completing the project.

List of Figures

List of Abbreviations

CTF	Capture The Flag
IEEE	Institute of Electical and Electronics Engineers
VM	Virtual Machine
IRC	Internet Relay Chat
ACI	Army Cyber Institute
CRI	Container Runtime Interface

Abstract

It can be difficult for any teacher to introduce technical concepts to students who have little or no technical expertise. Gamification is a term that has lately been used to describe a strategy for motivating students that involves incorporating various game-based strategies into instructional modules. This idea has led to the discovery that capture the flag (CTF)-style competitions are an effective approach to introduce students to a number of technical ideas within the traditional computer science curriculum. A CTF was conducted at numerous GenCyber camps across the nation with the main objective of educating high school students to various computer security and digital forensics issues without necessitating any prior knowledge on the part of the students.We discovered that this approach of breaking things down into discrete problems and linking these difficulties together in a competitive setting was remarkably successful at not just introducing kids to these concepts but also inspiring continuing learning after the camps concluded. This Paper will evaluate the effort's accomplishments as well as the drawbacks identified by applying this method.

Chapter 1

Introduction

More than 6000 security flaws were revealed in a variety of software products in 2020, including web browsers, the OS kernel, music players, and website content management systems. Many of these were brought on by flaws that have been thoroughly researched for more than ten years and have known remedies. Thus, it is essential to teach software engineers secure coding techniques in order to reduce losses and protect both corporate and public safety.Competitions like Capture the Flag (CTF) offer a fun way to learn about secure coding techniques and computer security concepts. Teams are given the same issues in this competition, and they must use computer security principles to solve them. They are particularly promising as pedagogical tools because of how practical and hands-on they are.

Due to its interdisciplinary character, computer security poses a problem for education. Computer security topics include anything from theoretical computer science concepts to practical information technology management concepts. This makes it challenging to capture the essence of what a computer security practitioner is.

The "capture the flag" competition is one approximation for this measurement that

has been developed. Attack-focused CTF competitions aim to condense the core of numerous facets of professional computer security work into a single, brief task that can be measured objectively. The focus areas that CTF contests typically measure include toolkit development, operational tradecraft, vulnerability finding, and exploit generation.

1.1 Objective of the project

A cybersecurity competition called Capture the Flag (CTF) is used to evaluate security expertise. The biggest cybersecurity conference in the US, DEFCON, held yearly in Las Vegas, Nevada, was where it was first created. The conference holds a weekend-long competition for cybersecurity, including CTF. CTF can be played in two different ways: Jeopardy and Attack-Defense. Although both formats assess participants' cybersecurity knowledge, they have different objectives. Participating teams in the Jeopardy format must finish as many tasks with various point values from a specific category as possible. Programming, networking, and reverse engineering are a few categories. Competing teams in the attack-defense format are required to assault each other while defending their weak computer systems. This is accomplished by attempting to overwrite the rivals' data file or "flag" with their own. Other CTF competitions, such as CSAW CTF and Plaid CTF, have been held since the CTF.

CTF competitions often follow one of two formats:

- **Jeopardy-style Challenges:** Teams are given a variety of tasks to complete in

a variety of fields, including cryptography, digital forensics, system security, and web application security. They can complete tasks in the order they see fit and work independently of other teams.

• **Attack-defense-style Challenges:** Teams are given a virtual machine with identically susceptible services in this strategy. The ultimate goal is to identify existing security flaws, correct them, and attack other teams' services using the vulnerabilities found.

Attack-defence CTFs are thought to produce better learning results between the two forms because of their interactive and practical nature. In contrast to Jeopardy-style CTFs, however, very few attack-defence CTFs are held annually and the number of participants is far fewer. Additionally, only three attack-defense CTFs were held online per year over the previous five years; the majority of these competitions were held in person.

1.2 Scope of the Project

Studies have shown that students respond better to interactive approaches like those used in CTF exercises than they do in a typical classroom setting, hence CTF is primarily utilised for cybersecurity education. According to a study by Adelphi University researchers, CTF games are a highly effective way to teach cybersecurity concepts in a fun way. They can also be used in a classroom setting; they have been used in University of Southern California's Introduction to Security, among other undergraduate computer science courses.

CTF is a favourite in military academies as well. They are frequently covered in the curriculum for courses on cybersecurity. For instance, CTF exercises are being conducted by students at the Air Force Academy and the Naval Academy who are members of cybersecurity clubs, according to an article published by the Cyber Defense Review, a publication published by the Army Cyber Institute (ACI) at West Point. Additionally, the Advanced Course in Engineering on Cyber Security, an intensive summer programme available to ROTC cadets, active duty personnel, and students, teaches numerous cybersecurity principles through CTF simulations.

