2023-08-01

Continuous Risk Assessment For Large-Scale Cyber Systems

Adeel A. Malik
University of Texas at El Paso

CONTINUOUS RISK ASSESSMENT FOR LARGE-SCALE CYBER SYSTEMS

ADEEL ASHRAF MALIK

Doctoral Program in Computer Science

APPROVED:

Deepak K. Tosh, Ph.D., Chair

Christopher D. Kiekintveld, Ph.D.

Mahmud Shahriar Hossain, Ph.D.

Jaime C. Acosta, Ph.D.

Eric D. Smith, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

CONTINUOUS RISK ASSESSMENT FOR LARGE-SCALE CYBER SYSTEMS

by

ADEEL ASHRAF MALIK

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

August 2023

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Deepak Tosh, my advisor, for his consistent support, encouragement, and patience throughout my academic journey. His unwavering dedication to my success has been instrumental in shaping me into the scholar I am today. With his guidance and counsel, I have learned to persevere in the face of adversity and to believe in myself even when doubt crept in. I am immensely grateful for Dr. Tosh's time and efforts in honing my talents and helping me achieve my academic goals.

In addition to my advisor, I would like to extend my gratitude to the other members of my committee, Dr. Christopher Kiekintveld, Dr. Mahmud Shahriar Hossain, Dr. Jamie Acosta, and Dr. Eric Smith. Their continuous support, suggestions, and guidance have been invaluable in my academic growth, pushing me beyond my limits and enabling me to expand my horizons.

Furthermore, I would like to express my appreciation to the faculty and staff of the Computer Science Department, as well as my fellow students, for their hard work and dedication in helping me complete my degree and prepare for my future career as a computer scientist.

Lastly, I would like to express my appreciation to my parents, Muhammad Ashraf Malik and Ghazala Yasmin, for their unconditional love, unwavering support, and constant prayers. Without their prayers and encouragement, none of this would have been possible. I also want to thank my wife, Nazia Sharmin, for her unwavering belief in me, constant support, and motivation throughout this journey. I am grateful to my daughter, Hareem Malik, for her unconditional love, Linda and Nodjimbadem for always making me feel welcome. I also extend my thanks to Jennifer for her 24/7 reviews and suggestions, my lab members Abel, Evan, William, and Asif for their unwavering support, and Mahmudulla Hassan for always being there for me from day one in El Paso.

Abstract

Cyberspace, with its multiple forms of device integration, is rapidly evolving and introducing loopholes within the cyber infrastructure, which creates opportunities for attackers. Despite the presence of network security devices such as firewalls, anti-virus, intrusion detection, and prevention systems, network intrusions still occur due to vulnerabilities within organizational assets or socially engineered cyber attacks. The lack of information about threats, vulnerabilities, and threat actors often leaves cyber defenders on a wild goose chase, making it critical to evaluate network security to mitigate adversarial threats periodically.

Various risk assessment frameworks, third-party tools, and online databases containing comprehensive threat information have been proposed in the past. However, obtaining infrastructure-specific information using these resources is challenging and laborious for a cyber defender. This dissertation focuses on equipping cyber defenders with the necessary, relevant, and infrastructure-specific information to better evaluate their cyber-risk posture and offer potential mitigation approaches to secure organizational security. We present Cyber-threats and Vulnerability Information Analyzer (CyVIA), a dynamic and scalable framework for conducting continuous risk assessments of any given cyber infrastructure.

CyVIA leverages concrete ways of analyzing anomalies and is designed to: 1) model the organizational security posture to evaluate security controls in place, 2) effectively combine vulnerability information from multi-formatted open-sourced vulnerability databases (VDBs) into a unified knowledge-base that is used to derive specific information, 3) map adversarial and control policies, services dependencies, applications, and vulnerabilities from the network nodes, 4) classify network nodes based on severities, and 5) provide consequences, mitigation, and relationship information of the found vulnerabilities.

CyVIA has been empirically evaluated on a simulated network environment containing various flavors of Microsoft Windows and Linux operating systems and compare the results with other state-of-the-art tools. The evaluation demonstrates the effectiveness of CyVIA

in providing relevant and infrastructure-specific information for evaluating and securing organizational security. CyVIA exhibits promising potential to assist cyber defenders in proactively identifying and mitigating vulnerabilities, thereby improving network security posture and reducing the risk of adversarial threats. This research's findings contribute to the cybersecurity field by addressing the challenges of obtaining infrastructure-specific information for effective risk assessment and mitigation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Exponential rise to cyber adoption has led significant expansion to threat landscape, thus impacting everything, including the public health sector, economics, electric grids, the Internet of Things (IoT), and many other sectors, including national security. The main reason organizations struggle to protect themselves is due to a lack of understanding about the importance and role of cybersecurity [2]. As reported by PwC [3], many organizations are actively seeking solutions to address cybersecurity issues.

Furthermore, with the transformation of cyber infrastructures into increasingly complex and competitive Cyber-Physical Systems (CPS), there are numerous uncertainties and challenges in modeling and analyzing cybersecurity, which prevent achieving 100% security [4, 5]. The number of reported vulnerabilities has also significantly increased in the past five years, as reported by the National Vulnerability Database (NVD) [Figure. 1.1], which further highlights the vulnerabilities within organizational infrastructures. The prevalence of IoT applications within CPS environments introduces additional uncertainties, as unanticipated or unmanaged risks create a highly competitive landscape for cyber defenders. Compromised security in such cases not only results in financial losses but also poses risks to human safety, particularly in the medical and healthcare sectors.

The structure of CPS (Cyber-Physical Systems) environments consists of two distinct parts: the cyber part and the physical part. However, most of the existing research in the field of CPS security has primarily focused on the cyber assessment aspects, neglecting the physical aspects. A typical CPS is comprised of various hardware and software components, including commercial and proprietary products, as well as embedded systems. This diversity in components introduces not only security concerns but also privacy concerns.

Despite the heterogeneity of CPS components, there is a lack of comprehensive review of CPS security literature, particularly in reference to [5]. Hence, there is a need for a systematic framework that can capture the essential aspects of any CPS. In the following sections of this chapter, we will discuss the critical aspects of CPS risk assessment, the motivation behind our research, the research objectives, the significance of our research, and provide an overview of this dissertation document.

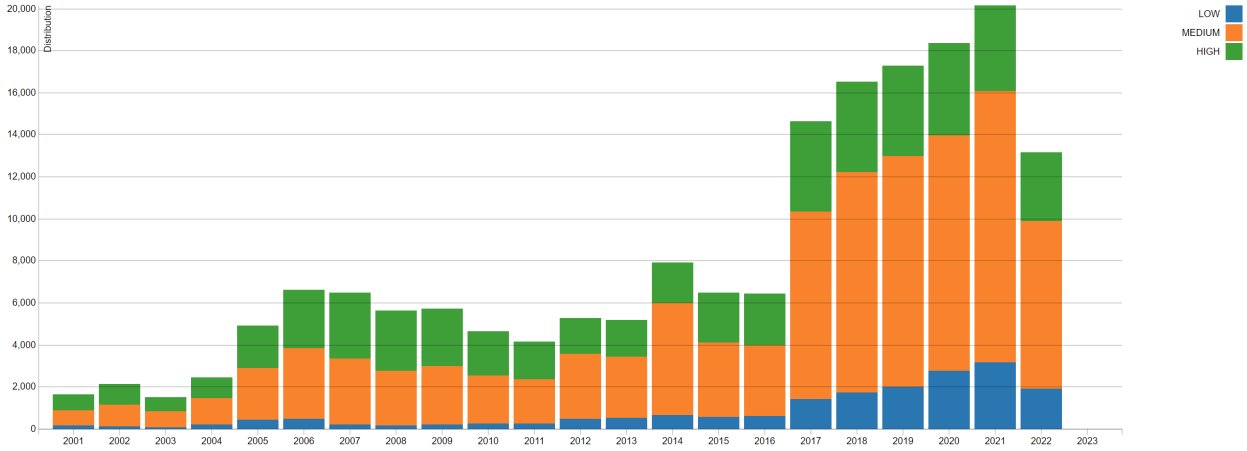

Figure 1.1: Vulnerability Severity Distribution Over Time [1]

1.1 Risk Assessment for Cyber-Physical Systems

The present-day threats posed by cyber attacks are of a grave nature and have a significant impact on various aspects of our surroundings. The continuously increasing number of evolving threats and reported vulnerabilities in recent years has become a severe concern, as malicious entities exploit every possible loophole to infiltrate organizational security. Keeping up with the rapidly changing threat landscape has become an onerous task for cyber defenders in such a scenario. Typically, organizational security is assessed through a combination of manual and semi-automated approaches or tools, with analyses generated periodically based on need. However, threat actors are constantly active and engaged

in adversarial activities. Therefore, it is crucial to monitor and track these activities in real-time to effectively prevent successful cyber attacks.

1.1.1 Real-Time Risk Assessment

Real-time risk assessment for CPS is a vital process to continuously identify how vulnerable the infrastructure is at any given point in time. Adversarial entities utilize vulnerabilities within the systems to gain unauthorized access and obtain sensitive information. The number of vulnerabilities reported to NIST's NVD has drastically increased starting from the year 2017 [6]. Various organizations like Cybersecurity Coalition [7], NCSC [8], GFCE [9], ENISA [10], Software Engineering Institute at Carnegie Mellon University [11] are working on emphasizing the importance Vulnerability management. Cyber attacks are either socially motivated for pleasure or politically motivated with specific goals. According to cybersecurity reports by Cisco, approximately one out of every three small and midsize businesses has experienced a cyber attack [12]. Furthermore, more than half of all cyber attacks result in financial damages totaling around $500,000 in US dollars. This highlights the significant financial impact that cyber attacks can have on businesses, emphasizing the importance of robust cybersecurity measures. The main reason many organizations fail to protect themselves is the lack of understanding of the importance and role of cybersecurity [4].

Ensuring cybersecurity has become a major challenge that requires ongoing efforts for a cyber defender, especially in the case of a large scale densely-connected environment such as a CPS mainly due to the complex and heterogeneous structure [5, 13, 14]. Periodic risk assessment plays an increasingly important role in securing any CPS and supports a cyber defender to identify critical areas of the infrastructure. Risk Assessment is also enforced by regulatory standards such as the FISMA [15], HIPAA [16], ISO 27001 [17], etc. Increasing cyber threats urge organizations to continuously emphasize information security. Commonly used risk assessment frameworks such as FRAP, OCTAVE, NIST's guide [18], ISO/IEC 27005, etc. define structured approaches and guidelines for risk assessment. How-

ever, these standards lack a metrics framework with attention to calculating risk. Other qualitative risk assessment techniques followed by organizations include What-if analysis, Checklists, HAZOP, Fault tree analysis (FTA), etc. However, the main challenge with any qualitative approach is that it is subjective and carried out by individuals based on their perceptions of the risk likelihood and consequences. Moreover, these approaches do not allow for the determination of probabilities and results using numerical measures.

As a result, a quantitative risk assessment model is required that not only is capable of providing asset-specific but also the overall risks for a particular CPS but also takes into account risk propagation among dependent nodes. On the other hand, vulnerabilities may also exist within highly secured facilities, offering adversarial entities the opportunity to infiltrate organizational security. Vulnerability assessment must also be included as part of a continuous risk assessment process.

1.1.2 Continuous Vulnerability Assessment

From our home networks to large-scale cyber-physical systems, vulnerabilities from software, hardware, and firmware are unavoidable. Traditional defenses, such as Firewalls, Antivirus, Internet Security suites, etc., may not offer complete protection against advanced cyber attacks including zero-days. Preventing adversarial access requires identifying attacker techniques, tactics, and procedures, employing security policies and controls to prevent such exploitation, and most importantly, patching security vulnerabilities on time. However, achieving effective cyber defense in large-scale computing environments could be challenging without a continuous risk monitoring approach that leverages up-to-date vulnerability information. Many vulnerability databases (VDBs) such as NIST's NVD [19], Symantec DeepSight, OSVDB, MITRE [20], etc. provide detailed information related to discovered vulnerabilities and their target products. However, obtaining, translating, and utilizing this information can be a very arduous task due to different data formats and consistency, scoring systems being used, and data integrity [21, 22]. Furthermore, vulnerabilities and cyber attacks vary from organization to organization, depending on the devices,

applications, security controls, and operating systems being used.

The number of reported vulnerabilities is increasing at an alarming rate [6] and it is quite challenging for cyber defenders to keep up with the emerging threats. To perform effective risk assessment, the availability of accurate and contextual cyber-threat information is vital. Although several vulnerability scanning tools [23–26] are available, these do not offer detailed insights into the overall risk of the computing infrastructure and mitigation approaches are primarily expert driven. Most tools are proprietary and business-driven where the implementation details are generally abstracted without any customization flexibility. Researchers developing vulnerability/risk assessment frameworks make use of the publicly available VDBs to fuel the assessment process [27, 28], and track the latest trends in cybersecurity vulnerabilities. Given the high need for continuous risk monitoring, it is vital to not only maintain consistency, quality, and integrity of the underlying vulnerability information set but also derive contextual information from the VDBs relevant to the computing infrastructure. Many studies [28–35] have identified various inconsistencies such as incompleteness, quality, and reliability of data. These inconsistencies exist mainly due to human error in various descriptive features of vulnerabilities which leads to incomplete or outdated information [21, 22]. Relying on and processing such inconsistent data can result in inaccurate results. The aforementioned studies solely address inconsistencies and do not include VDB integration or context-specific analysis in real-time.

Therefore, a data-driven framework that is capable of identifying infrastructure-specific vulnerabilities and obtaining vulnerability-specific information from various VDBs that can be used with the risk assessment framework is essentially needed. The knowledge driven from this framework can be further expanded to a Cyber Threat Intelligence (CTI) system that can be used not only within one organizational unit but across multiple units and multiple organizations.

1.1.3 Cyber Threat Intelligence

Awareness of a cyber defender plays a significant role in finding the potential attack paths an intruder might choose to invade organizational security. Securing digital assets depends on what kind of security controls are in place, and the degree of protection offered by such controls. However, inside these controls or other organizational assets, the presence of vulnerabilities or weaknesses can allow a threat actor to infiltrate highly protected facilities. A defender must not only rely on and retain information relevant to security controls but would also maintain an updated vulnerability information system in order to get a clear overall picture of the organizational security posture at regular intervals. According to National Vulnerability Database (NVD), the number of reported vulnerabilities is increasing at an alarming rate. In the year 2020 alone, not only the highest number of vulnerabilities (18,352) were reported to date [6], but also 57% of the reported vulnerabilities were classified as critical or high severity [36].

To perform vulnerability assessment, cyber defenders can either manually obtain and process information about the discovered computer security vulnerabilities from the publicly available vulnerability databases (VDBs) [27, 28, 37] such as NVD, Common Vulnerabilities And Exposures (CVE), Open Source Vulnerability Database (OSVDB), etc. or use vulnerability scanning tools [23–25]. Both options have their trade-offs [23]. In the case of VDBs, one can run into issues like data formats, data consistency and integrity, scoring systems, and metrics being used [21, 22]. The third-party vulnerability scanning tools on the other hand in most cases use the Common Vulnerability Scoring System (CVSS) [38] as a standard, however, they are still not widely adopted due to varied coverage, customization inflexibility, and the abstracted implementation details [23].

