



Saim Bin Zahid

01-134182-049

Sharjeel Sohail

01-134182-087

HACKADEMY - Vulnerable Machine Development for Cyber Drills

Bachelor of Science in Computer Science

Supervisor: Dr. Faisal Bashir

Department of Computer Sciences
Bahria University, Islamabad

June 2022

Certificate

We accept the work contained in the report titled “**Hackademy - Vulnerable Machine Development for Cyber Drills**”, written by **Sharjeel Sohail** and **Saim Bin Zahid** as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved By:

Supervisor: Dr. Faisal Basheer

Internal Examiner:

External Examiner:

Project Cordinator: Dr. Moazam Ali

May 27th, 2022

Abstract

With the advent of modern technologies and how everything is connected to the internet whether it is as personal as lifestyle products such as microwave ovens etc. or things that affects the masses such as nuclear powers, military operations and banking systems etc. All these are converting into IoT (Internet of Things) products. Even though there are countless benefits of them performing their designed task through the internet such as internet provides them reliability, compatibility and scalability etc. We still cannot ignore the fact that by getting more technologically advanced, the chances of becoming a victim of cyber crimes also increases. Cyber crimes refer to the attacks on a network, typically the internet. Cyber attacks are targeted on the confidentiality, availability and functionality of data. It is important to have professionals with enough capabilities to fight against these cyber crimes and help strengthen the internet and the byproducts of the internet in order to avoid becoming a victim of these cyber crimes. To overcome this need, Hackademy is created. Hackademy is a pen testing environment for such professionals or trainees to work on real life simulation of common cyber attacks so they can learn how to exploit the vulnerabilities in machines and systems in order to find out ways to strengthen those machines and systems. Hackademy is a user friendly platform that provides its functionalities to cyber security professionals of any level whether they are beginners or relatively professionals.

Acknowledgments

By the Grace of Allah Almighty we have completed this project. We might want to communicate our most profound thanks and appreciation towards Dr Faisal Bashir who allowed us the opportunity to deal with this Final year project, his supervision and his direction demonstrated extremely effective in making this Project Complete. He given us the reference material and numerous other undertaking related materials that not just helped us making the undertaking easily yet in addition finishing it on time.

We would also like to say Thanks to Sir Nouman Mushtaq (Team Lead CRC) and Sir Fawad (Research Assistant) for his consistent direction and his broad information regarding the Project that helped us a ton in the undertaking and was essential in its consummation.

SHARJEEL SOHAIL, SAIM BIN ZAHID
Bahria University Islamabad Campus
Islamabad, Pakistan

April 30, 2022

*“We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn’t anyone”*

Regina Dugan

Contents

Abstract	i
1 Introduction	1
1.1 Overview	1
1.2 Problem Description	3
1.3 Project Objective	3
1.4 Project Scope	3
2 Literature Review	5
2.1 Scholarly Contributions	5
2.2 Existing Systems	11
2.3 Proposed Systems	12
3 Requirement Specifications	13
3.1 Existing Systems	13
3.2 Proposed System	13
3.3 Requirements Specification	14
3.3.1 Functional Requirements	14
3.3.2 Non-Functional Requirements	14
3.4 Use Cases	14
3.4.1 General Workflow	15
3.4.2 Register User	15
3.4.3 User Login	16
3.4.4 View and Select machine	17
3.4.5 Configure machine	18
3.4.6 Exploitation	19
3.4.7 Verification	20
3.4.8 Update Record	21
4 System Design	23
4.1 System Architecture	23
4.2 Design Constraints	24
4.3 Design Methodology	24
4.4 Component Diagram	25
4.5 Process Interaction Diagram	25
4.6 Workflow Diagram	26
4.7 Sequence Diagram	27
4.7.1 Login Page	27
4.7.2 Operation of Hackademy	27
4.7.3 CTF Claiming	28

4.8	GUI Design	29
5	System Implementation	33
5.1	System Architecture	33
5.1.1	Presentation Layer (GUI)	33
5.1.2	Application Layer	33
5.1.3	Virtual Layer	33
5.2	System Requirements	34
5.2.1	Activity Requirement	34
5.2.2	Hardware Requirement	34
5.2.3	Software Requirement	34
5.3	Tools and Technologies	34
5.3.1	MongoDB Database	35
5.3.2	Linux OS	35
5.3.3	REST API	36
5.3.4	Angular	36
5.4	Development Environment and Languages	36
5.4.1	Microsoft Visual Studio Code	36
5.4.2	Typescript	36
5.4.3	Django	36
5.5	Vulnerable Machine Development Process	37
5.5.1	SecGen	37
5.5.2	Code Snippet for API	37
5.5.3	Code Snippet for Machine Configuration	38
5.5.4	Flow Chart for Machine Creation	41
5.6	FRONTEND DEVELOPMENT	41
5.6.1	Client-Side Dashboard	42
5.6.2	Attack Page	42
5.6.3	Machine Page	43
6	System Testing And Evaluation	44
6.1	Graphical User Interface GUI Testing	44
6.2	Usability Testing	44
6.3	Compatability Testing	44
6.4	Performance Testing	45
6.5	Exception Handling	45
6.6	Installation Testing	45
6.7	Test Cases	45
6.7.1	Application Starting test case	46
6.7.2	Successful Login	46
6.7.3	Pages are accessible	46
6.7.4	User Sees a list of Machines	46
6.7.5	Session Time Working	47
6.7.6	CTF Claimed	47
6.7.7	Feedback successfully saved	47
6.7.8	Admin Successfully adds, deletes machines	48
6.7.9	Admin giving permissions to sub admins	48
7	Conclusion	49
7.0.1	Future Work	49

A	User Manual	51
A.1	User’s Side	52
A.1.1	Landing Page	52
A.1.2	Dashboard	53
A.1.3	Machines	54
A.1.4	Challenges	55
A.1.5	Profile	56
A.1.6	Attacks	57
A.2	Administrators Side	58
A.2.1	Landing Page	58
A.2.2	Admins Landing Page	59
A.2.3	Machines Creation	60
A.2.4	CTF Challenges Page	61
A.2.5	Web Challenges Page	62
A.2.6	Attack’s List	63
	Bibliography	64

List of Figures

2.1	Starmine Visualization	6
2.2	Starmine Visualization (3D)	6
2.3	Attack Classification Diagram	7
2.4	KYPO ARCHITECTURE	8
2.5	CERN DEFACEMENT STATISTICS	9
2.6	InCTP Architecture	9
2.7	Comparative Study	10
2.8	Comparative Study continued	11
3.1	General Overview of the use case	15
3.2	Register User	16
3.3	Use case 1 Information	16
3.4	User Login	17
3.5	Use case 2 Information	17
3.6	View and select machines	18
3.7	Use case 3 Information	18
3.8	Configuring machine	19
3.9	Use case 4 Information	19
3.10	Exploitation	20
3.11	Use case 5 Information	20
3.12	Verification of Key	21
3.13	Use case 6 Information	21
3.14	Update Record	22
3.15	Use case 7 Information	22
4.1	Architecture of Hackademy	24
4.2	Component Diagram of Hackademy	25
4.3	Process Interaction Diagram of Hackademy	26
4.4	Workflow Diagram of Hackademy	26
4.5	Login Sequence	27
4.6	Sequence Diagram	28
4.7	Sequence Diagram of CTF Achieved	29
4.8	Hackademy Logo	29
4.9	Login and Sign Up Page	30
4.10	User Home Page	30
4.11	Attack Description Page	31
4.12	Available Machines List	31
4.13	Challenges Categories	32
4.14	User Profile	32

5.1	Linux Architecture	35
5.2	Django's Architecture	37
5.3	Flowchart of machine creation	41
5.4	Client Side Dashboard	42
5.5	Attack List Page	43
5.6	Available Machines Page	43
A.1	Login Page	52
A.2	User's Home Page	53
A.3	Machine List Page	54
A.4	Challenges Page	55
A.5	User Profile Page	56
A.6	Attack List Page	57
A.7	Login Page	58
A.8	Admin Home Page	59
A.9	Machine Generation Page	60
A.10	Active CTF Machines	61
A.11	Active Web Machines	62
A.12	Attacks	63

Chapter 1

Introduction

Throughout years mankind has always been a puppet to change, whether they like it or not. From the medieval times to the early modernization mankind has done many innovations. The reason for the advent of every innovation will always be deduced down to the need of that innovation. The word innovation was combined or compared with various synonyms throughout ages, In the medieval times Innovation was used with basic human needs for example, innovating different ways to use fire as a basic human need in making meals and keeping away the cold. The advent of a wheel was considered a major innovation in those times.

After the second industrial revolution, the usage of innovation was binded with the use of technology and till date in the 21st century, It is still used with technology, Digitalization is by far the greatest innovation category that mankind came up with. With each passing day human are using technology to ease up their everyday life chores. It is said that the amount and quality of innovation done in the 21st century is far more than any change happened in the past.

This chapter is a brief introduction of how the advent of innovation in 21st century is also giving way to cyber crimes and how our product can help fight it.

1.1 Overview

The word Cyber referred to the control and behavior of machines and it was formed in the 50s. Modern cyber refers to a network of computing nodes that are working with each other. Since they are working together then there is always a mean of communication that forms a network. This network shares data, resources, and system information etc. The best example of Cyber is the world wide web aka INTERNET. An internet is a global network therefore anything that has anything to do with the internet, falls under the category of CYBER.

It can be said that the internet is a collection or a set of many networks, thus it can be said that the internet requires exchange of data which can bring about risks and danger along with it. We will talk about how these risks gives its way to unauthorized data breaching and how that affects the confidentiality and privacy of its user's whether the user is general public, governmental offices or national security.

Cyber Threat or Cyber-attacks are the terms used for online attacks on the user data and control. A Cyber attack is an attack that cyber criminals inject within

the network by breaking through loopholes in the network or the program. The intensity of the attack depends on the intent of the malicious attacker/hacker. While most attacks are mostly nuisances, but some attacks are very intense and can even endanger human lives. The intention of these attacks can be broken down into two basic categories such as.

i.) Non-Malicious Attacks:

These kind of attacks are not intentional and can be caused due to negligence of the employee. Even though the intention of harming the network is not their, still it can cause great harm to the system if not dealt with.

ii.) Malicious Attacks:

These kind of attacks are intentional and they fall under criminal activity. The attackers generally attack the system in order to get revenge, get some ransom or just out of fun. These can be extremely deadly based on the intention of the attacker

There have been many cyber attacks throughout history and they are mostly targeted on the following applications.

i.) Military Equipment Failure

ii.) Power Blackouts

iii.) Banking and accounting sectors

iv.) Security Breaches etc.

Throughout years the internet has recorded many cyber attacks [4] and have given them specific titles based on their intentions and how they work. Following are a few common cyber attacks.

i.) Phishing:

Tricks email recipient into disclosing confidential information.

ii.) MitM:

Where the hacker finds an intermediate between sender and receivers.

iii.) Trojans:

Where the hacker enters the system in disguise (typically a standard piece of software).

iv.) Ransomware:

An attack where the hacker encrypts the victim's data and demands a ransom in return for the original data.

1.2 Problem Description

With the use of technology increasing day by day and almost everything is converting into the online world whether it is ecommerce, bank transactions, medical billing and history, Military data and many other. There will always be a risk of getting cyber attacked which if injected right can compromise the life of everyone since the internet is connected to everyone and everywhere. Thus there is a need for a security mechanism or strategy that provides everything in the pool of internet, a secure trading, communicating and transactions.

A need for professionals is arised. Professionals that are well qualified and well equipped to provide their services in order to keep the internet and also private networks secure. The job of these cyber security professionals is to keep check and balance to the systems and software. These security professionals are required to find the loopholes and any possible point of entry that attackers might use in order to have access to the network. They are also required to look out for vulnerabilities in the networks or softwares and give their report on how to tackle and fix these vulnerabilities so that it is prone to any cyber exploitation.

A question arises about how can these professionals be well qualified without trying black hat techniques to learn and grow. There needs to be a system or a platform where individuals who wants to be professionals in the field of cyber security can train themselves. There is a need for that platform where these individuals of any level whether they are beginners or experts are given accessibility to break the walls of custom designed networks or software.

1.3 Project Objective

The objective of this project is to.

- i.) Design and develop a solution or a platform where cyber security individuals and trainees can have hands on experience on real life cyber attacks.
- ii.) Provide a user friendly, fully functional application where trainees of any level can learn and perform their pen testing skills in order to exploit vulnerabilities without using any black hat means.
- iii.) Create a cyber range consisting of virtual machines that can work autonomously on the user's system without interfering with the host OS or host applications. This way the trainees can work on vulnerabilities on their locally integrated system without worrying about compromising their systems integrity.

1.4 Project Scope

This application aims to successfully create an operational simulation of common real life cyber threats within a virtual machine or docker. The idea is to organize each machine in terms of its level of difficulty and the vulnerability it have and then successfully providing the vulnerable machine to the end user (trainee).

Each user will have their own dashboard and profile where the score and statistics of their work will be tabled. Their practice history along with their achievement should also be listed successfully. Each user should only have access to their own instance of the machine and their work should not entangle with the work of a fellow trainee.

Each of these vulnerable machines are integrated in the host system as if they are working autonomously. These virtual machines with vulnerabilities cannot disturb the applications and the Operating system of the host machine and thus the trainee can perform their practice without worrying about the host system being compromised.

Chapter 2

Literature Review

In this chapter we will discuss past studies and research in the field of Cyber security. There has been many studies and scholarly papers in this regard. The purpose of this chapter is so we can learn about what different scholars has to say about said project and how their studies helps in the production of our project. We will also take into consideration about similar projects that were worked on in the past and how we can use that in this project. [7]

2.1 Scholarly Contributions

In this section, we will discuss the contributions and scholarly papers of different research where they have provided their findings and theoretical solutions in the field of the cyber security. With the increase in the use of internet, many researchers consider it extremely important to utilize technologies and knowledge in terms of cyber security.

Authors Solms R and van Niek J in their article [1] [11] researched about the definitions of cyber security and information security and how they are not that similar to each other. In their study they studied that although both these statements are different regions of the same world. They concluded that Information Security is the protection of individuals or organizational information that can be in any form of data. The term cyber security refers not only to the protection of information or cyberspace but also the protection of human targets. This scholarly article added an extra dimension to the cyberspace having ethical implications to humans and their unknowing contribution or participation in threats.

[6] Author Joseph S. Nye Jr. in his research told the readers about foreign policies needed and the comparison of two of the most groundbreaking technologies that are capable of disasters. The Inception of Nuclear arsenal and the world of Cyber Security. Joe wrote that the difference between the two worlds is great. He talked about how after nuclear inception, all nations had to come up with plans to cooperate with each other to avoid the usage of these nuclear arsenals, Cyberspace should be given the same importance by the world as it can cause harm that can be in some context worst than that of a nuclear threat. The biggest difference and the reason that gives cybersecurity an upper hand than nuclear arsenal is that in cyber technology the powers comes down to many nonstate actors and the threat is increasing day by day.

STARMINE [2] is a cyber threat monitoring system which is a more efficient version of traditional cyber threat monitoring systems like DShield and SANS. Traditional monitoring systems typically used either one or both of the following visualizations.

- i.) Geographical Visualization
This visualization was used to detect active attacks(sources/destination) geographically whether it is continent wise or country wise.
- ii.) Temporal Visualization
Used to answer the “When” of a attacks.

Nowadays cyber threats are advanced the attackers uses IP jumps for the source and destination of these attacks. To detect this there was a new visualization called “Logical Visualization”.

STARMINE uses all three of these visualizations for threat monitoring which helps in the analysis and predictive decisioning to avoid these attacks.

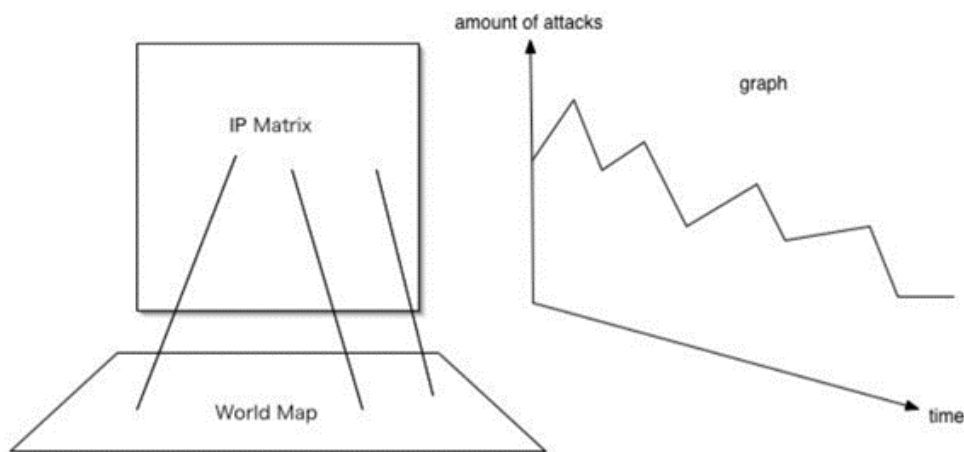


Figure 2.1: Starmine Visualization

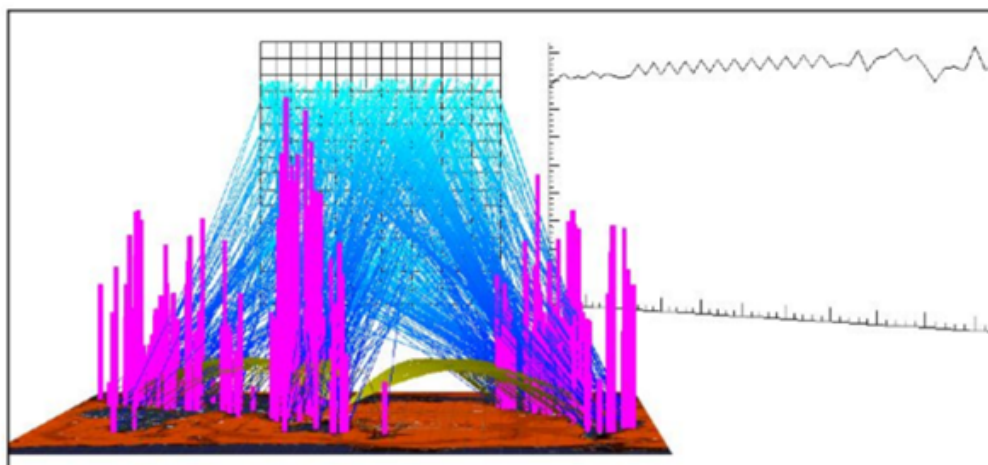


Figure 2.2: Starmine Visualization (3D)

M. Uma and G. Padmavathi in their paper researched [9] comprehensively on all of the recorded cyber-attacks and its classifications which makes it easier to analyze and study ways to create appropriate defense mechanisms. They also studied on the characteristics and motivations behind different attacks and how they work.