The assumption of a basic level of computer operational competence is a downside of CTF exercises. Since the exercises' primary goal is to impart cybersecurity concepts, basic computer skills like opening several tabs cannot be taught through them. As people need to be trained to comprehend the challenges' workflow, those managing CTF exercises have also had trouble supervising and managing competitions and training exercises. In an effort to help them grasp the exercise environments in advance, CTF competitions have attempted giving facilitators early access to them, however the majority of them still felt unprepared to oversee CTF events. Another issue is the generational divide between exercise designers and participants, which results in unworkable and occasionally outmoded difficulties.Without a clear understanding of the seriousness of the consequences caused by vulnerabilities, students may find it difficult to comprehend the significance of a security concept.

1.3 Module Description

Docker:

A virtualization technology at the operating system level is called Docker. Similar to hardware virtualization, Docker enables the execution of processes in constrained sandboxes known as containers. Similar to how virtual machines are launched from virtual disc images, these containers are built using container images. However, Docker virtualizes the host device's kernel as opposed to hardware virtualization. This offers comparable levels of isolation, security, and performance while dramatically reducing resource utilisation, image sizes, and startup times for containers compared to virtual machines. As a result, virtual machines can be replaced by Docker containers in attack-defence CTF infrastructures.

We use Docker Distribution and Portus in addition to Docker to handle exploits and patches for services that teams have submitted. A container image management solution called Docker Distribution makes it easier to store, update, and distribute images. Role-based access control is implemented by Portus, a front end for Docker distribution, for the images kept in Distribution and other resources. Teams, private and shared picture namespaces, and different access levels for various people can all be created by users. Because of this, Portus is the best option for handling all teams' container images.

Kubernetes :

A open source technology called Kubernetes (also known as K8s) orchestrates con-

tainer runtime systems across a cluster of networked resources. Docker is optional while using Kubernetes. Kubernetes is used to run and manage containers from various container runtimes. Numerous container runtimes, such as Docker, containerd, CRI-O, and any implementation of the Kubernetes CRI, are supported by Kubernetes (Container Runtime Interface).

1.4 Existing System

TryHackMe:

TryHackMe is an online platform that uses brief, gamified real-world labs to educate cyber security. We include material for both inexperienced and experienced hackers, as well as challenges and integrated guides to accommodate various learning methods.

Ashu Savani and Ben Spring, two cyber security enthusiasts who met while working a summer internship, founded TryHackMe in 2018. When they first entered the field, they discovered that learning security was a fragmented, challenging, and challenging experience; frequently, learning security by being given the IP of a vulnerable machine with no additional resources is not the most effective way to learn, especially if you have no prior knowledge. When Ben returned to school, he developed a method for deploying machines and gave it to Ashu. Ashu then recommended that they post all of their summer's worth of notes on a centralised site so that others might learn from them for free.

PicoCTF:

A set of CTFs called picoCTF is targeted towards high school pupils. Nowadays, aside from high school pupils, all players must overcome increasingly challenging hurdles in picoCTF. The majority of the tasks in picoCTF these days are provided by the picoCTF platform, and you can attempt to answer those challenges online. The difficulties range from extremely simple to difficult. Some basic CTF challenges are approachable for newcomers, while others are more difficult and may take some time to complete. The binary tasks are difficult for beginners, and picoCTF doesn't offer binary (pwn) challenges that are beginner-friendly. PicoCTF may not be the best platform for people just beginning out in the binary industry.

CTFd :

The Capture The Flag (CTF) framework CTFd was created with both administrators and users in mind. A strong yet user-friendly CTF website was determined to be necessary during the administration of CSAW CTF. I, as the organiser, had grown weary of writing PHP code that was a mess and hazardous SQL queries. These aspirations led to the creation of CTFd. Designed to function for one of the biggest CTFs in the world, yet simple enough to run for smaller groups. There are numerous hazards to be aware of while creating a CTF, and CTFd was created with these in mind.

Hack The Box :

Hack The Box is a sizable, online cybersecurity training platform that enables people, businesses, academic institutions, and all other kinds of organisations worldwide to advance their hacking abilities. Our online hacking laboratories include exercises

for all skill levels, from the easiest to the most challenging. Every week, fresh content is uploaded, covering the most recent exploits and vulnerabilities. We believe in effective, hands-on training where you learn by doing rather than only concentrating on theoretical learning and checkboxes. Over 1 million cybersecurity enthusiasts that use our platform to develop their abilities in an entertaining and gamified way make up the HTB community. Beyond the community, over 700 businesses, consulting firms, non-profits, and educational institutions use our b2b solution to teach their staff on cybersecurity.

Chapter 2

Literature Survey

The worldwide cybersecurity employment shortfall can be filled gradually but steadily by training security specialists. As a result, educational institutions, computer societies, governmental agencies, and commercial businesses have developed new curriculum, study programmes, and courses. ACM/IEEE Computing Curricula 2020 includes cybersecurity as a key component, and other specialised curricula, like CSEC2017, have started emerging recently. In addition to formal education, Capture the Flag (CTF) games and competitions are a growingly popular way for people to practise cybersecurity skills. Small groups of participants work together to complete various activities in an online learning environment as they practise their cybersecurity abilities at these events. CTF tasks, often known as challenges, include a variety of activities, such as hacking insecure networks and attacking websites. A text string called a flag is produced when a challenge is successfully solved and submitted online as proof of completion.