Ensuring cybersecurity is a major challenge that requires ongoing efforts for a cyber defender, especially in the case of a large scale densely-connected environment such as a CPS, mainly due to the complex and heterogeneous structure [5, 13, 14]. Periodic risk assessment supports a cyber defender in quantifying risks and identifying critical areas of the infrastructure. Relevant and timely received information about potential risks, threats,

and vulnerabilities aid the risk assessment process to derive more accurate and effective risk analysis on one hand, and an opportunity for a cyber defender to defend against these threats on the other. However, existing literature on cyber risk assessment often lacks consideration of vulnerabilities, or proposed frameworks are purely theoretical with limited implementation details. Additionally, contextual information related to the cyber infrastructure may be missing in current approaches. This highlights a gap in the current state-of-the-art, where a more holistic approach that incorporates vulnerabilities, practical implementation details, and contextual information is needed to accurately assess cyber risks. This lack of standardized contextual information creates blind spots in the defender's analysis of systems. Furthermore, highly secured organizational infrastructure can also get compromised by socially engineered cyber-attacks [39].

A threat intelligence system specifically tailored for large-scale environments that covers security for both, cyber and physical aspects of a CPS to provide contextual analyses and continuous risk assessment is required. For the cyber aspect, risks related to found vulnerabilities, whereas, the physical aspect should focus on risks related to the employed security controls, policies, network/service dependencies among network nodes, adversarial actors, and their capabilities. Furthermore, this system can be combined with an AI-Based model to make scenario-based predictions from the gathered knowledge and historical data.

1.1.4 Accelerating Cyber Risk Assessment with AI

Artificial intelligence (AI) in recent years has rapidly progressed in main domains such as robotics, personal assistants, smart homes, self-driving cars, etc. AI-Based systems can significantly reduce the amount of work for cyber defenders in case of conducting various analyses and predictions. AI on one hand is playing a major role in cybersecurity and is capable of making the most intelligent decisions a human would take [40]. Threat actors are constantly improving and updating their attack strategies using AI-Based cyber-attacks on the other hand [41], leaving no choice for cyber defenders but to develop intelligent systems that can utilize AI-Based solutions to anticipate risks in advance. Various AI techniques

such as data mining, natural language processing, machine learning, etc. can be used for solving cybersecurity issues like data/traffic analysis, risk predictions, behavioral patterns, and many more [42]. As the device heterogeneity in our networks is growing, the number of threats, types of cyber-attacks, and data produced by these devices are also growing, introducing various loopholes in the organizational infrastructures, constantly challenging cyber defenders [43]. With the exponential increase in the amount of data produced by several devices, it is essential to have an AI-Based intelligent system for monitoring and analysis of data. Commonly used AI applications in cybersecurity include spam filtering, fraud detection, botnet detection, hacking incidents, network intrusion detection, and many more [42].

Cybersecurity has become a major challenge in present times as our networks are not only limited to computers and networking devices but every other device is interconnected and requires internet connectivity. As a result of the rising interconnection and autonomy, there has been an increase in the number of cyber-attacks [41]. Conventional intrusion detection and prevention systems are not useful against zero-day attacks as the signatures/behaviors are not defined in the database [44]. AI-Based intelligent solutions on the other hand are gaining popularity in the cybersecurity domain and are very useful in such cases. Furthermore, 56% of cyber analysts are overwhelmed by the huge amount of data points they must monitor to detect and prevent intrusions, and 74% said that AI enables a faster response time [45]. AI can improve organizational security for cyber defenders in many ways such as monitoring incidents or intrusion detection, reducing the laborious human monitoring tasks [40]. Moreover, AI-Based techniques using machine learning algorithms are very helpful in bringing down the security breaches [42].

As a result, the present state-of-the-art requires a threat intelligence system that is not only capable of understanding threats and vulnerabilities for CPS but also capable of inferencing various threats using AI-based techniques to mitigate the found threats.

1.2 Motivation

The evaluation of cyber risk poses a significant challenge due to the heterogeneity and complexities of modern cyber infrastructures. The rapid growth of software and hardware components has made it difficult for cyber defenders to keep up with the latest cybersecurity trends. Furthermore, the ever-evolving nature of cyber threats and increasing sophistication of cyber attacks further complicates the accurate assessment of cyber risk. Existing risk evaluation frameworks and tools often lack practical industrial use cases and do not adequately incorporate vulnerability assessment as a critical component of risk assessment. Moreover, many of these frameworks are either purely theoretical or not publicly available, limiting their effectiveness in real-world cyber defense scenarios.

The use of AI in cyber risk evaluation is also limited to specific scenarios, despite its potential to significantly accelerate the risk analysis process. The lack of comprehensive and accessible AI-powered tools for cyber risk evaluation further adds to the challenges faced by cyber defenders. To address these limitations, there is a need for a framework that not only includes vulnerability assessment as a critical component of risk evaluation but also harnesses the power of AI to enhance the accuracy and efficiency of the risk assessment process. Such a framework would bridge the gap between the current state-of-the-art and the practical needs of cyber defenders, enabling them to not only evaluate cyber risk but also effectively mitigate it.

The main motivation of this research work is to fill the existing gaps in current cyber risk evaluation frameworks and propose a comprehensive framework that addresses the challenges faced by cyber defenders. The proposed framework aims to incorporate vulnerability assessment as an essential component of risk evaluation and leverage AI techniques to enhance the speed and accuracy of risk analysis. By including practical industrial use cases and making the framework publicly available, it aims to provide a practical and accessible tool for cyber defenders to evaluate and mitigate cyber risk in real-world scenarios. The proposed framework has the potential to significantly advance the field of cyber risk eval-

uation and contribute to improved cybersecurity practices in the face of the ever-evolving cyber threat landscape.

1.3 Research Objectives and Questions

This research aims to provide cyber defenders with the knowledge and information they need to protect their cyber infrastructures. As the modern cyber infrastructures comprise a diverse set of connecting devices as compared to a traditional computer network, it is quite challenging to gather device-specific information for every connecting device. This is mainly due to the kind of operating/embedded systems running on each device. Obtaining the infrastructure-specific information such as the employed security controls, organizational assets and their applications, network and service-based dependencies, running processes, and so on is essentially needed as the first step to assess risk for any given cyber infrastructure. These information pieces further allow the determination of various organizational loopholes and related mitigation techniques. Similarly, we use the obtained information to find the hidden patterns using various machine learning models, and use these models to predict potential cyber attacks within the organization in real-time. To summarize all of the above, we focus on the following objectives:

(A) Quantify the cyber asset's risk and security achieved through different controls.

Cyber defenders typically implement a range of software and hardware-based security controls to safeguard organizational infrastructures from intrusions. In a typical scenario, these defenders adjust the controls based on recommended security settings provided by vendors or security experts. However, the effectiveness of these controls in providing actual security is rarely evaluated. Therefore, the objective of this research is to address the following research questions in order to fill this gap and improve the understanding of cyber defense effectiveness:

Q1) How organizations can evaluate organizational security and quantify the security

risks related to organizational assets?

Q2) When adversaries are aware of the implemented security controls, how does this affect the overall security posture of the organizational cyber infrastructure?

Q3) How can we model the propagation of risks associated with controls and nodes to other nodes? Under this propagation scenario, how does it affect the overall risk of the infrastructure?

(B) Design and develop a scalable cyber-risk assessment and evaluation framework.

In a traditional industrial setup, cyber defenders employ a variety of tools, frameworks, manual, and automated approaches to detect anomalies within their organizational infrastructures that may impact the security of the organization. Additionally, they rely on various online information sources to gather information about the consequences of these threats and potential mitigation strategies. In light of this objective, this research aims to address the following research questions:

Q1) Given the node/service heterogeneity in standard infrastructures, how to track the dynamic system components including, but not limited to applications, processes, memory utilization, and port usage to effectively assess present security posture.

Q1a) How organizations can integrate information collected from external sources to accurately estimate the security risks in a continuous and real-time fashion.

Q2) How can we evaluate cyber risk considering adversarial capabilities, vulnerabilities, network dependencies, administrative policies, and provide mitigation techniques to produce cyber threat intelligence?

(C) Enabling accurate detection of risk anomalies from the operating environment and dynamically devising risk mitigation plans.

Artificial Intelligence (AI) has emerged as a crucial component in strategic decision-making, enabling organizations to make informed judgments with improved speed and accuracy. In the domain of risk assessment, the current state-of-the-art approaches

are often limited to risk-specific, vulnerability-specific, or threat intelligence-focused methodologies. However, there is a pressing need for AI-powered threat intelligence systems that can provide real-time risk assessment and mitigation capabilities. Therefore, the objective of this research is to address the following research questions in order to advance the field of AI-powered threat intelligence:

Q1) Identify and classify the risks associated with the cyber infrastructure into multiple uniform groups by leveraging the unstructured and unlabeled threat data.

Q2) Design a mitigation recommendation subsystem to assist in resolving anomaly alerts in real-time.

The primary objective of this research is to investigate and evaluate state-of-the-art frameworks, open-sourced and proprietary tools, and suggested techniques in the field of AI-powered threat intelligence for risk assessment and mitigation. By conducting a thorough examination of existing ideas, we aim to identify gaps in this domain and propose an innovative real-time risk assessment framework.

The research questions formulated for this study will shed light on the current state-of-the-art frameworks, open-sourced and proprietary tools, and suggested techniques related to AI-powered threat intelligence for risk assessment and mitigation. Through rigorous investigation and evaluation of these ideas, we will ascertain the limitations and shortcomings of existing approaches and propose a novel framework that addresses these gaps and offers real-time capabilities.

The ultimate goal of this research is to contribute to the advancement of AI-powered threat intelligence for risk assessment and mitigation by proposing a cutting-edge framework. This framework is expected to overcome the limitations of existing approaches and enable real-time risk assessment and mitigation, providing valuable insights and guidance to organizations and practitioners in the field of cybersecurity. The findings of this research have the potential to improve the effectiveness and efficiency of risk assessment and mitigation strategies in the dynamic landscape of cyber threats.

1.4 Contributions

This research introduces the Cyber-threats and Vulnerability Information Analyzer (CyVIA) framework, which is designed and implemented to enable continuous and comprehensive risk analysis of the target environment. CyVIA provides several key functionalities for effective risk assessment and mitigation:

1) Asset and infrastructure-wise risk assessment: CyVIA considers applied security controls, administrative policies, and adversarial capabilities to assess the risk associated with each asset and infrastructure in the target environment.

2) Asset and infrastructure-wise vulnerability assessment: CyVIA leverages multiple online sources, vulnerability types, severity, relationships, and computing products to assess vulnerabilities associated with each asset and infrastructure in the target environment.

3) Community-wide cyber threat intelligence sharing: CyVIA facilitates the sharing of cyber threat intelligence across the community, allowing for collaborative defense efforts and improved situational awareness.

4) Interdependencies between assets: CyVIA considers the interdependencies between assets, such as network and services, to identify potential risks and vulnerabilities associated with these interdependencies.

5) Identification of critical nodes on the network: CyVIA employs various factors, such as risk, vulnerabilities, severity scores, access vectors, and weakness types, to identify critical nodes on the network that require prioritized attention.

6) Consequences and mitigation information: CyVIA provides information on the consequences of found threats and suggests mitigation strategies to effectively address them.

The CyVIA framework offers a comprehensive and continuous risk analysis approach that encompasses various aspects of cyber threats, vulnerabilities, and interdependencies between assets. It provides valuable insights for cyber defenders to assess, prioritize, and mitigate risks in their target environment, ultimately enhancing the cybersecurity posture

of organizations.

1.5 Significance of the Research

The proliferation of new and evolving threats, as well as the increasing number of reported vulnerabilities, has seen a significant surge in recent years. Adversarial entities are constantly seeking loopholes to exploit organizational security, making it imperative for cyber defenders to stay vigilant. However, traditional approaches to organizational security assessment, which rely on manual and semi-automated methods, are typically generated periodically based on need, posing challenges in keeping up with the ever-changing threat landscape. Real-time monitoring of adversarial activities is crucial to prevent successful attacks. Therefore, this research underscores the importance of continuous risk and vulnerability assessment, cyber threat intelligence (CTI), and an AI-based prediction engine in the field of cybersecurity.

Need of Continuous Risk and Vulnerability Assessment: Continuous risk and vulnerability assessment are imperative in the realm of cybersecurity, as existing frameworks often remain theoretical, lacking in implementation details, and ill-suited for industrial scenarios. Moreover, these frameworks typically do not offer real-time protection and fail to integrate risk and vulnerability assessments into a unified approach. To address these limitations, this research proposes a novel framework called Cyber-threats and Vulnerability Information Analyzer (CyVIA). CyVIA is designed to integrate risk and vulnerability assessment into a single framework, providing continuous cyber risk assessment that captures all changes occurring within the network. By consolidating risk and vulnerability assessments and enabling real-time monitoring, CyVIA aims to overcome the limitations of existing frameworks and enhance the accuracy and effectiveness of cyber risk evaluation in practical, real-world scenarios.

Need of Cyber Threat Intelligence: CTI plays a pivotal role in educating cyber defenders about adversarial tactics, techniques, motives, targets, and attack behaviors. Existing CTI frameworks typically revolve around public or private knowledge bases and provide threat-specific data in its entirety. However, in order to gain more contextual information about specific threats, cyber defenders often need to manually search for related information, which can be laborious and time-consuming. In contrast, the proposed Cyber-threats and Vulnerability Information Analyzer (CyVIA) framework aims to streamline this process by automatically gathering operating environment-specific information and threat-specific data from multiple online databases. This allows CyVIA to provide relevant CTI, including information on applicable threats, their relationships with other threats, and associated consequences and mitigation techniques. By leveraging multiple data sources and automating the collection of threat-related information, CyVIA aims to enhance the comprehensiveness and efficiency of CTI, providing cyber defenders with a more comprehensive and contextual understanding of threats for effective risk assessment and mitigation.

Need of AI-Based Prediction Engine: AI-based solutions have been widely used in the field of cybersecurity, but they can have both positive and negative implications, as they can be leveraged by both attackers and defenders. Additionally, AI-based learning algorithms may exhibit erratic behavior when confronted with adversarial examples in datasets. Despite these challenges, the utilization of AI approaches is crucial for defenders to proactively anticipate risks based on threats, user behavior, network intrusions, and other factors. Therefore, the proposed Cyber-threats and Vulnerability Information Analyzer (CyVIA) framework seeks to combine CTI capabilities with an AI-based learning model to effectively anticipate future risks and enhance overall cybersecurity posture.

In summary, this research identifies the limitations of current state-of-the-art approaches in cybersecurity, which include the lack of continuous risk assessment, the need for contextual CTI, and the utilization of AI-based prediction engines. The CyVIA framework proposed in this study addresses these limitations by integrating risk and vulnerability

assessments, providing relevant CTI from multiple online databases, and incorporating an AI-based learning model. Furthermore, CyVIA takes into account risk propagation among dependent nodes at different layers, making it a comprehensive and effective framework for cyber risk management.

1.6 Dissertation Overview

The remaining sections of this dissertation are organized as follows: Chapter 2 provides an extensive discussion of related works in the fields of risk and vulnerability assessment, cyber threat intelligence, and AI-based solutions for cybersecurity, as well as their limitations. Chapter 3 presents the foundational quantitative risk model used in the presented framework. Chapter 4 outlines the initial work on vulnerability assessment. Chapter 5 integrates risk and vulnerability assessment in the CyVIA 2.0 framework. In Chapter 6, the AI-based prediction engine is presented, which allows for speedy analysis by predicting specific attack types from identified loopholes in the cyber infrastructure. Chapter 7 highlights the AI-based inference engine, which proposes mitigation strategies for identified risks. Finally, Chapter 8 concludes the dissertation by discussing the contributions of this research and potential future directions for further study.

