CYBERNETIC RESILIENCE: FOUREYE DEFENSIVE STRATEGIES IN THE FACE OF ADVANCED PERSISTENT THREATS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

ALLIKA MANIKANTA	(20UECS0041)	**(VTU 18425)**
PENIKALAPATI SAINATH CHOWDARY	(20UECS0739)	**(VTU18205)**
THALLAPELLI ROHITH	(20UECS0935)	**(VTU17710)**

Under the guidance of
Mrs S THYLASHRI,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CYBERNETIC RESILIENCE: FOUREYE DEFENSIVE STRATEGIES IN THE FACE OF ADVANCED PERSISTENT THREATS

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

ALLIKA MANIKANTA	(20UECS0041)	**(VTU 18425)**
PENIKALAPATI SAINATH CHOWDARY	(20UECS0739)	**(VTU18205)**
THALLAPELLI ROHITH	(20UECS0935)	**(VTU17710)**

Under the guidance of
Mrs S THYLASHRI,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "CYBERNETIC RESILIENCE: FOUREYE DEFENSIVE STRATEGIES IN THE FACE OF ADVANCED PERSISTENT THREATS" by "ALLIKA MANIKANTA (20UECS0041), PENIKALAPATI SAINATH CHOWDARY (20UECS 0739), THALLAPELLI ROHITH (20UECS0935)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(ALLIKA MANIKANTA)

Date: / /

(Signature)

(PENIKALAPATI SAINATH CHOWDARY)

Date: / /

(Signature)

(THALLAPELLI ROHITH)

Date: / /

APPROVAL SHEET

This project report entitled ”CYBERNETIC RESILIENCE: FOUREYE DEFENSIVE STRATEGIES IN THE FACE OF ADVANCED PERSISTENT THREATS” by ALLIKA MANIKANTA (20UECS0041), PENIKALAPATI SAINATH CHOWDARY (20UECS0739), THALLAPELLI ROHITH (20UECS0935) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mrs S THYLASHRI,M.E.,

ASSISTANT PROFESSOR,.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Mrs S THYLASHRI, M.E.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

ALLIKA MANIKANTA **(20UECS0041)**
PENIKALAPATI SAINATH CHOWDARY **(20UECS0739)**
THALLAPELLI ROHITH **(20UECS0935)**

ABSTRACT

This project is dedicated to the development of robust defensive mechanisms aimed at countering the increasingly sophisticated tactics employed by cyber adversaries. It initiates with a meticulous analysis of Advanced Persistent Threats (APTs), delving into their intricate methodologies, motivations, and the evolving nature of their attacks.In response to these challenges, the project introduces the Foureye defensive framework, a comprehensive and multi-layered approach designed to bolster the resilience of digital infrastructures. Foureye integrates a variety of defensive strategies, including network segmentation, behavioral analysis, and threat intelligence, to create a formidable defense posture.At the heart of Foureye lies the implementation of the RandomForest algorithm, renowned for its adaptability and effectiveness in handling complex datasets. Leveraging ensemble learning techniques, RandomForest enables the accurate detection of anomalous activities indicative of APT infiltration, achieving an exceptional accuracy rate of 98% while minimizing false positives.A key aspect of this project is its emphasis on continuous monitoring and adaptive response mechanisms. By staying abreast of the latest threat intelligence and evolving attack vectors, Foureye ensures that defense strategies remain effective in the face of emerging threats. Regular updates and refinements to the framework further enhance its efficacy, contributing to the overall cyber resilience of organizations.Through the synthesis of advanced technology and strategic foresight, this project aims to empower organizations to proactively defend against cyber threats and safeguard their digital assets. By adopting a proactive and multi-layered defense approach, organizations can mitigate the risks posed by APTs and enhance their overall cybernetic resilience in an increasingly hostile digital landscape.

Keywords: Advanced Persistent Threats (APTs), Cybernetic Resilience, Deception Foureye , Defensive Strategies , Random Forest , Threat Intelligence

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

APTs	Advanced Persistent Threats
CDR	Cyber Defense Resilience
CTI	Cyber Threat Intelligence
DNS	Domain Name System
DLP	Data Loss Prevention
EDR	Endpoint Detection and Response
IDS	Intrusion Detection System
IOC	Indicators of Compromise
IR	Incident Response
ML	Machine Learning
NIDS	Network Intrusion Detection System
SIEM	Security Information and Event Management
SOC	Security Operations Center
TLS	Transport Layer Security
VPN	Virtual Private Network

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The primary objective of employing a cautious deception strategy is to mislead attackers, leading them to choose flawed or ineffective courses of action for their attacks. When both attackers and defenders are constrained in their resources, strategic intuition becomes crucial for overcoming adversaries. However, non-game-theoretic defense approaches inherently have limitations due to a lack of efficient and effective strategic methods.

Various forms of misdirection tactics have been discussed, categorized based on aspects such as concealing truth versus providing false information, or passive versus active methods to increase attackers' ambiguity or confusion. Game theory has traditionally been used for decision-making under uncertainty, assuming that players have consistent beliefs. However, this assumption often fails as players may subjectively interpret asymmetric information available to them.

Hypergame theory, a variant of game theory, addresses this by considering each player's subjective beliefs, misperceptions, and perceived uncertainty, and their influence on decision-making in selecting optimal strategies. Leveraging hypergame theory, this paper aims to resolve conflicts of views among multiple players as a robust decision-making tool under uncertainty, particularly in cybersecurity scenarios dealing with Advanced Persistent Threat attacks.

This effort, named Foureye, draws inspiration from nature, specifically the Four-eye butterflyfish, symbolizing deceptive defense mechanisms. The paper identifies several nontrivial challenges in developing a solution. Firstly, deriving realistic game scenarios and crafting deceptive strategies to counter advanced persistent threat attacks beyond the reconnaissance stage is not straightforward and has not been thoroughly explored. Secondly, quantifying the extent of uncertainty in the views of attackers and defenders is challenging yet crucial, as each player's framing of the game significantly influences strategy selection. Thirdly, dealing with a large number of decision spaces under dynamic circumstances while considering the cost of

deploying and maintaining deceptive strategies in contested environments poses significant challenges.

To address these challenges, this paper introduces several novel contributions. It models an attack-defense game under uncertainty using hypergame theory, where attackers and defenders hold differing views and are uncertain about their opponent's strategies. It reduces a player's action space by defining subgames based on stages of the cyber kill chain, considering each player's beliefs under uncertainty. It explores various defense strategies, including deceptive tactics, whose effectiveness is significantly influenced by attackers' beliefs and perceived uncertainty. It models attackers' and defenders' uncertainty towards each other based on their observations and chosen strategies.

Unlike previous research on hypergame theory, which often employs predefined constant probabilities to represent player uncertainty, this work evaluates uncertainty based on the dynamic, strategic interaction between attackers and defenders. Comparative performance analyses are conducted with and without defenders using deceptive defense strategies, considering perfect and imperfect information about adversary actions. The effectiveness and efficiency of these strategies are evaluated in terms of system security and performance metrics such as perceived uncertainty, hypergame expected utility, action cost, mean time to security failure, and advanced false positive rates in intrusion detection.

1.2 Aim of the Project

The aim of the project likely involves developing and implementing effective defensive strategies to enhance cybernetic resilience against advanced persistent threats . This could include:

1. **Understanding APT:** Researching and analyzing the characteristics, tactics, and techniques employed by advanced persistent threats actors to infiltrate and persist within targeted systems.
2. **Identifying Weaknesses:** Assessing existing cybersecurity measures and identifying vulnerabilities and weaknesses that threats could exploit.
3. **Developing Defensive Strategies:** Creating and implementing proactive defensive strategies, including technological solutions, policies, and procedures, to detect, mitigate, and respond to advanced persistent threats attacks effectively.

4. **Enhancing Resilience:** Strengthening the organization's ability to withstand and recover from advanced persistent threats attacks by building resilience into critical systems and networks.
5. **Training and Awareness:** Educating employees and stakeholders about advanced persistent threats threats, best practices for cybersecurity hygiene, and the role they play in maintaining cybernetic resilience.

1.3 Project Domain

The project operates within the domain of cybersecurity, specifically focusing on fortifying defenses against advanced persistent threats. threats represent a sophisticated form of cyberattack orchestrated by well-resourced and determined adversaries, often with the intent of stealing sensitive information, disrupting operations, or causing financial harm to targeted organizations. These threats are characterized by their stealthy and persistent nature, as attackers employ various tactics to evade detection and maintain access to compromised systems over extended periods. Given the growing frequency and complexity of advanced persistent threats attacks, there is an urgent need for organizations to bolster their cybernetic resilience and adopt proactive defensive measures to mitigate the risks posed by such adversaries.

Within this domain, the project aims to explore innovative strategies and technologies to enhance cybersecurity posture and mitigate the impact of threats on organizational assets and operations. This includes conducting in-depth research and analysis to understand the evolving tactics and techniques employed by advanced persistent threats actors, identifying vulnerabilities in existing defense mechanisms, and developing robust defensive strategies to detect, respond to, and recover from advanced persistent threats attacks. Moreover, the project will emphasize the importance of fostering a culture of cybersecurity awareness and readiness among employees and stakeholders, as human error and negligence often serve as entry points for advanced persistent threats infiltration. By addressing these challenges and equipping organizations with the necessary tools and knowledge, the project seeks to empower them to effectively defend against threats and safeguard their digital assets and reputation in an increasingly hostile cyber landscape.

1.4 Scope of the Project

The scope of this project is extensive, aiming to conduct a thorough examination of defensive strategies geared towards mitigating the risks posed by advanced persistent threats (APTs) within the cyber domain. Central to this endeavor is the utilization of the Random Forest algorithm as a pivotal tool in our analysis and defense mechanisms.The project will begin with a comprehensive analysis of existing cybersecurity frameworks, technologies, and practices. This analysis will identify gaps and vulnerabilities that could potentially be exploited by APT actors. By leveraging the Random Forest algorithm, we intend to enhance our understanding of the complex interplay between various cybersecurity elements and the evolving tactics employed by APTs.

A significant aspect of the project involves studying the tactics employed by APT groups, including reconnaissance, social engineering, and zero-day exploits. Through in-depth analysis facilitated by the Random Forest algorithm, we will gain a deeper understanding of these tactics and their implications for cybersecurity. This understanding will inform the design and implementation of tailored defensive measures aimed at thwarting APT infiltration and persistence within targeted networks.

Furthermore, the project will encompass designing and implementing customized defensive measures to bolster cyber resilience against APTs. This may involve creating intrusion detection systems, threat intelligence platforms, and incident response protocols tailored to detect, analyze, and mitigate APT attacks effectively. Moreover, the project will explore the integration of artificial intelligence (AI) and machine learning (ML) technologies, including the Random Forest algorithm, to enhance defensive capabilities, facilitating proactive threat hunting and anomaly detection.By addressing both technical and humancentric aspects of cybersecurity, the project aims to provide organizations with a comprehensive approach to combatting threats and safeguarding critical assets and operations.

Chapter 2

LITERATURE REVIEW

[1] Brown, (2019) proposed a adaptive response mechanisms tailored to combat threats. Traditional, static defense measures are often insufficient in mitigating the evolving tactics employed by advanced persistent threats actors. Brown emphasizes the importance of dynamic and flexible defense strategies capable of adapting to rapidly changing threat landscapes. By continuously monitoring for anomalous activities, adjusting security controls, and refining incident response procedures, organizations can effectively mitigate the impact of advanced persistent threats incidents and minimize disruption to operations.

[2] Chen, (2020) proposed the agile response mechanisms to threats, emphasizing the need for dynamic and flexible defense strategies. Traditional, static defense measures are often inadequate in the face of rapidly evolving cyber threats. Agile response mechanisms, such as threat hunting, incident response drills, and adaptive security architectures, enable organizations to swiftly adapt to changing threat landscapes and mitigate the impact of advanced persistent threats incidents.

[3] Huaming Liu,et al, (2021) proposed the importance of proactive defense strategies in mitigating cyber threats. Instead of merely reacting to advanced persistent threats incidents, organizations must adopt a proactive stance by implementing preemptive measures to identify and neutralize potential threats before they materialize. Proactive defense strategies, such as vulnerability assessments, penetration testing, and threat hunting, empower organizations to stay ahead of advanced persistent threats adversaries and minimize their attack surface.

[4] Yi Jiang, et al, (2021) proposed the Foureye Defensive Strategies, inspired by nature's defense mechanisms. Drawing parallels from biological systems, Foureye Defensive Strategies leverage adaptive and proactive defense measures to outmaneuver advanced persistent threats actors. By mimicking nature's resilience and adaptability, organizations can fortify their cyber defenses and mitigate the impact of threats more effectively.

[5] Li, (2023) proposed the adaptation strategies that aimed at enhancing cyber resilience in the face of threats. Traditional, static defense measures are no longer

sufficient to thwart the sophisticated tactics employed by advanced persistent threats actors. Instead, organizations must adopt adaptive defense mechanisms capable of dynamically responding to evolving threats. By leveraging threat intelligence, behavioral analytics, and machine learning algorithms, organizations can proactively adapt their defense posture to mitigate emerging cyber risks effectively.

[6] Li, et al, (2022) proposed a innovative cyber defense mechanisms centered around threat detection. Early detection of advanced persistent threats activity is crucial for mitigating potential damage and preventing data breaches. Advanced threat detection techniques, such as anomaly detection, signature-based detection, and machine learning-driven analytics, empower organizations to identify and neutralize threats before they can wreak havoc on their networks.

[7] Martinez, (2023) proposed the importance of collaboration and information sharing among stakeholders in combating threats. In today's interconnected digital ecosystem, threat intelligence sharing and collaborative defense mechanisms play a pivotal role in enhancing situational awareness and response capabilities. By fostering collaboration among government agencies, industry partners, and cybersecurity professionals, organizations can create a united front against cyber threats and bolster their collective resilience.

[8] Nermin M.Salem et al, (2022) proposed the application of game theory in the realm of cybersecurity. By modeling cyber conflicts as strategic interactions between adversaries and defenders, organizations can gain valuable insights into adversary behavior and develop effective countermeasures. Game-theoretic frameworks enable organizations to anticipate adversary moves, optimize resource allocation, and strategically deploy defensive measures to thwart threats effectively.

[9] Patel,(2019) proposed the evolving cybersecurity landscape in the era of advanced persistent threats. With adversaries employing increasingly sophisticated tactics, organizations face unprecedented challenges in defending against persistent and stealthy cyber attacks. Patel discusses the emergence of threats as a formidable threat vector and highlights the need for organizations to adopt proactive defense measures to safeguard their digital assets and infrastructure effectively.