CTF was created by cybersecurity enthusiasts during the DEF CON hacker conference in 1996. CTF, however, is no longer the exclusive domain of hacking organisations. Teachers all around the world are now adopting this instructional game style to sup-

plement classroom instruction after it swiftly gained popularity. CTF has been used successfully in college courses and in security competitions for undergrads. Even internet behemoths like Facebook and Google hold CTFs that draw hundreds of participants each year. unlike conventional education methods like lectures and homework.

The majority of attack-defence CTFs use an architecture that was developed by iCTF over the course of numerous editions. All services are bundled into a virtual machine that is run either by the organisers on their server or by the participants on their hardware. Teams can access their virtual machines directly in the first scenario while doing so via SSH in the second scenario. Although practically all editions of iCTF include virtual machines as a fundamental part of the infrastructure, the organisation is also well known for experimenting with contest design and gameplay. In other game-based approaches to teaching computer and network security and other computer science courses, as well as targeted training programmes, virtual machines are also often employed.

2.1 Recommender System

A recommender system primarily serves as a tool to provide end users with recommendations for the goods and services that will work for them. By providing the customer with their preferred item, it encourages convenience. It also makes it simple to determine the interests of the final user. The primary goal of a recommender system is to provide recommendations to users who have specific preferences, and these preferences serve as the input. These systems base their recommendations on a user's

previous purchases or preferences for the product. A method called item-based filtering uses the user's previously chosen items to find comparable items. Other users are then advised to use these.

2.2 HackerBot

With regard to interaction, HackerBot was created to facilitate exercises in defence and investigation. An IRC chatbot called Hackerbot challenges students via instant messaging (IM), usually by assaulting or compromising the students' virtual machines (VMs) and asking them to defend, look into, or restore the VMs in exchange for CTF flags.

Each student has a desktop virtual machine (VM) for each lab, which comes with a client for chatting with the bot. Since Pidgin offers users a familiar IM interface (as opposed to a more conventional IRC-focused interface), it was chosen as the client. Firefox begins displaying the student's lab sheet, which contains login information for their VMs, as soon as the desktop VM starts.

Additionally, a "hackerbot server" virtual machine (VM) is running, hosting the Hackerbot and serving HTML lab worksheets (which are generated by the server). The Hackerbot attacked or interacted with at least one other server or desktop system in the majority of the weekly themes.

Chapter 3

Methodology

3.1 Attack Defence CTFs

3.1.1 Overview

Teams are given identical resources in attack-defence CTFs. virtual computers with specially created vulnerable services (or "apps"). Teams examine the service vices, identify security flaws, correct them, and launch an attack other teams scoring with the same security flaws. CTFs that test both offence and defence include attack-defence. in areas of network, system, and application security. As a result, attack-defence CTFs demand specialised cialized gaming systems that are challenging to implement neer and uphold consistently. The difficulties are greater. If the game is played online, the problem is exacerbated because players may be spread out geographically all over the world.

3.1.2 Services

A service is a specially created, risky application that is provided by the organisers. They can range from very basic network applications like a chat server for linked clients

to extremely complicated ones like banking and social networking software. All services offer the option to keep confidential data (referred to as a "flag") that can only be retrieved by entering a specific security code (e.g.: username and password). These services also have one or more vulnerabilities, including buffer overflows, SQL injection, and the intentional use of weak encryption keys. The services are carefully designed such that the flag can only be obtained by providing the relevant secret or taking advantage of the vulnerability.The flag and secret cannot be violated by bruteforce. surrendering a flag of an opposing team as proof that it was done fast that the group was able to utilise the corresponding assisting the opposing team.

3.1.3 Gameplay and scoring

CTFs that involve attack and defence typically last 8 to 12 hours. There are no flags stored in any of the services during the initial stage of the competition. Teams spend this time analysing the system and its services, fixing any security flaws they find, and creating exploits for these security holes so they can score points in the phase that follows.The second phase, known as the scoring phase, is divided into several rounds that last roughly equal amounts of time. Flags are stored in all services at the beginning of each round by scripts run by the organizers, who then retrieve them near the end. Updates to services that either add new functionality or change existing functionality occasionally may be introduced by organizers.Teams keep examining the services and network connections for fresh exploits while searching for any potential holes they might have overlooked. A team can use one of three different types of points during a round:

1. Offence points obtained by stealing flags from other teams.

2. Points are given for ensuring that services are accessible online.

3. points earned on defence by keeping rival teams from stealing flags.

The scoring system and gameplay closely mirror situations encountered in the real world:

1. Applications can fail gracefully with malicious input and without disclosing the secret information while still functioning as intended for genuine input.

2. Applications are constantly vulnerable to attack and compromise. Therefore, ongoing observation and prompt action are needed.

The offensive component also encourages players to think when creating secure software and creating security updates. from the viewpoint of an adversary.