Chapter 2

Related Work

Cyberspace has been expanding drastically integrating new generation hardware, software, and other devices. The scope of adversary risks expands exponentially as the depth and variety of this integration rises. Attackers can use sophisticated tools and strategies to continually investigate these systems for crippling flaws. Cyber defenders on the other hand are constantly challenged to find and patch these flaws. However, having limited resources and information, cyber defenders generally fail to identify intrusions in real-time. In this Chapter, we discuss the current state-of-the-art methodologies suggested by various studies to keep the organizational infrastructure safe. We investigate different options to address the research questions as described in Chapter 1, Section 1.3.

2.1 Cyber Risk Assessment

Cybersecurity is very broad and detecting malicious activities on the network is quite challenging [4]. Typically the cyber defenders have limited resources and awareness which makes it more difficult to analyze and model cybersecurity. The prevalent IoT applications, on the other hand, introduce numerous uncertainties in the CPS; unanticipated or unmanaged risks yield a highly competitive infrastructure for a cyber defender. Compromised security can cause not only financial loss but also endangers humans in the case of the medical and healthcare sectors. The adversarial entities either exploit the system vulnerabilities or use socially engineered attacks [46]. Detecting, reporting, and fixing vulnerabilities can be very tedious and a prolonged process. Preventing or reducing the impact of cyber-attacks largely depends on the quality of defense mechanisms for any cyber-terrain, and how well

the dependencies are modeled.

Reference [47] propose a framework that heavily relies on historical attack data and CVSS scores, and is limited to abstract analysis of situational awareness. Authors in [48] propose a CVSS v3 based risk assessment methodology that is limited to traditional computer networks. As in [49] propose a quantitative model where "Risk = Threat (T) x Vulnerability (V) x Consequence (C)". The model heavily relies on SME interaction and historical data to determine results. Reference [50] published their ongoing research model that maps the top-level business processes with the digital assets utilizing [51] to make better decisions. Working on similar lines, researchers from MITRE corporation, introduce crown jewel analysis (CJA) [52], the model is further updated to as Cyber Mission Impact Business Process Modeling tool (CMIA) [53], where the key accomplishment is representing the cyber dependencies between assets. Furthermore, in [4], the authors present a complete product that quantitatively identifies cybersecurity risks and provides suggestions on how to implement optimal security against the identified risks. Reference [54] presents a modeling and visualization tool, Cauldron, that maps the entire network, and the potential cyber threats with scenarios to improve overall security posture. The CMIA model and other models suggested by the MITRE corporation are only available to government agencies with more theoretical and no implementation details. The cauldron framework is also a part of military-funded research more focused on the implementation side, and missing granular work.

Authors in [55] present an information-sharing model, information sharing is first proposed by [56], and later by [57–59]. The main challenge with cybersecurity information sharing is that none of the organizations share attack information with others due to reputational damage. Similarly, other studies [60–63] suggest various quantitative approaches, whereas the industry typically follows a qualitative approach for risk assessment. Cyber insurance is also proposed as a potential and promising solution for risk elimination [64], and security spending optimization. However, risk interdependency introduces investment inefficiency and cyber insurance is ineffective in this case.

Understanding the challenges of CPS and performing the risk assessment is quite challenging and requires continuous efforts. The aforementioned quantitative models are relatively new and are not implemented in the industry directly as a case study. A publicly available model implemented on an industrial scenario such as IoT, transportation, manufacturing, supply chain, etc. can provide a clearer idea of the current state of the art, and what needs to be done to improve these models. Our goal is to develop a versatile industrial model that can be customized for implementation in any industrial environment, offering a transparent overview of the associated risks for the given CPS (Cyber-Physical System). This comprehensive model will encompass the evaluation of cyber risks from various dimensions, including but not limited to network nodes, security controls, and administrative policies. The outcomes of this model will specifically address the research questions Q1-Q3, as outlined in objective A in Chapter 1, Section 1.3. By fulfilling these objectives, we aim to provide valuable insights and actionable information to enable effective risk management and decision-making in industrial settings, contributing to the advancement of cybersecurity practices in the industrial domain.

2.2 Vulnerability Assessment

Several open VDBs including NVD, MITRE, and OSVDB provide information related to software and hardware security concerns thus enabling an opportunity for defenders to quickly detect and defend against network threats. Most VDBs use MITRE assigned CVE IDs and NVD assigned CVSS scores for vulnerabilities; therefore, inconsistencies in the data provided by NVD or MITRE can be devastating. Authors of [29] find inconsistencies in the publication dates, vendor and product names, severity scores, and vulnerability types in NVD data. 40.18% CVEs in the NVD have incorrect software names or versions [30,31]. Authors in [34] compare the NVD's scoring system with other VDBs and observe poor and uncertain results in the access complexity and authentication metrics. Authors in [35] propose an improved scoring system by adding the host environment (services and operating

systems) to the base metric group. Authors from MITRE [32], highlight the bias and noise in the vulnerability data and stress cautious use while performing studies. Several studies propose various Natural Language Processing and Machine Learning techniques to address the inconsistencies [21, 27, 28, 31], however, none of them focus on how to use the corrected vulnerability information to detect vulnerabilities in cyber infrastructures.

Similarly, several authors have compared and evaluated different VDBs. In [33], authors compare seven VDBs and observe that vulnerabilities in each database are listed under different classifications. This creates complications in conducting risk analysis based on information gathered from multiple sources. The authors propose CWE as the ideal classification taxonomy. Reference [65] compares four VDBs and observes 30%-35% missing CVE IDs in Symantec and Security Focus. Working on similar lines, [66] proposes combining CVE, NVD, and IBM X-Force in a local relational database, emphasizing on integrity and accuracy of CVE data. The quality of information maintained by VDBs is put to the test when researchers rely on this information to predict the trends and patterns in software vulnerabilities. The errors identified in the VDBs, particularly the NVD, are highlighted in the aforementioned studies. Therefore, it's vital to develop a dependable vulnerability analysis framework that is capable of merging contextual yet error-free data from multiple sources for accurate risk analysis of any given cyber infrastructure. We aim to answer Q1 and Q1a specifically under objective B as discussed in Chapter 1, Section 1.3.

2.3 Cyber Threat Intelligence

Traditional computer networks have transformed into Cyber-Physical Systems (CPS) with an ever-growing number of connected devices and increased numbers of various applications and services. Internet of things (IoT) and Industrial Internet of Things (IIoT) on the other hand are also reshaping our traditional networks to highly convoluted infrastructures introducing several uncertainties. Identification of cyber and physical aspects is extremely

important to evaluate network security. Authors in [67] propose a novel method that helps in solving the network structure identification problem by comparing various classical sparse recovery methods on noisy observed data. Similarly, authors in [68] use a similar approach to identify the bottlenecks within the given network. On the other hand, securing such a wide range of integration has become a major challenge in recent times where cyber defenders either have limited awareness or limited resources [2]. On average, organizations spend $18.4 million annually on cybersecurity tools [69] where 58% are willing to increase the budget by an average of 14% for the following years. However, 53% of information technology experts are unsure whether the cybersecurity tools are working as expected, and only 39% admit they are confident in the investment [70]. Global spending on cybersecurity products and services is expected to exceed $1 trillion in 2021 [71].

Vulnerability scanning tools provide insights into the cyber aspect of any network and proactive defense against application threats and are still not widely used as compared with malware or antivirus software. Authors in [23] provide a comparative evaluation of different tools and provide guidelines to practitioners for selecting the right tool. Authors in [38] evaluate nine different cybersecurity risk assessment tools. The study shows that most of these tools use the Common Vulnerability Scoring System (CVSS) as a standard and can integrate with other commercial technology partners for enhanced vulnerability management. Similarly, authors in [23–26] propose many other vulnerability scanning tools. However, the main issue with vulnerability scanning tools is that they do not offer insights about the overall infrastructural risk, and the implementation details on the other hand are generally abstracted.

Cybersecurity is an ongoing effort and organizations can not afford to look away in order to manage their cyber risk effectively. A cybersecurity evaluation tool (CET) is proposed in [72]. CET consists of 35 self-rate question survey that identifies organizational vulnerabilities based on a set of standard measures. CET helps in identifying the fundamental post-breach efforts that can proactively secure sensitive data. Romilla Syed proposes a cyber intelligence alert (CIA) system that informs common users about vulnerabilities and

their potential countermeasures [37]. CIA collects vulnerability from Twitter, CVE, NVD, vendor websites, and uses a machine-learning approach to reason if the alert should be raised for a vulnerability or not. Evaluating cybersecurity has also become a challenge with the increased number of cyber threats. Authors in [73] propose a cybersecurity audit model (CSAM) that implements the cybersecurity awareness training model (CATRAM). Similar to CET, CSAM also presents an ontology that can be used to evaluate cybersecurity assurance, however, the main challenge with these ontological schemes or tools is that they are subjective and carried out by individuals based on their perceptions of the risk.

Understanding the potential threats in CPS itself is challenging [74], authors in [5] present a security framework that studies the four main security concerns of CPS, i.e. threats, vulnerabilities, attacks, and controls. The proposed framework can be used to develop effective controls for CPS. The main challenge in CPS security is the increasing number of IoT devices that leads to a rise in the number of vulnerabilities, and eventually leading to successful exploitation [75]. Unlike [5], authors in [13] focus on the impact of cyber attacks on authenticity, confidentiality, reliability, resilience, and integrity. Similar to [5], the main challenges with CPS are raised in [13] and a tree of potential attacks on CPS is proposed. The difference between CPS, IoT, and Industry 4.0 is still very ill-defined, defining layers for each can help security researchers and professionals to develop more concrete security frameworks. Authors in [14] try to differentiate CPS from IoT and traditional information technology systems. The authors also present security issues at various layers of CPS, the affected security parameters, and the associated countermeasures to address these issues. Authors in [76] propose and implement a risk-informed approach that identifies critical CPS assets and the impact of affecting vulnerabilities on a smart grid system and plans to develop a tool to automate the process.

Cyber threat intelligence (CTI) sharing is another risk-informed approach that provides evidence-based knowledge about cyber threats that may exist within any cyber infrastructure. Utilizing such knowledge can be very beneficial in aiding the decision-making process to detect and prevent catastrophic events. However, how and what type of information to

share still remains unclear since there is no common definition or ontology available for CTI sharing [77, 78]. Most of the current CTI platforms operate manually and the slow sharing process becomes an obstacle for CTI sharing [79]. On the other hand, certain organizational risks such as free-riding, trust violation, negative publicity, reputational damage, etc. also prevent CTI sharing [80, 81]. Authors in [82, 83] stress the need for rules and regulations for CTI sharing in the existing policies.

Researchers at MITRE took a different approach to CTI. At first, they introduced Common Attack Pattern Enumeration and Classification (CAPEC) in 2007 that provides a range of commonly used attack patterns [84]. Later in 2015 MITRE introduced the Adversarial Tactics Techniques & Common Knowledge (ATT&CK) framework [85]. ATT&CK is a behavioral model that provides specific information on adversary tactics, techniques, and procedures as observed by the community for known actors. Which can be used for adversary emulation, red teaming, behavioral analytics development, defensive gap assessment, and cyber threat intelligence. The ATT&CK model consists of a set of techniques and sub-techniques that an adversary can take to accomplish their objectives which are represented in the ATT&CK Matrix as shown in [86]. ATT&CK also provides mitigation techniques for preventing the listed adversary techniques and sub-techniques. ATT&CK is further extended to focus on industrial control systems with additional use cases [87].

The aforementioned studies either do not satisfy the evolving security needs of CPS, highlight the security concerns related to CPS, or propose theoretical concepts to address the same. MITRE ATT&CK on the other hand is a community-based knowledge base with the focal point on adversary emulation and provides threat-actor-based information. A proactive cyber threat intelligence system specifically tailored for CPS to provide contextual information is critically needed. To ensure CPS or any infrastructural security it is vital to understand and identify the 1) various layers and the integrated devices in each layer as seen in Figure. 5.2, 2) assets that need protection, 3) controls protecting the assets and integrated devices, 4) threats, vulnerabilities, and VDBs, and finally, 5) users and other environmental variables such as running applications, open ports, processes, etc. A context-

aware framework that considers all of the above and can be used to mitigate malicious and harmful threats to answer specific research questions Q2 under objective B as mentioned in Chapter 1, Section 1.3.

2.4 AI-based Models for Cybersecurity

The accuracy and reliability of cybersecurity data are extremely important to derive the most useful and accurate decisions or analyses. Many researchers have addressed the inconsistencies found within the vulnerability data using various machine learning models [28–35]. AI-based solutions for cybersecurity are gaining popularity in recent years [45], but it is still relatively new. In a recent survey [88], authors discuss the current state of AI in cybersecurity, and conclude that AI will continue to grow not only for businesses but also for personal use. Authors in [89] highlighted potential AI-based solutions in cybersecurity such as user authentication, network situation awareness, dangerous behavior monitoring, and abnormal traffic identification. These applications can easily identify potential adversarial activity within the organizational infrastructure. Similarly, authors in [42] discuss various AI applications, algorithms, and libraries that can be used for implementing such solutions.

The AI discipline is divided into two categories: rule-based techniques and machine learning techniques, which allow computers to learn from vast amounts of data. Adversaries learn how to take advantage of AI-based learning methodologies such as deep learning, reinforcement learning, and support vector machines to weaponize them by automating the attack process. Authors in [41] explore various studies on AI-based cyber-attacks and classify several aspects of the malicious use of AI. The report [90] surveys the changing threat landscape and warns about potential AI techniques that can be harmful. The authors propose different ways to better forecast, prevent, and mitigate these threats. In another study [91], the authors compare different AI approaches in cybersecurity, their used methods, results, and advantages if any. The study concludes that AI-based approaches

need continuous updates, human interaction, and training. Furthermore, the authors in [89] also discuss the shortcomings of AI-based approaches such as interference of confusing data, maliciously modified model, lack of transparency in the AI decision-making process, etc., and propose a Human-in-the-loop method that combines AI with human wisdom to overcome the limitations.

A study conducted by Microsoft indicated that most of the attacks in 2018 lasted less than an hour [92]. It is quite difficult to prevent such attacks with traditional incident response. AI in such cases can be a very robust and resilient approach [93] to quickly prevent and recover from the attack state. On the other hand, by including a small number of adversarial examples in the datasets, AI-based learning models can be easily fooled from predicting desired outcomes [94–97]. Access to the datasets and the trained models must be carefully assigned as well. Therefore, it is critical to have an AI-powered risk assessment framework that is capable of highlighting anomalies in real-time to reduce the damage and eventually prevent cyber attacks. We plan to address the research questions Q1 and Q2 under objective C as discussed in Chapter 1, Section 1.3.

2.5 Summary

Cyber infrastructures in the present era exhibit significant diversity and uncertainty, posing challenges to modern defense mechanisms that are incapable of providing absolute security. Cyber defenders constantly face formidable adversaries who are proficient in conducting AI-based cyber-attacks utilizing cutting-edge technology. This chapter has highlighted several shortcomings in the current state-of-the-art approaches, including the lack of continuous risk assessment conducted only on demand, the exclusion of security controls and administrative policies from cyber risk assessment, separate treatment of vulnerability assessments despite their intrinsic connection to overall cyber risks, the labor-intensive and time-consuming nature of CTI information retrieval for threat intelligence, and the focus of existing AI-based cybersecurity solutions primarily on addressing inconsistencies

in vulnerability databases.