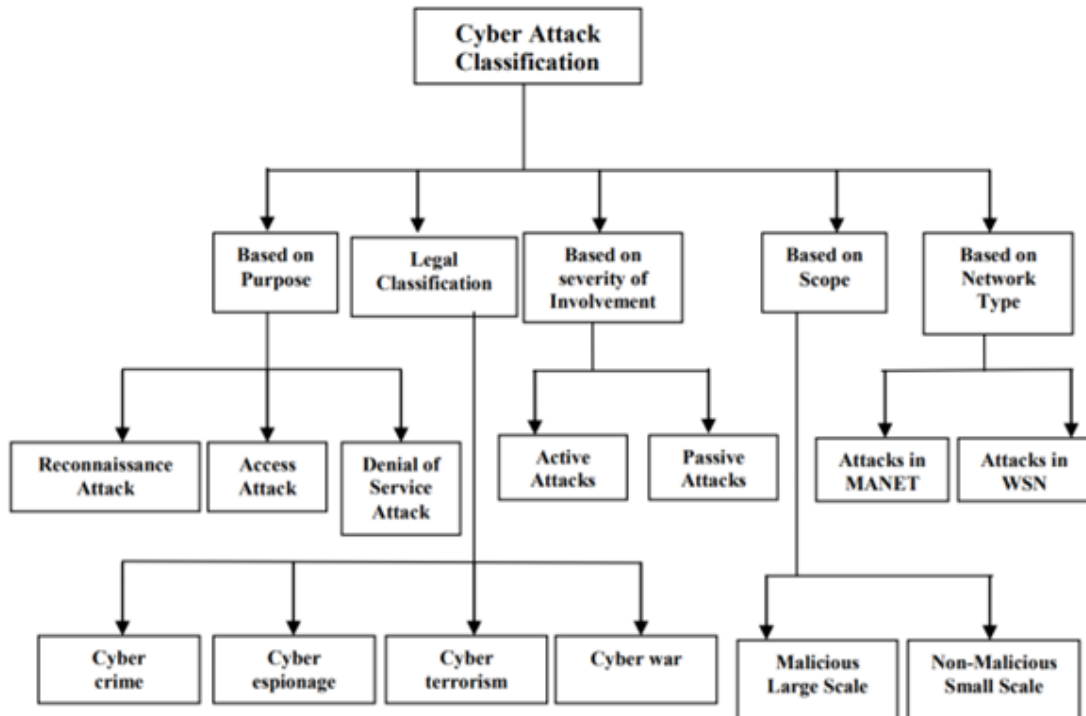


Figure 2.3: Attack Classification Diagram

A set of 18 countries or nations set out their respective strategies to avoid cyber-attacks and to enhance their cyber security. This set of strategies or NCSS (National Cyber Security Strategy) were researched on by Eric Luijff and Kim Besseling in their research paper [5]. This scholarly paper discusses about the strengths, weaknesses along with the similarities and differences of the routes taken by these nations for cyber security. The reason for letting out these strategies were so that analyst could work out ways that would be best in strengthening the security. This could help nations [2], organizations to adopt these strategies and to resolve any miss-happening by predicting them.

KYPO Cyber Range [12] is a PAAS system that creates a simulation of real-world computing systems on the cloud. It uses a virtualized distributed computing network used for the training of white hats. In this research paper by Vykopal, a comprehensive study about KYPO cyber range was done. KYPO gives their trainees a hands-on training environment where they can face real world cyber threat scenarios. KYPO uses CTF based exercises to teach their trainees on sandbox based virtual machines so that any activity stays inside the box and doesn't harm unnecessary files/data.

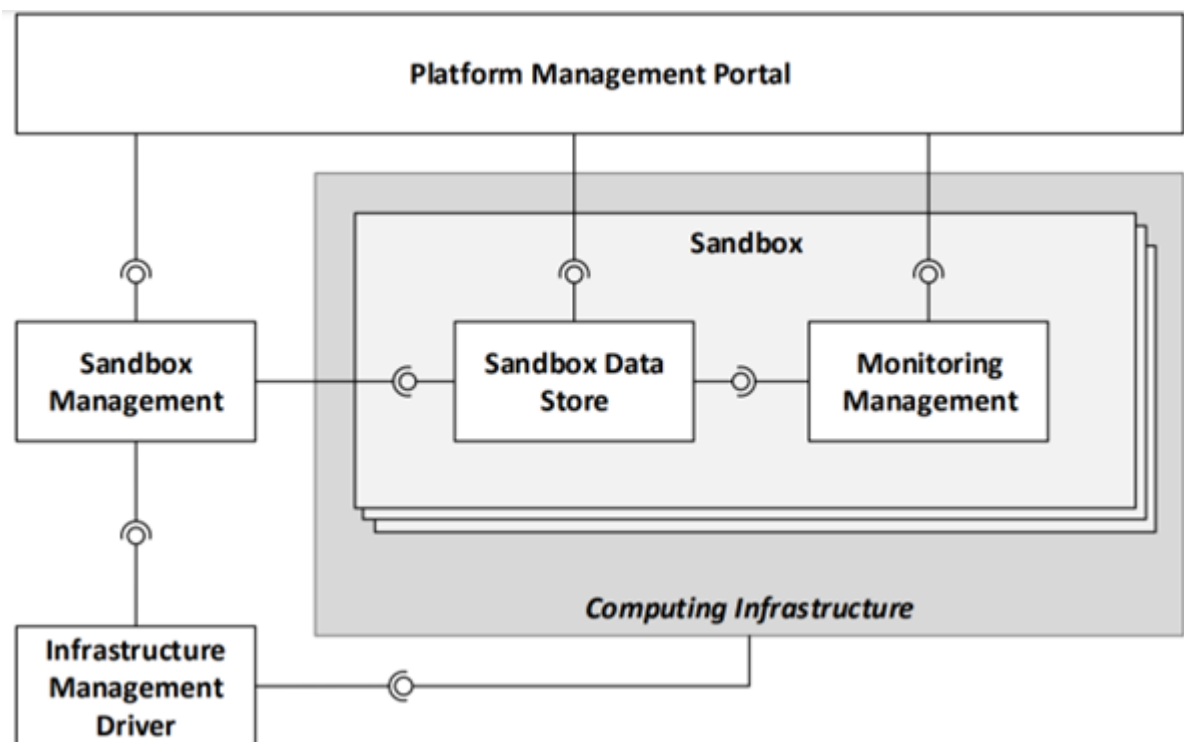


Figure 2.4: KYPO ARCHITECTURE

[13] Jan Vykopal along with fellow researchers had a comprehensive study about the creation of cyber drills and difficulties occurring in the creation of those drills. In this paper they presented their idea about the general exercise lifecycle which covered.

- i.) Preperation
- ii.) Dry Run
- iii.) Execution
- iv.) Evaluation
- v.) Repetition

This lifecycle was a to-do list while making any new activity/exercise for these trainees or to update current activities based on evaluation analysis.

A statistics study in Jan 2014 by CERN said about the defacements of domains on the world wide web.

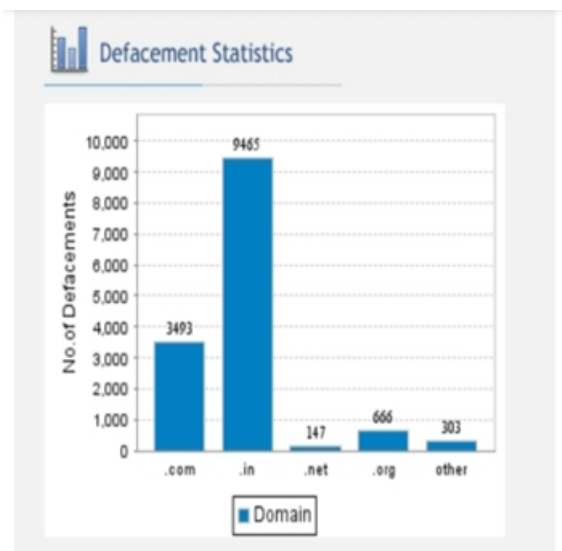


Figure 2.5: CERN DEFACTION STATISTICS

This alarming rate made researchers [1] K. Boopathi, S. Sreejith and A. Bithin wonder whether there are enough cyber security professionals with adequate knowledge who can fill the void in cyber market. In this paper they researched how they need to up the game of cyber range[3] [?] by bringing ‘InCTF’ based gaming approach to cyber security ranges. This gaming approach was to test cyber security trainees at various levels in a fun manner and to find out the loopholes accidentally made by the countries development team[14] .

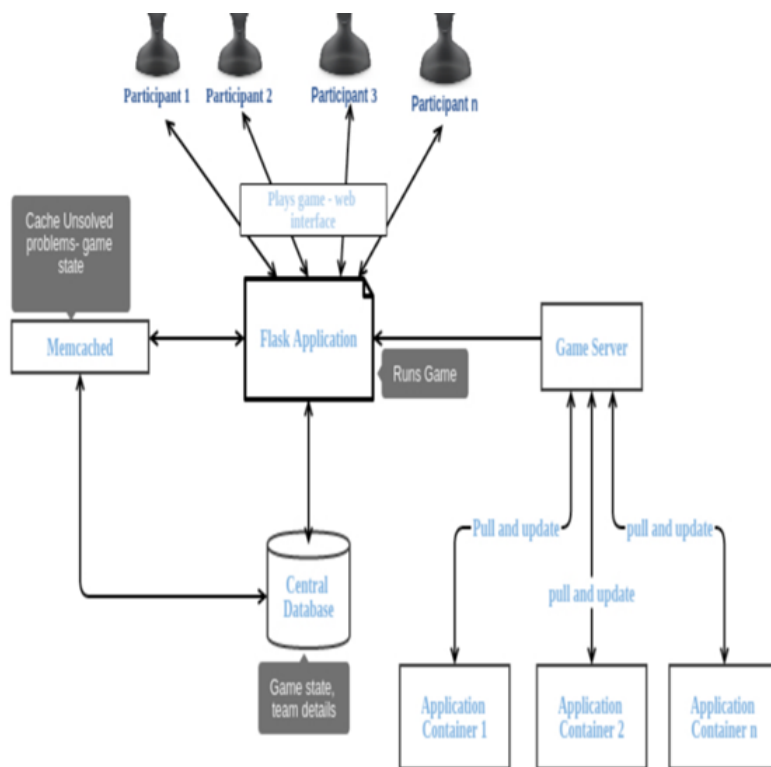


Figure 2.6: InCTP Architecture

SL No.	TITLE	YEAR OF PUBLICATION	DESCRIPTION	FEATURES
1	Cyber warfare: Issues and challenges	2015	Tells about the challenges faced in cyber war based on historical analysis	The paper identifies nine research challenges and analyzes contemporary work carried out in each along with future work to understand them better in order to avoid cyber warfare.
2	From Information Security to Cyber Security	2013	Additional dimension of human targets in the definition of Information Security and Cyber Security.	
3	Security Scenario Generator (SecGen)	2017	Improved cyber range simulation platform based on random scenarios.	Random Scenario Generator, Multiple Modules eg. Base, Network, Vulnerability, service, encoder etc.
4	Nuclear Lessons for Cyber Security	2011	Lessons from nuclear inception and foreign policies regarding cyber space.	
5	Starmine: A Visualization System for Cyber Attacks	2006	Added an upgrade to traditional cyber attacks monitoring systems	Integrates Geographical, Temporal and logical views for cyber attacks monitoring systems.
6	A Survey on Various Cyber Attacks and Their Classification	2013	Classified recorded attacks on the basis of motivation and characteristics	
7	Nineteen national cyber security strategies	2013	Strategies used to enhance cyber security.	
8	KYPO Cyber Range	2018		Uses Distributed Computing Networks for Cyber trainings

Figure 2.7: Comparative Study

9	Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range	2017	Made Exercise lifecycle for cyber drills	
10	AIT Cyber Range: Flexible Cyber Security Environment for Exercises, Training and Research	2021	Virtual Environments that supports the simulation of diverse infrastructures for cyber trainings.	Scenario Engine, Infrastructure Provisioning, Software Provisioning
11	Online Assessment for hands-on cyber security training in virtual lab	2012	Gives introduction to what cyber ranges are and how a traditional cyber range works	
12	CyRIS: A Cyber Range Instantiation System for Facilitating Security Training	2016	Prepares cyber ranges according to instructors specifications	System Configuration, Tools Installation, Incident Emulation, Content Management, Clone Management
13	Towards Pentesting Automation Using the Metasploit Framework	2020	Automated Penetration testing technique based on Metasploit framework	The model was trained by analyzing exploited machines from HacktheBox. The proposed model can exploit a fair number of systems.
14	Learning Cyber Security Through Gamification	2015	Made a fun Gaming UI for cyber trainees	
15	Stuxnet and the Limits of Cyber Warfare	2013	Talks about the first instance of a computer network attack and how to avoid from such attacks.	Analyzes Stuxnet attack and gives a theoretical estimate of the limits of cyber warfare based on probability calculations.

Figure 2.8: Comparative Study continued

2.2 Existing Systems

There are many cyber ranges in the market that works on CTF based cyber warfare training ground such as HacktheBox, Penteston and Vulnhub etc. These existing systems mostly use traditional technologies and methods to create host based virtual machines. The shortcoming of deploying virtualization in the hosts machine is that it will be costly and less reliable. Using this method increases dependency for the user/trainee who loses application traffic. Other than this some of the existing system's target's mostly on current cyber security professionals and does not focus much on beginners. For Example: HackTheBox requires its new user to use hacking tools and techniques in order to sign up in their dashboard for the first time, This step prevents beginners in getting access to their product in order to learn.

2.3 Proposed Systems

Hackademy caters to all these problems. Since availability of the system is the top priority of hackademy, thus hackademy uses tools and techniques to make the system available to its user irrespective of the infrastructure. Hackademy uses docker containers and images which makes the system scalable, available and fault tolerant. Dockerizing the vulnerable machines makes multiple instances of the vulnerable apps cluster of containers thus the resource sharing becomes flexible. Another important feature of hackademy unlike a few competitors is that hackademy is made for cyber trainees of any level unlike HackTheBox. Hackademy does not restrict its users to hack into the login system just to have access to the user dashboard.

Chapter 3

Requirement Specifications

In this chapter, the requirements and specifications of the system are discussed. Some already existing applications related to HACKADEMY are also discussed. An overview of proposed project along with its benefits are briefly discussed. Requirement specifications along with functional and non functional expectations of said project is also explained in the Requirement Specifications chapter.

3.1 Existing Systems

In this section we will take a look at the work that has already been done in selected field. There are many cyber ranges in the market that works on CTF based cyber warfare training ground such as HacktheBox, Penteston and Vulnhub etc. These existing systems mostly use traditional technologies and methods to create host based virtual machines. The shortcoming of deploying virtualization in the hosts machine is that it will be costly and less reliable. Using this method increases dependency for the user/trainee who loses application traffic. Other than this some of the existing system's target's mostly on current cyber security professionals and does not focus much on beginners. For Example: HackTheBox requires its new user to use hacking tools and techniques in order to sign up in their dashboard for the first time, This step prevents beginners in getting access to their product in order to learn.

3.2 Proposed System

HACKADEMY will be Pakistan's first cyber range web application which will consist of cyber drills for different level of trainees whether they are beginners or are experts. Hackademy will not use traditional virtual machines but will convert those vulnerable virtual machines into docker images. This method increases the reliability and efficiency of the system. Other than this, Hackademy is designed such that it can cater different levels of trainees/professionals whether they are beginning their career in cyber security or are already a part of it and wanting to grow better in their testing skills. Hackademy consists of interactive user friendly interface and contains a number of machines categorized in different difficulty level so any professional of any level can use said product.

3.3 Requirements Specification

Requirement Specifications For HACKADEMY – Vulnerable machine deployment for cyber drills are made by keeping in mind user interaction with the system, Admin’s interaction with the system and how these machines will work under certain constraints and conditions.

3.3.1 Functional Requirements

Functional requirements list down the activities that are mandatory for the system to perform.

- i.) Administrators should be able to create, add, deletedelete, and update vulnerable machines.
- ii.) Users/Trainees should be able to register and login to their dashboards.
- iii.) Users should have access to specified machines.
- iv.) System should be able to make instances of machines that does not disturb other instances
- v.) Sandbox environment should uphold in the training environment.

3.3.2 Non-Functional Requirements

Non-Functional requirements measure down how well our proposed system will work.

- i.) The system should be reliable when deploying machines.
- ii.) The system should have a controlled cost for maintenance.
- iii.) The system should be open to changes and upgrades.
- iv.) The system should keep in mind space and time constraints and should not be a toll on either.
- v.) The machines should be available for the users whenever required.

3.4 Use Cases

HACKADEMY is web application based on cyber range that is aimed at cyber trainees or individuals who want to learn pen testing / white hat hacking but cannot find a legal platform where they can get hands-on practice of real-world cyber space. The main idea is to create enough security professionals so that organizations whether private sector or public sector including governments, defense agencies can hire these professionals.

A user can login to their respective web portal with private credentials and choose from a list of activities/machines/vulnerable machines depending on the users’ level of experience. Once chosen they can continue to work on exploiting the said vulnerability and receive a flag if successful. Some of the use cases of the proposed system are provided below.

3.4.1 General Workflow

The general use case of hackademy has been described in the following use case diagram.