3.2 Docker

Docker is an operating system-level virtualization technique. Docker facilitates the running of processes in limited sandboxes known as containers, much to hardware virtualization. These containers are created using container images, much how virtual machines are started from virtual disc images. Contrary to hardware virtualization, Docker instead virtualizes the kernel of the host device. When compared to virtual machines, this delivers equivalent levels of isolation, security, and performance while drastically reducing resource usage, image sizes, and startup times for containers. Therefore, Docker containers can take the place of virtual computers in attack-defense CTF systems.

To handle exploits and patches for services that teams have submitted, we also employ Portus and Docker Distribution. It is simpler to store, update, and distribute images thanks to a tool called Docker Distribution for managing container images. Portus, a front end for Docker distribution, implements role-based access control for the images saved in Distribution and other resources. Users can build teams, private and shared picture namespaces, and varying access levels for different persons. Because of this, Portus is the ideal choice for managing the container pictures for all teams.

Chapter 4

Experimental Setup

4.1 Docker Desktop

You may create and distribute containerized applications and microservices with Docker Desktop, an easy-to-install programme for Mac, Linux, or Windows environments.

It offers a straightforward interface that lets you manage your containers, programmes, and images directly from your computer without having to resort to the CLI for basic operations.

4.2 Dockerfile

Docker can automatically create images by following the directions in a Dockerfile. All the commands a user could issue on the command line to put together an image are listed in a text file called a Dockerfile.

4.3 Docker Engine

Review of the Docker Engine An open source containerization solution for creating and containerizing your apps is called Docker Engine. An example of a client-server

application is Docker Engine, which:

a computer that runs the dockerd daemon process continuously. APIs that define the interfaces that software can use to communicate with and give commands to the Docker daemon. a docker client with a command line interface (CLI).

4.4 Architecture

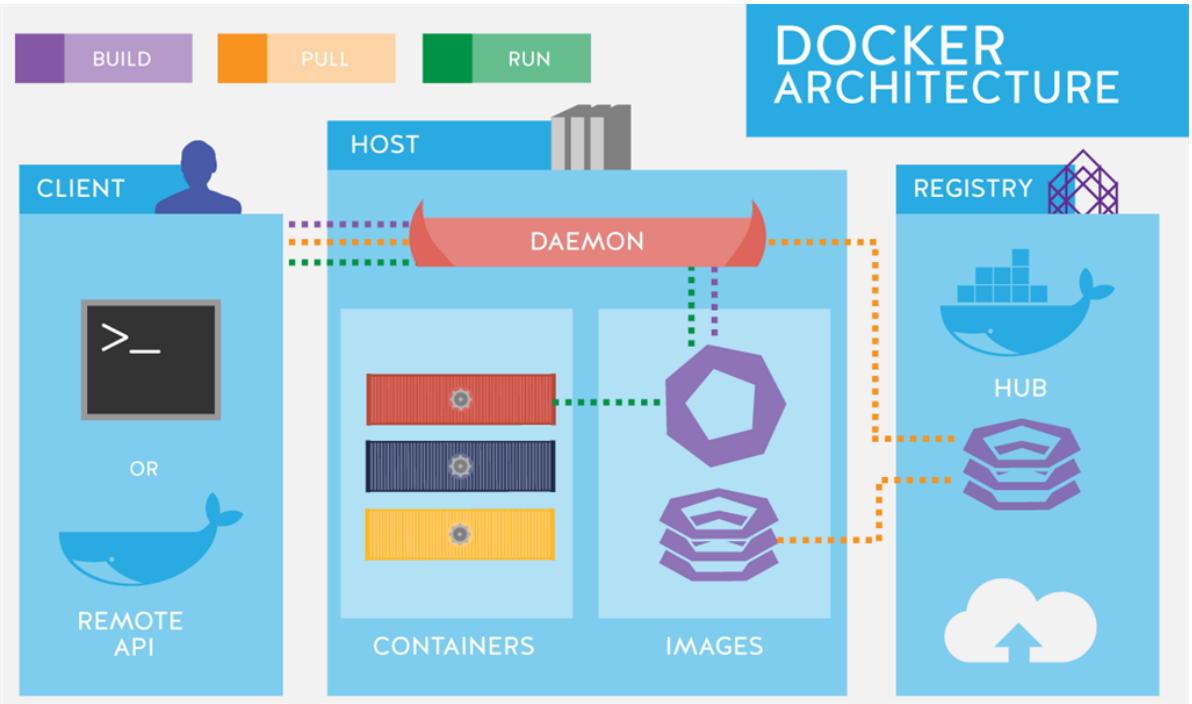

Figure 4.1: Architecture of a Docker Container

Build Command : This command used to build docker images from dockerfile.

Pull Command : This command used to pull a specific version or the latest version of the pre-built images from the registry.

Run command : This command used to start the container or to keep docker

container running for debugging mode.

4.5 Visual Studio Code

Microsoft created the source-code editor Visual Studio Code, generally known as VS Code, for Windows, Linux, and macOS using the Electron Framework. Debugging support, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git are among the features. The theme, keyboard shortcuts, options, and extensions that offer more functionality can all be changed by users.