In light of these limitations, we propose a three-dimensional security framework called CyVIA, which integrates cyber risk evaluation of network nodes, security controls, and administrative policies, vulnerability assessments encompassing hardware and software aspects of the cyber infrastructure, CTI for contextual analytics, and AI-based anticipation of future risks. By combining these dimensions, CyVIA aims to provide a comprehensive and holistic approach to cybersecurity, addressing the interconnected nature of various risk factors in cyber infrastructures and empowering cyber defenders with enhanced situational awareness, advanced analytics, and prediction capabilities to proactively mitigate emerging cyber threats.

In the next Chapter, we present our base risk assessment model which is used for risk quantification.

Chapter 3

Quantitative Risk Modeling and Analysis

Advancements in the Information Technology (IT) industry have enabled the expansion of the traditional organizational boundaries, integrating various new-generation devices into existing networks, such as IoT. The evolving IoT innovation has provided amplified connectivity for various business aspects, and helped reap more value for businesses. However, IoT devices increase the complexity of the operating environment, and securing such a diverse network becomes quite challenging. Furthermore, a highly interconnected environment provides opportunities for adversarial entities to gain command and control of the targeted network. Moreover, evolving network configurations, changing landscape of vulnerabilities, and inter-dependencies between cyber assets forfeits the traditional risk assessment techniques to protect the digital assets.

This Chapter addresses the research questions Q1-Q3 under the objective A, as discussed in Chapter 1, Section 1.3. We present our proposed model [98] which provides a cyber defender with the understanding of potential risks associated with the digital assets and enables an opportunity to reduce the impact of cyber-attacks. We start by defining the various components of the model. At first, we describe the organizational assets, followed by the defensive mechanisms in place, followed by the assumptions made by the model. To evaluate our model, we consider a diverse network shown in Fig. 3.1 where the proposed model provides full situational awareness of the risk.

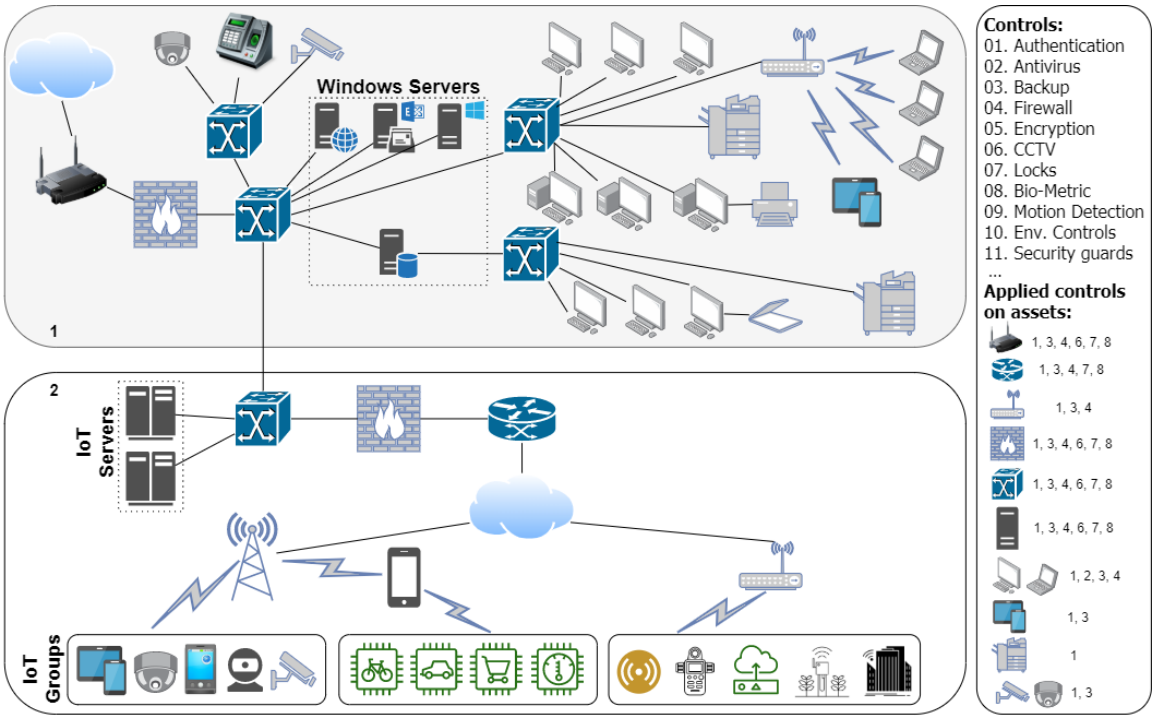

Figure 3.1: Industrial CPS environment.

3.1 Defining Organizational Assets

In the first stage, we identify organizational assets. Let K_i represent the organizational assets where K_i=$\{K_1, K_2, K_3, \ldots, K_n\}$. These assets are classified into seven major categories as described below:

1) Computers: Mainframes, Servers, Desktops, Laptops, Tablets, mobile devices, other computing devices. **2) Network Devices**: Firewall, Routers, Switches, Wi-Fi access points, other network equipment. **3) Security Devices**: Security Systems, CCTV, Locks (typical locks, Biometric, Face detection), other devices. **4) Data Storage**: Disk Storage Systems, Tape Storage Systems, Optical Storage Systems, Portable Data Storage, other storage devices. **5) Software**: Any licensed and open-source software in use by the company, Software Inventory. **6) Corporate**: Website, Social Media, Business Logo, other branding assets. **7) Others**: Peripherals (Display Monitors, Scanners, Printers, Projec-

tors, UPS, TV assets, etc), other devices.

Each of the above-mentioned categories is assigned a default criticality value based on priority, type, and the associated criticality as defined by the business. Let C_{K_i} represent the criticality value of how critical the asset K_i is on the network.

In the case of a cyberattack, a critical element to understand and evaluate is the risk propagation, where not only the targeted nodes suffer from a direct impact, but also the nodes that directly or indirectly depend on the targeted nodes. To address this concern, let D_{K_i} represent the dependency value, describing how many nodes are dependent on asset K_i.

3.2 Defining Defensive Mechanisms

To protect digital assets, cyber defenders apply various cutting-edge controls that lessen the associated risk factors. We identify and list the available controls in three major categories:

1. **Technical Controls**: $T = \{T_1, T_2, \ldots, T_8\}$.

   T_1: Strong Authentication, T_2: Antivirus/Internet Security, Security patches, fixes, updates, T_3: Disabling insecure and unneeded services, T_4: Intrusion Detection Systems, Intrusion Prevention Systems, T_5: Data and System Backups, File Integrity Monitoring, etc. T_6: Firewalls, T_7: Secure Protocols, T_8: Encryption.

2. **Physical Controls**: $P = \{P_1, P_2, \ldots, P_6\}$

   P_1: Video Surveillance, P_2: Locks and Bio-metric, Face detection access controls, P_3: Motion Detection Systems, P_4, Environmental Controls (Temperature, Humidity, Fire, etc.), P_5: Security Guards, dogs, P_6: Man-traps, Fences.

3. **Administrative Controls**: $A = \{A_1, A_2, \ldots, A_4\}$

   A_1: Comprehensive security policy, A_2: Periodic Security awareness training, A_3: Data classification (Encryption, Backup, etc.), A_4: NDA signing (newly hired staff).

The aforementioned controls integrate multiple layers of defense, however, do not guarantee 100% protection. Each control may not necessarily apply to every asset, e.g. a CCTV camera may only be required for a server computer. Adversarial entities can exploit loopholes in the known security controls, thus increasing overall risk. To address these concerns, we apply various controls and policies together as follows:

1. Since the control application varies from asset to asset, we introduce $M = \{M_1, M_2, \ldots, M_n\}$ representing the **must have controls**, mandatory security controls for a particular asset. Let $G = \{G_1, G_2, \ldots, G_n\}$ represent **good to have controls**, recommended controls that provide additional protection to particular assets. Let $O = \{O_1, O_2, \ldots, O_n\}$ represent **optional controls**, that are optional but not mandatory.

2. Knowing what type of security controls are implemented, it is easier to exploit known flaws. Hence, increasing risk, and decreasing the expected level of protection from the placed controls. To address this concern, we consider exposed (E) and not exposed (NE) situations for the security controls (M, G, O). Let MC_K represent the maximum control expected by the implemented security controls on assets.

3. The risk from humans (adversarial entities) can be divided into two major categories, a malicious insider, and an external adversary. An internal employee with some level of information can intentionally or unintentionally harm the CPS. The external adversaries on the other hand, always try to gain unauthorized access. In either case, the risk from humans should be considered. We classify these categories and represent the total risk, X_{K_i}, posed by humans on the asset K_i.

3.3 Assumptions

We have considered the following assumptions in the proposed model:

1. The model provides security standing based on the provided assets and the applied controls. If an in general security requirement is missing from the CPS, it is not

considered in the model e.g. if there is no backup system is not in place, the model assesses security based on the available controls.

2. The model assumes each asset is connected to the network with some level of dependency and applied controls. The standalone peripherals are not considered in this model.

3. In some cases such as calculating final values for M, G, O, the model places the value 0 instead of ∞ to avoid errors.

4. We have considered all network devices, security-related equipment, and corporate assets as highly critical assets.

In the following section, we estimate the overall security risk using the proposed model.

3.4 Risk Model for CPS Environment

Once we have C, D, M, G, O defined, the model can calculate total protection or total control provided by the applied security controls on a particular asset.

$$TC_{K_i} = (\frac{M^{ap}_{K_i}}{M^{av}_{K_i}} \times W_M) + (\frac{G^{ap}_{K_i}}{G^{av}_{K_i}} \times W_G) + (\frac{O^{ap}_{K_i}}{O^{av}_{K_i}} \times W_O) \tag{3.1}$$

where TC_{K_i} represents the weighted total control for the applied controls on the asset K_i, $M^{ap}_{K_i}$ are the applied must have controls out of the available must have controls ($M^{av}_{K_i}$) for the asset K_i. W_M, W_G, W_O represent the weights assigned to each category. The TC_{K_i} value provides the level of protection applied to the assets. Hence, we can calculate the final criticality FC, or the actual risk for K_i.

$$FC_{K_i} = 1 - TC_{K_i} \tag{3.2}$$

If two assets, K_i and K_j, are similar with same levels of protection and associated risks, they can be differentiated with the dependency factor, D_{K_i} and D_{K_j}, represents the number

of dependent nodes for asset K_i and K_j. We calculate the associated impact I as:

$$I_{K_i} = FC_{K_i} \times D_{K_i} \tag{3.3}$$

To get further insights and situational awareness of asset-wise risk, the model calculates the minimum and maximum risks as follows:

$$I_{K_i,min} = (1 - MC_K) \times D_{K_i} \tag{3.4}$$

where, $I_{K_i,min}$ is calculated assuming that all possible controls are applied to secure the asset K_i. $I_{K_i,max}$, on the other hand, is the maximum risk that the asset K_i can be exposed to, considering none of the controls are applied, $I_{K_i,max} = D_{K_i}$.

Once we derive the associated risks for individual assets, we can calculate the overall infrastructural risk for the CPS. To do this, we first normalize the calculated impact values as follows:

$$NR_{K_i} = \frac{I_{K_i} - I_{K_i,min}}{I_{K_i,max} - I_{K_i,min}} \tag{3.5}$$

And then, we calculate the final or overall risk, FR as:

$$FR = \frac{\sum_{i=1}^{n}[NR_{K_i} \times D_{K_i}]}{\sum_{i=1}^{n} D_{K_i}} \tag{3.6}$$

Risk propagation from one asset K_i to another K_j is a critical element, the overall risk does not include the propagated risk values. We calculate the propagated risk as follows:

$$PR_{K_i} = \frac{FC_{K_i}^{(S)} \times FC_{K_i}^{(C)}}{\sum_{i=1}^{n} FC_{K_i}^{(C)}} \tag{3.7}$$

where PR represents the propagated risk for the clients or dependent nodes, $FC_{K_i}^{(S)}$ refers the FC value of the serving node K_i, and $FC_{K_i}^{(C)}$ refers the FC value of the dependent node K_j. To evaluate the model, we consider a sophisticated industrial CPS as shown in Fig. 3.1 under the following four scenarios.

3.4.1 Not Exposed (NE) CPS Environment

This scenario assumes that the adversary does not have any information about security controls. The probability of attack success is low in this case, meaning the adversarial risk X_{K_i} is low, and the expected level of protection obtained from the security controls (MC_K) is high.

3.4.2 Exposed CPS Environment

In this case, we consider an attacker with some level of information about the employed security controls. A successful attack is more likely in this case, meaning the adversarial risk X_{K_i} is high, and the expected level of protection (MC_K) provided by the security controls is low.

3.4.3 Improved NE CPS Environment

This scenario refers to the case where a cyber defender has already evaluated the overall score from the NE and E scenarios and based on the knowledge gained, improved the infrastructure. This case provides an opportunity for the defender to reduce the overall risk to a minimum.

3.4.4 Risk Propagation in the CPS Environment

The adversarial entities can gain access to any node and move around the network from one node to another. This case assumes that an adversary with some level of information has gained access to the company router and firewall, and making her way to the end nodes using a top-down approach.

3.5 Evaluation Metrics

To identify, quantify, and mitigate risk considering the node/service heterogeneity in cyber infrastructures, and to reduce risk over time, we define the following metrics:

1. Risk Estimation Metric: The goal of this metric is to identify the potential susceptibilities within the given cyber infrastructure, and provide instructions on how to minimize them. This metric identifies possible paths an adversarial entity might choose to compromise organizational security.

2. Risk Preparedness Metric: The aim of the this metric is to focus on individual assets and highlight whether they are fully-patched and up-to-date.

3. Risk Adaptability Metric: Reduce the amount of risk over time, as compared with the initial state. For this metric, we evaluate the framework under the following three scenarios:

a. Not Exposed (NE) Environment: This scenario assumes that the adversary does not have any information about the implemented security controls.

b. Exposed (E) Environment: In this case, we consider an attacker with some level of information about the employed security controls, and a successful attack is more likely in this case.

c. Improved NE (IMP-NE) Environment: This scenario refers to the case where a cyber defender has already evaluated the overall risk score from the NE and E scenarios, and provides an opportunity for the defender to reduce the overall risk to a minimum. The risk is reduced based on the knowledge gained, and by applying the provided recommended security policies.

3.6 Results

To evaluate the proposed model, we start defining the required parameters used by the model. Once we have defined all assets, the model evaluates the security based on how

Table 3.1: Control weights.

Controls	**NE-Weights**	**E-Weights**
M	0.50	0.42
G	0.20	0.16
O	0.10	0.06
MC_K	**0.80**	**0.64**

critical the asset C_{K_i} is, the number of dependent nodes D_{K_i}, and what level of protection M, G, O is applied on the assets. C_{K_i} ranges from 0 to 1, $C_{K_i} = 0.7 - 1$ means the asset K_i is highly critical, $C_{K_i} = 0.4 - 0.69$ means medium, $C_{K_i} = 0.1 - 0.39$ means low, and $C_{K_i} = 0 - 0.09$ means very low or not critical at all. As an example, mainframes and other server computers are marked as highly critical since they have dependent nodes, hence $C_{K_i} = 0.7$. On the other hand, a desktop: $C_{K_i} = 0.3$, a laptop owned by a manager having important business details: $C_{K_i} = 0.5$. Similarly, other categories can be assigned values.