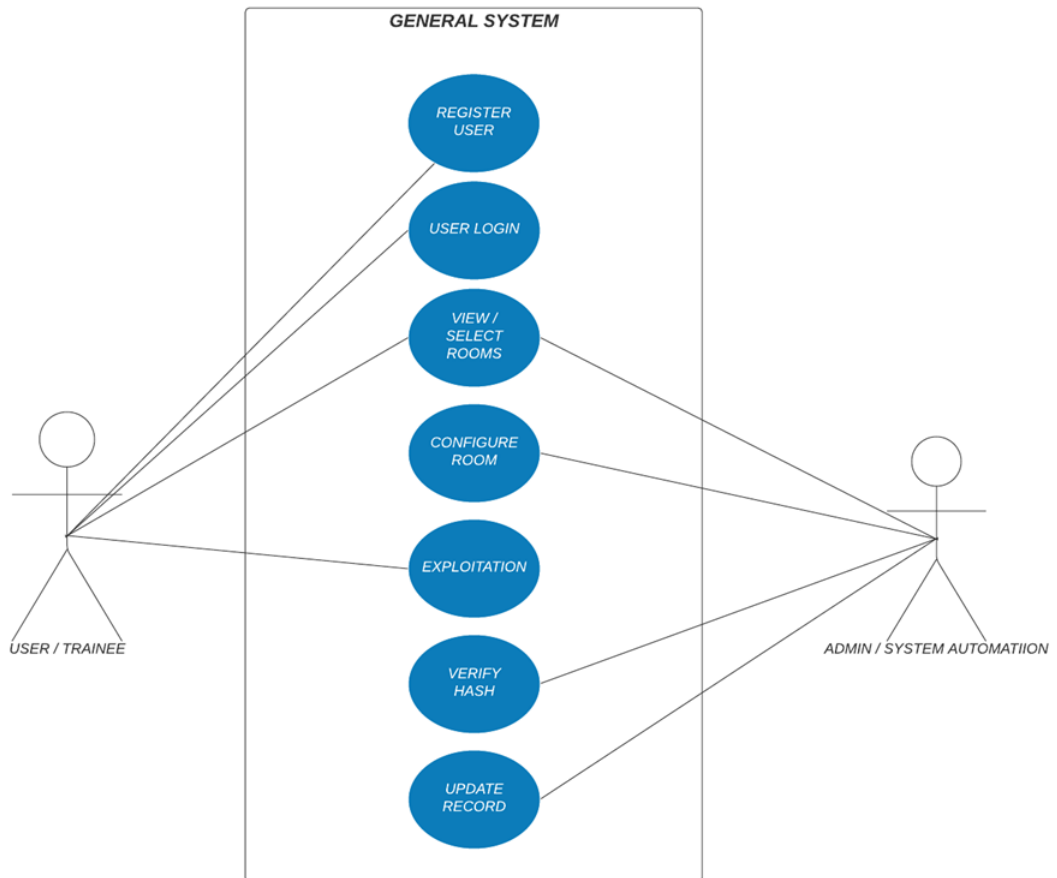


Figure 3.1: General Overview of the use case

3.4.2 Register User

New users / trainees have to register to the web portal before starting to work on the cyber drills.

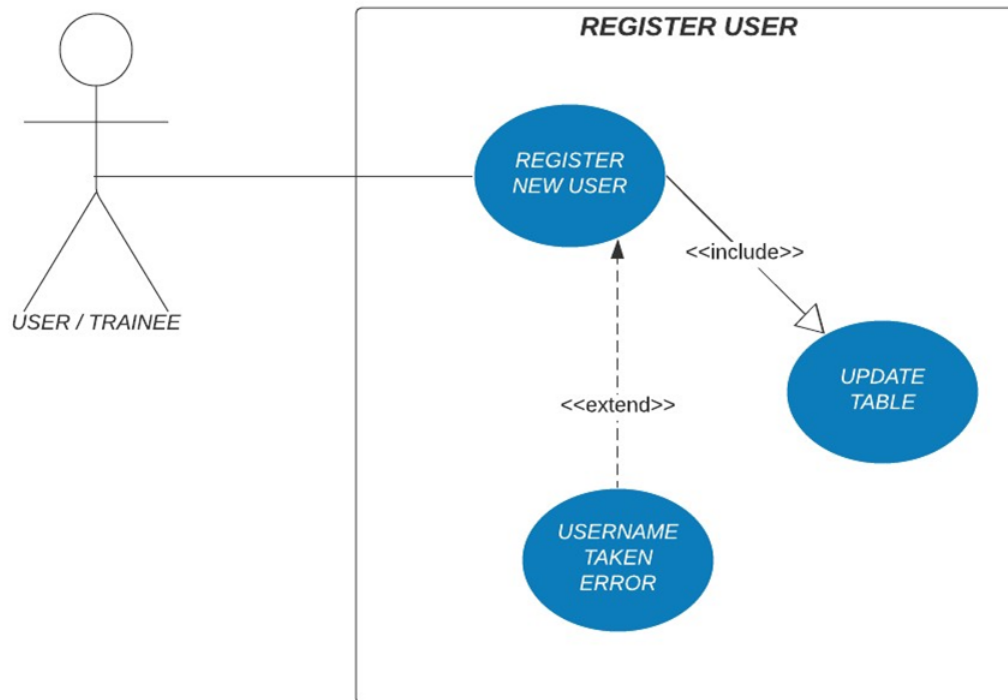


Figure 3.2: Register User

USE CASE ID	UC - 1
TITLE	Register User
ACTOR	User, Trainees
PRE-CONDITION	Start of Application
POST-CONDITION	Login Screen should be displayed
SUCCESS SCENARIO	Successfully creation of User portal.

Figure 3.3: Use case 1 Information

3.4.3 User Login

New users / trainees have to login to the web portal to see their dashboard and list of available activities / machines

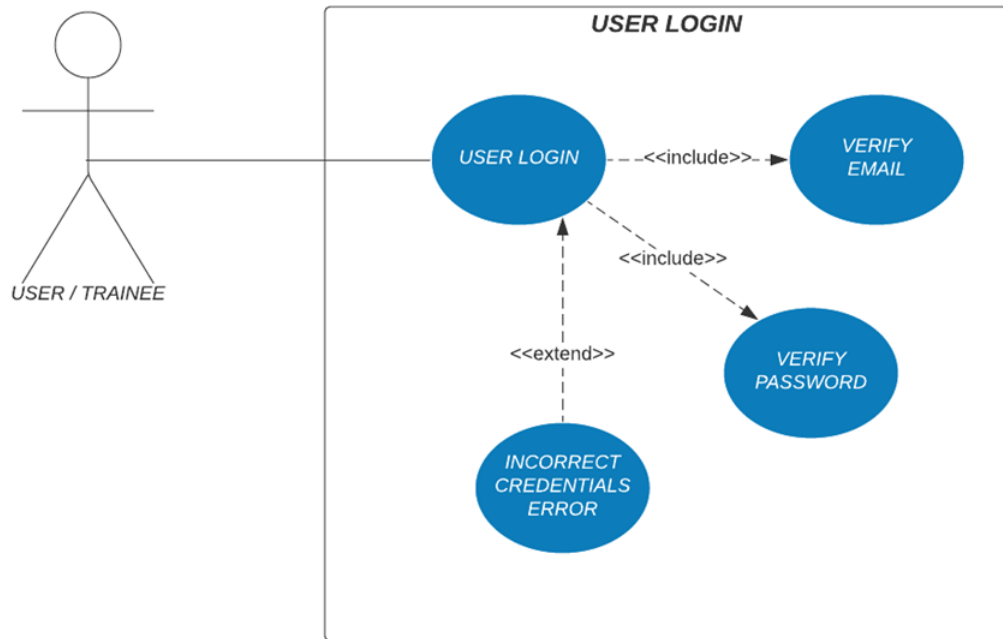


Figure 3.4: User Login

USE CASE ID	UC - 2
TITLE	User Login
ACTOR	User, Trainees
PRE-CONDITION	User account already created
POST-CONDITION	Profile/Dashboard should be displayed
SUCCESS SCENARIO	Successfully logged into the portal.

Figure 3.5: Use case 2 Information

3.4.4 View and Select machine

User will see a list of machine or exercises related to cyber drills and will select a machine to checkout their skills. The system will checkout whether the user is eligible to access the selected machines based on user’s level.

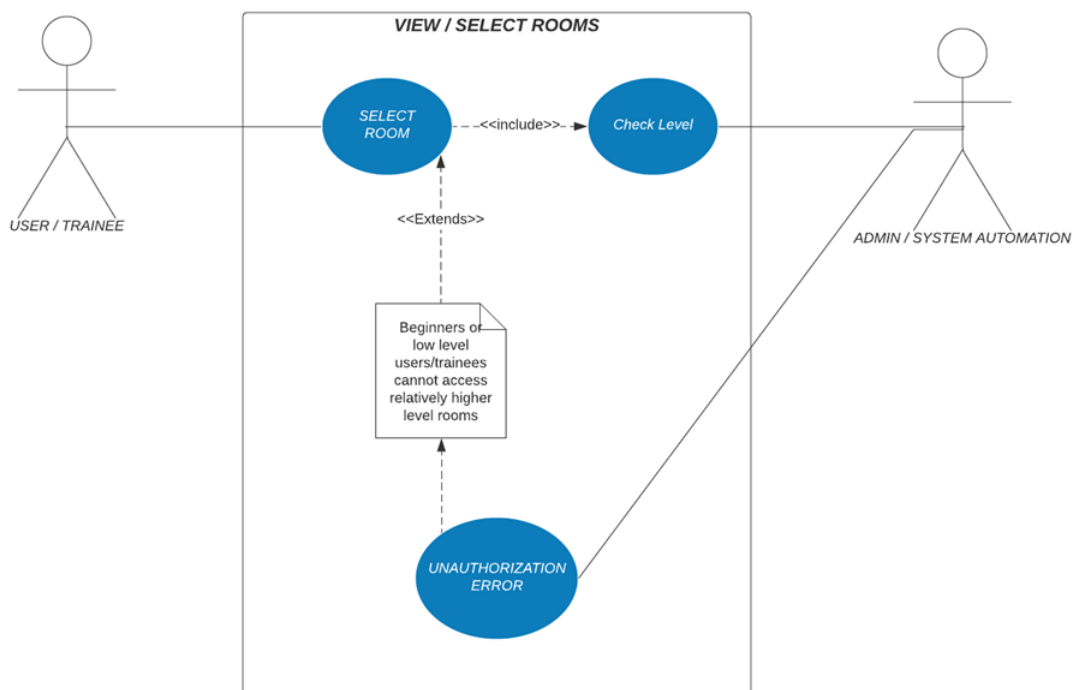


Figure 3.6: View and select machines

USE CASE ID	UC - 3
TITLE	View and Select Rooms
ACTOR	User, System
PRE-CONDITION	A list of rooms is available
POST-CONDITION	Configuration of room should available
SUCCESS SCENARIO	The user is eligible to access the selected room

Figure 3.7: Use case 3 Information

3.4.5 Configure machine

After the user selects a specified machine then a virtual environment for that machine is configured in the host machine.

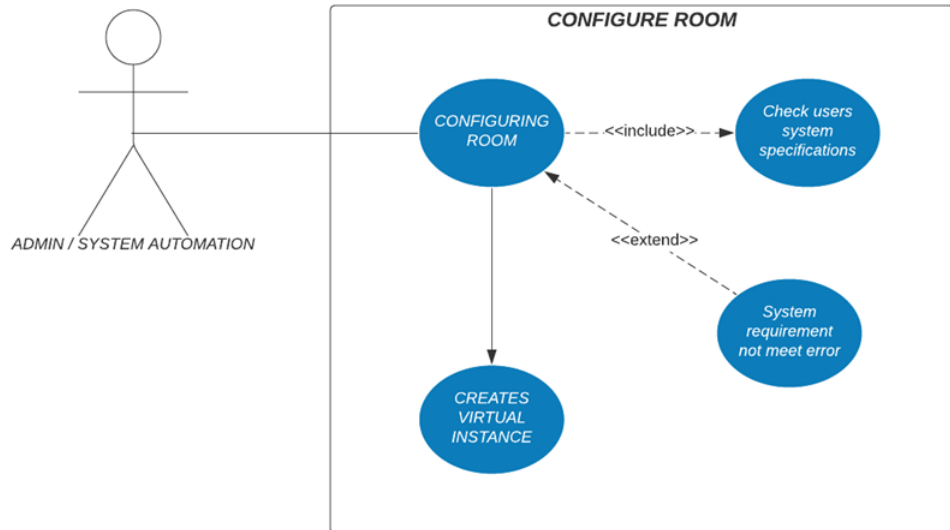


Figure 3.8: Configuring machine

USE CASE ID	UC - 4
TITLE	Configure Room
ACTOR	System
PRE-CONDITION	Selected room is accessible to the user
POST-CONDITION	
SUCCESS SCENARIO	Room successfully configured in the system

Figure 3.9: Use case 4 Information

3.4.6 Exploitation

After the virtual instance of specified machine is configured into the system, the user can finally work on the training exercises and work his way to exploit the vulnerability and figure out the hash code (CTF).

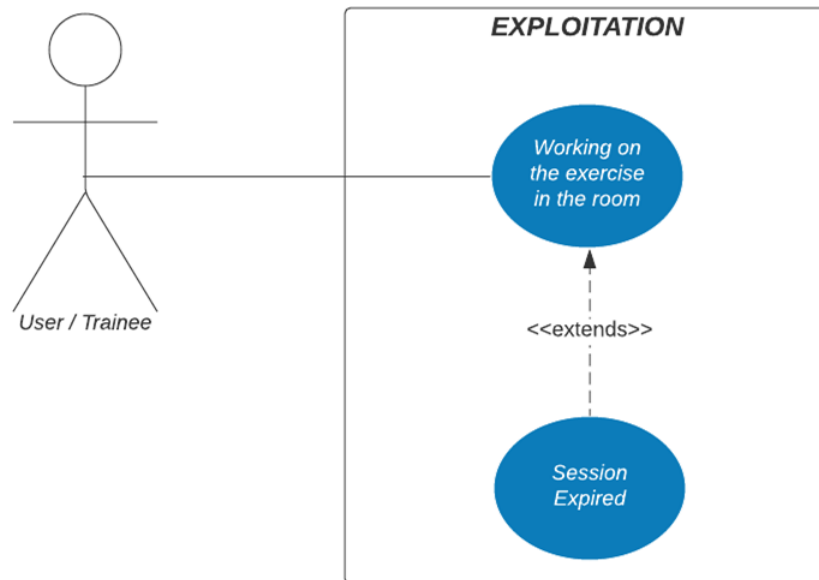


Figure 3.10: Exploitation

USE CASE ID	UC - 5
TITLE	Exploitation
ACTOR	User
PRE-CONDITION	Room successfully configured
POST-CONDITION	Flag(hash) captured Successfully
SUCCESS SCENARIO	Flag(hash) captured Successfully

Figure 3.11: Use case 5 Information

3.4.7 Verification

If the user is successfully able to catch the flag, then the system verifies the found hash code with the target hash code in his database.

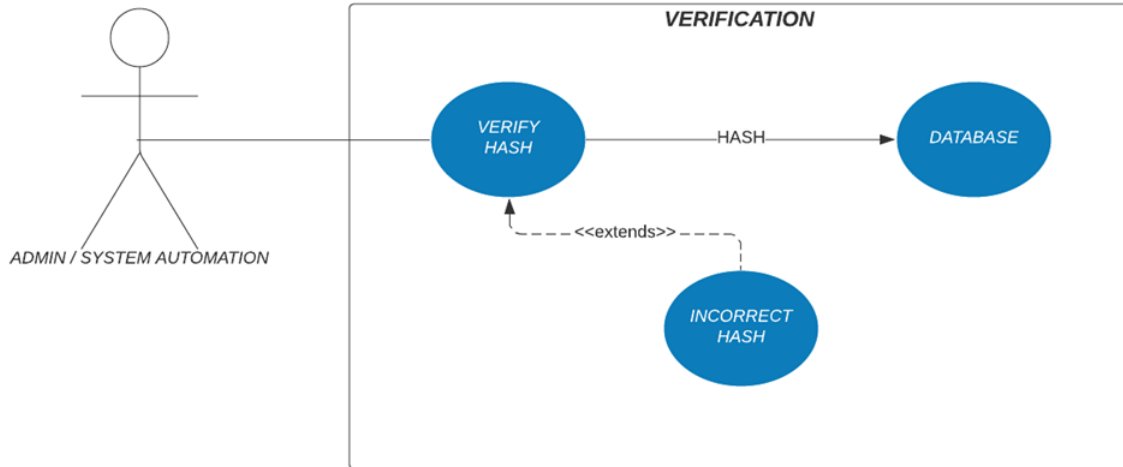


Figure 3.12: Verification of Key

USE CASE ID	UC - 6
TITLE	Verification
ACTOR	System, Admin
PRE-CONDITION	Finding of the hash
POST-CONDITION	
SUCCESS SCENARIO	The given hash is correct

Figure 3.13: Use case 6 Information

3.4.8 Update Record

After successful exploitation, the record of the user is updated in the database and log files and is displayed on user’s profile.

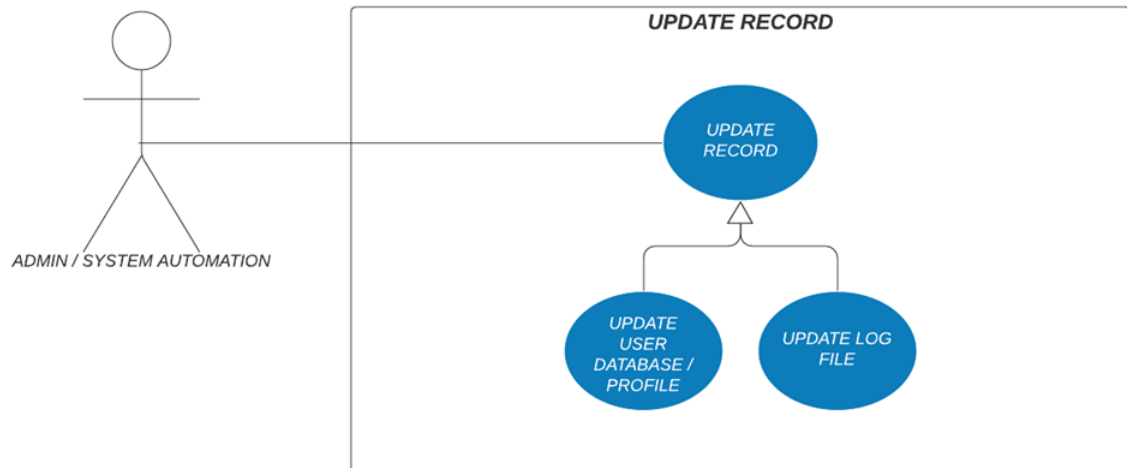


Figure 3.14: Update Record

USE CASE ID	UC - 7
TITLE	Update Record
ACTOR	Admin
PRE-CONDITION	Successful Verification
POST-CONDITION	Display on user's profile
SUCCESS SCENARIO	Record Updated Successfully

Figure 3.15: Use case 7 Information

Chapter 4

System Design

This chapter covers the design phase of the said project. All the specified requirements by the stakeholders and the developers of the system are discussed and brainstormed in this chapter by using models and diagrams. Examining different design aspects of Hackademy will be possible if the developers have adequate knowledge about the said system/product. Modeling allows designers to easily and precisely grasp and handle various aspects of the system. A successful phase results in a design that meets all of the requirements and meets the needs of the users. It must be simple to use and understand. The following chapter contains detailed diagrams for each system component.

4.1 System Architecture

The intended system is compatible with web applications and will be accessible through multiple architectures like Linux, macOS and windows etc. The system comprises a simple architecture with the following components.