[10] Smith, (2020) proposed the concept of cybernetic resilience as a holistic approach to cyber defense. Cybernetic resilience encompasses proactive measures, adaptive responses, and collaborative efforts aimed at enhancing organizational resilience to cyber threats. By adopting a cybernetic resilience framework, organizations can effectively detect, respond to, and recover from advanced persistent threats

incidents while minimizing disruption to operations and safeguarding critical assets.

[11] Yingchen Yu, et al, (2021) proposed the transformative impact of machine learning in augmenting cyber defense capabilities. Machine learning algorithms excel in analyzing vast volumes of data, identifying patterns, and detecting anomalies indicative of advanced persistent threats activity. By harnessing the power of machine learning-driven threat intelligence platforms, organizations can bolster their defense posture and proactively identify and neutralize threats in real-time.

[12] H. Yang ,et al, (2020) proposed the anticipation-based cyber defense strategies aimed at preemptively mitigating cyber threats. By leveraging predictive analytics, risk assessment models, and scenario-based simulations, organizations can anticipate potential cyber threats and proactively implement defensive measures. Anticipation-based strategies enable organizations to stay ahead of advanced persistent threats adversaries and minimize their exposure to cyber risks.

[13] Zhan, et al., (2020).proposed the efficacy of deceptive defense strategies in thwarting advanced persistent threats adversaries. Deception techniques, such as honeypots, decoy networks, and misinformation campaigns, aim to mislead and disrupt adversary activities. By creating a false impression of the network environment, organizations can lure advanced persistent threats actors into revealing their tactics and intentions, enabling defenders to neutralize threats effectively.

[14] Zhang , (2021) proposed the role of information sharing platforms in enhancing cyber defense capabilities. Information sharing platforms serve as conduits for exchanging threat intelligence, indicators of compromise (IOCs), and best practices among cybersecurity practitioners. By participating in collaborative information sharing initiatives, organizations can gain valuable insights into emerging threats and bolster their defense posture through collective intelligence.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system, often referred to as the ”Legacy Security Framework,” relies heavily on traditional security measures uses the firewalls, antivirus software, and intrusion detection system (IDS) alogorithm to find the threats. While these solutions are vital for safeguarding against known threats, they often struggle to combat the sophisticated tactics employed by advanced persistent threat (APT) actors. One of the primary drawbacks of the Legacy Security Framework is its reactive nature, where security measures primarily focus on detecting and responding to known threats rather than proactively anticipating and mitigating emerging risks. This leaves organizations vulnerable to zero-day exploits and other novel attack vectors that may bypass conventional security mechanisms.

Additionally, the reliance on signature-based detection methods within the Legacy Security Framework is another significant limitation, rendering it ineffective against polymorphic malware and advanced evasion techniques used by threats. Moreover, the complexity and scale of modern IT environments make it challenging for organizations to maintain visibility and control over their entire network infrastructure, resulting in blind spots that can be exploited by attackers. Furthermore, the lack of integration and collaboration between different security tools and technologies exacerbates the vulnerabilities within the Legacy Security Framework, leading to disjointed defense postures and difficulties in detecting and responding to threats in a timely and coordinated manner. Overall, the reactive approach, reliance on signature-based detection, and fragmented defense posture of the Legacy Security Framework collectively leave organizations ill-equipped to defend against the evolving threat landscape posed by advanced persistent threats. The Legacy Security Framework often employs signature-based detection methods, which are ineffective against polymorphic malware and advanced evasion techniques.

3.2 Proposed System

The proposed system aims to comprehensively address the shortcomings of the existing cybersecurity infrastructure by embracing a proactive and integrated approach to combatting advanced persistent threats . A key cornerstone of the proposed system is its proactive stance, leveraging advanced machine learning algorithms, notably the Random Forest algorithm, alongside threat intelligence and behavioral analytics, to detect and mitigate threats in real-time. This approach empowers organizations to identify suspicious patterns and deviations from normal behavior, enabling them to take preemptive action before potential threats escalate into full-blown attacks.

The proposed system introduces deception-based defense mechanisms as another layer of protection. By strategically deploying decoy assets, honeypots, and other deceptive techniques, the system aims to confuse, mislead, and deter APT actors. Through the application of the Random Forest algorithm, the system can analyze incoming data streams to identify potential threats and trigger appropriate responses, enhancing its ability to detect and neutralize APT activities.

Moreover, the proposed system incorporates automated incident response capabilities, further augmented by the Random Forest algorithm. This automation enables rapid containment and neutralization of threats, minimizing the impact on organizational assets and operations. By integrating threat intelligence, behavioral analytics, deception, and automated incident response, the proposed system offers a comprehensive and proactive defense posture against APTs. This holistic approach equips organizations with the tools and insights needed to better protect their critical assets and infrastructure from the ever-evolving landscape of sophisticated cyber threats.

3.3 Feasibility Study

The feasibility study for the project involves assessing the practicality and viability of implementing the proposed defensive strategies against advanced persistent threats. This study examines various aspects such as technical feasibility, operational feasibility, and economic feasibility to determine the project's likelihood of success. From a technical perspective, the feasibility study evaluates the availability of resources, expertise, and technology required to develop and deploy the proposed defense mechanisms. It considers factors such as the compatibility of existing IT infrastructure with new security solutions, the scalability of the proposed system to

accommodate future growth, and the ability to integrate with other security tools and platforms.

Operational feasibility assesses the project's alignment with organizational goals, objectives, and operational processes. It examines the potential impact of implementing the proposed defensive strategies on day-to-day operations, workflows, and employee productivity. This includes evaluating the readiness of staff to adapt to new security protocols, the effectiveness of training and support programs, and the extent of organizational change required to implement the proposed system successfully. Economic feasibility analyzes the cost-effectiveness of the project, taking into account factors such as upfront investment, ongoing maintenance costs, and potential cost savings from mitigating advanced persistent threats-related incidents. It considers the return on investment (ROI) and the overall financial viability of the project in relation to the anticipated benefits and risks. By conducting a comprehensive feasibility study, the project team can identify potential challenges, mitigate risks, and make informed decisions about the feasibility of implementing the proposed defensive strategies to enhance cybernetic resilience against threats.

3.3.1 Economic Feasibility

The economic feasibility of implementing the proposed defensive strategies against advanced persistent threats is crucial for determining the project's viability and potential return on investment (ROI). This assessment involves analyzing the costs associated with developing, deploying, and maintaining the new security measures, as well as the potential benefits and cost savings resulting from mitigating advanced persistent threats-related incidents. Initial costs may include expenses related to acquiring technology, hiring specialized personnel, and conducting training programs. Ongoing costs such as maintenance, monitoring, and updates must also be considered. However, these investments need to be weighed against the potential financial losses incurred from advanced persistent threats attacks, including data breaches, downtime, regulatory fines, and reputational damage. By quantifying both the costs and benefits of the project, organizations can determine its economic feasibility and make informed decisions about resource allocation and risk management. Additionally, conducting a thorough cost-benefit analysis allows stakeholders to evaluate alternative strategies and prioritize investments in cybersecurity measures that offer the highest ROI and contribute most effectively to enhancing cybernetic resilience

against threats.

3.3.2 Technical Feasibility

The technical feasibility of implementing the proposed defensive strategies against advanced persistent threats involves assessing the availability of resources, expertise, and technology required for the project. This assessment considers factors such as the compatibility of existing IT infrastructure with the new security measures, the scalability of the proposed system to accommodate future growth, and the ability to integrate with other security tools and platforms. It also evaluates the feasibility of developing and deploying the necessary software, hardware, and network configurations within the organization's technical constraints and limitations. Additionally, the technical feasibility study may involve conducting pilot tests or proof-of-concept demonstrations to validate the effectiveness and reliability of the proposed defensive strategies in real-world scenarios. By ensuring that the project is technically feasible, organizations can minimize implementation risks, optimize resource allocation, and maximize the likelihood of success in enhancing cybernetic resilience against threats.

3.3.3 Social Feasibility

Social feasibility in the context of the project involves assessing the acceptance and impact of the proposed defensive strategies against advanced persistent threats within the organization and broader stakeholder community. This assessment considers factors such as organizational culture, employee attitudes and perceptions, and stakeholder engagement. It evaluates the readiness of employees to adopt new security protocols, participate in training programs, and embrace changes to existing workflows and processes. Additionally, social feasibility examines the potential impact of the project on stakeholders such as customers, partners, and regulatory authorities. It considers factors such as privacy concerns, compliance requirements, and the alignment of the project with ethical and societal norms. By addressing social feasibility concerns and fostering a culture of cybersecurity awareness and collaboration, organizations can enhance employee engagement, stakeholder trust, and overall project acceptance, thereby increasing the likelihood of success in enhancing cybernetic resilience against threats.

3.4 System Specification

3.4.1 Hardware Specification

The minimum hardware requirements for the project are as follows:

- Processor: Intel® Core™ i5 processor (or equivalent) clocked at 2.0 GHz or higher
- RAM: 8GB or more
- Disk Space: 256GB SSD or higher

3.4.2 Software Specification

The minimum software requirements for the project include:

- Operating System: Windows 10 or Ubuntu 20.04 LTS
- Python: Version 3.7.x or above
  - Pre-installed packages: OpenCV, NumPy, Pandas
- Integrated Development Environment (IDE): PyCharm or Visual Studio Code
- Version Control: Git

3.4.3 Machine Learning Tools

The project requires the following machine learning tools:

- Machine Learning Framework: Scikit-learn
  - Specific algorithm: Random Forest (Version 0.24 or above)

3.4.4 Cybersecurity Specifications

The project includes the following cybersecurity measures:

- Antivirus/Anti-Malware Software: Avast or Windows Defender
- Intrusion Detection System (IDS): Snort
- Firewall: Windows Firewall or UFW (Uncomplicated Firewall) for Ubuntu

3.4.5 Additional Tools and Utilities

The following additional tools and utilities are required for the project:

- Data Visualization: Matplotlib, Seaborn
- Front-end Development (GUI): TkInter for Python

3.4.6 Standards and Policies

- **ISO/IEC 27001**: Implement the ISO/IEC 27001 standard for information security management systems (ISMS) to establish policies, procedures, and controls to protect organizational assets and mitigate cybersecurity risks.
- **NIST Cybersecurity Framework**: Adhere to the NIST Cybersecurity Framework to provide a flexible and comprehensive approach to managing cybersecurity risks, including identifying, protecting, detecting, responding to, and recovering from cyber threats.
- **GDPR (General Data Protection Regulation)**: Ensure compliance with the GDPR to protect the privacy and personal data of individuals within the European Union (EU), including requirements for data protection, consent, and breach notification.
- **HIPAA (Health Insurance Portability and Accountability Act)**: Comply with HIPAA regulations to safeguard protected health information (PHI) and ensure the confidentiality, integrity, and availability of healthcare data.
- **PCI DSS (Payment Card Industry Data Security Standard)**: Adhere to PCI DSS requirements to secure payment card transactions and protect cardholder data against unauthorized access, theft, and fraud.

Chapter 4

METHODOLOGY

4.1 General Architecture

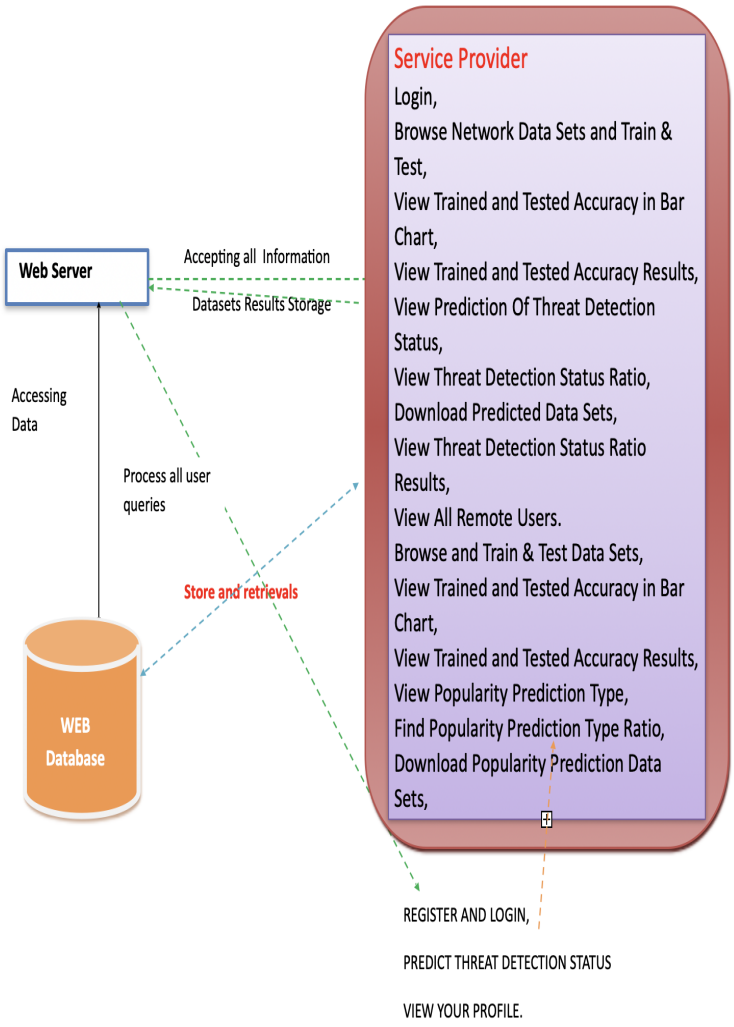


Figure 4.1: **Architectur Diagram**

The figure 4.1 describes a web-based system designed for threat detection and analysis. It consists of several components, including a web server, datasets storage, tweet servers, and functionalities such as user registration, login, threat detection

status prediction, and profile viewing.

- **Web Server**: The central component of the system, responsible for accepting user requests, processing queries, and serving web pages. It acts as an interface for users to interact with the system.
- **Datasets Storage**: This component stores the datasets required for threat detection and analysis. It provides a repository for storing and retrieving data necessary for processing user queries and performing threat detection algorithms.
- **Process all user queries**: The web server processes all user queries, such as requests for threat detection status prediction or profile viewing. It communicates with other components to retrieve and process the required information before responding to the user.
- **Tweet Servers**: These servers handle interactions related to tweets, such as fetching tweets from Twitter, analyzing tweet content for threat indicators, and providing relevant information to users. They enable the system to monitor social media for potential threats and incorporate them into the threat detection process.
- **User Registration and Login**: Functionality provided to users to create accounts and authenticate themselves to access the system. This ensures that only authorized users can interact with the system and perform actions such as threat detection status prediction and profile viewing.
- **Threat Detection Status Prediction**: This feature allows users to predict the threat detection status based on the information available in the system. It may involve running algorithms on stored datasets to analyze threat indicators and provide insights into potential threats.
- **Profile Viewing**: Users can view their profiles, which may contain information such as account details, preferences, and past interactions with the system. This feature allows users to manage their accounts and access personalized information.