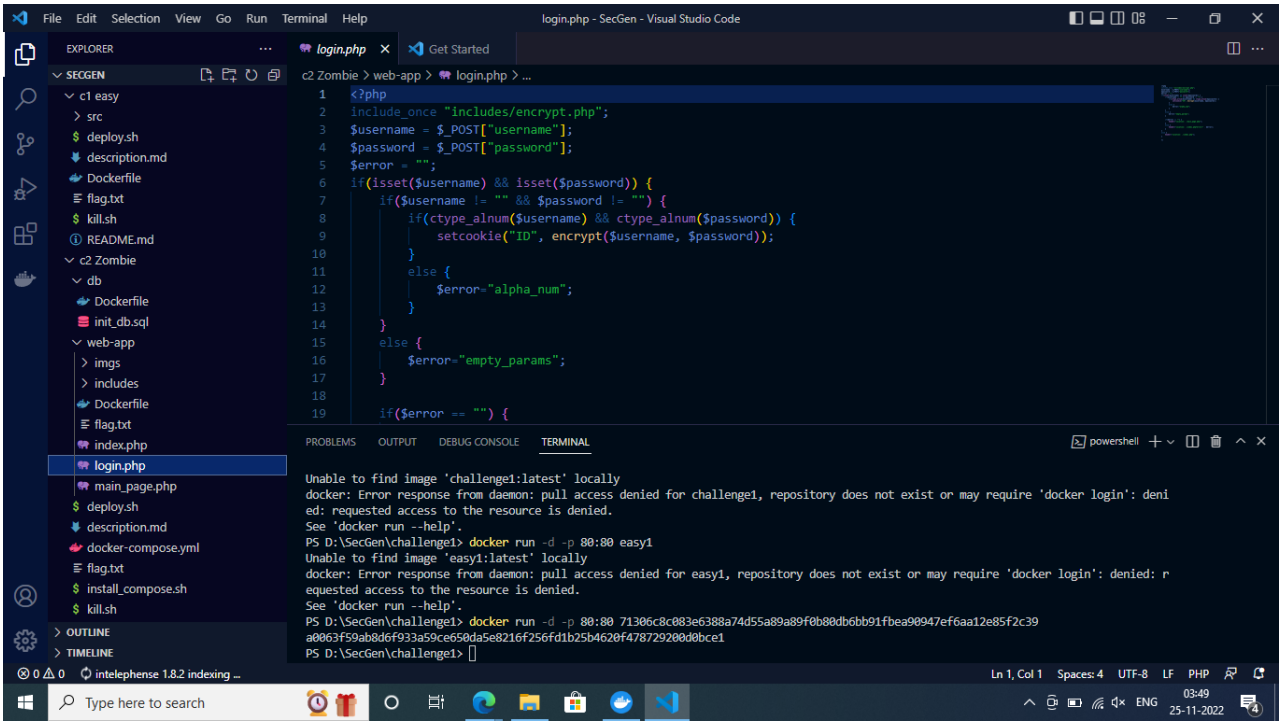

Figure 4.2: Source Code

Chapter 5

Results and Discussions

Based on figures 5.1, 5.2, 5.3, and 5.4, we believe it is feasible to operate a competition utilising the container-based infrastructure and much less resources while maintaining a comparable number of teams and services to other attack-defence CTFs. However, we were unable to test this experimentally since the single Docker daemon became overloaded due to the simultaneous launch of too many exploit containers. We believe that by properly planning the execution of exploit containers on the container host machine, this issue can be fixed. Scaling out, on the other hand, can aid in distributing the load over numerous daemons and workstations. Additionally, using numerous machines in-

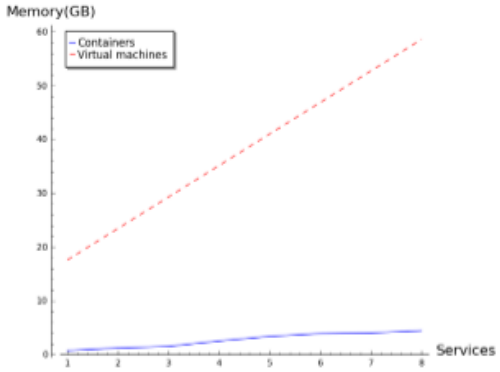

Figure 5.1: With 30 teams

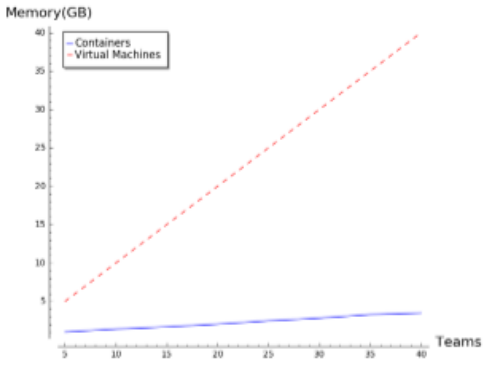

Figure 5.2: With 3 services.