- i.) PRESENTATION LAYER (GUI)
This layer contains an eye-catching user interface that is easy to interact with.
- ii.) APPLICATION LAYER
This layer contains the logical reasoning of different components such as managing user roles, web pages, connection with DB etc.
- iii.) VIRTUAL LAYER
This layer controls the management and deployment of Virtual Machines or Docker machines in hosts system

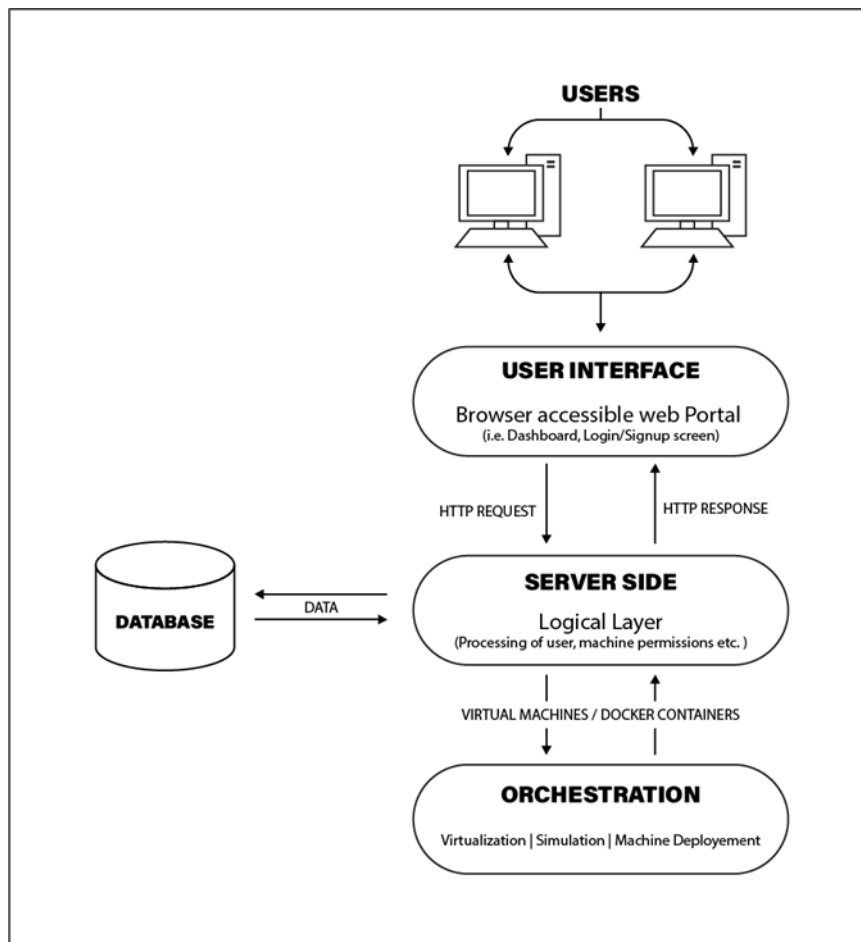


Figure 4.1: Architecture of Hackademy

4.2 Design Constraints

Following are the major constraints of our system.

- i.) Different actors will be given different rights and permissions to perform certain functionalities within the system, i.e., Users will not be allowed to have modifications regarding the virtual machines, admins will be given permissions or privileges to read, write a particular functionality like creating logs, managing machines and managing users etc.
- ii.) The machines will be manually given specific vulnerabilities in development phase using SecGen technology.
- iii.) Machines session will expire due to inactivity of the user.
- iv.) Specific number of machine instances will be configured due to hardware limitations.

4.3 Design Methodology

Throughout the development phase of Hackademy, we might evolve and work on our functional and non-functional requirements, the best way to keep on improving

is to use the Iterative methodology where we can keep improving the system with each step by taking points from previous iterations. This step would involve,

- i.) Requirement Analysis
- ii.) Designing
- iii.) Development
- iv.) Testing
- v.) Evaluating
- vi.) Repeat

4.4 Component Diagram

The component diagram depicts how the system's various components interact. It was created to assist developers with the project's implementation.

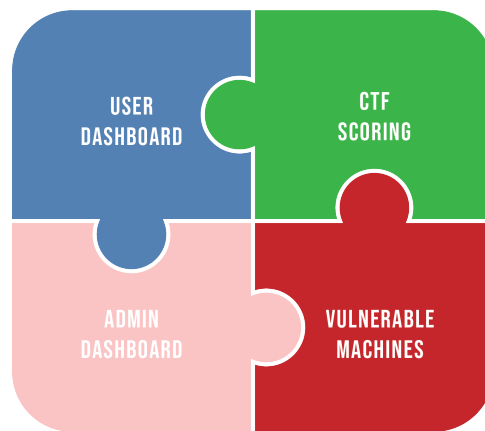


Figure 4.2: Component Diagram of Hackademy

4.5 Process Interaction Diagram

The process interaction diagram shows the processes in chronological sequence as well as their interactions. The program is started by the user. He starts the first procedure by selecting a machine from a list of options. When you click the create button, a new process is started to setup the machine in the host system. If the supplied hash code was right, the procedure will change the CTF status to claimed, indicating that the process was successful. The CTF status will be failed if the hash code provided was wrong.

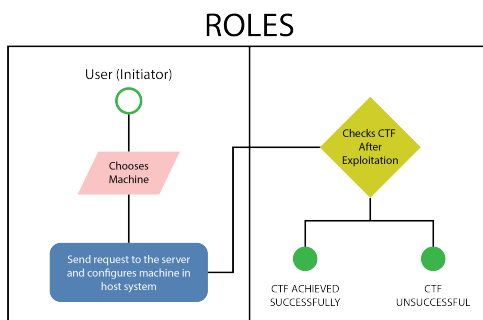


Figure 4.3: Process Interaction Diagram of Hackademy

4.6 Workflow Diagram

How our system works, The functionalities and responsibilities of ‘Hackademy – Vulnerable Machine Development for Cyber Drills’ are explained using the workflow diagram Figure 4.4.1, which determines how different activities are triggered and how different component work.

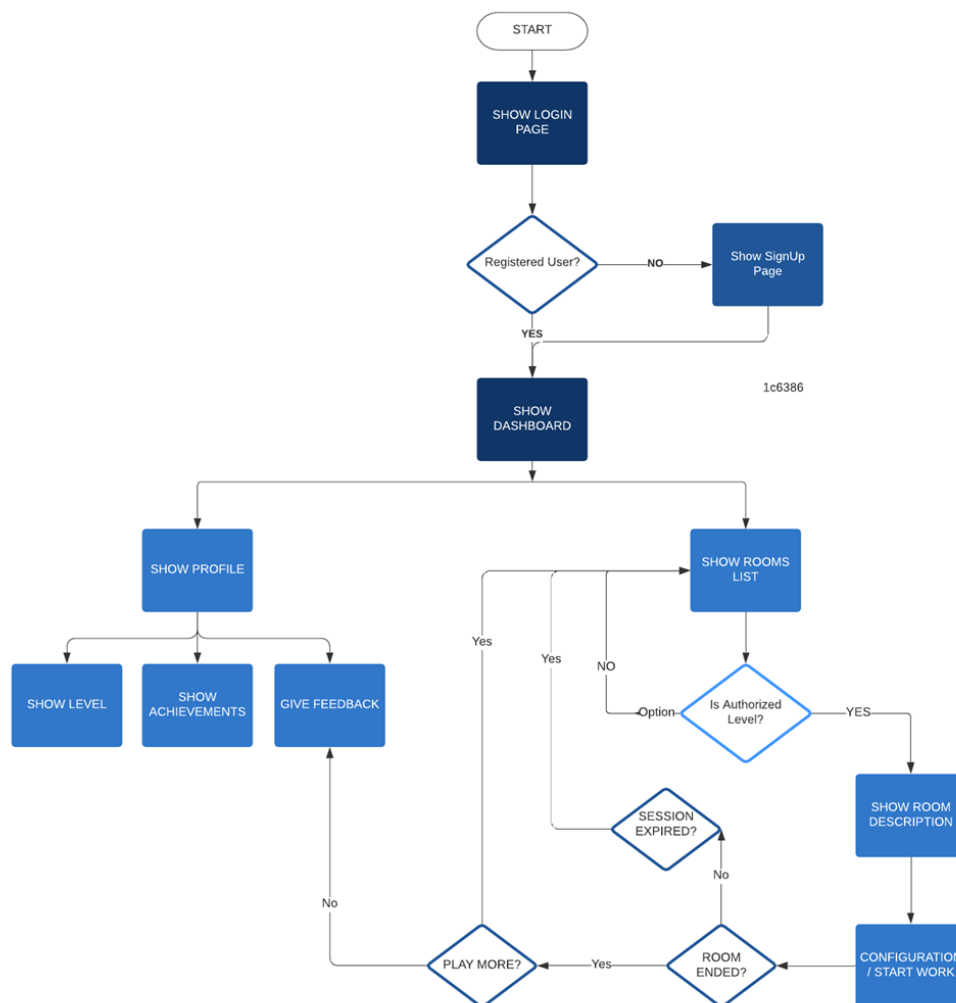


Figure 4.4: Workflow Diagram of Hackademy

4.7 Sequence Diagram

Following are the system diagram for Hackademy, It shows the series of messages between components (objects) and actors (users) of the system.

4.7.1 Login Page

According to system requirements, all users need to login to the web portal before performing any activity. Figure 4.4.2.1 is the representation of how login messages and access is passed through the user to the login page and how it is then authenticated and validated from the database.

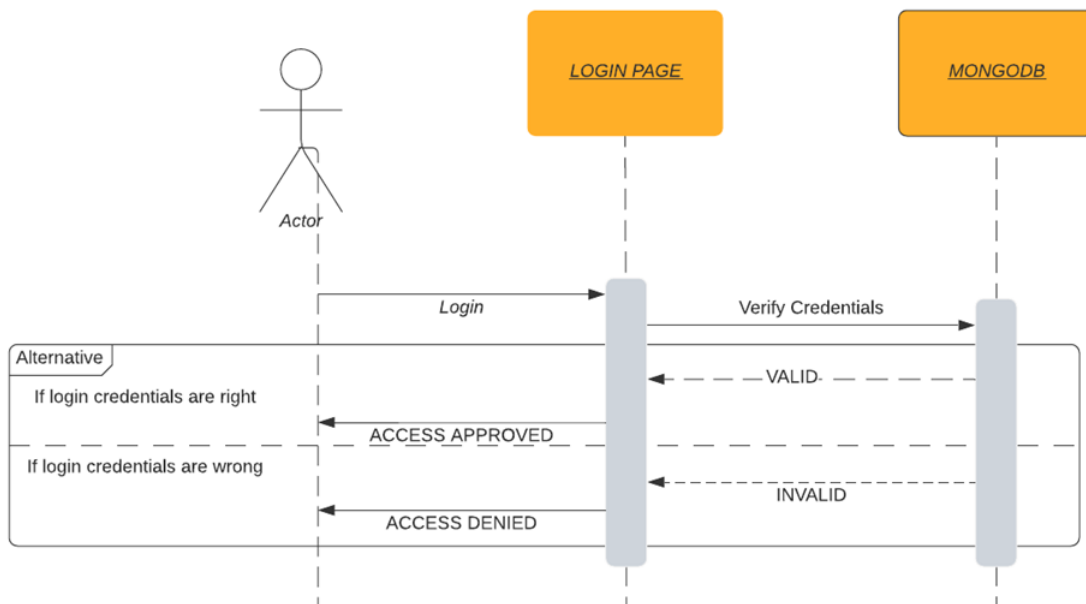


Figure 4.5: Login Sequence

4.7.2 Operation of Hackademy

The User will ask the server through user interface about the available rooms, The server will then gather the lists of available rooms from the database and return a list of rooms to the user using UI. The user will choose a room and request the specific room page. Server will check whether the user has the authorized level to request the specific room, if yes then the server will accept the request otherwise the user has to choose again. Once the room is selected, the user will request the server to download and configure the room (machine).

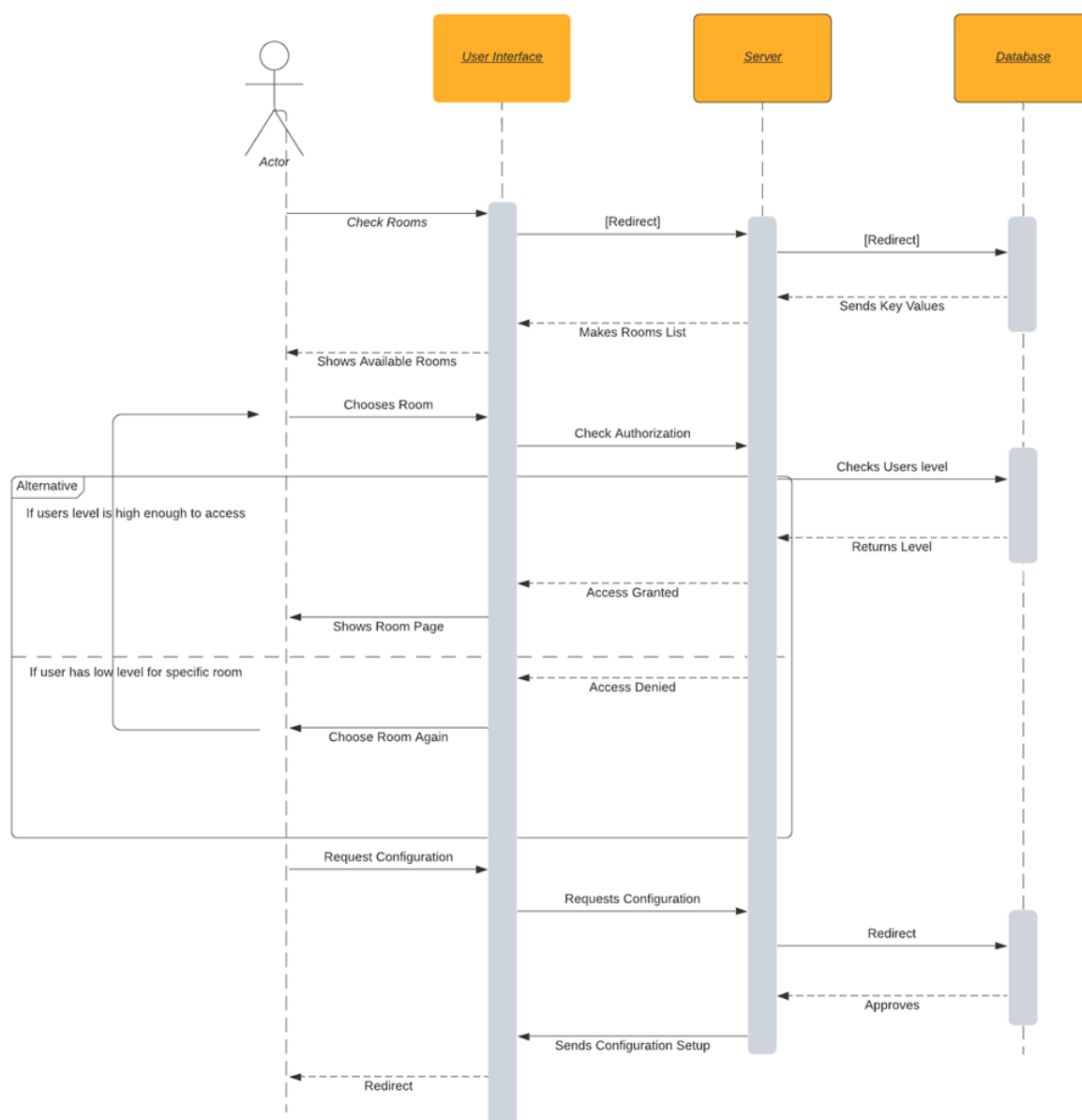


Figure 4.6: Sequence Diagram

4.7.3 CTF Claiming

The user will Enter the hashcode he achieved by exploiting the vulnerability in the machine he worked on, Once the hashcode is entered it will be sent to the backend through an API and it will compare the HASH entered by the user with the actual hashcode of that exploited vulnerability.

If the Entered hashcode is exactly what was required then the user have achieved the CTF and he will be redirected to his dashboard with a success Message, If the hashcode is different than he will be redirected to the machines list and an unsuccessful attempt message will be shown.

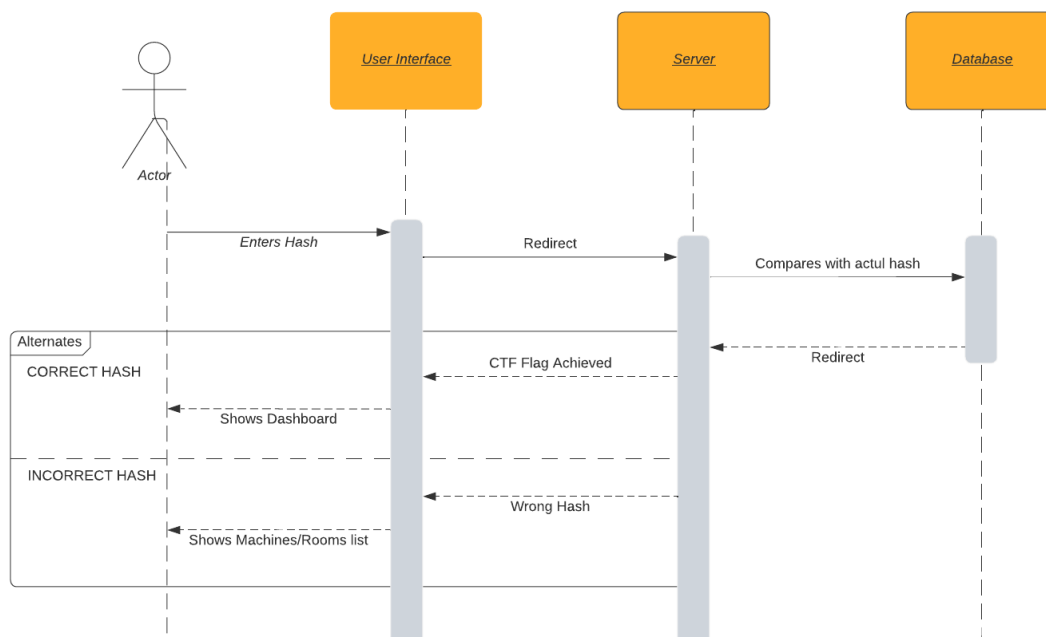


Figure 4.7: Sequence Diagram of CTF Achieved

4.8 GUI Design

In this section, the GUI Design of Hackademy would be portrayed. The GUI of Hackademy was carefully brainstormed so that it is both interactive to the user and also follows the mood of cyber security. Due to this, we are using a dramatic dualtone color scheme. We are also using secondary colors and iconography in order to make the GUI seamless and attractive.