The dependency element differentiates two assets K_i, K_j with similar specifications, and applied controls from one another. $D_{K_i} = 50$, if 50 nodes are depending on the asset K_i, and $D_{K_i} = 1$, in case of no dependents or 1 dependent node.

We assign weighted values for $M, G,$, and O under NE, and E situations, representing the expected level of protection, as shown in Table 3.1. The maximum protection MC_K for the NE scenario is 0.8, meaning, if all recommended controls are applied, the system is 80% secure. Whereas in the case of E, the maximum protection reduces to 64%, i.e. a 20% drop. The remainder is the risk from humans as shown in Table 3.2. Please note that each control value and adversarial risk value can be adjusted over time and these are the initial starting point values.

Total control, TC_{K_i} ranges from 0 to MC_K, representing the protection level provided by the applied controls. $TC_{K_i} = 0$ means the applied controls are providing no security, whereas $TC_{K_i} = 0.8$ means 80% security from adversarial acts. $I_{K_i}, I_{K_i,min}, I_{K_i,max}$ repre-

Table 3.2: Risk from humans.

Controls	**Internal**	**External Adversaries**			X_{K_i}
	Employees	**Novice**	**Intermediate**	**Expert**	
Not Exposed	0.03	0.02	0.05	0.10	0.20
Exposed	0.05	0.04	0.09	0.18	0.36

sent the actual, minimum, and maximum risks posed to the organizational assets where $I_{K_i,min} \leq I_{K_i} \leq I_{K_i,max}$. The model computes normalized risk NR_{K_i} using min-max normalization, to calculate the infrastructural risk (FR_K) of the CPS. NR_{K_i} is a positive decimal number and FR_K ranges between 0 to 1. To evaluate the scenarios described in the previous section, we apply the proposed model as follows.

3.6.1 Not Exposed CPS Environment

This scenario assumes that the adversary has no information about the network topology or applied controls. Hence, the applied security controls are providing the desired level of protection. The calculated infrastructural risk (FR) in this case is 37.72%. As seen in Fig. 3.2, the highest risk $I_{K_i} = 43.89$ is posed to the asset IoT Server 1, followed by $I_{K_j} = 43.88$, posed to the asset Router01. It can also be observed from the graph that there is an opportunity to improve the risk for both assets since $I_{K_i,min} = 16$ for IoT Server 1, and $I_{K_j,min} = 13$ for Router01.

3.6.2 Exposed CPS Environment

When the adversary has potential information about the network, the probability of a successful attack increases, thus reducing the expected level of protection from the security controls. Fig. 3.3 demonstrates that the risk values for all assets and the overall risk ($FR = 40.32$) increase when the controls are exposed, and at the same time, there is an increase in asset-wise risk. $I_{K_i} = 51.11$ for IoT Server 1, and $I_{K_j} = 48.10$ for Router01.

3.6.3 Improved NE CPS Environment

Based on the analysis from NE and E scenarios, the defender can apply the recommended set of controls to tighten the CPS security. If applied, the overall risk drops to 11.80%, and as seen in Fig. 3.4, the asset-related risk also drops, $I_{K_i} = 27.89$ for IoT Server 1, and $I_{K_j} = 18.20$ for Router1. It can also be observed from the graph that the risk levels for most of the assets are close to the minimum. If we compare the aforementioned scenarios, we see that the impact level and final risk reduces over time if the controls are tuned as shown in Fig. 3.5.

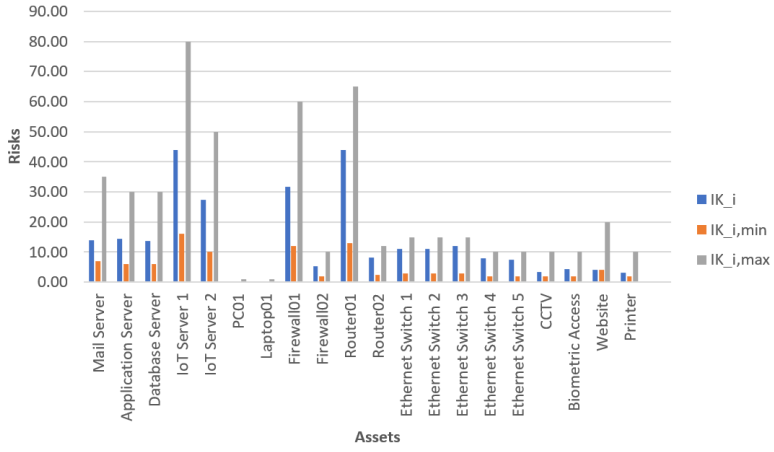

Figure 3.2: Not Exposed CPS: $I_{K_i}, I_{K_i,min}, I_{K_i,max}$

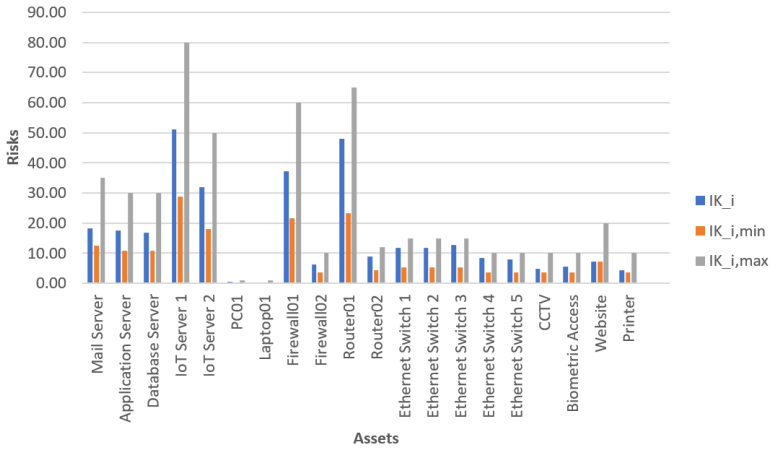

Figure 3.3: Exposed CPS: $I_{K_i}, I_{K_i,min}, I_{K_i,max}$

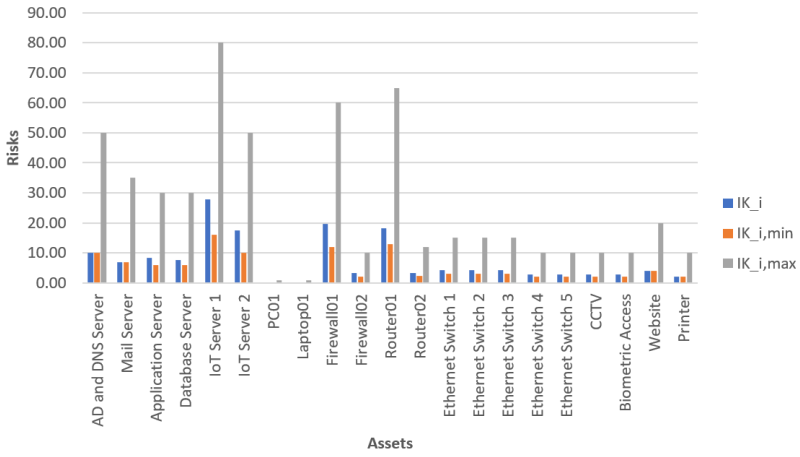

Figure 3.4: Improved Not Exposed CPS: $I_{K_i}, I_{K_i,min}, I_{K_i,max}$

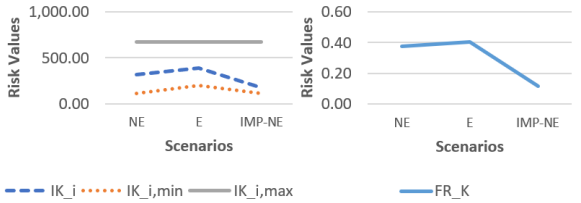

Figure 3.5: Overall security posture

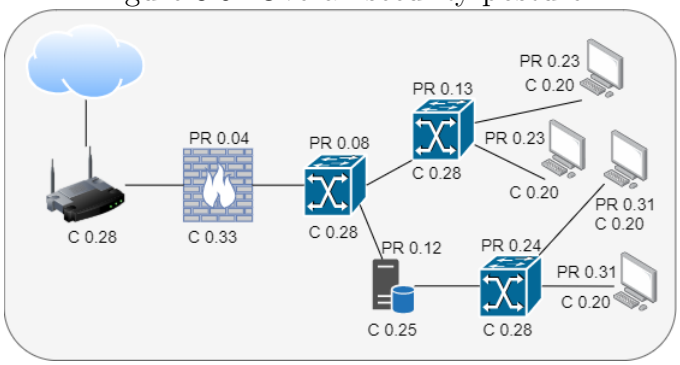

Figure 3.6: Risk propagation

3.6.4 Risk Propagation in the CPS Environment

Inter-dependency among digital assets introduces new kinds of risk and often induces firms to invest inefficiently in IT security. Our model is capable of demonstrating the risk propagation from parent nodes to child nodes and visa-versa. Fig. 3.6 shows a portion of the network with propagated risk values. The propagated risk values for this particular case

represent the risk of child node exploitation if the parent node is exploited.

3.7 Summary

CPS environments today are increasingly convoluted and highly competitive with a lot of uncertainties in terms of cybersecurity. It is imperative and quite challenging for a cyber defender to have full cyber situational awareness. We present a generic risk assessment model to address research questions Q1-Q3 under the objective A. The presented model provides an understanding of the current security posture of the given CPS, and the recommended set of controls to reduce the risk from adversarial attacks. We test and evaluate the proposed model for a blending IoT and traditional network, and show how the risk is reduced over time utilizing recommended controls. This work is published at the 29th International Conference on Computer Communications and Networks (ICCCN 2020). The current discussion of cyber risk assessment has excluded vulnerability assessment. Therefore, in the subsequent chapter, we will introduce our foundational vulnerability assessment framework, CyVIA.

Chapter 4

Cyber-threat and Vulnerability Information Analyzer (CyVIA)

Cyber infrastructures today are an amalgam of various technologies that create a wide spectrum of adversarial threats for a cyber defender to defend against. In such a scenario, a dynamic risk assessment framework, capable of assessing the protection offered by the employed security controls, and able to recognize the vulnerable loopholes within the broad range of installed products, is of the highest need. To address the research questions Q1 and Q1a under objective B of Chapter 1, Section 1.3, the following two main research questions were identified: *1) How can dynamic system components, such as applications, processes, memory utilization, and port usage, be tracked to effectively assess the security posture of cyber infrastructures, taking into account node/service heterogeneity? 2) How can information from external vulnerability sources be integrated to estimate security risks in real-time?*

We present and discuss our proposed framework CyVIA 1.0 [99] as seen in Fig. 4.1 in the following sections of this Chapter.

4.1 CyVIA System Architecture

In this chapter, we introduce the architecture of CyVIA, which encompasses the identification of security vulnerabilities, classification of risk types, and provision of product-wise and overall risk assessment for the given cyber infrastructure. We will discuss the various phases of the CyVIA architecture in the following sections.

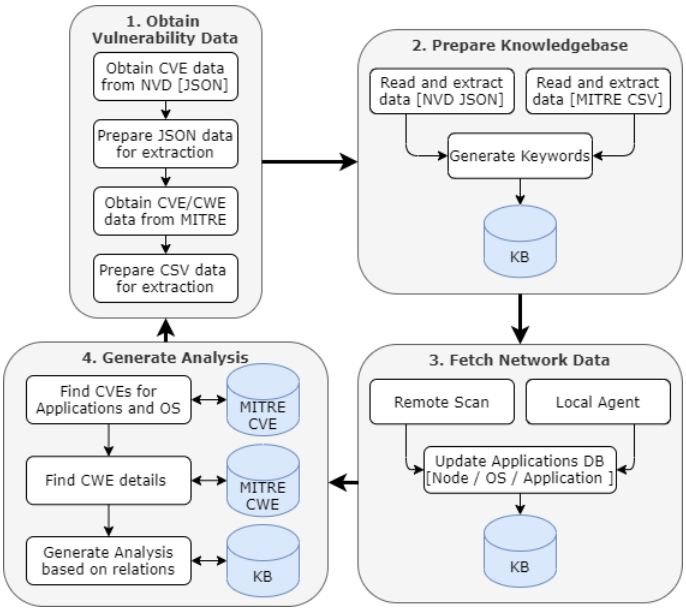

Figure 4.1: CyVIA Architecture (1.0)

4.1.1 Phase 1: Obtaining Vulnerability Data

NVD provides public access to 1) Vulnerabilities - CVE, 2) Products - CPE (Common Platform Enumerations), and 3) Checklists - NCP (National Checklist Program) data. CVE data is bundled in a JSON file for each year starting from 2002. MITRE, on the other hand, provides 1) CVE, 2) Common Weakness Enumeration (CWE), and 3) Common Attack Pattern Enumeration and Classification (CAPEC) data. As of May 2021, the NVD database contains 163,159, and the MITRE database contains 154,045 publicly known cybersecurity vulnerabilities. In our case, we initially focus on building a comprehensive knowledge-base that contains detailed information for each vulnerability. To do this, the framework initially obtains CVE information from NIST's NVD and CWE lists from MITRE. Irrespective of the data format provided by these two sources, CyVIA incorporates a generalized fetching procedure to access, read, and extract relevant information from all disparate sources. This information is further combined based on the relationships found among the data elements in Phase 2.

4.1.2 Phase 2: Preparing Knowledge-Base

From the datasets obtained in Phase 1, CyVIA extracts the following from the NVD data: CVE ID, Description, Lang, CWE ID, Severity, CVSS V2, CVSS V3, Access Vector, User Interaction, Publish Date, Modification Date, URLs, and Tags. The extracted data is then combined with CWE data, based on the found CWE ID relationships. For each CWE, we match and extract the CWE ID, Description, Platform, Consequences, Mitigation, and Affected Resources. Once the CVE and CWE data are combined, the keywords for each CVE entry are generated. These keywords can be of any attribute, such as Operating Systems, Software Names and Versions, and Port Numbers associated with the CVE entry. We use spaCy's rule-based matcher engine and regular expression for keyword extraction. Furthermore, for each CWE, parent, and children relationships are also mined from the MITRE website for relational analysis. The outcome of this phase is a comprehensive knowledge-base that is used in Phase 4 for finding vulnerabilities within the target network and for future machine learning developments. A knowledge-base entry for each CVE holds all of the above-mentioned attributes.

4.1.3 Phase 3: Fetching Network Data

The objective of this phase is to collect network node information. CyVIA is capable of discovering active network nodes through a remote server and detailed information of these nodes is captured by running a local agent on the discovered nodes. The objective of this phase is to collect network node information. This phase captures hostname, IP address, gateway, installed OS, and applications to generate node profiles using this information. These profiles are used in Phase 4 for vulnerability analysis. CyVIA can capture host information from any node (physical or virtual) as long as there is network connectivity, and each profile is ≈ 0.15MB.

4.1.4 Phase 4: Generating Analysis

This phase starts evaluating the network nodes based on the installed products (operating systems and installed applications). Upon receiving the product list for each node, CyVIA returns a list of vulnerabilities that may exist within the given products. This process is repeated for each product in the product list; keywords are matched with the knowledge-base and MITRE's CVE database for increased accuracy in the results. After retrieving a list of possible CVEs, we match the remaining features and relationships from the knowledge-base. As seen in Fig.4.1, the CWE database is also referenced for collecting the missing information and classification purposes. The generated vulnerability list for the target network is further classified based on severity and weakness types in this phase. CyVIA is capable of providing multiple analyses of the found vulnerabilities as discussed in Section 4.3. CyVIA's goal is not only to provide researchers, developers, system administrators, and cyber defenders the capability of interfacing external VDBs and evaluating network configurations for vulnerabilities but also to raise awareness of the inconsistencies within these VDBs.