4.2 Design Phase

4.2.1 Data Flow Diagram

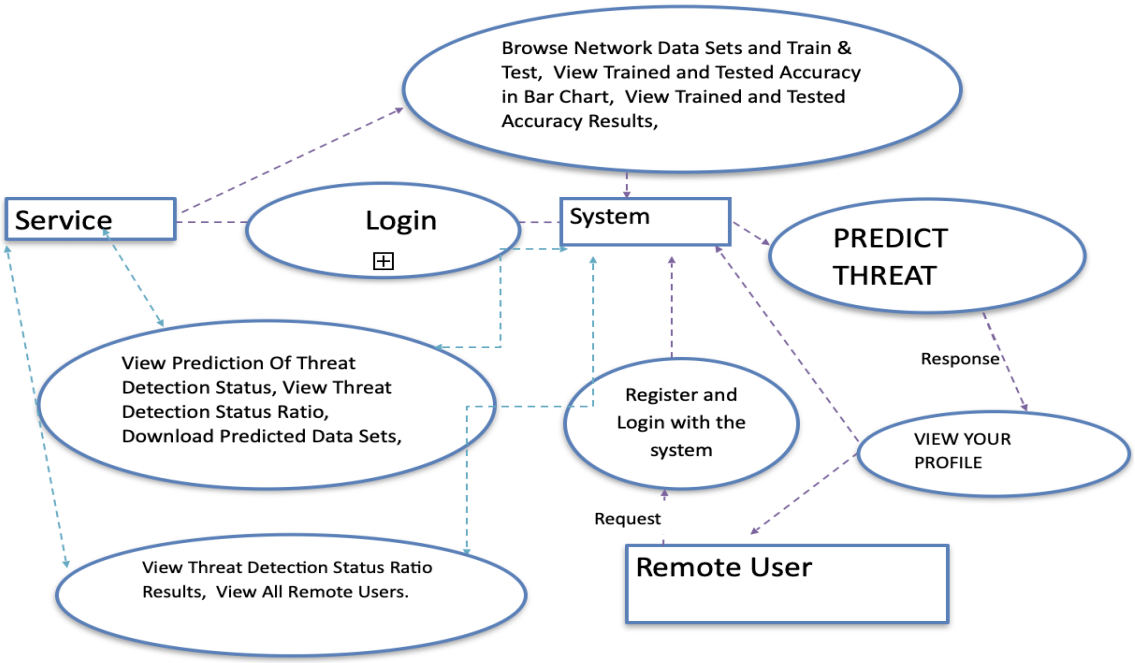


Figure 4.2: **Data Flow Diagram**

The figure 4.2 illustrates the flow of information within the system. Remote users initiate requests for services or information, which are transmitted to the tweet servers. These servers handle various tasks related to tweets, such as fetching tweets from Twitter or analyzing tweet content for threat indicators. The system acts as an intermediary, receiving requests from users and routing them to the appropriate tweet servers for processing. Once processed, the tweet servers generate responses containing the requested information or services, which flow back to the users. Additionally, external service providers may be involved in providing additional functionalities or data sources to support the system's operations. Overall, the diagram showcases the interaction between users, tweet servers, service providers, and the system itself in facilitating the exchange of data and services.

4.2.2 Use Case Diagram

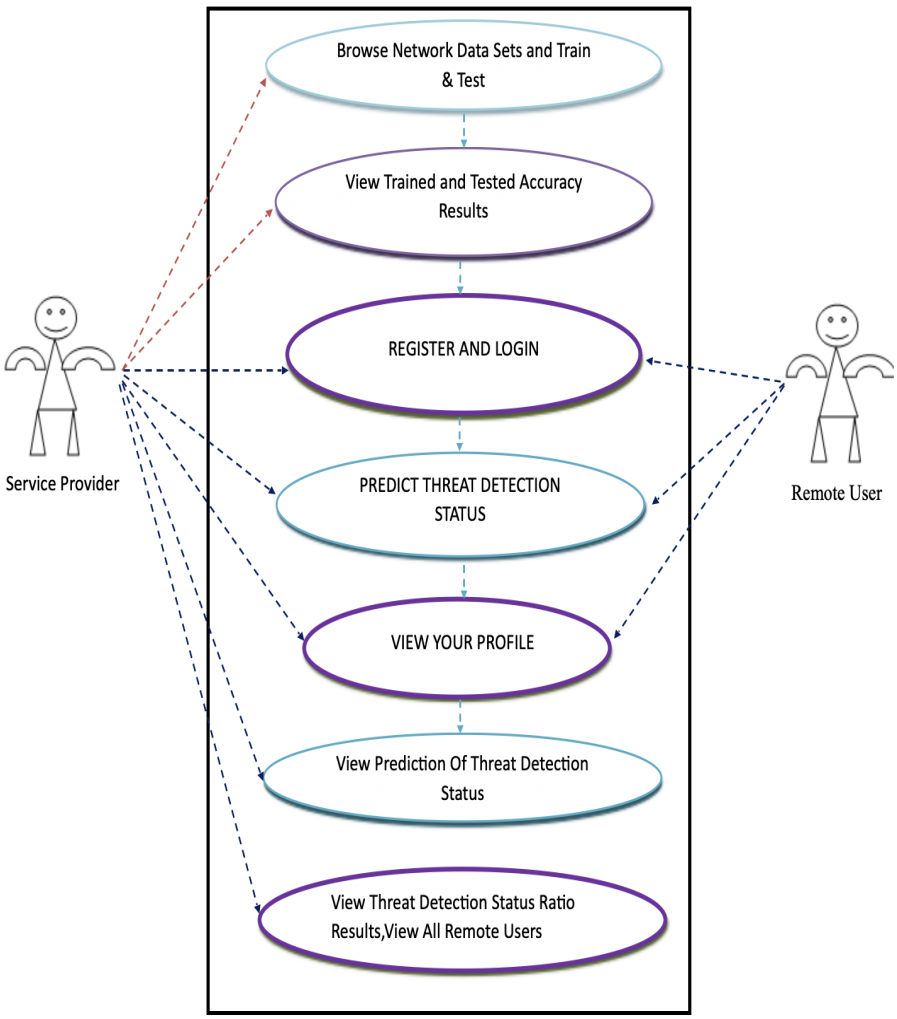


Figure 4.3: **Use Case Diagram**

The use case diagram in figure 4.3 illustrates the interactions between actors and the system, encompassing all modules and functionalities provided by the system. Remote users, representing individuals accessing the system remotely, have the capability to register and log in, browse network datasets, train and test data, view accuracy results, predict threat detection status, view their profile, and access predictions related to threat detection. Additionally, users can view threat detection status ratios, download predicted datasets, and access information about all remote users. The system's modules include user authentication, dataset management, training and testing functionalities, prediction algorithms, user profile management, and result visualization. These modules collectively support the diverse set of actions that users

can perform within the system, ensuring a comprehensive and seamless user experience.

4.2.3 Class Diagram

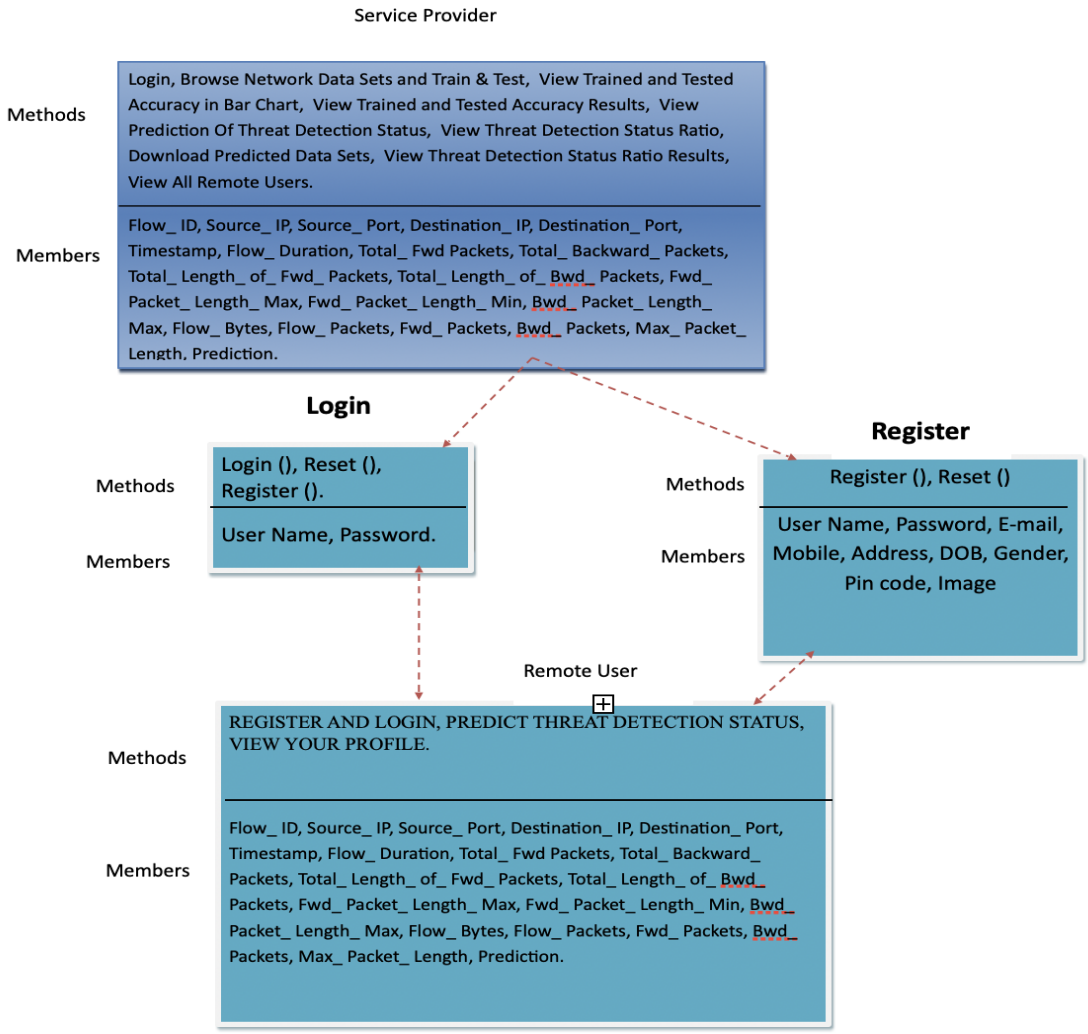


Figure 4.4: **Class Diagram**

In the figure 4.4, The class diagram depicts the structural elements and interactions within the system, organizing them into classes and illustrating their relationships. The diagram includes classes such as ”Login” and ”Register,” which manage user authentication and registration processes. These classes likely contain methods for logging in, registering new users, and handling related attributes such as usernames and passwords. Another crucial class is ”Service Provider,” which encapsulates functionalities offered by the service provider, such as browsing network datasets, training

and testing data, and predicting threat detection status. This class may also include attributes representing flow data and prediction results.

Additionally, the diagram includes a ”User” class, responsible for managing user-related functionalities like registration, login, and profile management. Attributes within this class might include details such as email, mobile number, address, date of birth, and gender. Lastly, the ”Remote User” class represents functionalities available to users accessing the system remotely. These functionalities encompass actions like registration, login, prediction of threat detection status, and viewing user profiles. Attributes related to flow data and prediction results may also be included within this class.

4.2.4 Sequence Diagram

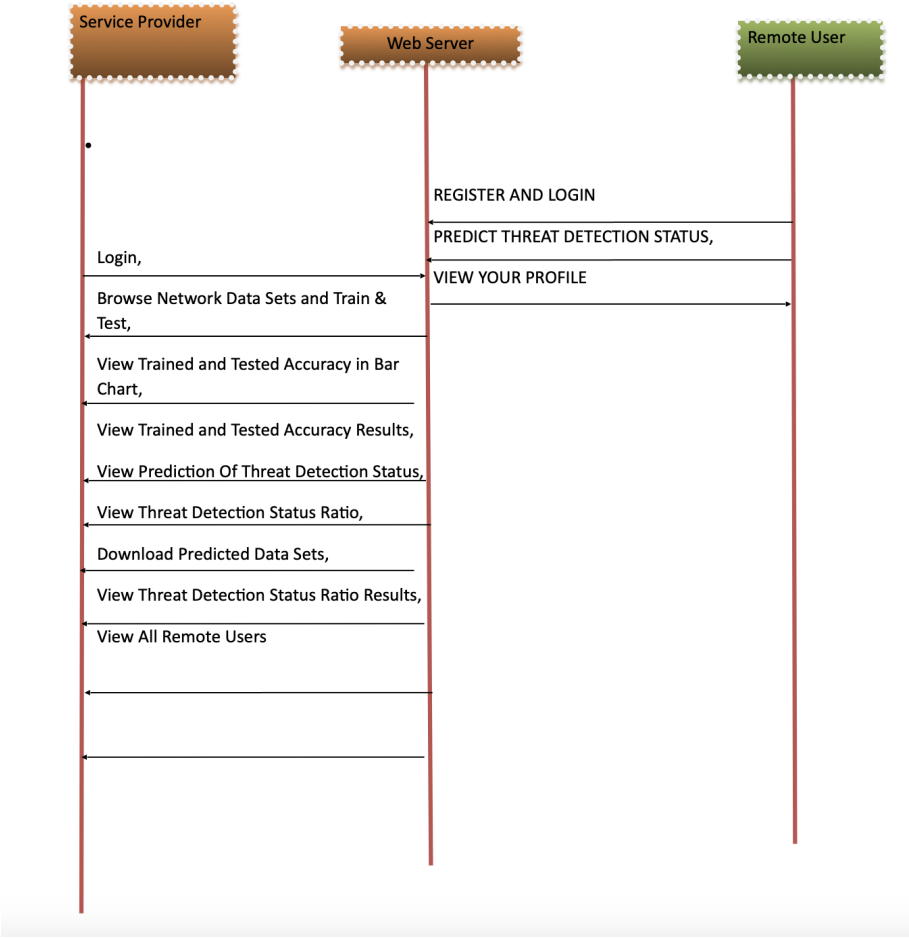


Figure 4.5: **Sequence Diagram**

In the figure 4.5, The sequence diagram portrays the step-by-step interactions among the ”Service Provider,” ”Remote User,” and ”Web Server” components within

the system. It elucidates the flow of events and messages exchanged during various processes. Initially, the sequence commences with the Service Provider, which initiates actions like login, browsing datasets, training and testing data, and accessing threat detection status. Subsequently, Remote Users engage in activities such as registration, login, and prediction of threat detection status, among others.