Figure 4.8: Hackademy Logo

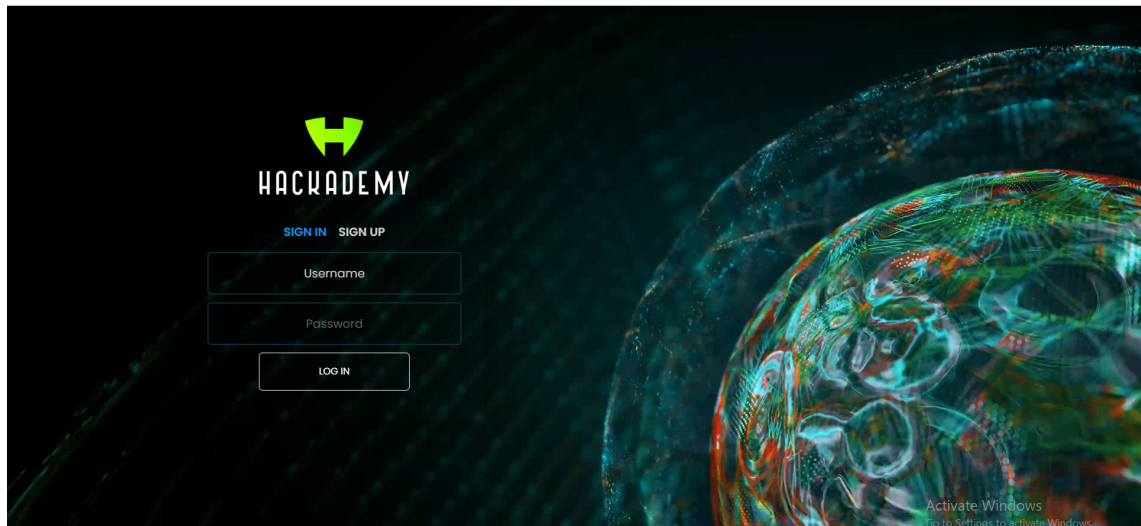


Figure 4.9: Login and Sign Up Page

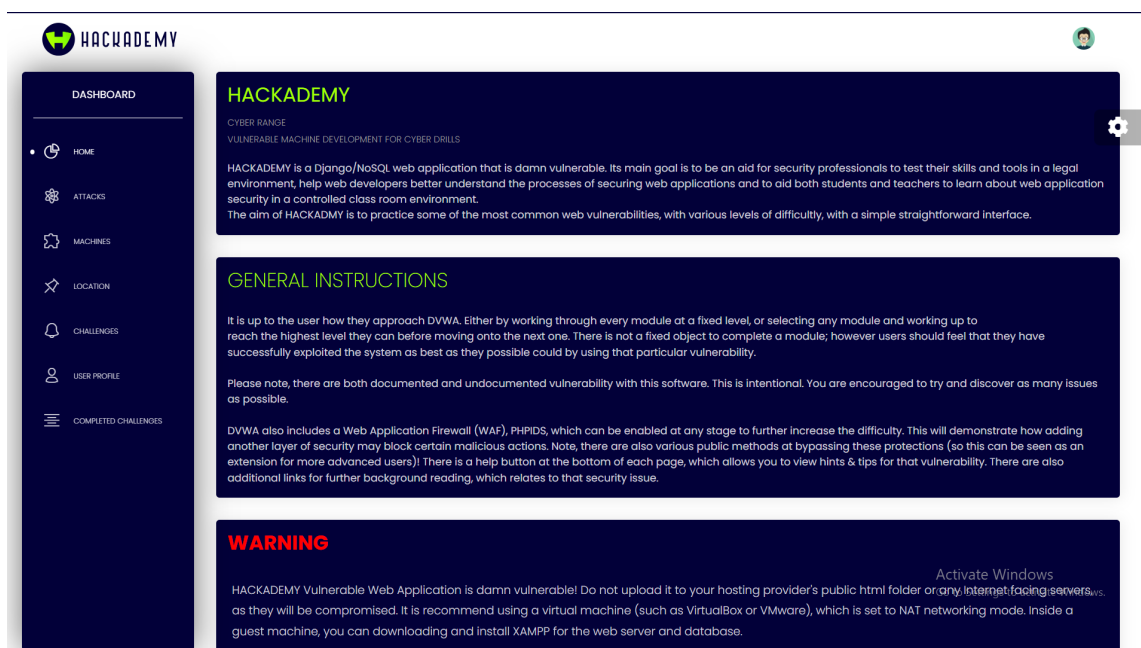


Figure 4.10: User Home Page

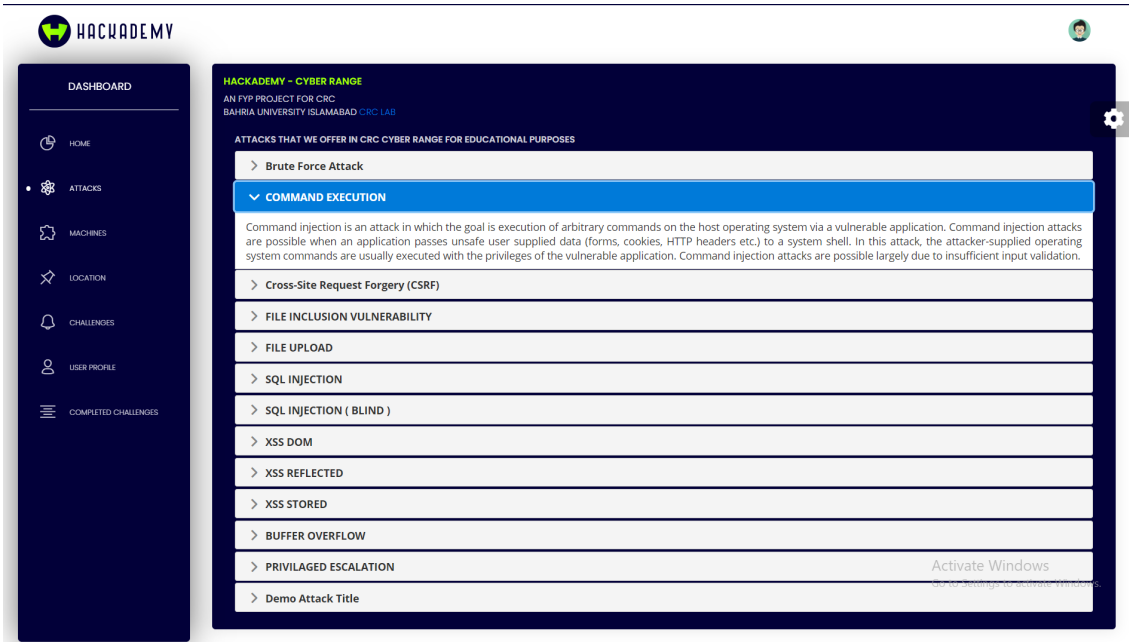


Figure 4.11: Attack Description Page



Figure 4.12: Available Machines List

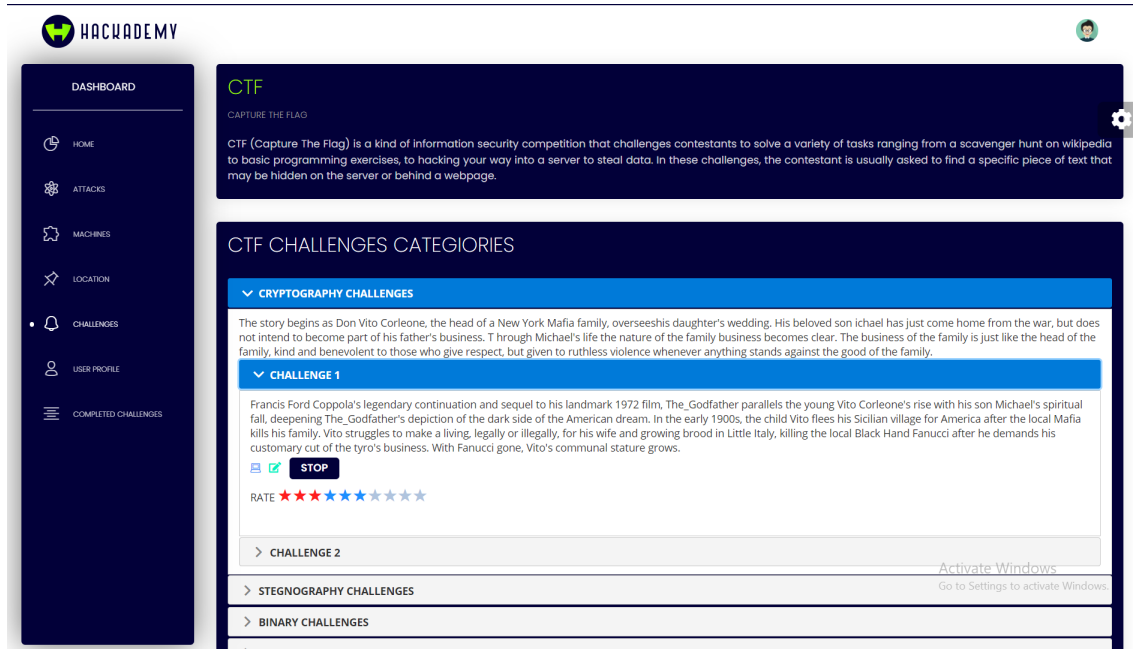


Figure 4.13: Challenges Categories

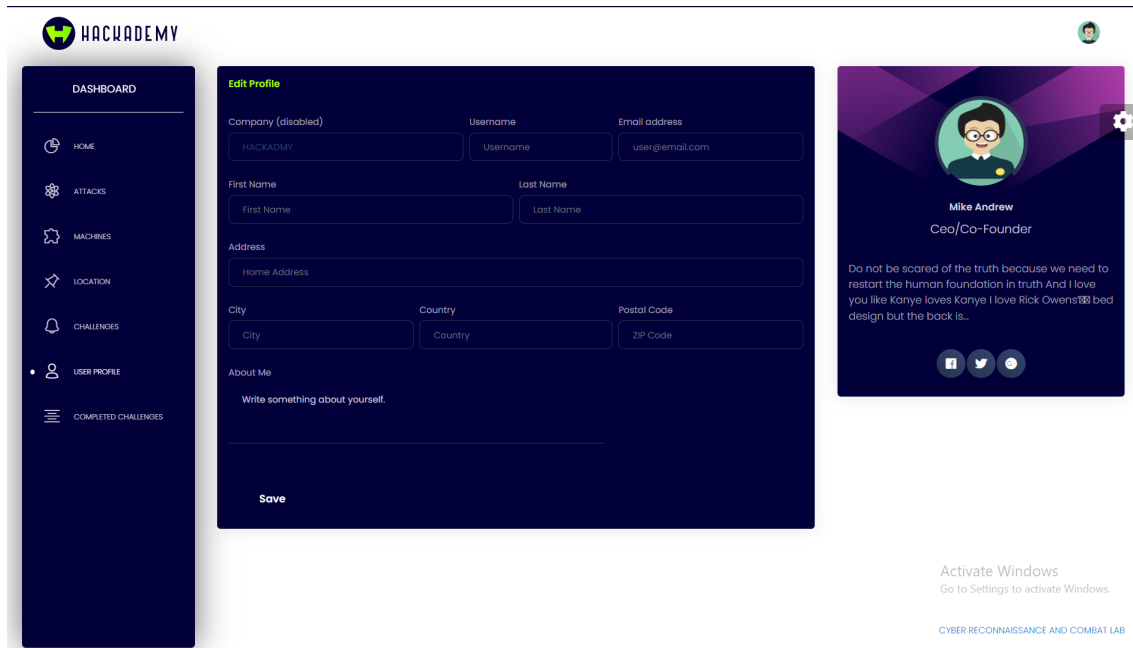


Figure 4.14: User Profile

Chapter 5

System Implementation

The purpose of this chapter is to provide a detailed understanding and working of the proposed project. In system implementation phase we use logical understanding and development models to make a physical implementation of the proposed system by using relevant languages, environments and framework libraries. We develop the intended system by keeping in mind the system requirement and system designs along with all the stakeholders and the system users.

5.1 System Architecture

The intended system is compatible with web applications and will be accessible through multiple architectures like Linux, macOS and windows etc. The system comprises a simple architecture with the following components.

5.1.1 Presentation Layer (GUI)

This layer contains an eye-catching user interface that is easy to interact for the users. For the presentation layer we have used html for the structure, scss for designing the layout and making it user friendly. And for quick accessibility and management we have used Angular framework which makes it easier to access services and define functionalities. The color combinations we used are those similar to creating a hacking academy mood.

5.1.2 Application Layer

This layer contains the logical reasoning of different components such as managing user roles, web pages, connection with Database etc. Application layer is the area where we define how our system will behave in certain conditions and certain commands and also give pathways to http requests. We wrote our system's api's in application layer.

5.1.3 Virtual Layer

This layer controls the management and deployment of Virtual Machines or Docker machines in hosts system. These machines contains the configured virtual environment for specific vulnerabilities.

5.2 System Requirements

For the creation of our product 'Hackademy', following were our system's requirement in order for smooth development and testing.

5.2.1 Activity Requirement

The Environment or platforms that are needed in order to have a smooth and successful development of Hackademy.

- i.) Computer OS - Both Windows and Linux
- ii.) Database - MongoDB

5.2.2 Hardware Requirement

The hardware requirements here should also be kept in mind because developing such a project would utilize hardware resources up to an extent.

- i.) Computing Device - Laptop or Desktop
- ii.) RAM - minimum 8GB - RAM would be shared with virtual machines so they can work autonomously
- iii.) Storage - Storage would be shared with virtual machines so they can work autonomously

5.2.3 Software Requirement

The following are the required software's to be installed in the developer's system to write the code effectively and run it.

- i.) Code Editor - Preferably Visual Studio Code
- ii.) Terminal - To write scripts
- iii.) Version Control GUI - Preferably Git, but not necessary
- iv.) Postman - To check API calls

5.3 Tools and Technologies

The tools and technologies used for the development of HACKADEMY – vulnerable machine development for cyber drills are as following.

5.3.1 MongoDB Database

MongoDB is a cross platform, NOSQL database that uses BSON document oriented program. MongoDB has gained quite some popularity since it benefits more than traditional SQL based relational databases. MongoDB can store and manage tons of complex data and provides many characteristics and services like data accessibility, reliability, data integrity and scalability. The reason we used MongoDB in our project is due to the fact that security is an important factor for the system we intend to build. Since mongoDB is a safe and secure database than most traditional DBs and also the fact that mongoDB is a NOSQL database so we can also avoid attacks like SQL Injections by using such DB.

5.3.2 Linux OS

Linux is a UNIX based operating system with many variants such as Ubuntu or Kali Linux. These are open source operating systems meaning they can be customized according to the user's requirement. Since our project requires the generation or vulnerable machines. We cannot simply inject these vulnerabilities in the user's machine since it might compromise these systems. Therefore we are working towards creating virtual machines that will host these vulnerabilities in them. This way everything related to injecting and exploiting these threats will be done in separate sandboxes or virtual machines this way it wont harm the host machine. We are using linux OS in these virtual machines since it is easier to inject and configure vulnerabilities in it because of linux being an open source Operating System

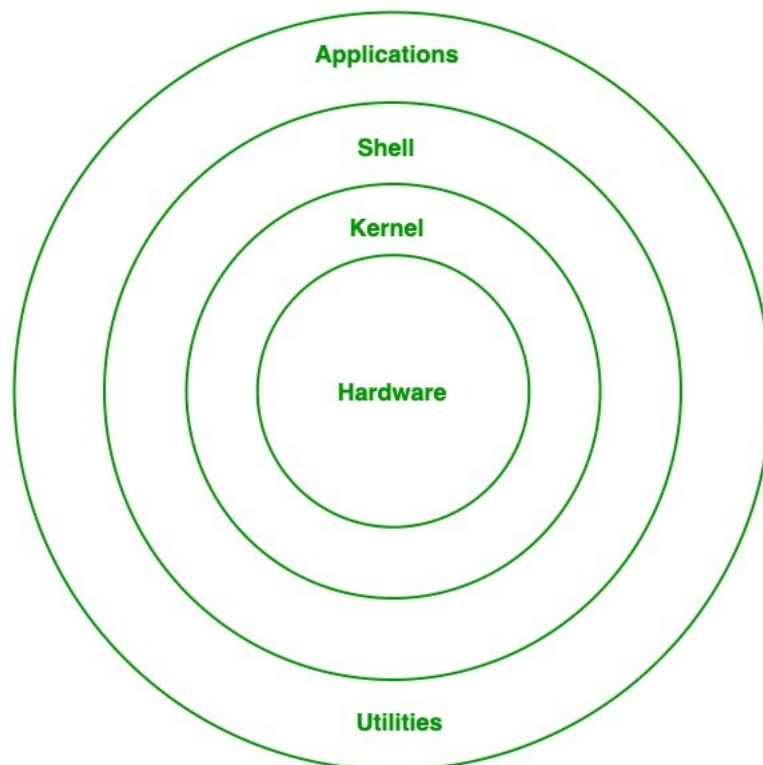


Figure 5.1: Linux Architecture

5.3.3 REST API

API or Application Programming Interface is an interface or a piece of code through which multiple programs or services talk to each other. In this project we are using RESTful API since it is the most popular API out there. REST stands for Representational State Transfer, this API helps best in handling user requests typically sent and received through http or https. We are using REST API in both admin side and user side of our project, using REST API in user's panel, we are only using it to POST data and GET data using POST and GET Request. In the admin's panel, admins are allowed to manipulate data the way they want so they can use all REST requests such as GET, POST, PUT, DELETE.

5.3.4 Angular

Angular is an open source project initiated by google and contributed by many open source developers. It is used to develop single page client side applications using html and typescript. We have used angular in the front end of our application so that it is easier for the user to have fast and reliable access to our system.

5.4 Development Environment and Languages

Development environment and languages used for the development of HACKADEMY – vulnerable machine development for cyber drills are as following.

5.4.1 Microsoft Visual Studio Code

We are using Microsoft's Visual studio code which can be worked as an IDE after installing all dependencies in it. It has built in support for front end technologies i.e. html, css, JavaScript and typescript etc. The reason we used VS code as our code editor is because VSC is compatible with windows, Linux and macOS and our project requires both working of the windows and for Linux.

5.4.2 Typescript

Typescript is a superset of javascript with added functionalities like optional static typing and much more. The reason we chose typescript over javascript is the fact that typescript typically gives compile errors thus reducing the possibility for runtime errors. It is a lightweight language that we are using in the front end of our project along with angular framework for easy accessibility.

5.4.3 Django

Python is considered to be the father of all programming languages due to its diversity and general role. Simply put, python is and can be used everywhere in any type of system. It is a high level interpreted language and uses an object oriented approach. We are using python in the backend or the server side of our application

along with Django framework. We are also writing our APIs in Django python.