4.2 Challenges, Limitations, and Advantages of CyVIA

Gathering vulnerability data is not straightforward because different VDBs provide different formats of data. Obtaining, processing, and integrating such data requires customized modules. In this section, we discuss the inconsistencies and challenges with obtaining and processing data from VDBs. We also discuss assumptions, limitations, and integration of the CyVIA framework.

4.2.1 Inconsistencies within the NVD and MITRE Data

Data in NVD and CVE databases is entered by humans and is stored in natural language plaintexts making it challenging for any automation tool to infer from the provided de-

scriptions rather a human expert is required to make a decision.

4.2.2 Assumptions, Limitations, and Integration of CyVIA

a) Assumptions: We have considered the following assumptions for CyVIA: 1) we assume that various CVE features, such as CVSS scores, CWE IDs, Severity values, etc., stored in the NVD are correctly assigned. 2) Because NVD is fed by MITRE data, and CWE is managed by MITRE, we take the final CWE features from MITRE. 3) The final list of possible vulnerabilities is matched with MITRE's CVE search engine.

b) Limitations: CyVIA at this point is limited to: 1) NVD and MITRE VDBs integration, however, JSON/CSV data can be imported from other VDBs. 2) Providing only the reported vulnerabilities within the operating systems and installed applications. 3) CVSS v2 for severity and score calculations since CVSS v3 scores are not available for all CVEs.

c) Integration Overview: In our previous work [98], we proposed a generic risk assessment model that assesses and quantifies the current security posture of any given cyber infrastructure based on the applied security controls. Security controls, however, are not capable of providing 100% protection; we need to consider the vulnerabilities that exist within the system configurations. CyVIA is an attempt to not only revamp our risk assessment framework, but also help other researchers, developers, system administrators, and cyber defenders to perform vulnerability analysis and assessment. CyVIA requires a product list as the input and returns the possible list of vulnerabilities along with other analyses as the output. This output is then fed to the risk assessment model to provide an enhanced security posture of the given network.

4.2.3 Advantages of using CyVIA

To conduct an effective cyber risk and vulnerability assessment, a cyber defender can either 1) use an open-sourced or a proprietary tool to gather vulnerability information for the network nodes, or 2) manually collect node configurations from the network and related

vulnerability information from the VDBs to derive further analysis. With option 1, the cyber defender is required to translate the tool-generated results into an acceptable form that can be plugged into the risk assessment framework being used. Option 2, on the other hand, is very laborious because network and vulnerability information is captured manually, and with constantly changing network configurations, repeating the entire process over and over again becomes burdensome.

CyVIA provides continuous risk monitoring by automating the entire process from data gathering to analyses generation. The entire process is repeated to capture the changing network configurations irrespective of time and space constraints. Depending on the risk assessment framework being used, a cyber defender can prioritize any of the produced results. CyVIA is capable of 1) finding vulnerabilities for each product, 2) classifying the found vulnerabilities based on weakness type, severity, and access vector, 3) spotlighting products based on the mean severity and CVSS scores, 4) pointing out the high priority vulnerabilities associated with the environment that the defender should be targeting, and 5) generating relational analysis between vulnerabilities, products, and weakness types. The aforementioned analyses for a real computing infrastructure are extensively discussed in Section 4.3.

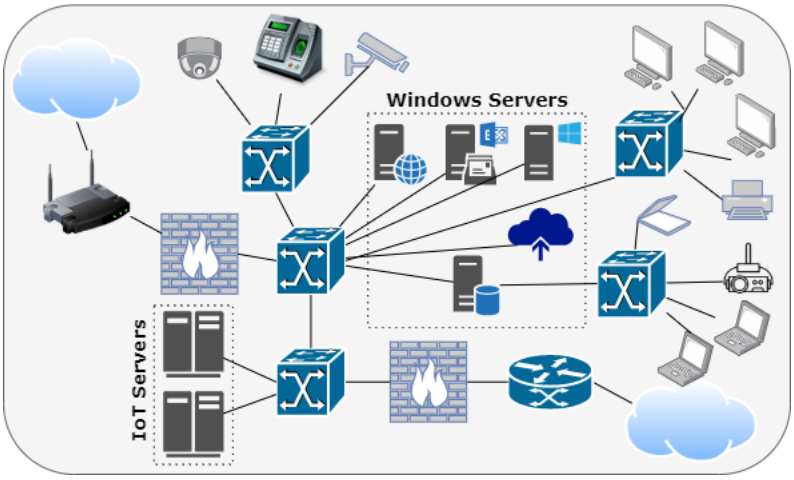

Figure 4.2: Target Cyber Infrastructure

4.3 Results

Acquiring contextual information to perform cyber risk assessment is a major challenge; CyVIA[1] is capable of generating this information, making the process much easier for cyber defenders. CyVIA knowledge-base contains CVE information from the year 1999 to the current year. To evaluate CyVIA, we used Oracle VM VirtualBox to mimic a virtual industrial network with a variety of hosts, applications, and services. Fig. 4.2 illustrates the target network, composed of heterogeneous components where each node contains a default set of installed applications. CyVIA is open source, screenshots and code are available on the repository.

4.3.1 Vulnerability Severity Groups

NVD provides three severity rankings of vulnerabilities for CVSS v2.0 (Low=0-3.9, Med=4-6.9, High=7-10) and five for CVSS v3.0 (None=0, Low=0.1-3.9, Med=4-6.9, High=7-8.9, Critical=9-10). CyVIA highlights the severity of vulnerabilities based on CVSS v2.0. As seen in Fig. 4.3, 42.35% of the vulnerabilities within the target infrastructure have High, 43.61% have Medium, and 14.03% have the severity level of Low out of 3,327 total CVEs found.

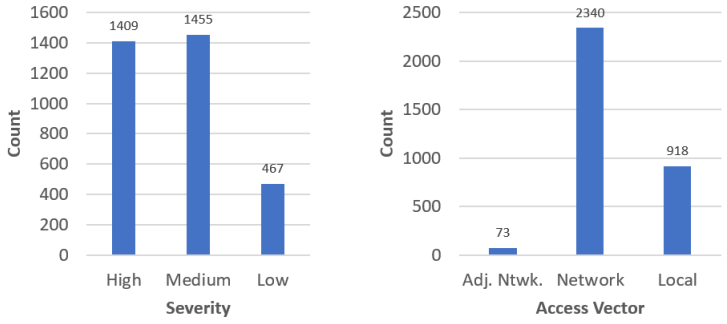

Figure 4.3: Severity and Access Vectors of found vulnerabilities

[1]https://github.com/trucyber/Risk-Assessment-Framework

4.3.2 Vulnerability Access Vectors

How vulnerable a network infrastructure is becoming more evident when a cyber defender is clear on how the discovered vulnerabilities can be exploited. For example, whether a vulnerability can be exploited with local or remote access. To avoid such infiltration, CyVIA is capable of imparting access vector information of the found vulnerabilities. As seen in Fig. 4.3, any adversarial entity can exploit 70.24% of the found vulnerabilities by gaining network access, 27.56% by gaining particular node access, and the remaining 2.19% by gaining adjacent network access.

4.3.3 Most and Least Vulnerable Products

Table 4.1 itemizes the master product list with each product's identification number, number of vulnerabilities, and weakness types found for each product. Among these products, Microsoft Windows 8.1 is the most vulnerable product with 691 vulnerabilities, and Ubuntu Core 16, FortiGate 2.8, and MongoDB 3.6 are the least vulnerable products with 2 vulnerabilities each. Fig. 5.8 illustrates the top 10 vulnerable products within this list.

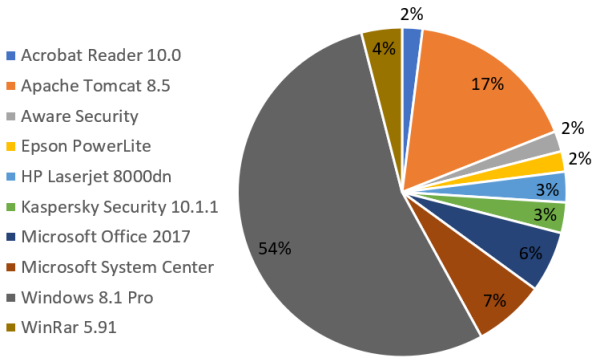

Figure 4.4: Top 10 vulnerable products

Table 4.1: Product list with number of CVEs and CWEs present

#	**Product**	**CVEs**	**CWEs**
1	Acrobat Reader 10.0	29	7
2	Alpine Linux 3.10	16	11
3	Apache Tomcat 8.5	213	41
4	Aware Security	22	12
5	Canon imageRUNNER 1643i	3	1
6	Cisco NX-OS 5.2	16	6
7	Epson PowerLite	26	15
8	FortiGate 2.8	2	2
9	HP LaserJet 8000dn	37	12
10	Kaspersky Security 10.1.1	42	16
11	Microsoft Office 2017	76	6
12	Microsoft System Center build 5.0.8412.1309	93	27
13	Microsoft Windows 7 Professional 6.1.7601	12	5
14	Microsoft Windows 8.1 Pro	691	39
15	MongoDB 3.6	3	2
16	Ubuntu Core 16	2	2
17	Windows Server 2008 build 6002	8	5
18	WinRAR 5.91	47	7
19	Wordpress 5.3	5	4
20	Zoneminder 1.30	6	4

4.3.4 Product severity observations

CyVIA attempts to identify the most vulnerable products within the network in many ways; one of them is to rate each product based on the mean severity score value of the observed vulnerabilities under each product. Table 4.2 lists the top 10 products based on

mean severity score. We can see that Fortigate 2.8 has the highest score value of 8.75 out of 10.0, and Canon imageRUNNER 1643i has the least mean score of 5.97. It is interesting to note that Windows 8.1 has the highest number of reported vulnerabilities (691) and is 9th on this list.

Table 4.2: Top 10 Products with high scores

Product	**Mean**	High
FortiGate 2.8	8.75	10.00
Acrobat Reader 10.0	8.53	9.30
Microsoft Windows 7 Professional 6.1.7601	7.92	9.30
Microsoft Office 2017	7.37	9.30
Windows Server 2008 build 6002	7.28	10.00
Cisco NX-OS 5.2	6.49	9.00
HP LaserJet 8000dn	6.28	10.00
Microsoft System Center build 5.0.8412.1309	6.28	10.00
Microsoft Windows 8.1 Pro 6.3.9600	6.21	10.00
Canon imageRUNNER 1643i	5.97	7.50

Table 4.3: Product-to-CVE and CVE-to-Product grouping

Key	Values
Wordpress 5.3	CVE-2019-20043,CVE-2019-20042,CVE-2019-16780..
FortiGate 2.8	CVE-2005-3058,CVE-2005-3057.
...	...
CVE-2017-8682	Office 2017, Windows 7 Pro, Windows 8.1 Pro.
CVE-2010-3227	Windows 7, 8.1, System Center, Server 2008
...	...

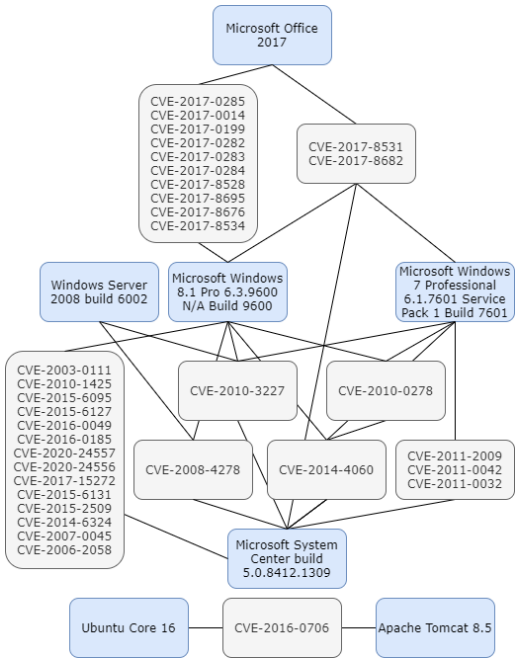

Figure 4.5: CVE to Product Relationships (Target Infrastructure)

4.3.5 Product, CVE, and CWE mapping

The ability to map relationships between products, vulnerabilities and weakness types in a particular context makes CyVIA more dynamic and robust. Table 4.3 shows a few examples of Product-to-CVE, and CVE-to-Product mapping (Fig. 4.5) where repeating vulnerabilities among products, and common products among weakness types are also mapped.

4.3.6 Top 10 Weakness Categories

Table 4.4 consists of a salient summary of the top 10 weakness categories existing within the target infrastructure. As we can see, CWE-119 (Buffer Overflow) is on the top of the list containing 19.7%, followed by CWE-200 (Information Disclosure) containing 18.1% of the found vulnerabilities. Among the top 10, only CWE-20 has a parent CWE i.e. CWE-707, all other CWEs do not have any parents. Table 4.4 also lists the affected products under each category, and we can see that CWE-119 is present among 70% of the products.

Table 4.4: Summary of Top 10 CWEs present in the targeted cyber infrastructure

CWE ID	**Names**	**Affected Products**	**%**
CWE-119	Buffer Overflow	1,2,3,4,6,7,9,10,11,12,13,14,17,18	19.7
CWE-200	Unauthorized Access	2,3,4,7,9,10,11,12,14	18.1
NVD-Other	Other	1,2,3,4,5,7,9,10,12,13,14,18	16.0
CWE-264	Permissions,Privileges & Access Controls	1,3,4,7,8,9,10,12,14,15,17,18	14.6
CWE-20	Improper Input Validation	1,2,3,4,6,9,10,11,12,13,14,18	11.2
NVD-noinfo	Insufficient Information	1,3,4,6,7,8,9,10,11,14,18	9.7
CWE-79	Cross-site Scripting	1,3,7,9,11,12,14,17,19,20	3.4
CWE-284	Improper Access Control	3,4,10,12,14	3.0
CWE-399	Resource Management Errors	2,3,4,6,10,14	2.6
CWE-22	Path Traversal	3,6,9,12,14,18,19	1.7

4.4 Summary

Traditional networks have evolved into more sophisticated infrastructures increasing the attack surface for adversaries. Defending against such events necessitates the identification of potential attack paths. Vulnerability assessment can help uncover areas that require immediate attention. To addresses the research questions Q1 and Q1a under the objective B, we present CyVIA framework in this Chapter, which effectively incorporates vulnerability information from major VDBs and prepares a comprehensive knowledge-base that is used further to provide continuous risk assessment of any cyber infrastructure. We discuss challenges in obtaining and processing this information and further evaluate the proposed framework on a real-world industrial network to find the most crucial vulnerabilities, computing products, and their relationships. This work is published at the 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom).

In the next chapter, we showcase the integration of the base risk and vulnerability assessment frameworks with additional enhancements, aiming to provide a comprehensive cyber situational awareness to cyber defenders. We will present the modifications and

improvements made to the CyVIA framework to further enhance its capabilities in assisting cyber defenders in effectively managing cyber risks.