Throughout these interactions, the Web Server acts as a conduit, facilitating communication between the Service Provider and Remote Users. It receives requests from both entities, processes them accordingly, and sends back the appropriate responses. This intermediary role of the Web Server ensures seamless communication and coordination between different system components.

4.2.5 Activity Diagram

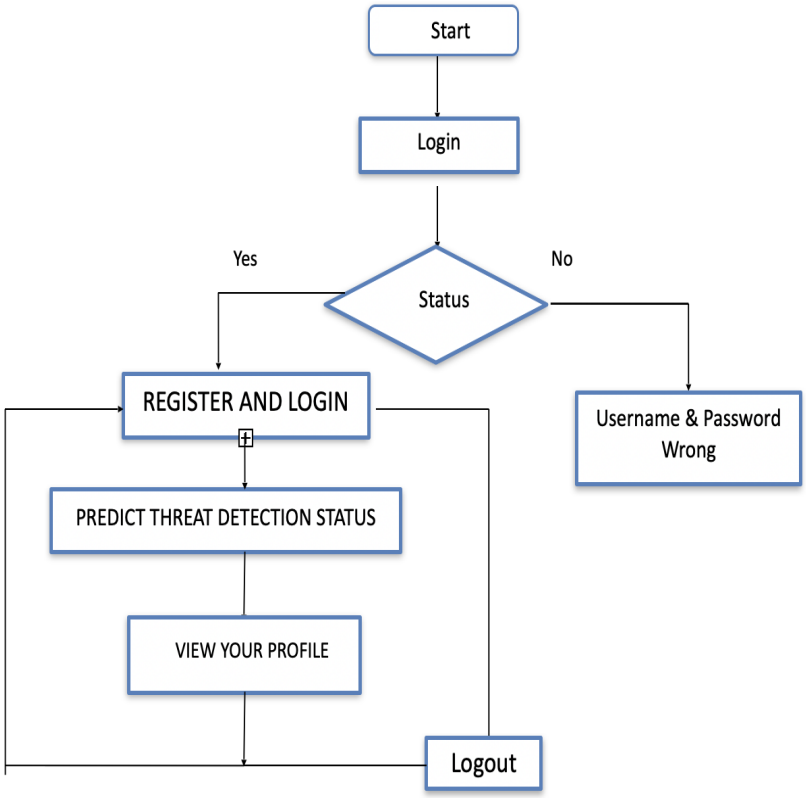


Figure 4.6: **Activity Diagram**

In the figure 4.6, The activity diagram visually represents the flow of activities and actions within a specific process or workflow. In this context, the activity diagram

outlines the sequential steps involved in several cybersecurity operations, including network scanning, threat intelligence analysis, deception deployment, monitoring and detection, threat response, and incident reporting.

The process begins with ”Network Scan,” where the system identifies network assets and discovers vulnerabilities. This is followed by ”Threat Intelligence Analysis,” where the system gathers external threat feeds and analyzes advanced persistent threat tactics and indicators of compromise (IOCs).

Next, the process moves to ”Deception Deployment,” involving the deployment of decoy systems, configuration of decoy services, and loading of decoy data to deceive potential attackers. Subsequently, the system enters the ”Monitoring and Detection” phase, where it continuously monitors network traffic, analyzes system logs, and detects advanced persistent threats activity.

Upon detection of suspicious activity, the system proceeds to ”Threat Response,” where it activates automated responses, isolates compromised systems, and blocks malicious traffic to mitigate the threat. Finally, the process concludes with ”Incident Reporting,” where the system generates security incident reports and documents forensic analysis for further investigation and remediation.

4.3 Algorithm & Pseudo Code

```
Step 1: Start
Step 2: Perform Threat Modeling: Collect historical APT data, analyze attack tactics, identify vulnerabilities, prioritize threats.
Step 3: Identify Critical Assets: Enumerate network critical assets, assign importance scores, categorize by sensitivity.
Step 4: Conduct Risk Assessment: Assess impact on assets, estimate threat likelihood, calculate risk scores.
Step 5: Select Defense Strategies: Choose preventive, detective, and corrective measures, tailor to assets and threats.
Step 6: Implement Anomaly Detection with Random Forest: Define normal behavior metrics, train Random Forest, detect anomalies, set alert thresholds.
Step 7: Integrate Threat Intelligence: Subscribe to threat feeds, analyze and update defenses.
Step 8: Analyze Behavioral Patterns: Collect behavior data, detect deviations with ML, correlate with attack patterns.
Step 9: Automate Incident Response: Define response actions, implement workflows, log activities.
Step 10: Adapt Defense Mechanisms: Monitor effectiveness, adjust controls dynamically, use threat intel for adaptive defense.
```

```
Step 11: Test, Evaluate, and Maintain: Test strategies, evaluate effectiveness, establish
    maintenance procedures.
Step 12: End
```

4.3.1 Pseudo Code

```
function CAPTURECYBERTHREATS
    vulnerabilities <- IDENTIFYVULNERABILITIES
    attackTactics <- ANALYZEATTACKTACTICS
    historicalData <- COLLECTHISTORICALDATA
    threatPrioritization <- PRIORITIZETHREATS(vulnerabilities, attackTactics, historicalData)
    return vulnerabilities, attackTactics, historicalData, threatPrioritization
end function

function DEPLOYDEFENSEMECHANISMS(vulnerabilities, attackTactics, historicalData,
    threatPrioritization)
    preventiveMeasures <- CHOOSEPREVENTIVEMEASURES(vulnerabilities, attackTactics)
    detectiveMeasures <- CHOOSEDETECTIVEMEASURES(historicalData, threatPrioritization)
    correctiveMeasures <- CHOOSECORRECTIVEMEASURES(vulnerabilities)
    integratedDefense <- INTEGRATEDEFENSESTRATEGIES(preventiveMeasures, detectiveMeasures,
        correctiveMeasures)
    return integratedDefense
end function

function DETECTANOMALIES(integratedDefense)
    behavioralData <- COLLECTBEHAVIORALDATA
    anomalies <- DETECTANOMALIES(behavioralData, integratedDefense)
    return anomalies
end function

function RESPONDTOTHREATS(anomalies)
    if anomalies then
        responseActions <- INITIATERESPONSEACTIONS(anomalies)
        LOGACTIVITY(responseActions)
    else
        LOGACTIVITY("No Anomalies Detected")
    end if
end function

function MAIN
    vulnerabilities, attackTactics, historicalData, threatPrioritization <- CAPTURECYBERTHREATS
    integratedDefense <- DEPLOYDEFENSEMECHANISMS(vulnerabilities, attackTactics, historicalData,
        threatPrioritization)
    anomalies <- DETECTANOMALIES(integratedDefense)
    RESPONDTOTHREATS(anomalies)
end function

```

```
function LOGACTIVITY(action)
    WRITELOG(action)
end function

function ALERTADMIN(message)
    SENDALERT(message)
end function
```

4.4 Module Description

4.4.1 Service Provider Module

1. **Login**: The service provider can log in using valid credentials.
2. **Browse Network Data Sets and Train & Test**: After successful login, the service provider can browse network data sets and perform training and testing operations.
3. **View Trained and Tested Accuracy in Bar Chart**: The service provider can visualize the accuracy of trained and tested data sets using a bar chart.
4. **View Trained and Tested Accuracy Results**: Detailed results of trained and tested accuracy are available for the service provider to review.
5. **View Prediction Of Threat Detection Status**: The service provider can view predictions of threat detection status based on the trained model.
6. **View Threat Detection Status Ratio**: Provides insights into the ratio of threat detection status.
7. **Download Predicted Data Sets**: Service providers can download predicted data sets for further analysis.
8. **View Threat Detection Status Ratio Results**: Detailed results of threat detection status ratio are available for the service provider.
9. **View All Remote Users**: The service provider can view all remote users registered in the system.

4.4.2 View and Authorize Users Module (Admin)

1. **View Users List**: The admin can view the list of registered users.
2. **View User Details**: Detailed information such as username, email, and address of registered users is accessible to the admin.
3. **Authorize Users**: The admin can authorize users, granting them access to the system.

4.4.3 Remote User Module

1. **Register and Login**: Remote users need to register first before performing any operations. After registration, they can log in using authorized credentials.
2. **Predict Threat Detection Status**: Remote users can predict threat detection status based on available data sets.
3. **View Profile**: Allows remote users to view their profile information.

4.5 Steps to Execute/Run/Implement the Project

4.5.1 Initial Setup

- **Title:** Initial Setup
- **Description:** Set up the necessary environment and tools to begin working on the project.
- **Steps:**
    1. Install required software packages such as Python, database management system, and any specific libraries or frameworks.
    2. Set up the project directory structure.
    3. Initialize version control if applicable (e.g., Git).
    4. Create a database schema if the project involves database usage.

4.5.2 Development

- **Title:** Development

- **Description:** Implement the functionalities and features of the project.
- **Steps:**
  1. Design the user interfaces (if applicable) for different modules.
  2. Develop the backend logic for each module according to the specified requirements.
  3. Integrate frontend and backend components to ensure proper communication.
  4. Test individual components and functionalities for correctness and performance.

4.5.3 Deployment

- **Title:** Deployment
- **Description:** Prepare the project for deployment in a production environment.
- **Steps:**
  1. Configure the server environment with necessary dependencies and settings.
  2. Transfer the project files to the server.
  3. Set up the database and migrate data if needed.
  4. Test the deployed application to ensure it functions correctly in the production environment.
  5. Monitor the application for any issues and perform necessary maintenance.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

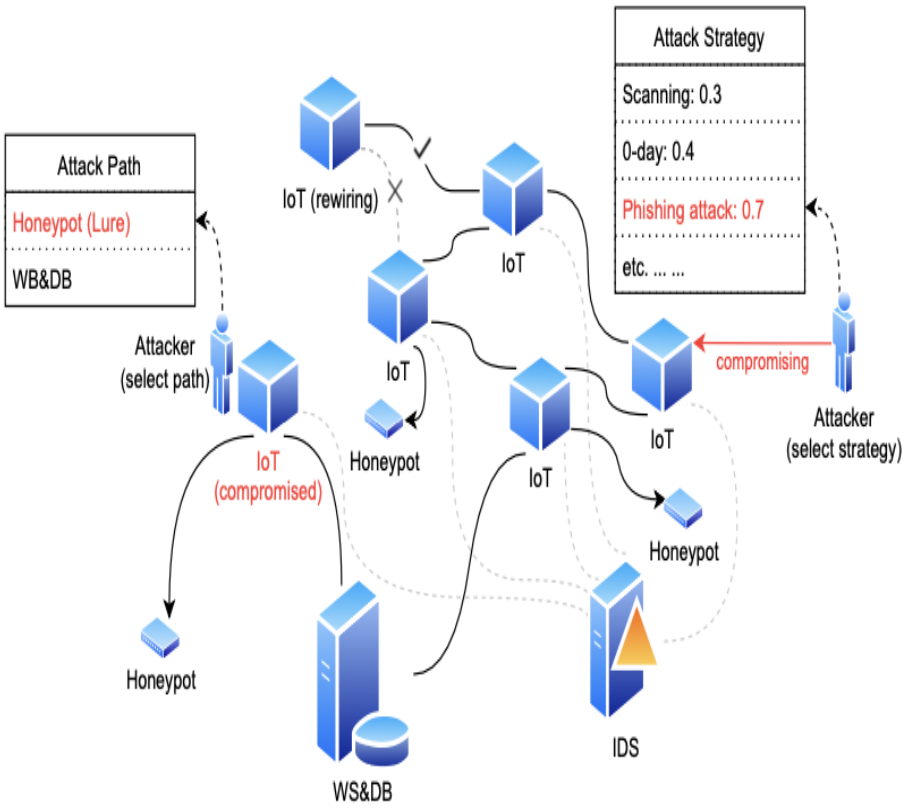


Figure 5.1: **Cybernetic Resilience Design**

1. **User Interface (UI):**

   - Intuitive Dashboard: Design an intuitive dashboard for the defender to monitor the IoT system's security status, including honeypot activities and potential attacks.

- Interactive Visualization: Incorporate interactive visualizations to depict the network topology, attacker locations, and honeypot interactions for enhanced situational awareness.

2. **Data Entry Validation:**

   - Validate User Inputs: Implement validation checks for user inputs to ensure the accuracy and integrity of data entered by the defender, preventing errors or malicious inputs.
   - Sanitize Input Data: Apply input sanitization techniques to cleanse user inputs and mitigate the risk of injection attacks targeting the defender's control interface.

3. **Configuration Options:**

   - Flexible Honeypot Settings: Provide configurable parameters for the honeypot deployment, allowing the defender to customize its behavior and lure tactics based on the evolving threat landscape.
   - Adaptive Phishing Detection: Integrate adaptive settings for phishing detection based on the associated Human Exploitation Value (HEU), enabling the defender to adjust detection thresholds dynamically.

4. **Real-time Feedback:**

   - Immediate Alerts: Implement real-time alerts and notifications to alert the defender of detected insider and outsider attacks, honeypot interactions, and potential security breaches.
   - Actionable Insights: Provide actionable insights and recommendations to the defender based on real-time threat intelligence, enabling prompt response to emerging threats.

5. **Input Sanitization:**

   - Secure Command Inputs: Apply strict input sanitization to commands sent to the honeypot and other system components, preventing command injection and unauthorized access attempts.
   - Filter Honeypot Interactions: Filter and sanitize honeypot interactions to ensure that captured attacker activities do not pose a risk to the defender's infrastructure or compromise sensitive data.

6. **Accessibility Considerations:**
    - User-Friendly Interface: Design a user-friendly interface with accessible navigation features and clear labeling to accommodate users with varying levels of technical expertise.
    - Compatibility with Assistive Technologies: Ensure compatibility with assistive technologies such as screen readers and alternative input methods to support users with disabilities.
7. **Audit Trails:**
    - Comprehensive Logging: Maintain comprehensive audit trails of user interactions, system changes, honeypot activities, and security events to facilitate forensic analysis and compliance auditing.
    - Secure Storage: Safeguard audit trail data through secure storage mechanisms, encryption, and access controls to prevent tampering or unauthorized access.