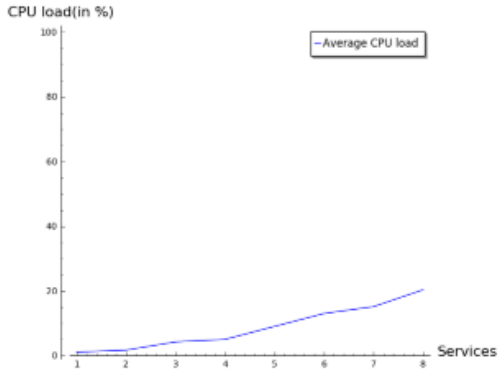

Figure 5.3: CPU Utilization.

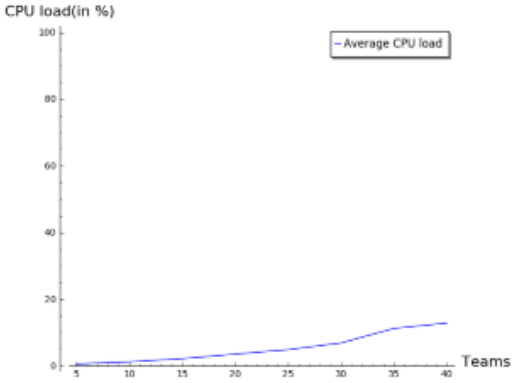

Figure 5.4: Main memory utilisation.

creases the number of ports that are accessible, making it possible to accommodate more service containers and consequently more teams.Container management may be made much easier with the use of clustering technologies like Docker Swarm. Another option is to run numerous Docker daemons simultaneously on the same machine, but this would necessitate manually bridging the containers run by various daemons, which is not an easy operation.