Chapter 5

Towards Building Cyber Threat Intelligence (CyVIA 2.0)

The current chapter presents the CyVIA 2.0 architecture [100], which builds upon the base vulnerability assessment model (CyVIA 1.0) introduced in the previous chapter. Unlike the previous model, which followed a step-by-step process, CyVIA 2.0 provides a continuous real-time evaluation of the target network by dynamically integrating different components that interact with each other to create an effective cyber threat intelligence system. CyVIA 2.0 comprises four main modules. The first module is a vulnerability database wrapper module that acquires the latest vulnerability information from external sources in a timely manner. The second module is a knowledge-base generation module that keeps the CyVIA knowledge-base current. The third module is an environmental data collection module that continuously tracks infrastructural changes. Finally, the fourth module is a threat modeling and risk analysis module that prepares various analytics for cyber defenders based on the identified anomalies.

To address the research question Q2 under the objective B, as discussed in Chapter 1, Section 1.3, which is *"How can we evaluate cyber risk considering adversarial capabilities, vulnerabilities, network dependencies, administrative policies, and provide mitigation techniques to produce cyber threat intelligence?"*, we present CyVIA 2.0. CyVIA 2.0 inputs data from three sources: 1) multiple VDBs, 2) network nodes (configurations, services, running processes, open ports, and so on), and 3) the security policies keeping the network nodes secure on the network such as the applied security controls and other administrative policies. CyVIA 2.0 produces two types of output: 1) dynamic informed analysis of chang-

ing network configurations and vulnerabilities, and 2) comprehensive analysis of network infrastructure based on the applied security controls and discovered vulnerabilities. In the following sections of this Chapter, we first go over each individual component and then describe different phases from the CyVIA 2.0 architecture as seen in Figure. 5.1.

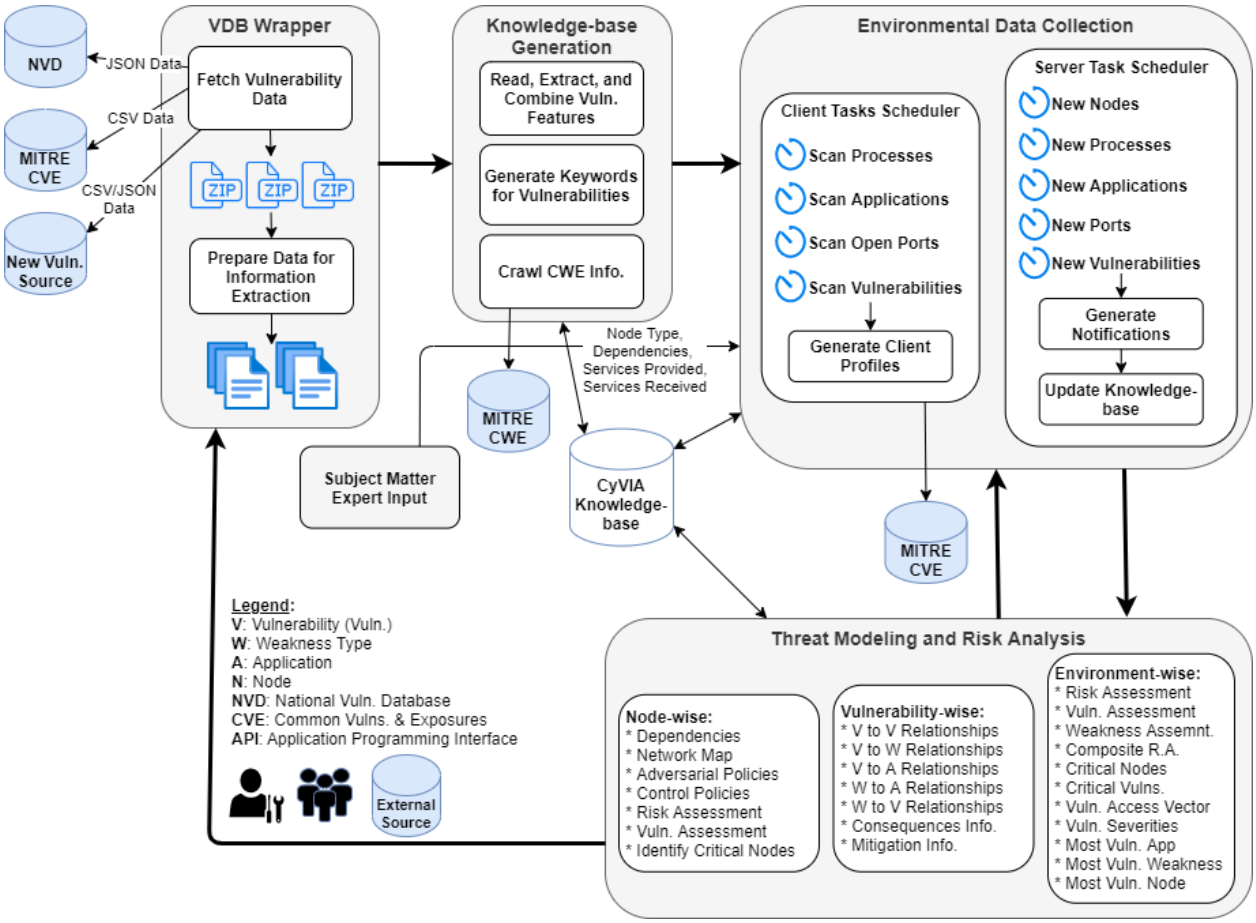

Figure 5.1: CyVIA Architecture (2.0)

5.1 Vulnerability Database (VDB) Wrapper

CyVIA is capable of collecting vulnerability data from multiple sources and multiple formats. At present, we collect data from NVD and MITRE, however, CyVIA is capable of integrating data from other sources. As of October 2021, the NVD database contains

172,427 publicly known vulnerability reports. These reports are bundled together in yearly JSON compressed files starting from the year 2001 to date. MITRE on the other hand provides vulnerability groups by weakness types and other attributes such as weakness type description, applicable platforms, modes of introduction, and more in a CSV file format. During this phase, CyVIA collects the multi-formatted datasets from NVD and MITRE and prepares data for extraction during the next phase.

5.2 Knowledge-Base Generation

This phase is responsible for generating a knowledge-base from the collected datasets. This knowledge-base is used by all other components of CyVIA. During this phase, each report item is analyzed and categorized, vulnerability features are extracted, and keywords for each vulnerability are generated. Various information pieces are combined into a comprehensive knowledge-base based on the found relationships in the data points, irrespective of the different data formats. This phase also crawls additional related information from the MITRE website such as parent and child relationships among weakness types. Once the dataset is prepared, the environmental data is collected during the next phase.

5.3 Environmental Data Collection

In this phase, the computing environment or digital assets information is collected. This process has two sub-components (schedulers), a server component that runs on any of the administrator servers, and a client component that runs on all clients. The client and server scheduler communicate and exchange information with each other. The components and sub-components of this phase are discussed in detail as follows:

5.3.1 Schedulers

Providing up-to-date analysis strictly depends on the following factors: 1) how updated the obtained vulnerability information is, and, 2) how updated the network node profiles are. To ensure the up-to-date analysis, CyVIA integrates a scheduler module that has two sub-components:

(i) **Client Task Scheduler**: For command and control, adversaries employ a variety of tactics and protocols after a successful attack to maintain persistence within the target environment. In such cases, most of the related processes execute in the background without user awareness. CyVIA monitors running processes in real-time to alert administrators of any newly detected processes on any of the network nodes. The recorded information for each process includes but is not limited to process id, executing file path, process owner, number of threads, CPU, memory used by the process, etc. Similarly, processes using high memory and CPU are also highlighted during this process for the administrators to take necessary actions if required. Furthermore, any newly installed application, open port, or a vulnerability associated with any of the installed applications is also reported. A client-side scheduler is responsible to keep track of processes, applications, open ports, and vulnerabilities to ensure updated client/node profiles and informed administrator.

(ii) **Server Task Scheduler**: The server-side scheduler captures the changes in information between the server and clients, validates the information, and generates notifications for the administrator about the newly discovered nodes on the network, processes, applications, ports, and vulnerabilities on the network nodes. The server-side scheduler is also responsible to keep the knowledge-base up to date with the latest vulnerability information.

5.3.2 Node Profiling

Any cyber threat intelligence system must collect environmental data specific to the computing environment in order to generate contextual analysis. CyVIA can not only capture changing network configurations on the go, but it can also notify administrators of the changes so that they can take appropriate actions where needed. With the help of a remote agent, CyVIA initially captures the active nodes on the network and their associated information. And, with a local agent running on the detected nodes, this information is refined even further. This process captures and generates node profiles and the IT administrators can fine-tune the profiles as needed. Based on the acquired node information, a node profile contains information such as hostname, IP address, gateway, installed OS, installed apps, open ports, and running processes.

5.4 Subject Matter Expert Input

The CyVIA framework provides flexibility to the subject matter experts or administrators to modify various elements and override the default security setup recommended for each type of network node, as and when required. For example, assigning security controls and adversarial risks to nodes on the network, changing the control and adversary weights, overriding the final risk values to get more realistic scores. Once the node profiles are generated, the administrator can define the following information:

(i) **Asset Type**: whether the node is a computer (server, workstation, etc.), a network device (firewall, router, etc.), etc.

(ii) **Control Policy**: states the defensive mechanisms or controls such as technical, physical, or administrative, that are applied on the current node.

(iii) **Adversarial Policy**: defines which types of adversarial risks are applicable on this particular node.

(iv) **Services Provided**: lists the number of services offered by the current node to other nodes on the network.

(v) **Services Received**: if the current node is receiving any services from other nodes on the network, it must be recorded in the node profile.

In the next Section, we discuss controls and policies in detail.

5.5 Control and Adversary Mapping

To protect digital assets and mitigate associated risk factors, cyber defenders deploy several cutting-edge security controls. It is critical to consider these controls while performing cyber risk analysis. CyVIA keeps a record of detailed control information such as control type, assigned weight for each control, a recommended set of controls for different types of network devices, and the administrator-defined control set for a particular type of digital asset. Similarly, different types of adversaries (internal and external) can be defined and assigned weights based on their assumed capabilities. These information pieces are maintained under the control master, and the various attributes of the control master are as follows:

5.5.1 Control and Adversary Definition

Control definition document contains the master list of available security controls that can be used to secure digital assets. At present, we classify these controls into three main types. 1) Technical Controls ($T = \{T_1, T_2, \ldots, T_8\}$), where T_1=Strong Authentication, T_2=Antivirus/Patches/Updates, ..., T_8=Encryption. 2) Physical Controls ($P = \{P_1, P_2, \ldots, P_6\}$), where P_1=Video Surveillance, P_2=Locks, ..., P_6=Man-traps, and 3) Administrative Controls ($A = \{A_1, A_2, A_3, A_4\}$), where A_1=Security Policy, A_2=Security Training, A_3=Data classification, A_4=NDA Signing [98]. This document is used to specify the control set for each node on the network, representing administrator efforts for securing network nodes or digital assets. And the purpose of the adversary definition document is

to define the types of adversaries that the organizational assets are exposed to. At the moment we have four types of adversarial actors: internal employees, and external adversaries with novice, intermediate, and expert expertise. Both of these documents can be expanded as per the organizational needs.

5.5.2 Control and Adversary Weights

Each of the defined controls is assigned a weight value and since the control application varies from asset to asset, we further introduce control application categories M (must have), G (good to have), O (optional) for different types of digital assets. Similarly, the level of protection provided by these controls will vary if the applied controls are exposed to adversarial entities. We assign two different types of weights, 1) NE (not exposed): when the controls are not exposed to the adversarial entities, and 2) E (exposed): when the adversaries are aware of what controls are applied to protect organizational assets. These weights are used to calculate the level of protection that can be expected by the applied controls.

Similarly, the threat posed by humans or adversarial entities is determined by the threat actor's level of access and skill set and it is critical to categorize individuals based on their competence and access location. An inside employee with a given level of access, for example, may pose a different risk than an external experienced attacker. Similar to controls, we categorize adversaries and assign weights based on their skill-set and location.

5.5.3 Master and User-Defined Policies

Master policy document contains the ideal or recommended control configurations for different types of devices on the network. The controls are categorized further into three more categories M, G, and O as explained earlier. Network devices are categorized into seven different types: 1) Servers: server computers providing services to other nodes on the network, 2) Workstations: client computers receiving services from servers, 3) Portables:

portable devices such as laptops, tablets, etc., 4) Network: networking equipment such as routers, switches, access points, etc., 5) Network Security: firewall, IPS/IDS, etc.), 6) Storage: USB, Optical Disk, SAN, NAS, etc., 7) IoT: any device connecting to the network not classified in above categories.

For each type of device, the master policy holds a recommended M, G, O control that determines how secure the node is in terms of control security. For example, a server device must have the controls T1-T3, whereas T4 is good to have: "Server": ["T1:M", "T2:M", "T3:M", "T4:G", ...]. Each node profile specifies whether these recommended controls are applied or not. For example when T1-T3 are applied and T4 not applied: "ControlPolicy": ["T1:1", "T2:1", "T3:1", "T4:0", ...]. Similar to control mapping, adversarial threats are also mapped within node profiles for each node. If a particular control or threat is applied or applicable to a node, it will be represented by the value 1, otherwise by 0 stating that the control or threat is not applied or applicable. For example, a CCTV control and an external adversarial threat may not be applicable for a standalone scanner.

Ideally, each device under the same device category should have the same controls applied as per the defined control policy, however, it can change as per the network administrator's approval. CyVIA allows the administrators to have custom user-defined policies as per their needs. Another use case for this scenario is the third-party devices with limited access rights and policy options such as a DVR for CCTV recording. Administrators can further secure these devices by employing custom physical (locks) or administrative controls (policies).

5.6 Threat Modeling and Risk Analysis

This phase is mainly responsible for generating contextual analyses for the computing environment being analyzed.

5.6.1 Interdependency Between Nodes - Service Mapping

Dependencies between network nodes present a different set of challenges for a cyber defender. Because risk scores are usually centered on network/infrastructure, we add the dependency factor for nodes, which represents the number of service dependents for a node [98]. The higher the number of dependents, the more important the node is in the network. CyVIA is capable of generating the network map of the given infrastructure as well as service dependencies. The recorded information under each node's profiles is used to map the services that node K_i delivers to node K_j on the network. CyVIA's dependency map illustrates the service dependencies between network nodes and aids the administrator in identifying crucial network nodes. We keep track of services provided (service:port) and services received (IP:port) by every node on the network.

5.6.2 Severity of Nodes

How critical a node on the network is, can be determined by what risk the network node is introducing to the infrastructure. In our case, we consider the following factors while calculating risk scores:

(i) **Control-Based Risk**: This risk informs the administrator about what amount of protection should be expected from the applied security controls in light of adversarial threats.

(ii) **CVSS-Based or Vulnerability-Based Risk**: How vulnerable each node on the network and the overall infrastructure is seeing the discovered vulnerabilities.

By aggregating both scores, we can label the most critical nodes on the network that require urgent attention from the administrator to improve the general welfare of the network. Furthermore, the critical nodes can also be identified by analyzing the number of open ports vs actual dependents.

5.6.3 Potential Consequences and Mitigation

Once the vulnerabilities within the specific infrastructure have been identified, CyVIA can educate the administrator about the potential consequences of the discovered vulnerabilities as well as mitigation strategies that may be utilized to prevent such exploitation. For example, vulnerabilities under the category CWE-5, i.e. "J2EE Misconfiguration: Data Transmission Without Encryption" target the "Integrity" metric and are capable of modifying the application data. Using SSL or encryption for all access-controlled sites is a mitigation strategy that can be utilized to avoid such exploitation.