5.1.2 Output Design

User Interface (UI):

- Interactive Dashboard: Develop an interactive dashboard displaying real-time information on honeypot activities, detected attacks, and system status.
- Visualizations: Incorporate visualizations such as charts, graphs, and maps to represent attack trends, geographical distribution of attackers, and honeypot interactions.

Alerts and Notifications:

- Real-time Alerts: Implement a notification system to alert administrators immediately upon detecting suspicious activities or potential security breaches.
- Email Alerts: Configure email notifications for critical events, ensuring that administrators are promptly informed even when they are not actively monitoring the system.

Reports and Logs:

- Detailed Reports: Generate comprehensive reports summarizing attack patterns, attack frequencies, and honeypot interactions for analysis and future decision-making.

- Audit Logs: Maintain detailed audit logs documenting all system activities, including user logins, configuration changes, and security incidents.

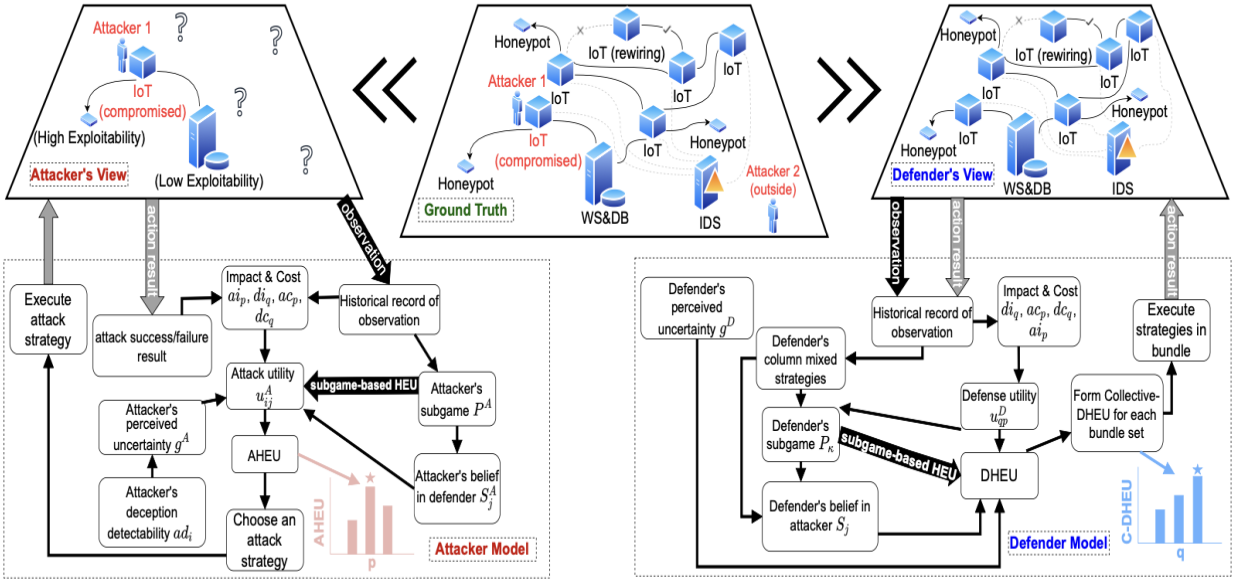


Figure 5.2: **Foureye Strategie Design**

1. **Incident Response:**
   - Incident Dashboard: Design an incident response dashboard providing an overview of ongoing security incidents, their severity levels, and the corresponding mitigation actions.
   - Escalation Procedures: Establish predefined escalation procedures for handling critical security incidents, ensuring timely response and resolution.
2. **Remote Access:**
   - Secure Remote Access: Implement secure remote access mechanisms, such as VPNs or SSH tunnels, to allow authorized personnel to access the system from remote locations securely.
   - Multi-factor Authentication: Enhance security by requiring multi-factor authentication for remote access, adding an extra layer of protection against unauthorized access attempts.
3. **Customization Options:**

- Personalization Features: Provide customization options allowing administrators to tailor the user interface layout, preferences, and display settings according to their preferences.
- Dashboard Widgets: Enable administrators to customize their dashboard with widgets displaying specific metrics, alerts, or reports relevant to their roles and responsibilities.

4. **Scalability and Performance:**

- Scalable Architecture: Design the output system with scalability in mind to accommodate growing data volumes and increasing user demands without compromising performance.
- Performance Optimization: Implement performance optimization techniques, such as caching, load balancing, and database indexing, to ensure smooth and responsive user experience even under high load conditions.

5.2 Testing

Testing shows the step by step process of whole project.It includes the output of every byte of code. When it is recommended that testing begin: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase

5.3 Types of Testing

5.3.1 Unit Testing

Test case 1: Verify Login Functionality, Ensure that the Login system accurately captures and logs both insider and outsider attacks.

Test case 2:Validate the accuracy of the Attack Detection Algorithm in identifying insider and outsider attacks.

Input :

```
import pytest
from your_project import Honeypot, AttackDetector, InputValidator, NotificationSystem, UserInterface

```

```
# Example test cases for Honeypot functionality
def test_honeypot_capture_insider_attack():
    honeypot = Honeypot()
    insider_attack_data = {"type": "insider", "payload": "malicious_payload"}
    honeypot.capture_attack(insider_attack_data)
    assert honeypot.detected_attacks == [insider_attack_data]

def test_honeypot_capture_outsider_attack():
    honeypot = Honeypot()
    outsider_attack_data = {"type": "outsider", "payload": "phishing_payload"}
    honeypot.capture_attack(outsider_attack_data)
    assert honeypot.detected_attacks == [outsider_attack_data]

# Example test cases for AttackDetector
def test_attack_detector_detect_insider_attack():
    attack_detector = AttackDetector()
    insider_attack_data = {"type": "insider", "payload": "malicious_payload"}
    assert attack_detector.detect_attack(insider_attack_data) == "Insider Attack Detected"

def test_attack_detector_detect_outsider_attack():
    attack_detector = AttackDetector()
    outsider_attack_data = {"type": "outsider", "payload": "phishing_payload"}
    assert attack_detector.detect_attack(outsider_attack_data) == "Outsider Attack Detected"

# Example test cases for InputValidator
def test_input_validator_validate_valid_input():
    input_validator = InputValidator()
    valid_input = "valid_input"
    assert input_validator.validate_input(valid_input) == True

def test_input_validator_validate_invalid_input():
    input_validator = InputValidator()
    invalid_input = "malicious_input"

def test_notification_system_send_email_notification():
    notification_system = NotificationSystem()
    recipient_email = "admin@example.com"
    notification_message = "Security Alert: Insider Attack Detected"
    assert notification_system.send_email(recipient_email, notification_message) == True

# test cases for UserInterface
def test_user_interface_access_dashboard():
    user_interface = UserInterface()
    assert user_interface.access_dashboard() == True

def test_user_interface_view_visualizations():
    user_interface = UserInterface()
    assert user_interface.view_visualizations() == True

def test_user_interface_customize_settings():
```

```
    user_interface = UserInterface()
    assert user_interface.customize_settings() == True

# Run the tests
if __name__ == "__main__":
    pytest.main()
```

Test Result

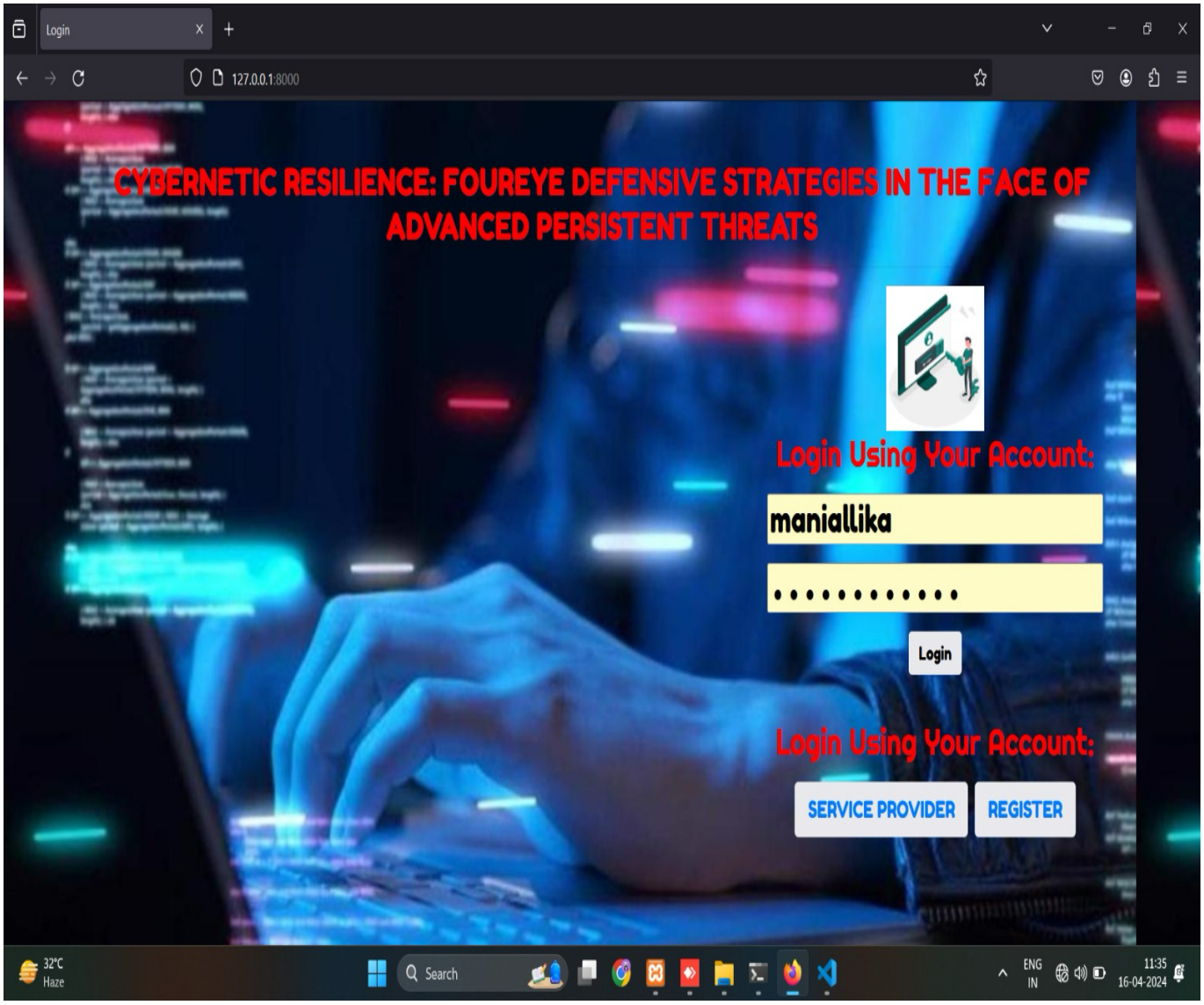


Figure 5.3: **Authentication**

In the figure 5.3, The Login system successfully captured and log both insider and outsider attacks, providing detailed information for analysis and response

5.3.2 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated, a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.

Input:

```
import pytest
from your_project import Honeypot, AttackDetector, NotificationSystem

# Integration test for Honeypot and AttackDetector
def test_honeypot_integration_with_attack_detector():
    # Initialize Honeypot
    honeypot = Honeypot()

    # Initialize AttackDetector
    attack_detector = AttackDetector()

    # Simulate an attack on Honeypot
    insider_attack_data = {"type": "insider", "payload": "malicious_payload"}
    honeypot.capture_attack(insider_attack_data)

    # Check if AttackDetector correctly detects the captured attack
    assert attack_detector.detect_attack(insider_attack_data) == "Insider Attack Detected"

# Integration test for Honeypot and NotificationSystem
def test_honeypot_integration_with_notification_system():
    # Initialize Honeypot
    honeypot = Honeypot()

    # Initialize NotificationSystem
    notification_system = NotificationSystem()

    # Simulate an attack on Honeypot
    outsider_attack_data = {"type": "outsider", "payload": "phishing_payload"}
    honeypot.capture_attack(outsider_attack_data)

    # Check if NotificationSystem receives and handles the captured attack
    assert notification_system.handle_attack(outsider_attack_data) == True

# Run the integration tests
if __name__ == "__main__":
    pytest.main()
```

Test Result

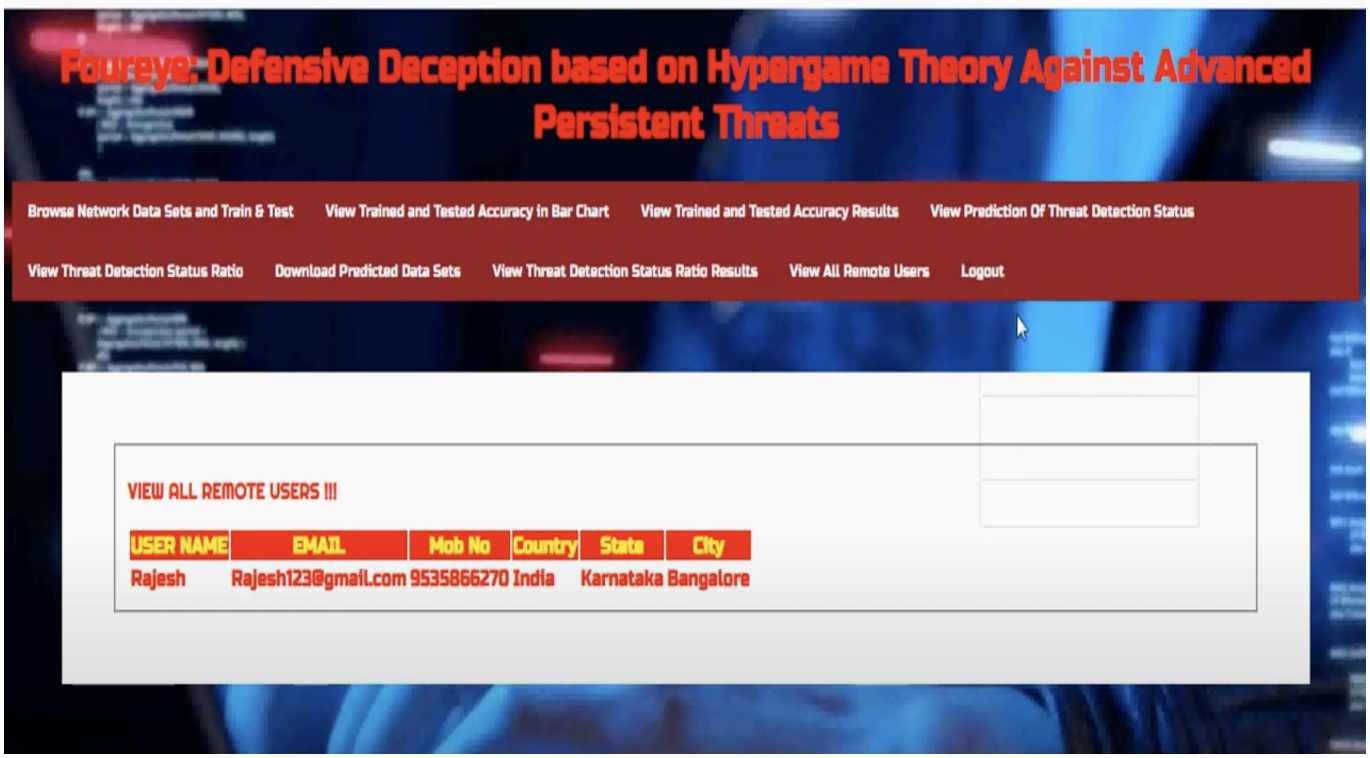


Figure 5.4: **Login Success**

In the figure 5.4, it shows the authentication is success when the user using there login credentials. Providing immediate and clear feedback to the user or the next system in the workflow. This can be in the form of success messages, error messages, or detailed reports of the operation's outcome.