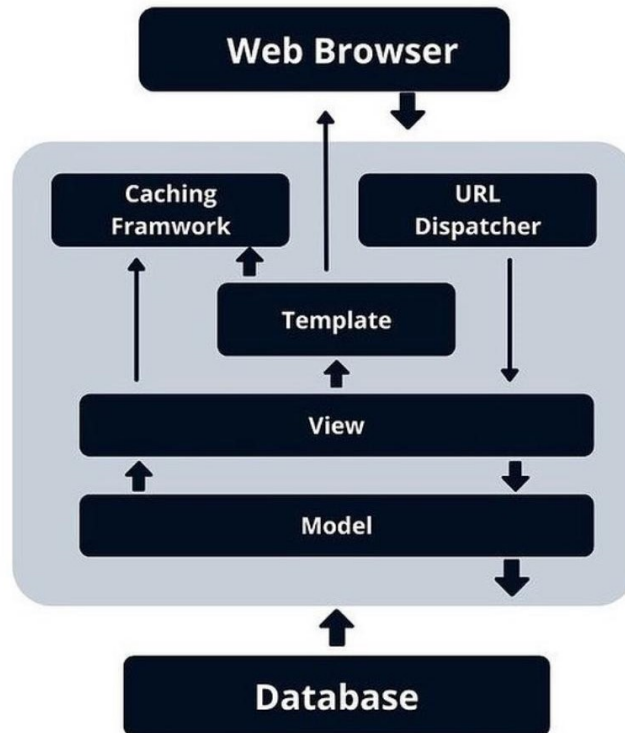


Figure 5.2: Django's Architecture

5.5 Vulnerable Machine Development Process

5.5.1 SecGen

[8] Hands-on experience with security tools and strategies to attack and protect vulnerable systems benefits computer security students. Virtual machines (VMs) are a useful tool for sharing hacker targets. However, creating these hacking tasks takes time and is virtually static once completed. That is, after a task has been "solved," the student is no longer faced with a difficulty, and if the challenge was developed for a competition or evaluation, it cannot be reproduced without risking plagiarism and collusion.

Security Scenario Generator (SecGen) may generate sophisticated VMs based on random scenarios, with a variety of use-cases, including: constructing networks of VMs with random services and in-the-wild vulnerabilities and themed content, which can form the foundation of penetration testing [10] operations; VMs for educational purposes

5.5.2 Code Snippet for API

Below is the python code snippet for handling the machine creation API from the angular front-end and configuring the virtual machine according to the configura-

tions entered by the admin.

```

import os
import itertools
import threading
import time
import sys
import xml.etree.ElementTree as ET
ET.register_namespace("", "http://www.github/cliffe/SecGen/scenario")
ET.register_namespace("", "http://www.w3.org/2001/XMLSchema-instance")
ET.register_namespace("", "http://www.github/cliffe/SecGen/scenario")
pathM = 'sudo_ruby_secgen.rb-s-scenarios/examples/vulnerability_examples/
insecure_web_applications/commando/'
newfile = ("Updated.xml")
tree = ET.parse('HackademyAPI\impossible.xml')
root = tree.getroot()

def MachineCreationAPI(request, id=3):
    if request.method=='POST':
        if 'MachineName' in request.POST:
            MachineName = request.POST['MachineName']
        if 'MachineType' in request.POST:
            MachineType = request.POST['MachineType']
        if 'CreationDate' in request.POST:
            CreationDate = request.POST['CreationDate']
        if 'ExpiryDate' in request.POST:
            ExpiryDate = request.POST['ExpiryDate']
        if 'MachineDescription' in request.POST:
            MachineDescription = request.POST['MachineDescription']
        if 'Vulnerability' in request.POST:
            Vulnerability = request.POST['Vulnerability']
        if 'Difficulty' in request.POST:
            Difficulty = request.POST['Difficulty']
        if 'Encoders' in request.POST:
            Encoders = request.POST['Encoders']
        if 'Network' in request.POST:
            Network = request.POST['Network']
        if 'Service' in request.POST:
            Service = request.POST['Service']
        if 'webapp' in request.POST:
            webapp = request.POST['webapp']
        if 'generator' in request.POST:
            generator = request.POST['generator']
        if 'Service' in request.POST:
            datastore = request.POST['datastore']

    for x in root.findall("./{http://www.github/cliffe/SecGen/scenario}vulnerability"):
        x.attrib['module-path']=f'{Vulnerability}'

    for x in root.findall("./{http://www.github/cliffe/SecGen/scenario}difficulty"):
        x.text = f'{Difficulty}'

    for x in tree.findall("./{http://www.github/cliffe/SecGen/scenario}encoder"):
        x.attrib['name'] = f'{Encoders}'

    for x in tree.findall("./{http://www.github/cliffe/SecGen/scenario}network"):
        x.attrib['type'] = f'{Network}'

    for x in root.findall("./{http://www.github/cliffe/SecGen/scenario}type"):
        x.text = f'{MachineType}'

    tree.write(newfile)

```

5.5.3 Code Snippet for Machine Configuration

Below is the SecGen Machine code written in XML. In said code we add custom configurations to create custom vulnerable machines based on the configurations

entered by the admin. This XML code will generate a virtual machines based on the changes.

```
<?xml version="1.0"?>

<scenario xmlns="http://www.github/cliffe/SecGen/scenario"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.github/cliffe/SecGen/scenario">

  <name>Secure Commando Scenario</name>
  <author>Joshua Hickling</author>
  <description>A web server with a vulnerable web server</description>

  <type>ctf</type>
  <type>attack-ctf</type>
  <difficulty>impossible</difficulty>

  <system>
    <system_name>web_server</system_name>
    <base type="server" distro="Debian_9"/>

    <input into_datastore="IP_addresses">
      <value>172.10.0.2</value>
      <value>172.10.0.3</value>
    </input>

  <vulnerability module_path=".*commando">

    <input into="verbose_error_deactivation">
      <value>true</value>
    </input>

    <input into="default_admin_deactivation">
      <value>true</value>
    </input>

    <input into_datastore="customer_feedback_table_headings">
      <generator module_path=".*customer_feedback_table_headings" />
    </input>

    <input into_datastore="product_table_headings">
      <generator module_path=".*product_table_headings" />
    </input>

    <input into="database">
      <generator module_path=".*sql_table_setup">
        <input into="customer_feedback_table_headings">
          <datastore>customer_feedback_table_headings</datastore>
        </input>
        <input into="product_table_headings">
          <datastore>product_table_headings</datastore>
        </input>
        <input into="field_to_leak">
          <generator type="flag-generator" />
        </input>
      </generator>
    </input>
  </vulnerability>
</scenario>
```

```

</input>

<input into="sqli">
<generator module_path=".*sqli_template">
  <input into="difficulty">
    <value>impossible</value>
  </input>
  <input into="table_headings">
    <datastore>customer_feedback_table_headings</datastore>
  </input>
</generator>
</input>

<input into="search">
<generator module_path=".*xss_search_template">
  <input into="difficulty">
    <value>impossible</value>
  </input>
  <input into="blacklist">
    <generator module_path=".*xss_blacklist" />
  </input>
  <input into="table_headings">
    <datastore>product_table_headings</datastore>
  </input>
  <input into="strings_to_leak">
    <generator type="flag_generator" />
  </input>
</generator>
</input>

</vulnerability>

  <network type="private_network">
    <input into="IP_address">
      <datastore access="next">IP_addresses</datastore>
    </input>
  </network>

<build type="cleanup">
  <input into="root_password">
    <generator type="strong_password_generator" />
  </input>
</build>
</system>

<system>
  <system_name>desktop</system_name>
  <base distro="Kali" name="MSF" />

  <network type="private_network">
    <input into="IP_address">
      <datastore access="next">IP_addresses</datastore>
    </input>
  </network>
</system>
</scenario>

```

5.5.4 Flow Chart for Machine Creation

This subsection explains the workflow of how a vulnerable machine is created.

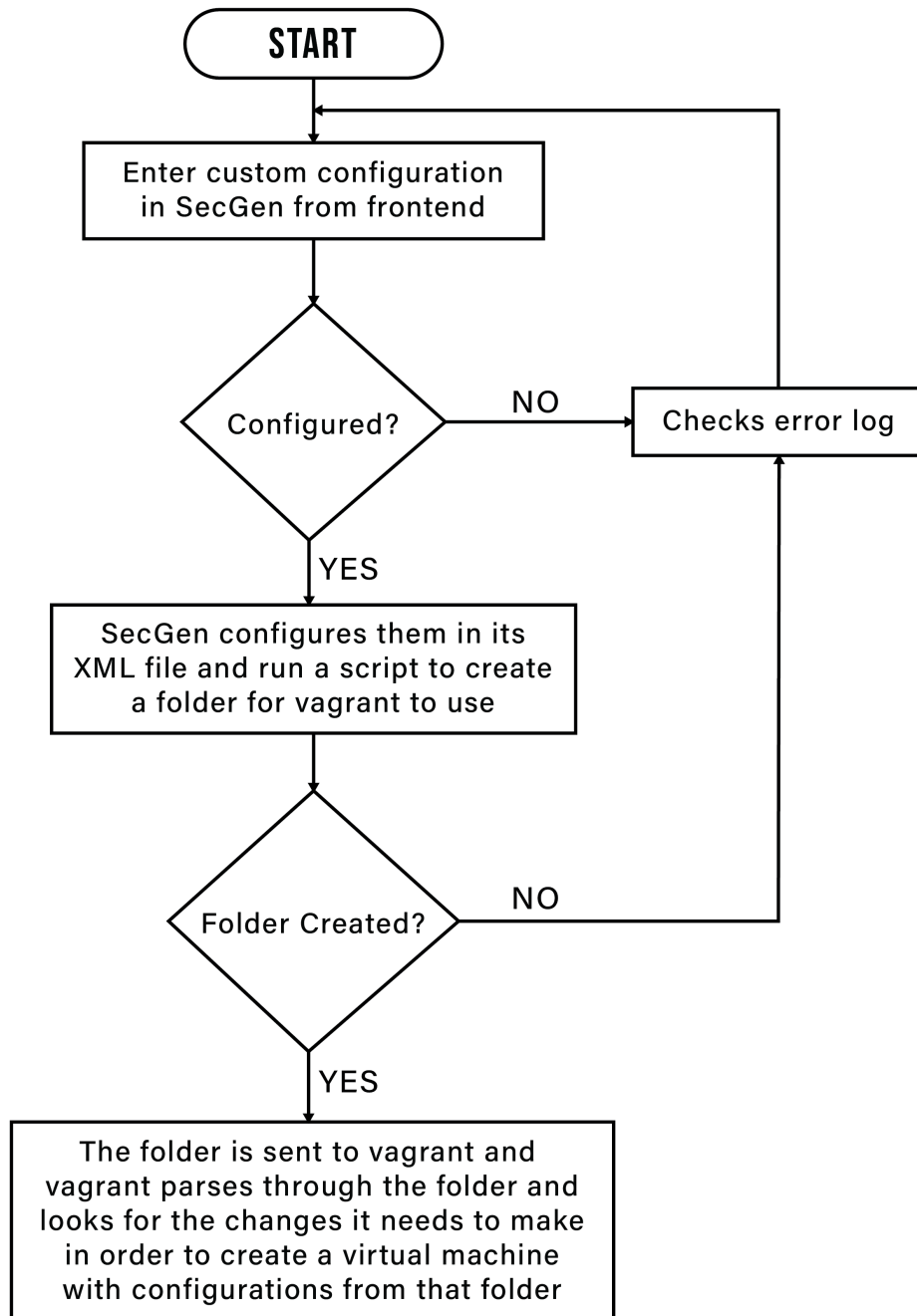


Figure 5.3: Flowchart of machine creation

5.6 FRONTEND DEVELOPMENT

Our front end needs a generated vulnerable machine which can be in the form of a virtual machine. The frontend of client side will then list down authorized machines

to the user they can work on, our front end consists of many components such as a blog or a user manual which explains different types of threats/attacks, how these attacks can be avoided, it can also contain the installation manual for the user to learn how to use the platform.

We are implementing our projects frontend on angular which creates single paged web applications. Attached below is the front end of our application for client side.

5.6.1 Client-Side Dashboard

The Dashboard contains the a brief information about the project and some general instruction. The page consists of 3 base components, one is the left menu bar which contains different pages such as home, attacks etc. The content of the menu is yet to be finalized so we have currently added dummy menu options. Then there is the body component which contain the body of the triggered page from the menu. In the dashboards home page, the body simply consists of information related to the system.

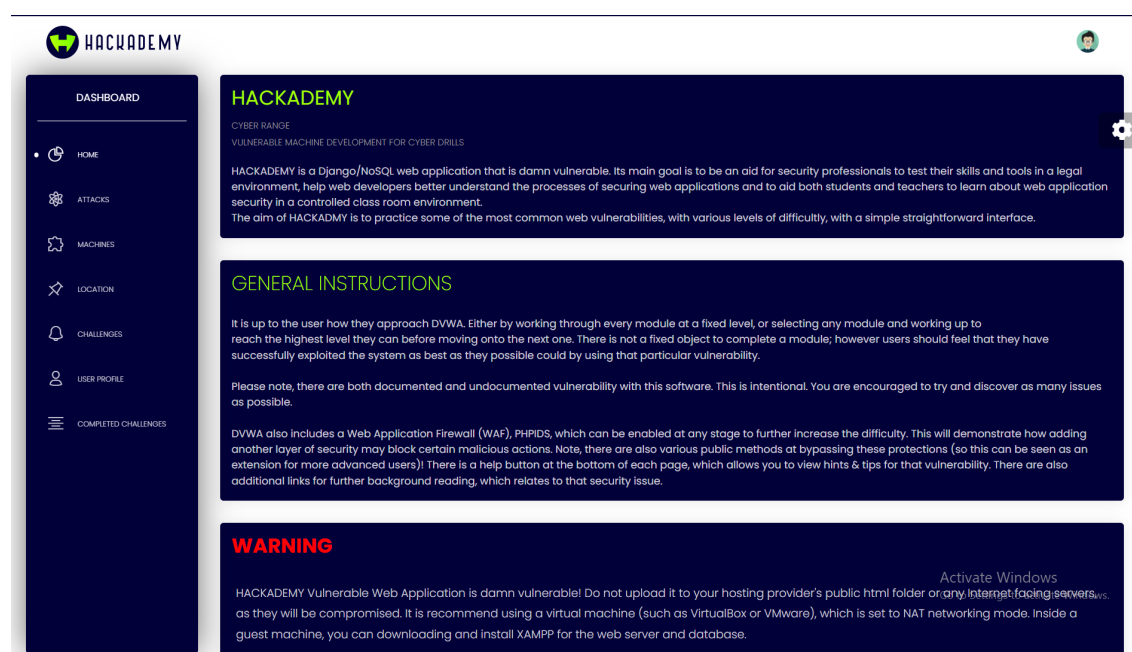


Figure 5.4: Client Side Dashboard

5.6.2 Attack Page

Attack's page contains the list of common attacks that the internet faces.

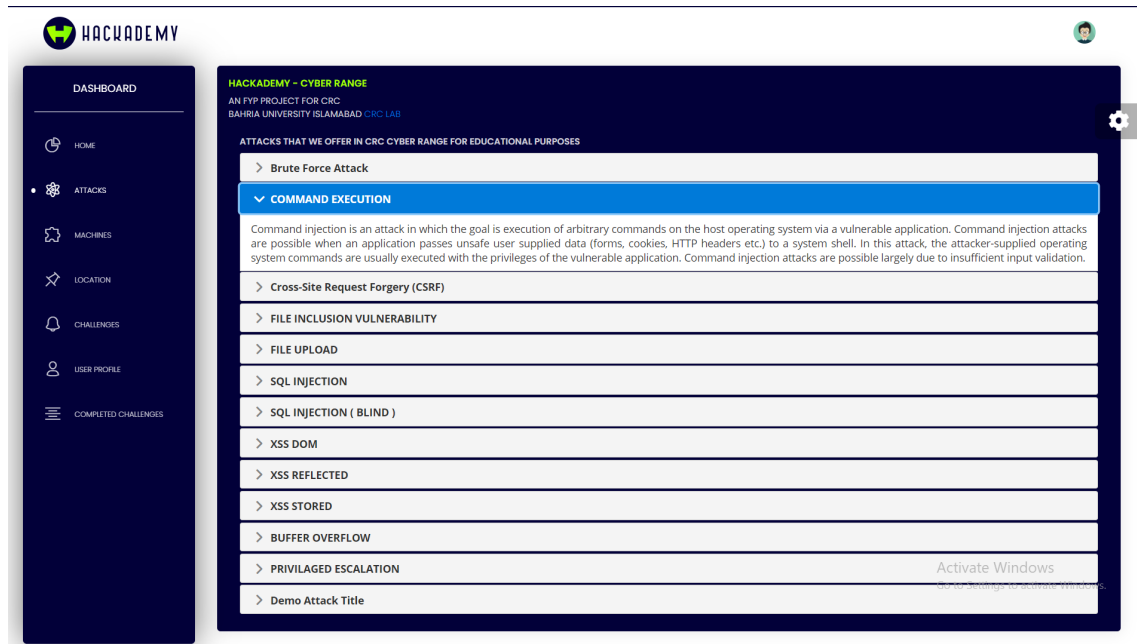


Figure 5.5: Attack List Page

5.6.3 Machine Page

Machines page contains the authorized machines added by the administrators of the system, The user will only be able to have access to the machines that are authorized to them. They can view machines details such as name type, launch date, expiry date and description. They can do actions such as integrate the specified machine and use it to exploit the vulnerability in the said machine. The user can then rate the machine based on experience.



Figure 5.6: Available Machines Page

Chapter 6

System Testing And Evaluation

In this chapter we will do a thorough testing of functional and non functional requirements of the system.

6.1 Graphical User Interface GUI Testing

We have created an eye catching, user friendly interface for the trainees. The GUI does not consist of any complex designing therefore the chances of getting bugs were lowered. Even so, many human subjects were given a chance to test the intractability of the user interface. They later gave a positive feedback and declared that the GUI of our proposed project is infact easy to use.

6.2 Usability Testing

Usability testing refers to the testing of systems interactions with its users. It also refers to how simple or complex was it for a user to operate the system and complete designated tasks. The initial usability testing was done by our close peers and supervisor to checkout whether the proposed system is easy to interact with. These subjects tested to see whether the machines were accessible to them without going into complexities. During the testing phase they realized that the system also provides them with important notices such as session expiry time etc. The overall usability testing helped us find out some setbacks in the system and we fixed those setbacks to test them again. We end result responded with user satisfaction.

6.3 Compatability Testing

Compatibility testing refers to the system being accessible and easy to use in different platforms i.e. on different screen sizes, different browsers, different Operating systems etc. The purpose of this testing is to work ways to provide our proposed system to users working on different types of systems or architectures. Since our project is for pen testers and hacking students therefore we have applied a constraint to the system that only desktop computers or laptop users are allowed to have access to the vulnerable machines, therefore mobile users whether android or IOS are not compatible. Other than this our proposed system works on Windows and Linux. And we have also tested to check whether it works cross browsers and does it provide all of its functionalities and services in other browsers as well or not.