Chapter 6

Deployment Process

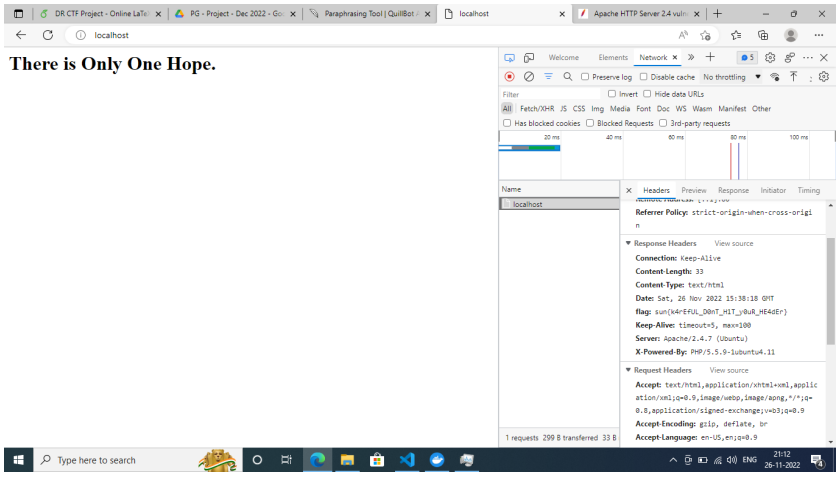

Figure 6.1: Challenge 1 : Default Misconfiguration

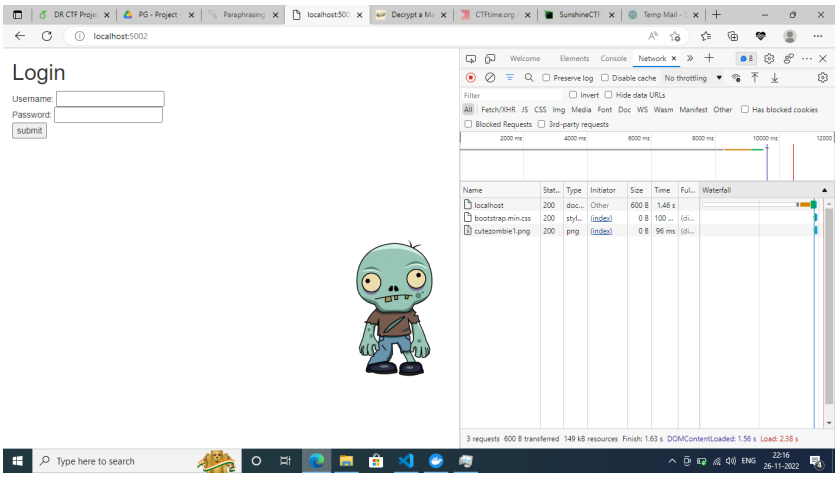

Figure 6.2: Challenge 2 : Blind SQL Injection

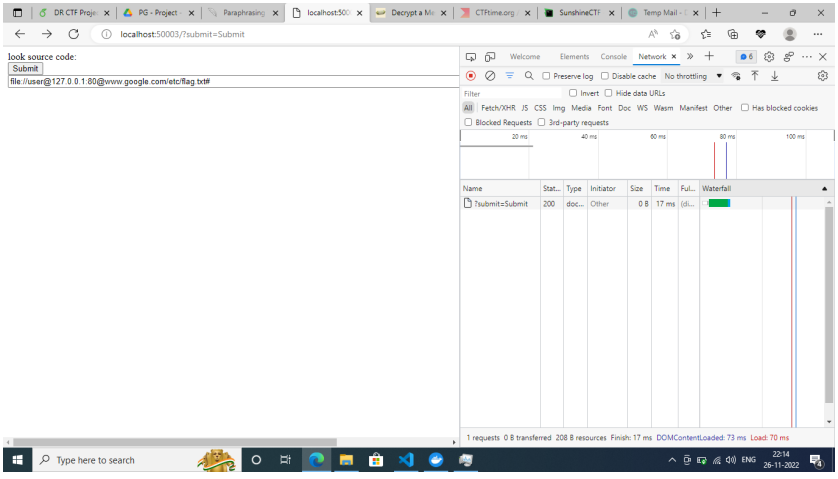

Figure 6.3: Challenge 3 : Command Injection

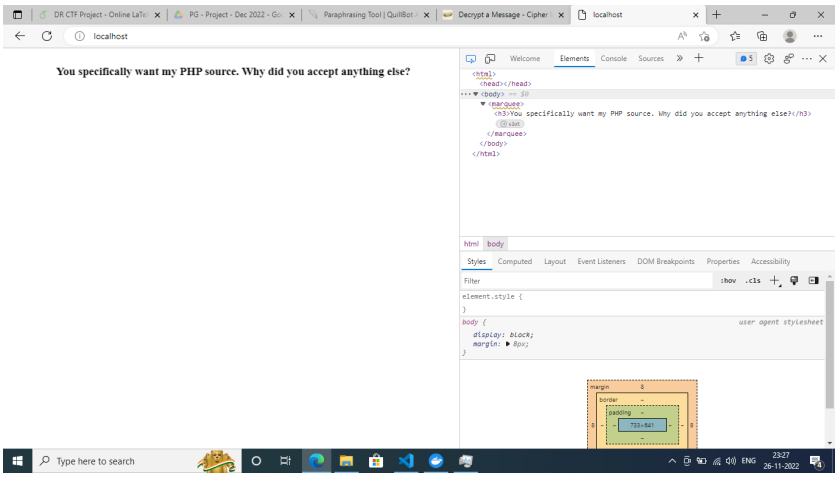

Figure 6.4: Challenge 4 : Server Side Request Forgery

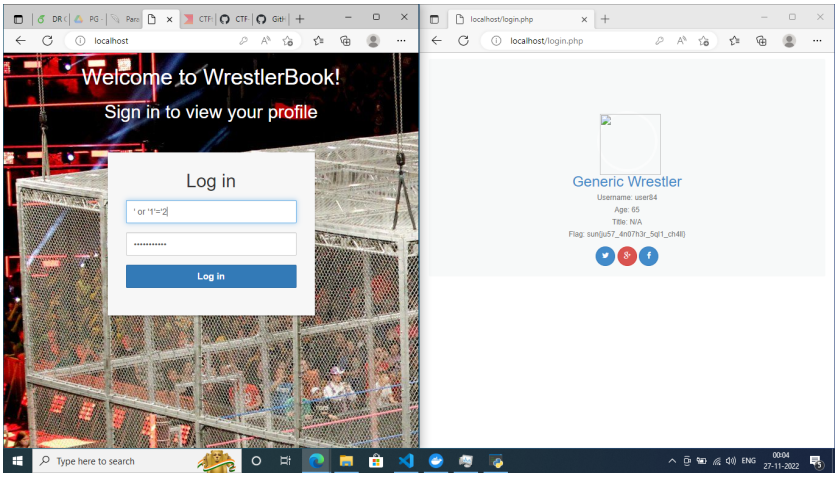

Figure 6.5: Challenge 5 : SQL Injection

Chapter 7

Conclusion

Attack-defence The dynamic and realistic gaming environments of CTFs are thought to produce higher learning outcomes. However, its mainstream acceptance has been hindered by the gameplay and infrastructure setup complexity. We provide an unique CTF infrastructure that greatly lowers resource requirements and handles several system administration duties by using Docker containers rather than virtual machines. The resource-efficient infrastructure, together with a number of readily accessible third-party tools, makes it much easier to organise such CTFs, scale them to include multiple teams, and allow players to concentrate more on learning secure coding techniques.

CTF challenges are great for honing technical abilities, but they don't cover issues like phishing and basic cybersecurity awareness. However, these issues are crucial for reducing the present sophisticated cyber threats.

Our paper makes various contributions to the fields of cybersecurity, education, and CTF participants and designers.

First, we gathered information about CTF-practiced cybersecurity subjects. Then, we talked about the ramifications of these findings and related them to the

latest developments in cybersecurity education. Finally, we provided suggestions for more researchers and instructions for future effort to inspire additional study.

7.1 Limitations & Future Works

Teams cannot download exploits from the network and reverse engineer them to find new vulnerabilities, which is one of the system's major drawbacks. To guarantee the infrastructure and exploits operate properly, another difficulty is that the exploit scheduling method should dynamically adjust as more exploits are uploaded. As part of our ongoing effort, we intend to address these problems and get a user evaluation of the system.