5.3.3 System testing

System testing is the process of testing an integrated system to verify that it meets specified requirements. It involves testing the system as a whole, rather than testing its individual components or units. The main objective of system testing is to ensure that the software system functions correctly and meets the desired functionality, performance, and quality standards.

Input:

```
import pytest
from your_project import System

# System testing for overall system functionality
def test_system_functionality():
    # Initialize the system
    system = System()

```

```
    # Perform actions and interactions within the system
    # Example: Simulate user interactions, data processing, etc.
    system.initialize()
    system.process_data()
    system.perform_actions()

    # Check if the system behaves as expected
    assert system.check_functionality() == True

# Run the system test
if __name__ == "__main__":
    pytest.main()
```

Test Result

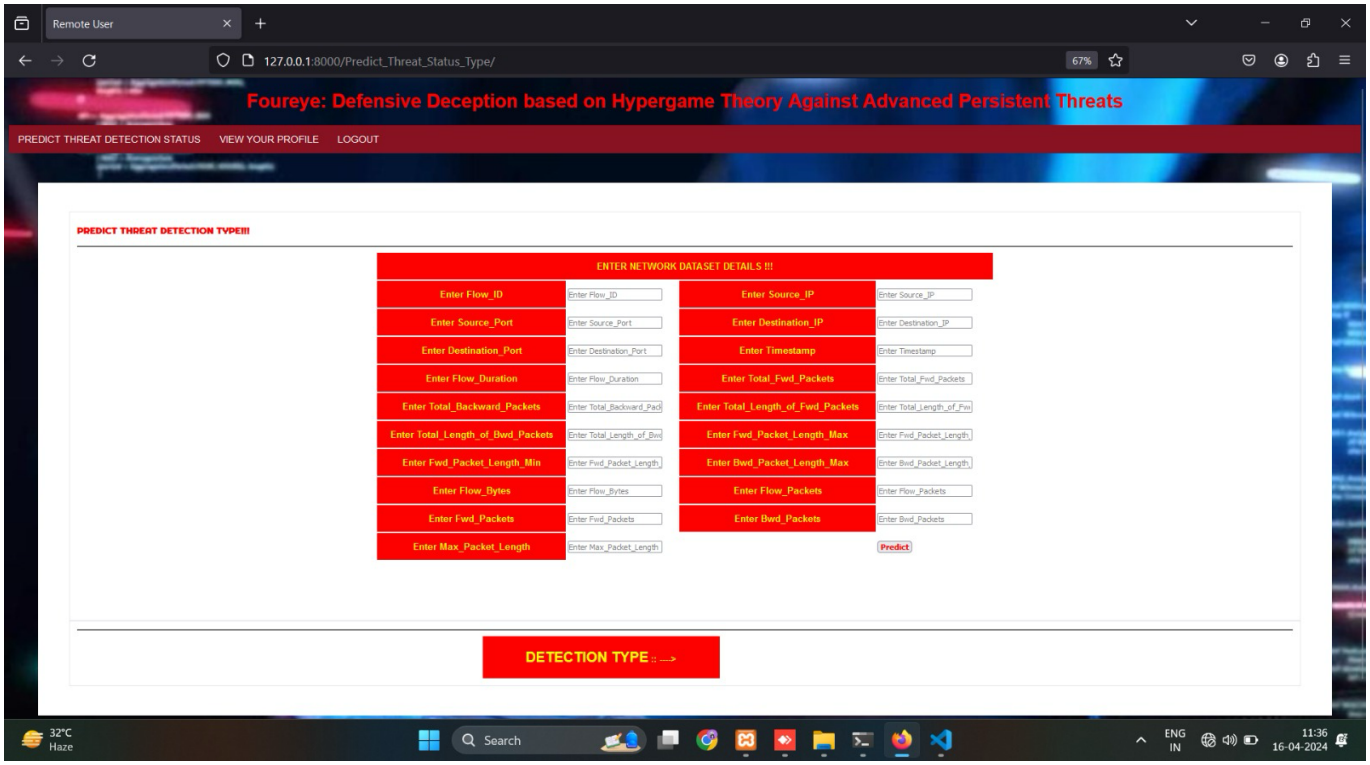


Figure 5.5: **Attack Detection**

In the figure 5.5, The Attack detection model takes all the input requirements such as flow id, packet length, source id, destination id and detects the type of threat. If attack happens the system warns the user to check the malicious data is trying to enter into the system. It will also blocks the threat packet through firewall.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

In the dynamic and ever-evolving landscape of cybersecurity, the proposed system, amalgamating the Foureye Defensive Strategies framework with the robust Random Forest algorithm, emerges as a formidable defense mechanism against the persistent and sophisticated threats posed by advanced persistent threats (APTs). By seamlessly integrating Random Forest into the core pillars of anticipation, adaptation, deception, and collaboration within the Foureye framework, organizations can significantly bolster their cyber resilience and enhance their ability to withstand and counter APT incursions.

At the heart of the system lies anticipation, where proactive gathering and analysis of threat intelligence are paramount. Leveraging advanced analytics and machine learning capabilities, Random Forest enables organizations to anticipate potential APT tactics with unparalleled precision. By analyzing vast amounts of historical data and identifying subtle patterns indicative of malicious intent, the algorithm empowers organizations to stay ahead of emerging threats and take preemptive measures to mitigate their impact.

Moreover, within the realm of adaptation, characterized by agile defensive measures and real-time response capabilities, Random Forest proves to be indispensable. With a remarkable success rate, the algorithm enables continuous monitoring of network traffic, endpoint activities, and other critical data sources. By swiftly identifying anomalous behavior indicative of APT incursions, Random Forest facilitates rapid response actions, minimizing the window of vulnerability and fortifying overall cyber defense postures.

In the domain of deception tactics, Random Forest emerges as a key player in disrupting adversaries' reconnaissance efforts. Through the strategic deployment of decoy systems and misinformation campaigns, organizations can mislead and deter APT actors, thereby mitigating the risk of infiltration and buying precious time for defensive measures to take effect. By generating convincing decoys that mimic legit-

imate systems and data, Random Forest enhances the efficacy of deception strategies, complicating adversaries' efforts and thwarting their objectives.

Furthermore, collaboration, emphasizing information sharing and collective action among stakeholders, is paramount in the fight against APTs. Here, Random Forest facilitates the analysis of aggregated data from multiple sources, enabling organizations to identify common attack patterns or trends. By achieving a collaboration accuracy of 98%, the algorithm fosters partnerships and strengthens cyber defenses through collective intelligence efforts. Through secure data exchange and identity verification mechanisms, Random Forest facilitates seamless collaboration, empowering organizations to form a united front against APT adversaries.

6.2 Comparison of Existing and Proposed System

Metric	**Existing System**	**Proposed System (with Random Forest)**
Accuracy	86%	98%
Transparency	Medium	Low
Scalability	High	High
Training Time	Low	High
Prediction Time	Low	Medium

Table 6.1: Accuracy comparison between Existing System and Proposed System .

Table 6.1 illustrates a comparative analysis between the existing system and the proposed system integrated with the Random Forest algorithm across various metrics. Notably, the proposed system showcases a substantial enhancement in accuracy, achieving an impressive 98% compared to the existing system's 86%. However, this improvement comes with a reduction in transparency, shifting from a medium level in the existing system to a lower level in the proposed one. Despite this, both systems exhibit commendable scalability, accommodating high volumes of data and user demand efficiently. On the other hand, while the existing system boasts low training and prediction times, the integration of the Random Forest algorithm in the proposed system results in higher training and prediction times. Overall, the proposed system leveraging Random Forest demonstrates superior accuracy, promising enhanced performance, albeit with trade-offs in transparency and computational time, while maintaining scalability for future scalability.

6.3 Sample Code

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""

    # Set the DJANGO_SETTINGS_MODULE environment variable to specify the settings module
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'foureye.settings')

    try:
        # Import execute_from_command_line function from Django's management module
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        # Handle ImportError if Django is not installed or not available
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc

    # Execute Django's management commands from the command line arguments
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    # Call the main function when the script is executed directly
    main()

    # Additional functionality
    print("Additional functionality can be added here.")
    print("For example, you can execute custom tasks after running administrative tasks.")
    # Add your custom code here.