6.4 Performance Testing

Performance testing refers to the efficiency of the system, through performance testing it can be find whether a system is fast enough to load resources along with how well does a system performs under stress conditions. Our system uses Angular framework in frontend which promises to accessibility of all the components on a single reload thus the performance of fetching web pages were quite fast after thorough testing. Our resources does not use much memory power and what little memory resource is used is managed well using mongoDB which is known for its efficiency in stress and scalability. Although there are still some setbacks in the retrieval of vulnerable machines and integrating it in the host system, but we are working on it.

6.5 Exception Handling

Exception handling refers to how well a system works when exceptions are thrown to the system. Exceptions in our system are usually occurring due to human errors. We learned this after usability testing therefore we have handled these exceptions.

We have handled exceptions in the login and registration page so that a user can only enter the system if.

- i.) The user enters correct format of credentials, if an exception is thrown where the user enters incorrect format in the id, a handler prompts the user to type correct credential format.
- ii.) The user enters correct credential, User with incorrect private key tries to login, then the system will prompt them to type again.

6.6 Installation Testing

Since our system requires the user to install and integrate the specified or chosen machine in its system, therefore we tested this action using various types of host systems with different specifications. We came up with the following outcomes.

- i.) Only Host systems whose specifications meet with the minimum requirement for the vulnerable machines can fully use the system.
- ii.) Since the host need to install a virtual machine, thus it might take some time to download and install.

6.7 Test Cases

The following test cases were implemented.

6.7.1 Application Starting test case

Test Case ID	01
Description	Application Starting test
Applicable for	Runs on all browsers
Pre Condition	None
Post Condition	Shows landing page (login)
Steps	Execute the program
Expected	Application should start
Actual	Applications Starts

6.7.2 Successful Login

The dashboard of user should successfully reach after entering correct credentials

Test Case ID	02
Description	Panel Successfully logs in
Applicable for	Runs on all browsers
Pre Condition	Application starts
Post Condition	Shows Dashboard page
Steps	Enter Credentials
Expected	Dashboard is accessible
Actual	Dashboard accessed
Result	Pass

6.7.3 Pages are accessible

After accessing the dashboard, testing to see whether different pages on the side menu such as attacks, machines, user profile etc. are accessible.

Test Case ID	03
Description	Pages should be accessible
Applicable for	Runs on all browsers
Pre-Condition	User logs in
Post Condition	Pages gets open
Steps	Click on each page heading in the side menu
Expected	Page Opens
Actual	Page Opens
Result	Pass

6.7.4 User Sees a list of Machines

User is able to see and access authorized machines

Test Case ID	04
Description	Machines are accessible to the user
Applicable for	Runs on all browsers
Pre-Condition	Opens Machines page
Post Condition	Machine gets accessed
Steps	User triggers a machine he chose
Expected	Machine Setup is accessible
Actual	Machine Setup is accessed
Result	Pass

6.7.5 Session Time Working

Test case to see if the session expiry time works for machines

Test Case ID	05
Description	Machines are expired after session expiry
Applicable for	Runs on all browsers
Pre-Condition	Access a machine
Post Condition	Goes back to the Machine Page
Steps	If the machine session time is up
Expected	Machine should automatically expire
Actual	Machine expires
Result	Pass

6.7.6 CTF Claimed

Test case to see if CTF claimed and changes occur in user profile after the hash is approved

Test Case ID	06
Description	CTF claiming in user profile
Applicable for	Runs on all browsers
Pre-Condition	Hash code found
Post Condition	User statistics updates, achievement updates
Steps	User successfully exploits the vulnerability
Expected	CTF updates the users' statistics and achievements
Actual	Statistics updated; achievement unlocked
Result	Pass

6.7.7 Feedback successfully saved

Test case to see if the users feedback on an exploited machine is saved or not

Test Case ID	07
Description	Feedback Saved
Applicable for	Runs on all browsers
Pre-Condition	Feedback given
Post Condition	Feedback saved
Steps	Feedback is given after successfully capturing the flag
Expected	Feedback is saved
Actual	Feedback saved
Result	Pass

6.7.8 Admin Successfully adds, deletes machines

Test Case ID	08
Description	Admin adds or deletes machines
Applicable for	Runs on all browsers
Pre-Condition	Admin Panel Logs In
Post Condition	Machine successfully added or deleted
Steps	Admin adds or deletes a machine in the system
Expected	Machine added/deleted
Actual	Machine added/deleted
Result	Pass

6.7.9 Admin giving permissions to sub admins

Admin giving various permissions to different sub admins. Eg. Giving a permission to an admin to add delete users. Or to allow admins to change add delete machines etc.

Test Case ID	09
Description	Giving roles and permissions to sub admins
Applicable for	Runs on all browsers
Pre-Condition	Creation of sub admins
Post Condition	Roles and Permissions granted
Steps	Admin checks specified read, write permissions
Expected	Roles granted
Actual	Role granted
Result	Pass

Chapter 7

Conclusion

We developed our project called Hackademy – Vulnerable machine development for cyber drills. We have worked on creating a system for individuals, trainees, pen testers where they can get skilled enough to work in offices to protect their respective organizations from cyber threats. We have provided a solution where these individuals or trainees can get a platform where they can have hands on experience on real life cyber threats that are happening in the world of internet. Our project will make these trainees into cyber security experts by working on different types of threats and different levels of threats. This will make them market ready and to be hired by public and private organizations to secure their day-to-day online operations. The objective was to create a cyber range which will consist of virtual machines or virtual docker containers that can work autonomously on locally on one's system without interfering with the host OS. This way we can inject vulnerabilities in these machines for the trainees to exploit without actually exploiting the host system. This application aims to successfully create an operational simulation of common real life cyber threats within a virtual machine or docker. The idea is to organize each machine in terms of its level of difficulty and the vulnerability it have and then successfully providing the vulnerable machine to the end user (trainee). Each user will have their own dashboard and profile where the score and statistics of their work will be tabled. Their practice history along with their achievement should also be listed successfully. Each user should only have access to their own instance of the machine and their work should not entangle with the work of a fellow trainee.

This project gave us an insight on how projects work in the market and how one ought to work in order to achieve skilled in the market.

7.0.1 Future Work

Working on Hackademy made us realize that such applications are a great way in helping new and current cyber security professionals, Hackademy is a great platforms for these individuals to enhance their cyber security skills and learn the kind of cyber attacks are happening worldwide and how these cyber attacks can cause massive problem for everyone using the world wide web. Following are a few ideas we would love to work on in the future to improve Hackademy.

- i.) More Vulnerable virtual machines
- ii.) A discussion channel to improve the cyber community

- iii.) Blogs and cyber tutorials
- iv.) CTF Competitions

Appendix A

User Manual

User manual is a complete guide for users to understand the working flow of the application. Below is a pictorial user manual in which we have discussed how a user should operate Hackademy in order to perform the work required. We have created a user friendly, easy to learn interactive design so that the user does not feel lost while using the application.

Hackademy has two types of user's:-

1. Administrators
2. Trainees

A.1 User's Side

A.1.1 Landing Page

The first page that the user will see when they enter the website would be a login and sign up page. On this page they can see the application's logo, some animation that will not disturb the user and an input form field where they can enter their credentials to login if they are already a user, Incase they are new user's they can sign up using the same page. .

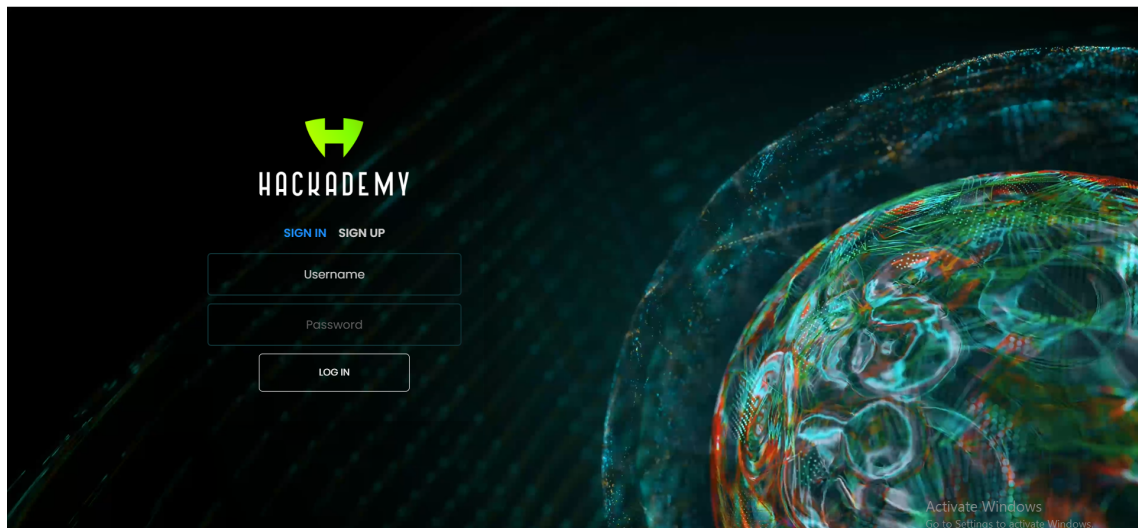


Figure A.1: Login Page

A.1.2 Dashboard

After a user successfully logs in the application after authentication, they land on the user dashboard's home page. On this page they can see a few details of the application. .

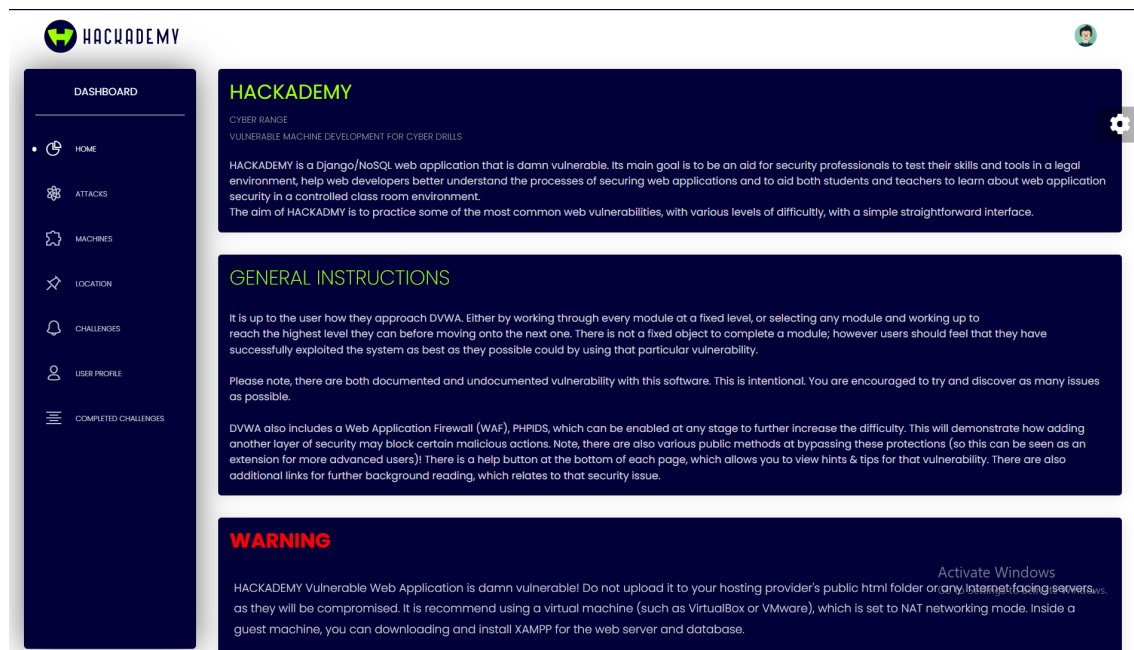


Figure A.2: User's Home Page

A.1.3 Machines

To get started, the user need to initiate a vulnerable virtual machine, to do that he will have to select the machine's page from the left sidebar where he can see multiple pages, each designed for various purposes. When the user triggers the machine of their choice, it automatically downloads the machine in the host system and configure's it for the user. .



The screenshot shows the HackAcademy dashboard with a sidebar on the left and a main content area. The sidebar includes links for Dashboard, Home, Attacks, Machines (selected), Location, Challenges, User Profile, and Completed Challenges. The main content area displays a table titled 'VULNERABLE MACHINES' with columns for #, Machine Name, Machine Type, Launch Date, Expiry Date, and Actions. The table lists 10 machines, all of which are currently 'Off'. The last machine, OWASPBICKS, has a warning message about Windows activation.

#	MACHINE NAME	MACHINE TYPE	LAUNCH DATE	EXPIRY DATE	ACTIONS
1	MUTILLIDAE_2	WEB APP	2019-01-01	2019-02-01	Off
2	DVWA	WEB APP	2019-01-01	2019-02-20	Off
3	BWAPP2	WEB APP	2019-01-01	2021-01-01	Off
4	JUICE SHOP	WEB APP	2019-01-01	2021-01-01	Off
5	GRUVERE	WEB APP	2019-01-01	2021-01-01	Off
6	BADSTOR-DOCKER	WEB APP	2019-01-01	2021-01-01	Off
7	HACKAZONE	WEB APP	2019-01-01	2021-01-01	Off
8	XVWA	WEB APP	2019-01-01	2021-01-01	Off
9	WACKOPICKO	WEB APP	2019-01-01	2021-01-01	Off Activate Windows Go to Settings to activate Windows.
10	OWASPBICKS	WEB APP	2019-01-01	2021-01-01	Off

Figure A.3: Machine List Page

A.1.4 Challenges

The user can also trigger machine's from the challenges page. On this page there are a few challenges organized in their own categories like CTF or Web based challenges.

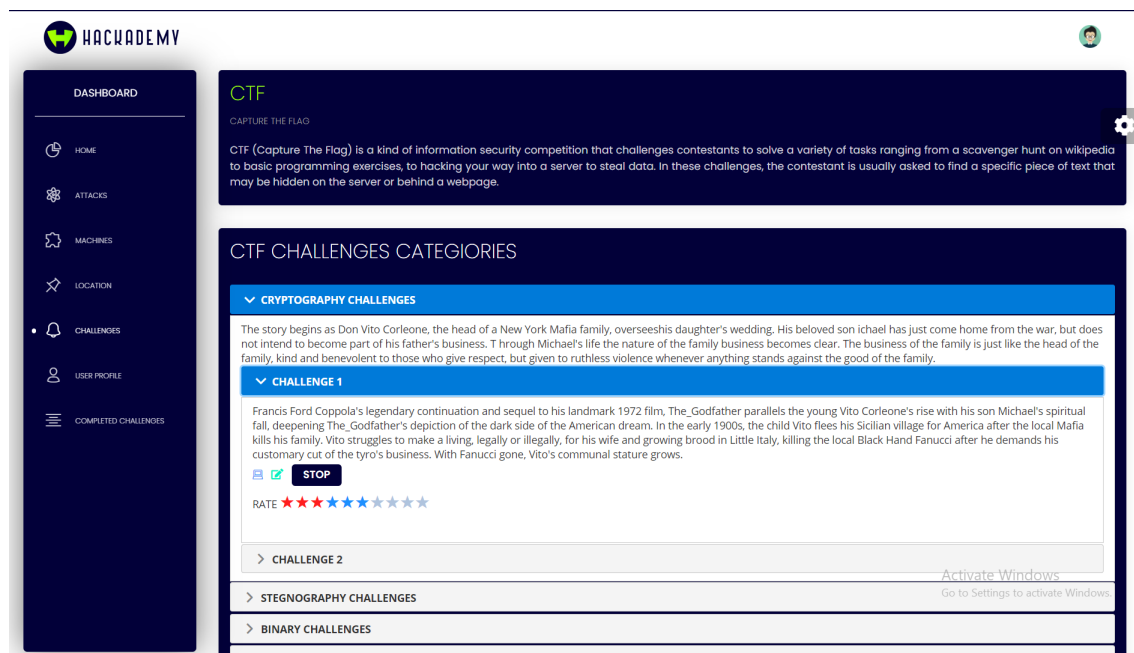


Figure A.4: Challenges Page

A.1.5 Profile

User also has the privilege to update their personal information such as Name, email etc. This can be done through the User Profile Page .

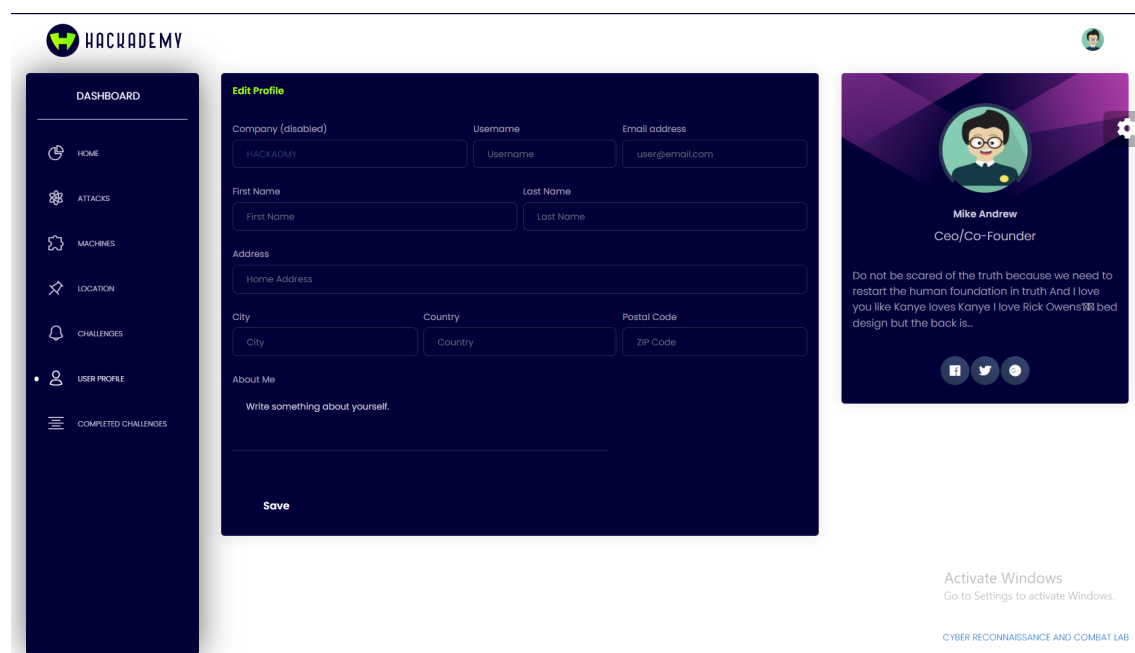


Figure A.5: User Profile Page

A.1.6 Attacks

User can view other information such as the attack's list which are updated on the attacks information page by the administrators .