References

1. Arvind S Raj, Bithin Alangot, Seshagiri Prabhu and Krishnashree Achuthan (arvindsraj,bithina,seshagiriprabhu)@am.amrita.edu, krishna@amrita.edu Amrita Center for Cybersecurity Systems and Networks Amrita Vishwa Vidyapeetham: Scalable and lightweight CTF infrastructures using application containers.

2. Theodoor Scholte, Davide Balzarotti, and Engin Kirda. Have things changed now? An Empirical Study on Input Validation Vulnerabilities in Web Applications. Computers Security, 31 (3):344–356, 2012.

3. Victor van der Veen, Nitish dutt Sharma, Lorenzo Cavallaro, and Herbert Bos. Memory Errors: The Past, the Present, and the Future. Research in Attacks, Intrusions, and Defenses, pages 86-106, 2012.

4. Clark Taylor1,3, Pablo Arias2,3, Jim Klopchic3, Celeste Matarazzo3, and Evi Dube3: CTF: State-of-the-Art and Building the Next Generation

5. Vykopal, J., and Barták, M. On the design of security games: From frustrating to engaging learning. In 2016 USENIX Workshop on Advances in Security Education (ASE 16) (Austin, TX, 2016), USENIX Association.

6. Werther, J., Zhivich, M., Leek, T., and Zeldovich, N. Experiences in cyber security education: The mit lincoln laboratory capture-the-flag exercise. In CSET (2011).

7. Zendler, A. Computer science education teaching meth- ods: An overview of the literature. International Journal of Research Studies in Computing 4, 2 (2015).

8. Datta, A., Hatti, A., Ajith, Mahajan, A., Shri- vastava, A., Bhargava, A., Shah, A., Machiry, A., Jakhar, A., Das, H., Forshaw, J., Bharmal, M., Mahajan, P., Walikar, R., Saint, Mittal, S., and Chauhan, S. Nullcon hackim 2017. http://ctf.nullcon.net/.

9. Danihan, M., and Duggan, S. Owasp security shepherd. https://www.owasp.org/index.php/OWASP_Security_Shepherd.

10. Hamari, J., Koivisto, J., and Sarsa, H. Does gam- ification work?–a literature review of empirical studies on gamification. In System Sciences (HICSS), 2014 47th Hawaii International Conference on (2014), IEEE, pp. 3025–3034.

11. R. Sahay et al.The application of software defined networking on securing computer networks: a survey J. Netw. Comput. Appl.(2019)

12. U. Sarmah et al. A survey of detection methods for xss attacks J. Netw. Comput. Appl.(2018)

13. Stela Kucek, Maria Leitner, An Empirical Survey of Functions and Configurations of Open-Source Capture the Flag (CTF) Environments.(2020)

14. SeongIl Wi, Jaeseung Choi, and Sang Kil Cha, KAIST: Git-based CTF: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition.

15. MATIAS, P., BARBOSA, P., CARDOSO, T., MARIANO, D., AND ARANHA, D. NIZKCTF: A non-interactive zero-knowledge capture the flag platform. CoRR abs/1708.05844 (2017).

16. PHAM, C., TANG, D., CHINEN, K.-I., AND BEURAN, R.CyRIS: A cyber range instantiation system for facilitating security training. In Proceedings of the Symposium on Information and Communication Technology (2016), pp. 251–258.

17. TRICKEL, E., DISPERATI, F., GUSTAFSON, E., KALANTARI, F., MABEY, M., TIWARI, N., SAFAEI, Y., DOUPE´, A., AND VIGNA, G. Shell we play a game? CTF-as-a-service for security education. In Proceedings of the USENIX Workshop on Advances in Security Education (2017).

18. BEURAN, R., PHAM, C., TANG, D., ICHI CHINEN, K., TAN, Y., AND SHINODA, Y. CyTrONE: An integrated cybersecurity training framework. In Proceedings of the International Conference on Information Systems Security and Privacy (2017), pp. 157–166.

19. BURNS, T. J., RIOS, S. C., JORDAN, T. K., GU, Q., AND UNDERWOOD, T. Analysis and exercises for engaging beginners in online CTF competitions for security education. In Proceedings of the USENIX Workshop on Advances in

Security Education (2017)

20. VYKOPAL, J., OSLEJ ˇ SEK ˇ , R., C ˇ ELEDA, P., VIZVARY, M., AND TOVARN ˇ AK´ , D. KYPO cyber range: Design and use cases. In Proceedings of the International Conference on Software Technologies (2017)

21. Yang, S.; Wang, X.; Wang, X.; An, L.; Zhang, G. High-performance docker integration scheme based on OpenStack. World Wide Web 2020, 23, 2593–2632.

22. de Leon, D.C.; Goes, C.E.; Haney, M.A.; Krings, A.W. ADLES: Specifying, deploying, and sharing hands-on cyber-exercises. Comput. Secur. 2018, 74, 12–40.