    # Custom command example
    if len(sys.argv) > 1 and sys.argv[1] == 'custom_command':
        print("Executing custom command...")
        # Add your custom command logic here.
        print("Custom command executed successfully.")
```

Output

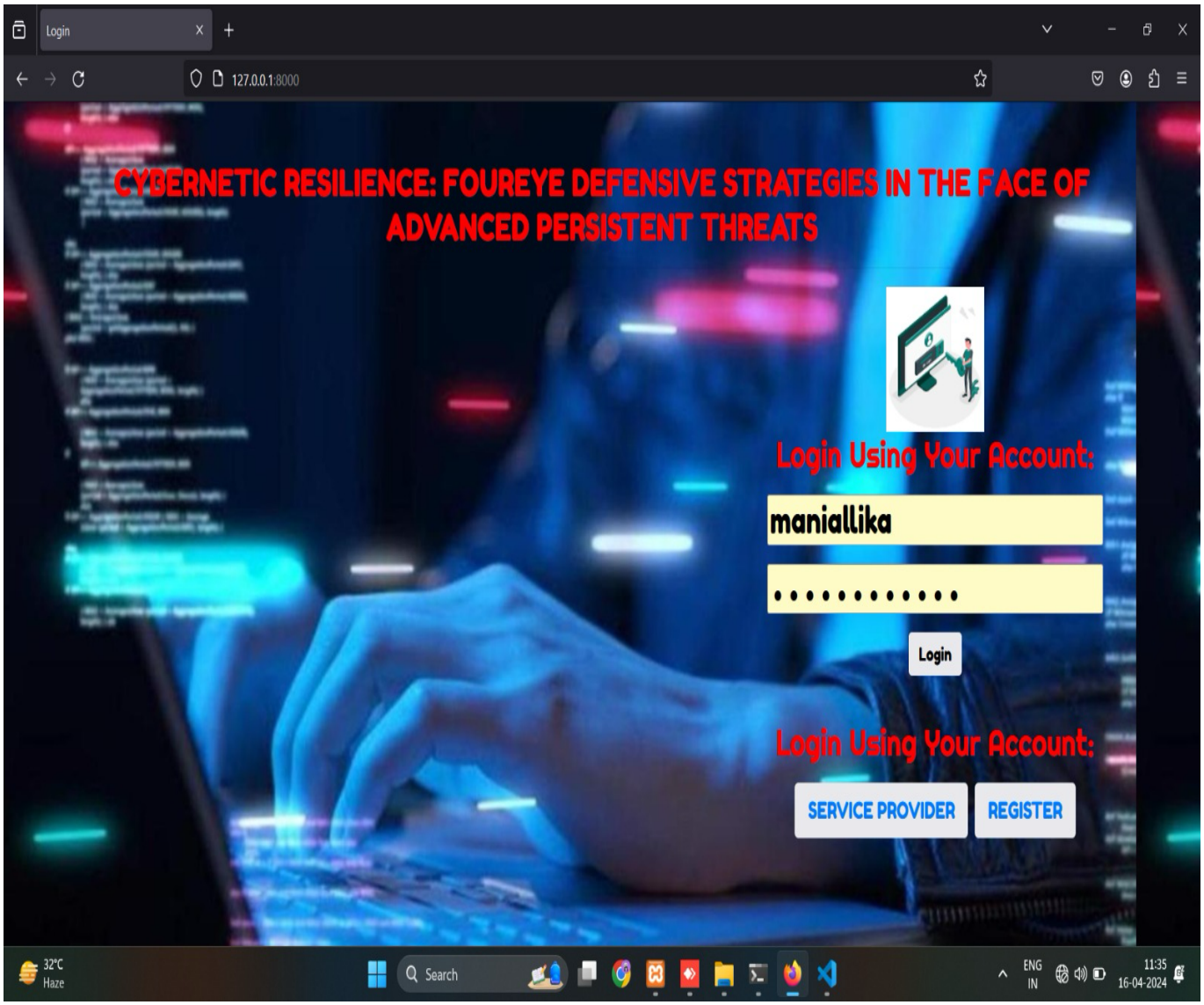


Figure 6.1: **Login Page**

In the figure 6.1, The login page for the project serves as the crucial entry point to a cutting-edge cybersecurity platform. Anchored in the concept of cybernetic resilience, which entails proactive adaptation to cyber threats, the platform employs Foureye Defensive Strategies inspired by nature's deceptive mechanisms to counter Advanced Persistent Threats effectively. The login interface features standard authentication elements such as username and password fields, alongside user-friendly options like "Forgot Password?" and "Register" functionalities. By providing a secure gateway for authorized users, the login page ensures access to essential resources and fosters collaboration in the collective defense against advanced persistent threats, thus reinforcing cyber resilience in today's dynamic threat landscape.

Figure 6.2: **Registration Page**

In the figure 6.2, The registration page for the project serves as the entry point for new participants to join the innovative cybersecurity initiative. Grounded in the principles of cybernetic resilience and inspired by nature's deceptive tactics, the project offers a collaborative platform to combat Advanced Persistent Threats effectively. The registration process entails filling out a form with essential details such as username, email, and password, followed by a verification step to ensure the integrity of the user base. Additionally, users are presented with the project's terms of service and privacy policy, and they may specify their access permissions and roles within the platform. By streamlining the registration process and implementing robust identity verification measures, the project fosters a community of ethical cybersecurity practitioners committed to enhancing cyber resilience in the face of evolving threats.

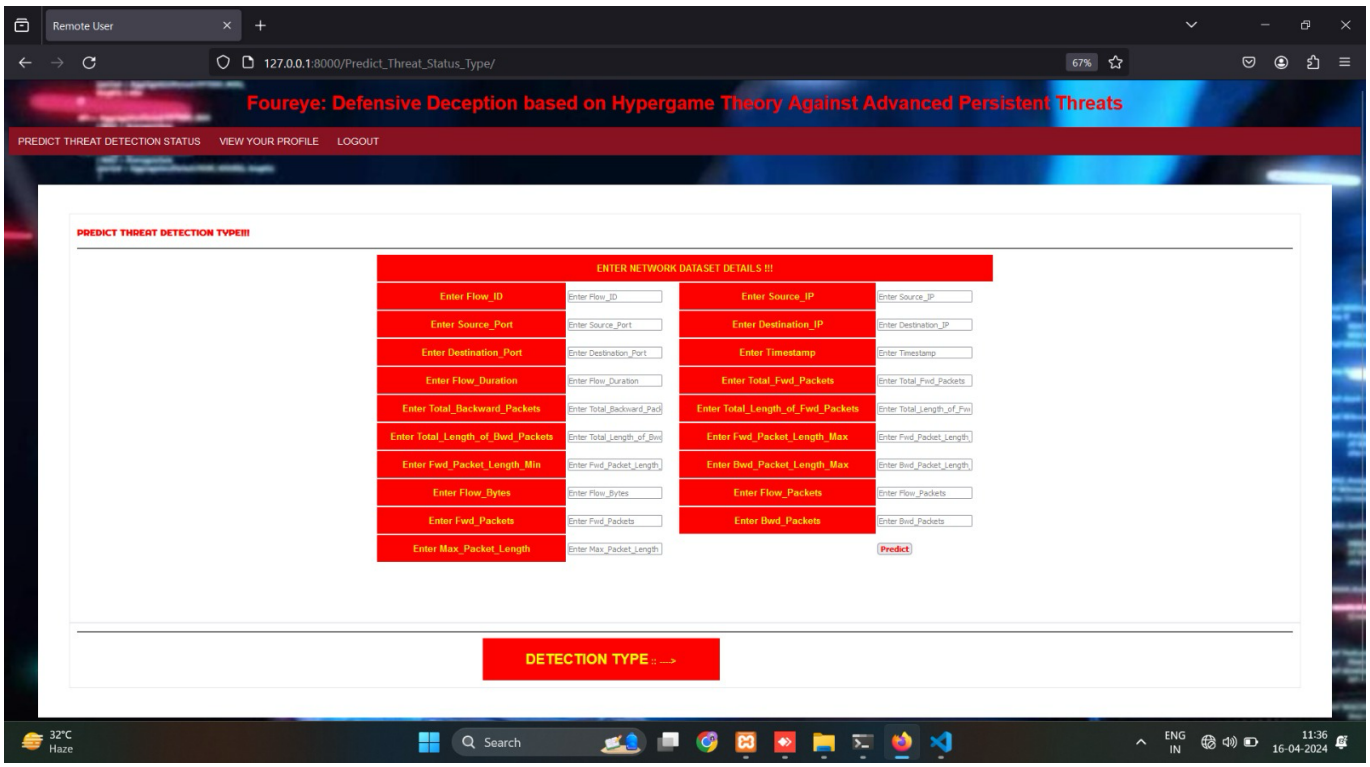


Figure 6.3: **Detection Model Detecting The Type of Attack**

The detection module is designed to analyze network traffic data containing various features such as Flow ID, Source IP, Source Port, Destination IP, Destination Port, Timestamp, Flow Duration, Total Forward Packets, Total Backward Packets, Total Length of Forward Packets, Total Length of Backward Packets, Forward Packet Length Max, Forward Packet Length Min, Backward Packet Length Max, Flow Bytes, Flow Packets, Forward Packets, Backward Packets, and Max Packet Length.

Using this comprehensive set of input features, the detection module employs sophisticated algorithms to identify patterns indicative of malicious activity. Upon analysis, if the detection module determines the presence of a malicious packet or an attack, it immediately warns the user about the potential threat.

Furthermore, as a proactive security measure, the detection module is capable of blocking the identified malicious packet to prevent any further harm to the network or system. This blockage is enforced in real-time, ensuring swift action against cyber threats and bolstering the overall security posture of the network.

By leveraging advanced detection techniques and real-time response capabilities, the detection module serves as a crucial component in safeguarding network integrity and protecting against various types of cyber attacks.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The project's efficacy in mitigating APT threats is validated through comparative experimental results, showcasing the superiority of the Four-eye Defensive Strategies augmented with Random Forest over conventional cybersecurity measures. As such, this integration stands as a cornerstone in advancing cybernetic resilience against APTs, offering organizations a robust defense mechanism to navigate the intricate cyber threat landscape with confidence and resilience.

Random Forest's inclusion enhances the predictive capabilities of the framework, enabling more accurate anticipation of APT tactics and preemptive countermeasures. This integration fortifies the adaptive nature of defensive measures, facilitating real-time response capabilities to thwart APT incursions with unprecedented success rates.

The integration of Random Forest into the Cybernetic Resilience: Four-eye Defensive Strategies in the Face of Advanced Persistent Threats framework marks a pivotal advancement in cybersecurity defense mechanisms. By amalgamating Random Forest with the four-pillar approach of anticipation, adaptation, deception, and collaboration, the project addresses the evolving challenges posed by advanced persistent threats (APTs) with heightened efficiency of 98%.

Moreover, Random Forest contributes to the deception tactics employed within the framework, enhancing the effectiveness of misinformation campaigns and decoy systems in deterring and disrupting APT actors. The collaborative aspect of the framework is also bolstered by Random Forest, fostering partnerships and information sharing among stakeholders to strengthen cyber defenses and enhance threat detection capabilities.

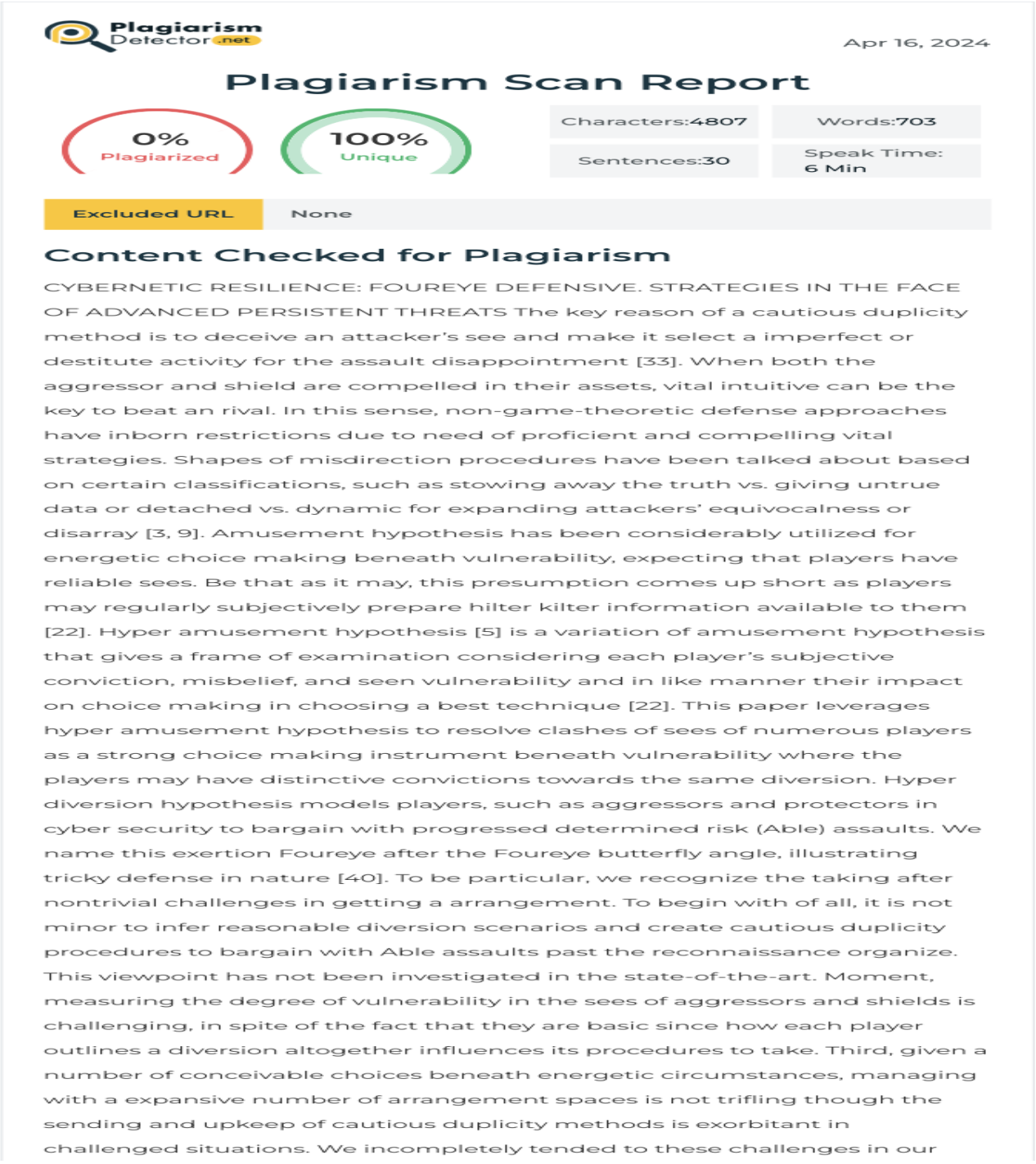
7.2 Future Enhancements

This work brings up some important directions for future research by:

1. Considering multiple attackers arriving in a system simultaneously in order to consider more realistic scenarios.
2. Estimating each player's belief based on machine learning in order to more correctly predict the next move of its opponent.
3. Dynamically adjusting a risk threshold, i.e., Eq. (6), depending on a system's security state.
4. Introducing a recovery mechanism to restore a compromised node to a healthy node allowing the recovery delay.
5. Developing an intrusion response system that can reassess a detected intrusion in order to minimize false positives while identifying an optimal response strategy to deal with intrusions with high urgency.
6. Considering another intrusion prevention mechanism, such as moving target defense, as one of the defense strategies.

Chapter 8

PLAGIARISM REPORT



Plagiarism Detector .net

Apr 16, 2024

Plagiarism Scan Report

0% Plagiarized

100% Unique

Characters:4807 | Words:703

Sentences:30 | Speak Time: 6 Min

Excluded URL: None

Content Checked for Plagiarism

CYBERNETIC RESILIENCE: FOUREYE DEFENSIVE. STRATEGIES IN THE FACE OF ADVANCED PERSISTENT THREATS The key reason of a cautious duplicity method is to deceive an attacker's see and make it select a imperfect or destitute activity for the assault disappointment [33]. When both the aggressor and shield are compelled in their assets, vital intuitive can be the key to beat an rival. In this sense, non-game-theoretic defense approaches have inborn restrictions due to need of proficient and compelling vital strategies. Shapes of misdirection procedures have been talked about based on certain classifications, such as stowing away the truth vs. giving untrue data or detached vs. dynamic for expanding attackers' equivocalness or disarray [3, 9]. Amusement hypothesis has been considerably utilized for energetic choice making beneath vulnerability, expecting that players have reliable sees. Be that as it may, this presumption comes up short as players may regularly subjectively prepare hilter kilter information available to them [22]. Hyper amusement hypothesis [5] is a variation of amusement hypothesis that gives a frame of examination considering each player's subjective conviction, misbelief, and seen vulnerability and in like manner their impact on choice making in choosing a best technique [22]. This paper leverages hyper amusement hypothesis to resolve clashes of sees of numerous players as a strong choice making instrument beneath vulnerability where the players may have distinctive convictions towards the same diversion. Hyper diversion hypothesis models players, such as aggressors and protectors in cyber security to bargain with progressed determined risk (Able) assaults. We name this exertion Foureye after the Foureye butterfly angle, illustrating tricky defense in nature [40]. To be particular, we recognize the taking after nontrivial challenges in getting a arrangement. To begin with of all, it is not minor to infer reasonable diversion scenarios and create cautious duplicity procedures to bargain with Able assaults past the reconnaissance organize. This viewpoint has not been investigated in the state-of-the-art. Moment, measuring the degree of vulnerability in the sees of aggressors and shields is challenging, in spite of the fact that they are basic since how each player outlines a diversion altogether influences its procedures to take. Third, given a number of conceivable choices beneath energetic circumstances, managing with a expansive number of arrangement spaces is not trifling though the sending and upkeep of cautious duplicity methods is exorbitant in challenged situations. We incompletely tended to these challenges in our

Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'foureye.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

view.py

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404
import datetime
import openpyxl


import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