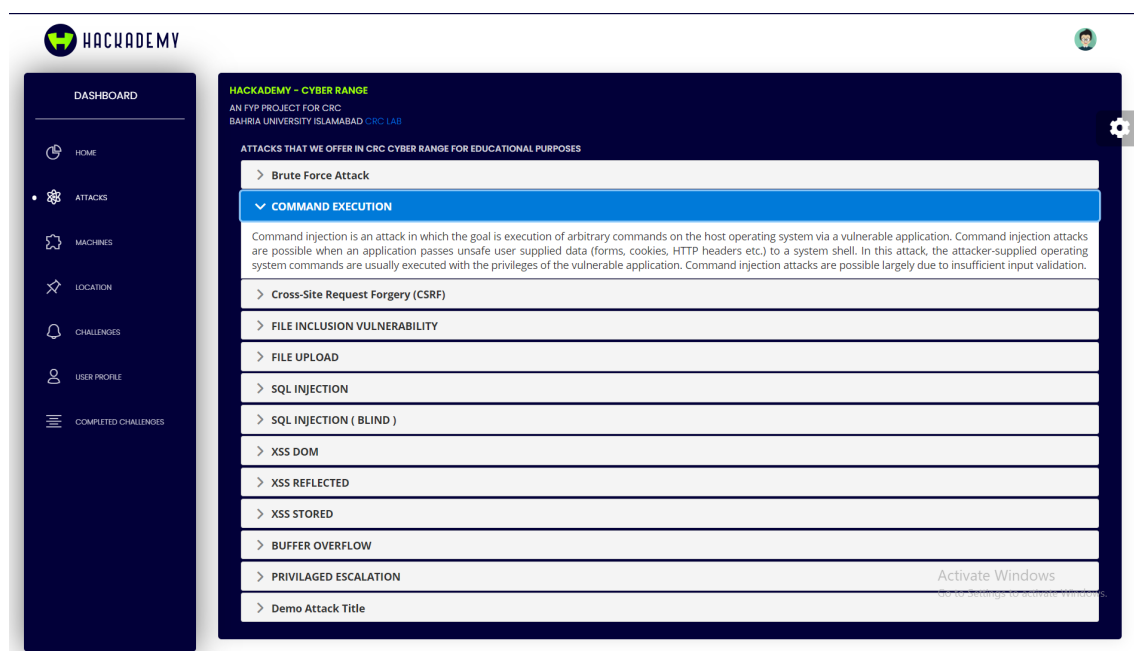


Figure A.6: Attack List Page

A.2 Administrators Side

A.2.1 Landing Page

The first page that the administrators will see when they enter the website would be a login and sign up page. On this page they can see the application's logo, some animation that will not disturb the user and an input form field where they can enter their credentials to login the admins panel. .

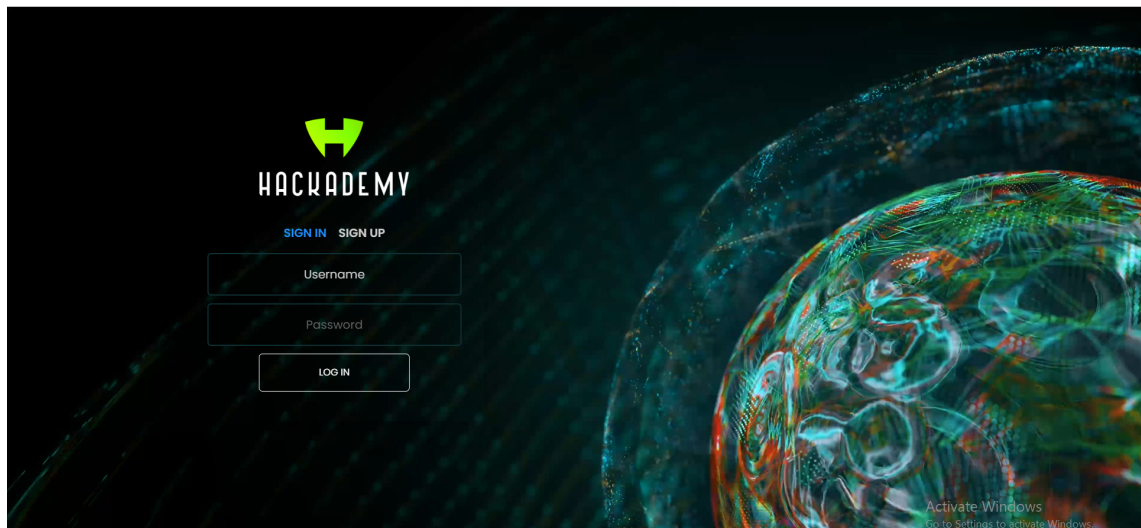


Figure A.7: Login Page

A.2.2 Admins Landing Page

After successfully logging in the system the administrator will see a dashboard. On the home page of this dashboard there will be an organized information about user's and machine's .

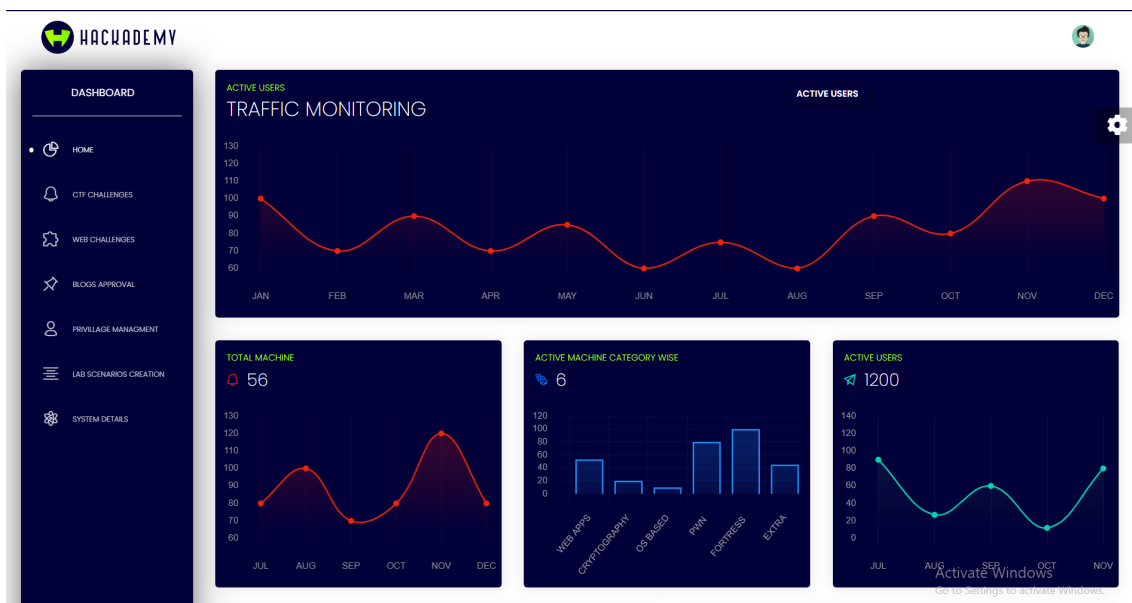


Figure A.8: Admin Home Page

A.2.3 Machines Creation

One of the main task of an administrator is to create vulnerable virtual machines, to do so, they will open the machine creation page and fill the form field with information such as type of machine, type of vulnerability, level of difficulty etc. After filling the required fields, they will press generate which will run a script in the backend and create a vulnerable virtual machine in a specified amount of time.

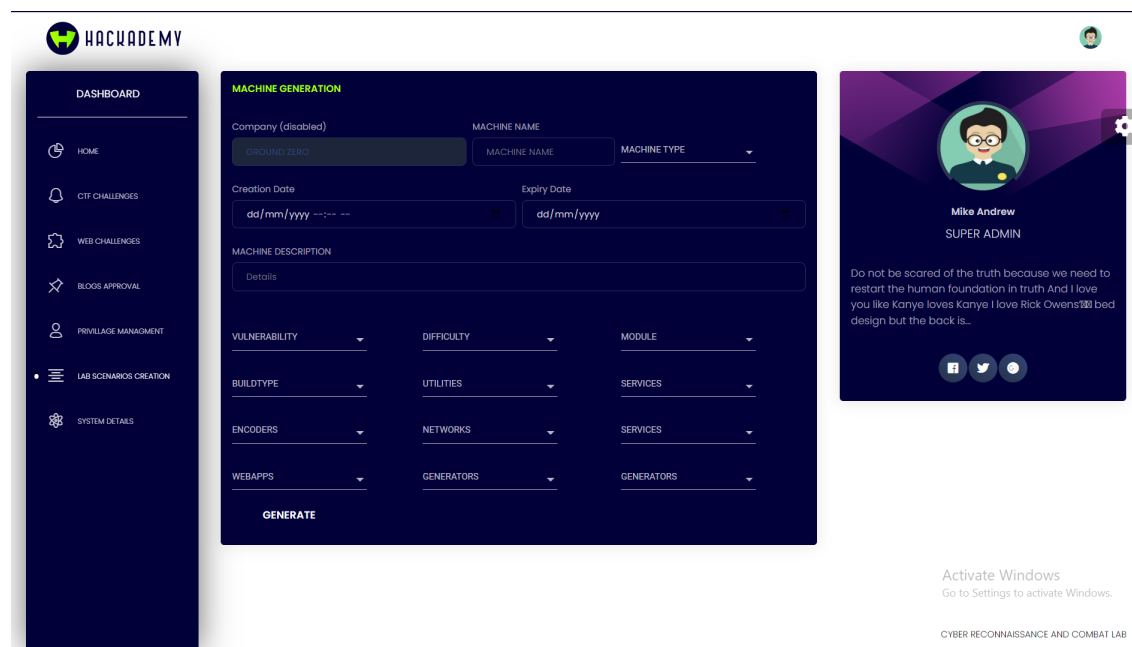


Figure A.9: Machine Generation Page

A.2.4 CTF Challenges Page

Once a machine is successfully created, if the type of the machine created is CTF, then that machine is listed in the CTF Challenges page. .

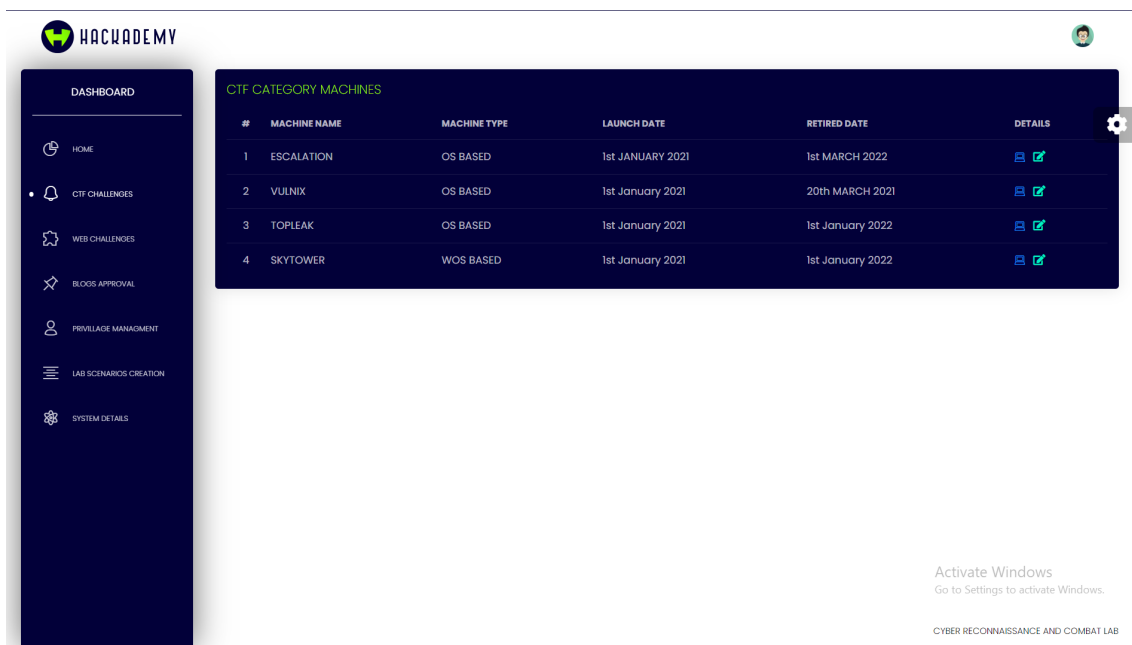





Figure A.10: Active CTF Machines

A.2.5 Web Challenges Page

Once a machine is successfully created, if the type of the machine created is Web based, then that machine is listed in the Web Challenges page. .



The screenshot displays the Hacking Academy dashboard. On the left is a dark sidebar with navigation options: DASHBOARD, HOME, CTF CHALLENGES, WEB CHALLENGES (highlighted), BLOODS APPROVAL, PRIVILEGE MANAGEMENT, LAB SCENARIOS CREATION, and SYSTEM DETAILS. The main content area is titled 'WEB CATEGORY MACHINES' and contains a table with the following data:

#	MACHINE NAME	MACHINE TYPE	ATTEMPTED USERS	USER RATINGS	DETAILS	CONFIGURATION
1	ESCALATION	OS BASED	60	★★★★★★	 	Update Delete
2	VULNIX	OS BASED	80	★★★★★★	 	60
3	TOPLEAK	OS BASED	50	★★★★★★	 	8
4	SKYTOWER	WOS BASED	100	★★★★★★	 	20

At the bottom right of the dashboard, there is a Windows watermark: 'Activate Windows. Go to Settings to activate Windows. CYBER RECONNAISSANCE AND COMBAT LAB'.

Figure A.11: Active Web Machines

A.2.6 Attack's List

Admins can also add in newer attacks that have been discovered. They can add these in the attack's page. .

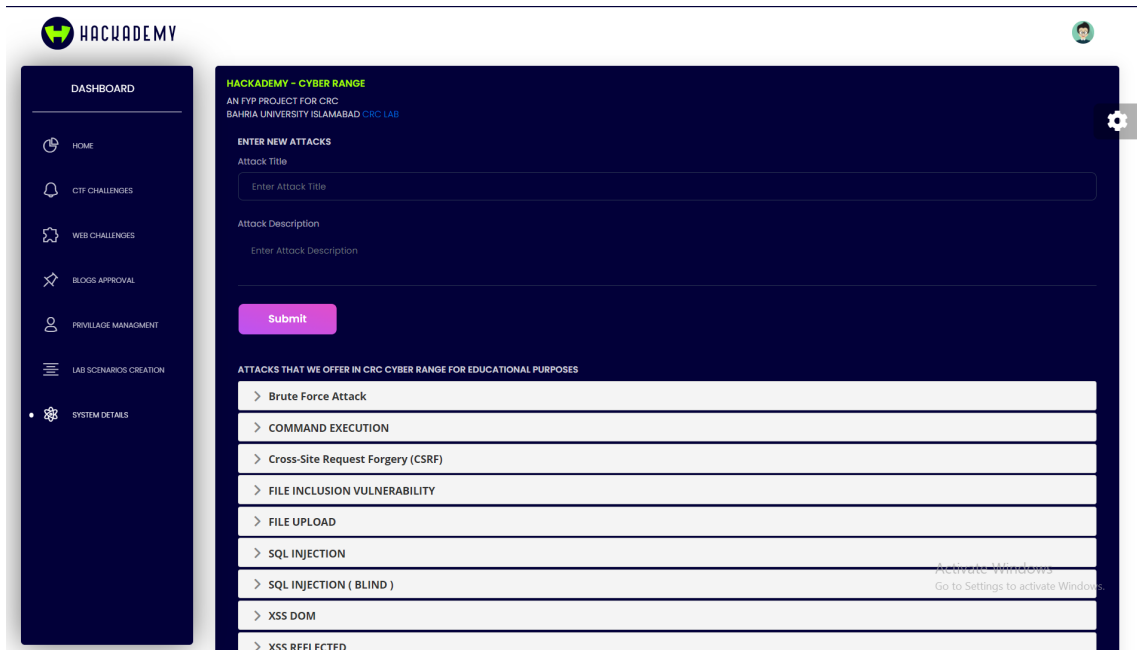


Figure A.12: Attacks

Bibliography

- [1] K Boopathi, S Sreejith, and A Bithin. Learning cyber security through gamification. *Indian Journal of Science and Technology*, 8(7):642–649, 2015.
- [2] Yusuke Hideshima and Hideki Koike. Starmine: A visualization system for cyber attacks. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation-Volume 60*, pages 131–138, 2006.
- [3] Maria Leitner, Maximilian Frank, Wolfgang Hotwagner, Gregor Langner, Oliver Maurhart, Timea Pahi, Lenhard Reuter, Florian Skopik, Paul Smith, and Manuel Warum. Ait cyber range: flexible cyber security environment for exercises, training and research. In *Proceedings of the European Interdisciplinary Cybersecurity Conference*, pages 1–6, 2020.
- [4] Jon R Lindsay. Stuxnet and the limits of cyber warfare. *Security Studies*, 22(3):365–404, 2013.
- [5] Eric Luijff, Kim Besseling, and Patrick De Graaf. Nineteen national cyber security strategies. *International Journal of Critical Infrastructures* 6, 9(1-2):3–31, 2013.
- [6] Joseph S Nye. Nuclear lessons for cyber security? *Strategic Studies Quarterly*, 5(4):18–38, 2011.
- [7] Michael Robinson, Kevin Jones, and Helge Janicke. Cyber warfare: Issues and challenges. *Computers & security*, 49:70–94, 2015.
- [8] Z Cliffe Schreuders, Thomas Shaw, Mohammad Shan-A-Khuda, Gajendra Ravichandran, Jason Keighley, and Mihai Ordean. Security scenario generator ({{{{{{SecGen}}}}}}): A framework for generating randomly vulnerable rich-scenario {VMs} for learning computer security and hosting {CTF} events. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*, 2017.
- [9] M Uma and Ganapathi Padmavathi. A survey on various cyber attacks and their classification. *Int. J. Netw. Secur.*, 15(5):390–396, 2013.
- [10] Ovidiu Valea and Ciprian Oprîşa. Towards pentesting automation using the metasploit framework. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 171–178. IEEE, 2020.
- [11] Rossouw Von Solms and Johan Van Niekerk. From information security to cyber security. *computers & security*, 38:97–102, 2013.
- [12] Jan Vykopal, Radek Ošlejšek, Pavel Čeleda, Martin Vizvary, and Daniel To-varňák. Kypo cyber range: Design and use cases. 2017.

- [13] Jan Vykopal, Martin Vizváry, Radek Oslejsek, Pavel Celeda, and Daniel Tovar-nak. Lessons learned from complex hands-on defence exercises in a cyber range. In *2017 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE, 2017.
- [14] Christian Willems and Christoph Meinel. Online assessment for hands-on cyber security training in a virtual lab. In *Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE, 2012.