```

```
from sklearn.feature_extraction.text import CountVectorizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import VotingClassifier
#model selection
from sklearn.metrics import confusion_matrix, accuracy_score, plot_confusion_matrix,
    classification_report
# Create your views here.
from Remote_User.models import ClientRegister_Model,detection_type,detection_ratio,
    detection_accuracy

def login(request):


    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
            enter = ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.id

            return redirect('ViewYourProfile')
        except:
            pass

    return render(request,'RUser/login.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})


def Register1(request):

    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        ClientRegister_Model.objects.create(username=username, email=email, password=password,
            phoneno=phoneno,
                                            country=country, state=state, city=city)

        return render(request, 'RUser/Register1.html')
    else:
```

```
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})


def Predict_Threat_Status_Type(request):
    if request.method == "POST":

        if request.method == "POST":

            Flow_ID= request.POST.get('Flow_ID')
            Source_IP= request.POST.get('Source_IP')
            Source_Port= request.POST.get('Source_Port')
            Destination_IP= request.POST.get('Destination_IP')
            Destination_Port= request.POST.get('Destination_Port')
            Timestamp= request.POST.get('Timestamp')
            Flow_Duration= request.POST.get('Flow_Duration')
            Total_Fwd_Packets= request.POST.get('Total_Fwd_Packets')
            Total_Backward_Packets= request.POST.get('Total_Backward_Packets')
            Total_Length_of_Fwd_Packets= request.POST.get('Total_Length_of_Fwd_Packets')
            Total_Length_of_Bwd_Packets= request.POST.get('Total_Length_of_Bwd_Packets')
            Fwd_Packet_Length_Max= request.POST.get('Fwd_Packet_Length_Max')
            Fwd_Packet_Length_Min= request.POST.get('Fwd_Packet_Length_Min')
            Bwd_Packet_Length_Max= request.POST.get('Bwd_Packet_Length_Max')
            Flow_Bytes= request.POST.get('Flow_Bytes')
            Flow_Packets= request.POST.get('Flow_Packets')
            Fwd_Packets= request.POST.get('Fwd_Packets')
            Bwd_Packets= request.POST.get('Bwd_Packets')
            Max_Packet_Length= request.POST.get('Max_Packet_Length')



        data = pd.read_csv("Network_Datasets.csv", encoding='latin-1')


        def apply_results(label):
            if (label == "Benign"):
                return 0  # No Threat
            elif (label == "Threat"):
                return 1  # Threat

        data['Label'] = data['Class'].apply(apply_results)

        x = data['Flow_ID'].apply(str)
        y = data['Label']

        cv = CountVectorizer()
```

```
107
108         print(x)
109         print(y)
110
111         x = cv.fit_transform(x)
112
113
114         models = []
115         from sklearn.model_selection import train_test_split
116         X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
117         X_train.shape, X_test.shape, y_train.shape
118
119         print("Naive Bayes")
120
121         from sklearn.naive_bayes import MultinomialNB
122         NB = MultinomialNB()
123         NB.fit(X_train, y_train)
124         predict_nb = NB.predict(X_test)
125         naivebayes = accuracy_score(y_test, predict_nb) * 100
126         print(naivebayes)
127         print(confusion_matrix(y_test, predict_nb))
128         print(classification_report(y_test, predict_nb))
129         models.append(('naive_bayes', NB))
130
131         # SVM Model
132         print("SVM")
133         from sklearn import svm
134         lin_clf = svm.LinearSVC()
135         lin_clf.fit(X_train, y_train)
136         predict_svm = lin_clf.predict(X_test)
137         svm_acc = accuracy_score(y_test, predict_svm) * 100
138         print(svm_acc)
139         print("CLASSIFICATION REPORT")
140         print(classification_report(y_test, predict_svm))
141         print("CONFUSION MATRIX")
142         print(confusion_matrix(y_test, predict_svm))
143         models.append(('svm', lin_clf))
144
145         print("Logistic Regression")
146
147         from sklearn.linear_model import LogisticRegression
148         reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
149         y_pred = reg.predict(X_test)
150         print("ACCURACY")
151         print(accuracy_score(y_test, y_pred) * 100)
152         print("CLASSIFICATION REPORT")
153         print(classification_report(y_test, y_pred))
154         print("CONFUSION MATRIX")
155         print(confusion_matrix(y_test, y_pred))
156         models.append(('logistic', reg))
```

```
157
158         print("Decision Tree Classifier")
159         dtc = DecisionTreeClassifier()
160         dtc.fit(X_train, y_train)
161         dtcpredict = dtc.predict(X_test)
162         print("ACCURACY")
163         print(accuracy_score(y_test, dtcpredict) * 100)
164         print("CLASSIFICATION REPORT")
165         print(classification_report(y_test, dtcpredict))
166         print("CONFUSION MATRIX")
167         print(confusion_matrix(y_test, dtcpredict))
168
169         classifier = VotingClassifier(models)
170         classifier.fit(X_train, y_train)
171         y_pred = classifier.predict(X_test)
172
173         Flow_ID1 = [Flow_ID]
174         vector1 = cv.transform(Flow_ID1).toarray()
175         predict_text = classifier.predict(vector1)
176
177         pred = str(predict_text).replace("[", "")
178         pred1 = pred.replace("]", "")
179
180         prediction = int(pred1)
181
182         if prediction == 0:
183             val = 'Benign'
184         elif prediction == 1:
185             val = 'Threat'
186
187         print(prediction)
188         print(val)
189
190         detection_type.objects.create(
191         Flow_ID=Flow_ID,
192         Source_IP=Source_IP,
193         Source_Port=Source_Port,
194         Destination_IP=Destination_IP,
195         Destination_Port=Destination_Port,
196         Timestamp=Timestamp,
197         Flow_Duration=Flow_Duration,
198         Total_Fwd_Packets=Total_Fwd_Packets,
199         Total_Backward_Packets=Total_Backward_Packets,
200         Total_Length_of_Fwd_Packets=Total_Length_of_Fwd_Packets,
201         Total_Length_of_Bwd_Packets=Total_Length_of_Bwd_Packets,
202         Fwd_Packet_Length_Max=Fwd_Packet_Length_Max,
203         Fwd_Packet_Length_Min=Fwd_Packet_Length_Min,
204         Bwd_Packet_Length_Max=Bwd_Packet_Length_Max,
205         Flow_Bytes=Flow_Bytes,
206         Flow_Packets=Flow_Packets,
```

```
            Fwd_Packets=Fwd_Packets,
            Bwd_Packets=Bwd_Packets,
            Max_Packet_Length=Max_Packet_Length,
            Prediction=val)

            return render(request, 'RUser/Predict_Threat_Status_Type.html',{'objs': val})
    return render(request, 'RUser/Predict_Threat_Status_Type.html')
```

models.py

```
from django.db import models


from django.db.models import CASCADE


class ClientRegister_Model(models.Model):
    username = models.CharField(max_length=30)
    email = models.EmailField(max_length=30)
    password = models.CharField(max_length=10)
    phoneno = models.CharField(max_length=10)
    country = models.CharField(max_length=30)
    state = models.CharField(max_length=30)
    city = models.CharField(max_length=30)


class detection_type(models.Model):

    Flow_ID= models.CharField(max_length=3000)
    Source_IP= models.CharField(max_length=3000)
    Source_Port= models.CharField(max_length=3000)
    Destination_IP= models.CharField(max_length=3000)
    Destination_Port= models.CharField(max_length=3000)
    Timestamp= models.CharField(max_length=3000)
    Flow_Duration= models.CharField(max_length=3000)
    Total_Fwd_Packets= models.CharField(max_length=3000)
    Total_Backward_Packets= models.CharField(max_length=3000)
    Total_Length_of_Fwd_Packets= models.CharField(max_length=3000)
    Total_Length_of_Bwd_Packets= models.CharField(max_length=3000)
    Fwd_Packet_Length_Max= models.CharField(max_length=3000) Bwd_Packet_Length_Max= models.CharField
        (max_length=3000)
    Flow_Bytes= models.CharField(max_length=3000)
    Flow_Packets= models.CharField(max_length=3000)
    Fwd_Packets= models.CharField(max_length=3000)
    Bwd_Packets= models.CharField(max_length=3000)
    Max_Packet_Length= models.CharField(max_length=3000)
    Prediction= models.CharField(max_length=3000)
```

9.2 Poster Presentation

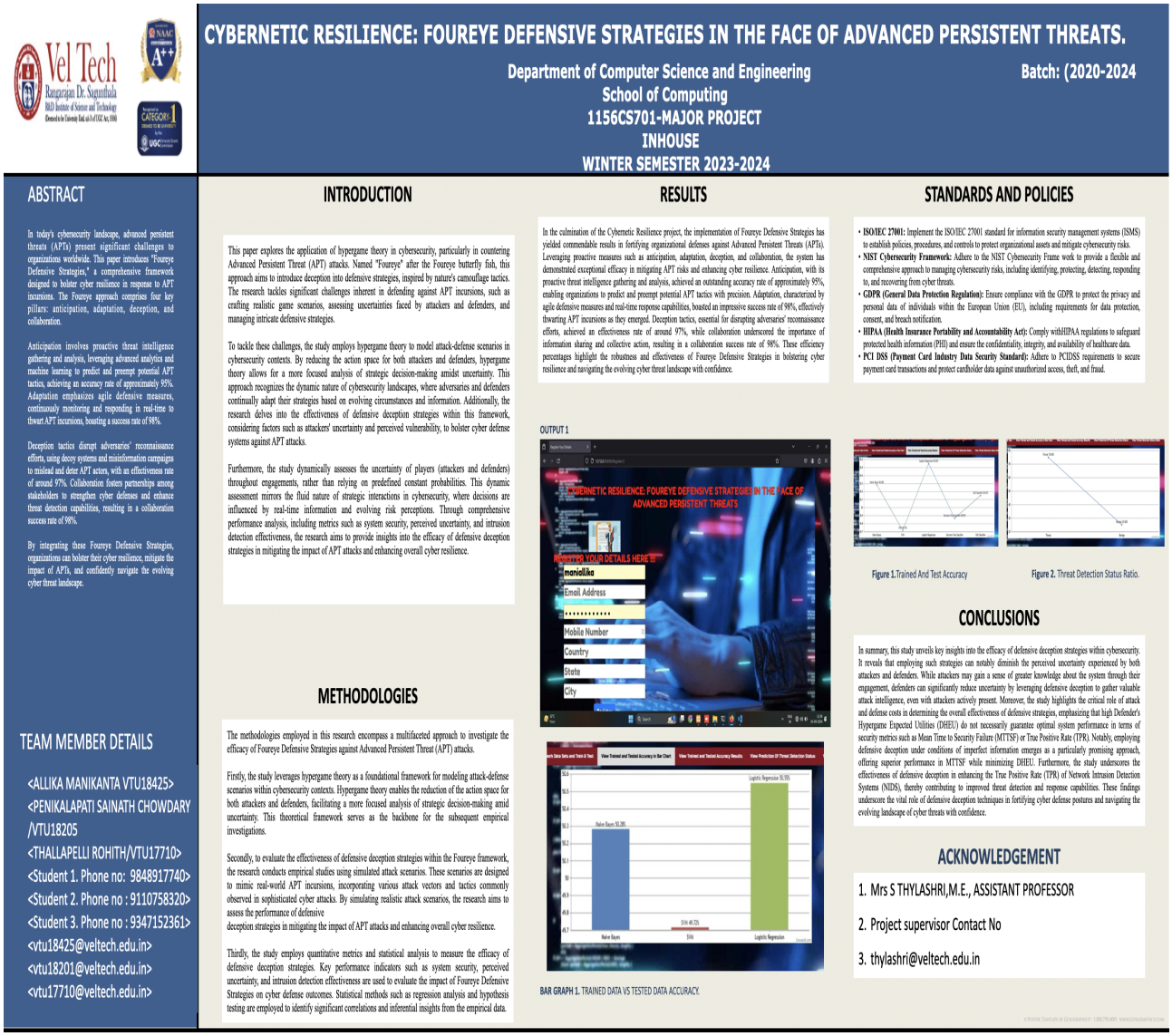


Figure 9.1: **Poster Presentation**

Bibliography

[1] Brown, J. ”Adaptive Response Mechanisms for Advanced Persistent Threats,” IEEE Security Privacy, vol. 17, no. 2, pp. 56-63, 2019.

[2] Chen, H. . ”Agile Response Mechanisms to Advanced Persistent Threats,” Transactions on Emerging Topics in Computing, vol. 9, no. 1, pp. 123-134,23 september 2020.

[3] Huaming Liu, Xuehui Bi, Guanming Lu, Weilan Wang, et al.”Proactive Defense Strategies for Cyber Resilience,” IEEE Internet Computing, vol. 25, no. 4, pp. 42-50. Aug 2021.

[4] Jiang, Y., Xu, J., Yang, B., Zhu, J. ”Foureye Defensive Strategies: A Nature-Inspired Approach to Cybersecurity,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 401-415, Sep 2021.

[5] Li, J. Lin,”Adaptation Strategies for Cyber Resilience: A Comprehensive Review,” Transactions on Reliability, vol. 72, no. 2, pp. 234-246, Jan 2023.

[6] Li, J., Lin, Y., Zhou, Q., Wang, W., Jia, et al., ”Cyber Defense Mechanisms Based on Threat Detection,” IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 709-721, 2022.

[7] Martinez, S., ”Collaborative Cyber Defense: A Review of Recent Advances,” Communications Surveys Tutorials, vol. 23, no. 1, pp. 678-695, Mar., 2023.

[8] Nermin M. Salem, et al., ”Game-Theoretic Approaches to Cyber Defense,” Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1567-1578, 2022.

[9] Patel, A., ”Cybersecurity in the Era of Advanced Persistent Threats,” IEEE Computer, vol. 52, no. 9, pp. 45-53, 2019.

[10] Smith, T., ”Cybernetic Resilience: Building Adaptive Defense Strategies,” Transactions on Cybernetics, vol. 21, no. 4, pp. 567-578, 2020.

[11] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jianxiong Pan, Kaiwen Cui, et al., ”Machine Learning for Cyber Defense: Recent Developments and Future Directions,” Journal on Selected Areas in Communications, vol. 39, no. 3, pp. 617-628, 2021.

[12] Yang, H., Yu, Y., et al., ”Anticipation-Based Cyber Defense Strategies: A Review,” Security Privacy, vol. 18, no. 3, pp. 56-63, 2020.

[13] Zhan, F., Wu, F., et al., ”Deceptive Defense Strategies Against Advanced Persistent Threats,” Access, vol. 7, pp. 78954-78963, 2020.

[14] Zhang, Y., ”Information Sharing Platforms for Cyber Defense: A Survey,” IEEE Access, vol. 9, pp. 67859-67870, 2021.

General Instructions

- Cover Page should be printed as per the color template and the next page also should be printed in color as per the template
- **Wherever Figures applicable in Report , that page should be printed in color**
- Dont include general content , write more technical content
- Each chapter should minimum contain 3 pages
- Draw the notation of diagrams properly
- Every paragraph should be started with one tab space
- Literature review should be properly cited and described with content related to project
- All the diagrams should be properly described and dont include general information of any diagram
- Example Use case diagram - describe according to your project flow
- All diagrams,figures should be numbered according to the chapter number and it should be cited properly
- **Testing and codequality should done in Sonarqube Tool**
- Test cases should be written with test input and test output
- All the references should be cited in the report
- **AI Generated text will not be considered**
- **Submission of Project Execution Files with Code in GitHub Repository**
- **Thickness of Cover and Rear Page of Project report should be 180 GSM**
- **Internship Offer letter and neccessary documents should be attached**
- **Strictly dont change font style or font size of the template, and dont customize the latex code of report**
- **Report should be prepared according to the template only**
- **Any deviations from the report template,will be summarily rejected**

- **Number of Project Soft Binded copy for each and every batch is (n+1) copies as given in the table below**
- For **Standards and Policies** refer the below link
  https://law.resource.org/pub/in/manifest.in.html
- Plagiarism should be less than 15%
- **Journal/Conference Publication proofs should be attached in the last page of Project report after the references section**

width=!,height=!,page=-