5.7 Assumptions, Limitations, and Integration Overview

Assumptions

We have considered the following assumptions for CyVIA 2.0: 1) we assume that various CVE features, such as CVSS scores, CWE IDs, Severity values, etc., stored in the NVD are correctly assigned. 2) Because NVD is fed by MITRE data, and CWE is managed by MITRE, we take the final CWE features from MITRE. 3) The final list of possible vulnerabilities is matched with MITRE's CVE search engine. 4) We use a raspberry pi as a device on the perception layer that represents IoT devices and communicates with different sensors for data collection. 5) Due to limited resources, we are unable to deploy CyVIA 2.0 on a live large network, however, we have conducted several trials of CyVIA 2.0 on various network clusters containing different versions of Microsoft Windows and Linux, and we are confident that it can be deployed on any large network.

Limitations

CyVIA at this point is limited to: 1) Local agent that can capture information from nodes running Windows 7 onward, having power-shell script execution enabled. And for Linux, we have tested agents on Ubuntu, Kali, Debian, and Fedora. 2) Services offered by nodes are

captured through the remote scan, however, the nodes utilizing these services are identified by the administrator.

Integration Overview

A cyber defender present within the target network is capable of interacting with all components of CyVIA whereas limited interaction with different components is available from outside the network using the API.

5.8 Results

We evaluate CyVIA on a large VM setup having different clusters of nodes, representing different parts of the network. Nodes are mapped and evaluated during this process. Table 5.1 lists the subset cluster being evaluated in this Section, its nodes, their IP addresses, and the installed OS. All nodes have a default set of applications installed and a few custom applications such as MySQL, SQL Server, etc. to create dependencies between nodes. The node cluster includes nodes from each layer as seen in Figure 5.2. We selected three state-of-the-art vulnerability scanning tools, Nessus Essentials by Tenable, InsightVM by Rapid7, and Greenbone Security Manager (GSM) by Greenbone, and scanned the network using these tools. We also scanned the network using CyVIA.

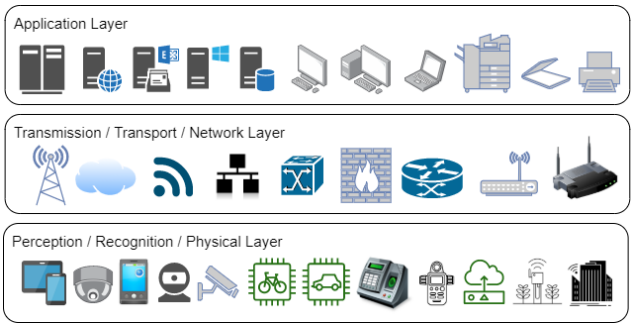

Figure 5.2: Layers

In the following subsections, we initially discuss the findings by CyVIA and then for

Table 5.1: Network Node List

Node	**IP**	**OS**
Win7	50.50.50.4	Windows 7 ENT
Win81	50.50.50.5	Windows 81 ENT
Win10	50.50.50.6	Windows 10 ENT
Windows11	50.50.50.7	Windows 11 Pro
Server2012	50.50.50.8	Server 2012 R2
Server2016	50.50.50.9	Server 2016 Datacenter
Centos	50.50.50.23	Centos 8.3.1
Debian	50.50.50.24	Debian 10
Fedora	50.50.50.25	Fedora 33
OpenSUSE	50.50.50.26	OpenSUSE 15.2 1
Raspbian	50.50.50.27	Raspbian
Ubuntu16	50.50.50.28	Ubuntu 16 LTS
Ubuntu18	50.50.50.29	Ubuntu 18 LTS
Ubuntu20	50.50.50.30	Ubuntu 20 LTS

each tool followed by a comparison between the four. Please note that we only provided the node IPs and OS credentials to each tool for scanning and kept everything else as default. Each tool was installed on a fresh virtual machine with no other application installed or running, and assigned 8GB of RAM and 2 threads of Intel i7 processor.

5.8.1 Analysis by CyVIA

CyVIA is capable of generating contextual information based on the network nodes, applied security controls and policies on these nodes, and the found vulnerabilities within the installed OS and applications on these nodes. Therefore, the execution process is slightly different as compared with other tools. In the following subsections, we discuss the major

components, their execution, and responsibilities.

a) Node Profiling

CyVIA is capable of detecting network nodes using the scheduler module. Once a node is detected, CyVIA tries to obtain node information remotely using a profiling agent. Based on the information captured in this process, further analyses are generated, therefore, it is critical to verify and update each node profile to have the most accurate results. The scheduler module has two sub-components, a client-side scheduler, and a server-side scheduler, responsible for evaluating the changes in node profiles. These schedulers work closely with the profiling agents. A server-side profiling agent captures node profiles remotely, and a client-side profiling agent runs on each client.

(i) **Server Side Scheduler**: CyVIA keeps track of changes by closely monitoring the recorded node profiles and any new observed changes on the network. For example, any newly discovered node(s), process(es), application(s), or vulnerabilities are highlighted in this process. The server-side scheduler relies more on the recorded information and the remote profiling agent. The following output sample shows the server-side scheduler execution where a network id is required to start monitoring the specified network. The recorded information is displayed for each node and in case of any change, it is highlighted for consideration. The server-side scheduler schedules tasks to run after every few minutes to keep track of changes.

```
Please provide network id:  50.50.50.0
Server scheduler started at 21:21:19
21:21:19 Fetching existing data...
Win10:  [Processes:  55 , Users:  3 , Apps:  5 , Open Ports:  19 , Vulner-
abilities:  227]
...  more ...
Ubuntu20:  [Processes:  187 , Users:  12 , Apps:  9 , Open Ports:  2 , Vul-
```

```
nerabilities:  430]
** Starting network scanner at 21:22:19
Found 10 alive hosts.  Newly discovered node(s) 2
** New host(s):  ['50.50.50.90', '50.50.50.99']
21:26:20 Looking for changes in node processes, applications, and ports...
** 4 New process(es) found **
Win10:  ['SystemSettingsBroker.exe', 'sppsvc.exe', 'SppExtComObj.Exe', 'Ap-
plicationFrameHost.exe']
** 1 New application(s) found **
Win10:  ['Free Cam 8']
No new open ports found.
No new vulnerabilities found.
...  more ...
```

We can see that 2 new nodes on the network are found, and 4 new processes with 1 new application on the Win10 node are detected and prompted in the above sample.

(ii) **Remote Profiling Agent**: CyVIA initially detects network nodes remotely and tries to obtain individual node information using a remote profiling agent as shown previously in the output sample. During this process, not necessarily all nodes are discovered depending on the security settings on each node. The undiscovered node(s) information is further captured with the help of the local profiling agent discussed next. This process took $\approx$ 10 minutes in our case of 14 nodes network. The information captured is stored and the sample output is as follows:

```
Please provide router IP / Network ID: 50.50.50.1
Scanning network please wait...
Found host:  50.50.50.1
Found host:  50.50.50.5
```

```
...  more ...
Total alive hosts:  9
Scanning hosts, please wait...
Collecting information for the IP 50.50.50.1
Host:  50.50.50.1, State:  up
OS Vendor:  Linux, OS: Linux, OS Ver:  2.6.X, OS Type:  general purpose, Ac-
curacy percent:  100.
Running protocol(s) :  tcp
port :  22 state :  open
port :  80 state :  open
port :  443 state :  open
...  more ...
```

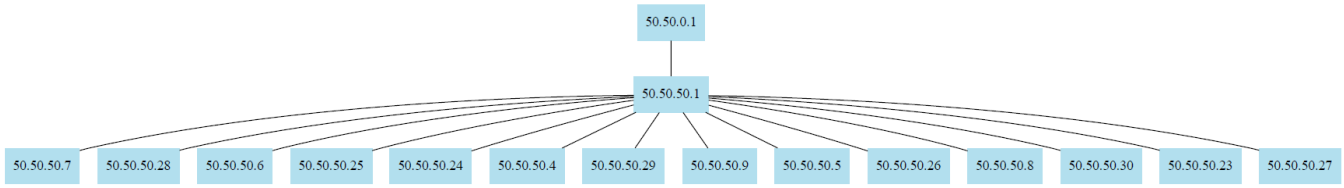

Figure 5.3: CyVIA Network Map

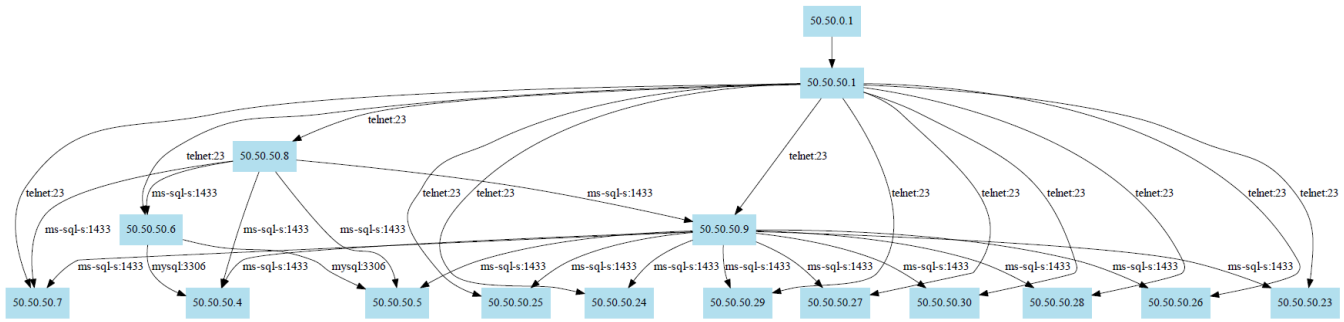

Figure 5.4: CyVIA Dependency Map

(iii) **Client Side Scheduler**: Client Scheduler is responsible for monitoring any new process, application, or vulnerability on the client-side. The discovered items are

reported to the server scheduler for further action. The client-side scheduler also schedules tasks to run after every few minutes to keep track of changes. The sample output is shown below.

```
Client scheduler started at 20:21:03
** Starting process scanner at 20:22:03
HostName:  Win10 HostIP: 192.168.0.199
Running Processes:  55 Previously recorded:  55
Finding new processes if any...
## New application ApplicationFrameHost.exe found (not recorded previously)
with 55 processes:
."ApplicationFrameHost.exe":  [
...  "pid":  3796,
...  "exe":  "C:/Windows/System32/ApplicationFrameHost.exe",
...  "username":  "WIN10/IEUser",
...  "num_threads":  1,
...  "cpu_percent":  0.0,
...  "memory_percent":  0.5453594831106481,
...  "cpu_times":  [
......  0.125,
......  0.21875,
......  0.0,
......  0.0
...  ]
.  ]
...  more ...
** New PID 2100 under MsMpEng.exe
** New PID 4996 under NisSrv.exe
```

```
** New PID 1444 under OneDrive.exe
Found 1 new application(s) and 3 new process(es) among the 55 currently run-
ning.
Recording newly discovered process...  Done.
```

(iv) **Local Profiling Agent**: With the help of the local administrator, a local agent can be deployed and executed on each node on the network that captures the remaining pieces of information required to complete the node profiles. This process takes ≈ 1 minute on each node and ≈ 14 minutes for the entire network. The administrator can verify the captured information and fine-tune node profiles as discussed in Sections 5.3.2 and 5.4.

b) Interdependency Between Nodes - Service Mapping

Complete node profiles allow CyVIA to generate network and dependency diagrams as seen in Figure. 7.3 and Figure. 7.4. This allows the administrator to understand the network hierarchy and service load on each node. We can see that node 50.50.50.1 and 50.50.50.9 has a higher number of service dependants as compared with other nodes on the network. This process takes a few seconds to generate the analysis.

c) The severity of Nodes

Nodes can be flagged critical in several ways, as discussed above, a node with the highest number of dependents is also critical for the network. Two major risk categories used by CyVIA to flag critical nodes are as follows:

(i) **Control-Based Risk**: During this process, CyVIA at first ensures that the control documents exist and respective weights are assigned to each of the categories. After this, each node is analyzed in terms of control security based on the control and adversarial policy applied to each node. The following output sample shows an analysis of three different cases during this process.

Table 5.2: CyVIA Node and Infrastructure-based Control Risk

IP	Node	M_Av	M_Ap	G_Av	G_Ap	O_Av	O_Ap	Deps.	NR	CR
50.50.50.1	Router1	12	12	2	2	2	2	13	0.00	0.00
50.50.50.9	Win2016	12	12	3	3	2	2	12	0.00	0.00
50.50.50.8	Win2012	12	5	3	2	2	1	6	0.51	0.11
50.50.50.6	Win10	7	3	2	2	7	3	3	0.43	0.10
50.50.50.27	Raspbian	7	0	2	0	7	0	1	1.00	0.22
50.50.50.5	Win8	7	4	2	0	7	6	1	0.54	0.12
50.50.50.24	Debian10	7	2	2	2	7	4	1	0.51	0.11
50.50.50.4	Win7	7	4	2	1	7	5	1	0.44	0.10
50.50.50.29	Ubuntu18	7	6	2	0	7	5	1	0.37	0.08
50.50.50.25	Fedora33	7	7	2	0	7	4	1	0.29	0.06
50.50.50.23	Centos	7	6	2	1	7	7	1	0.21	0.04
50.50.50.30	Ubuntu20	7	5	2	2	7	6	1	0.20	0.04
50.50.50.7	Win11	7	7	2	2	7	7	1	0.00	0.00
50.50.50.28	Ubuntu16	7	7	2	2	7	7	1	0.00	0.00
50.50.50.26	openSUSE	7	7	2	2	7	7	1	0.00	0.00

```
Host:  Win11, IP: 50.50.50.7
Must have controls:  7 applied, out of 7
Good to have controls:  2 applied, out of 2
Optional controls:  7 applied, out of 7
All recommended controls applied
Host:  Win10, IP: 50.50.50.6
Must have controls:  3 applied, out of 7
Good to have controls:  2 applied, out of 2
Optional controls:  3 applied, out of 7
Recommended controls not applied:  ['T2:M -> T2:0', 'T6:M -> T6:0', 'A2:M
-> A2:0', 'A4:M -> A4:0']
Matched controls:  ['T1:M', 'T3:G', 'T4:O', 'T5:O', 'T7:G', 'T8:O', 'P1:O',
'P2:O', 'P3:O', 'P4:O', 'P5:N', 'P6:N', 'A1:M', 'A3:M']
Host:  Raspbian, IP: 50.50.50.27
Must have controls:  0 applied, out of 7
```

```
Good to have controls:  0 applied, out of 2
Optional controls:  0 applied, out of 7
Recommended controls not applied:  ['T1:M -> T1:0', 'T2:M -> T2:0', 'T3:G
-> T3:0', 'T6:M -> T6:0', 'T7:G -> T7:0', 'A1:M -> A1:0', 'A2:M -> A2:0',
'A3:M -> A3:0', 'A4:M -> A4:0'] Matched controls:  ['T4:O', 'T5:O', 'T8:O',
'P1:O', 'P2:O', 'P3:O', 'P4:O', 'P5:N', 'P6:N']
...  more nodes ...
```

Table 5.3: CyVIA Infrastructure-based Risk Summary

IP	**Node**	**O.P.**	**Apps.**	**High**	**Med.**	**Low**	**Total**	**CR**	**VR**	**TR**
50.50.50.27	Raspbian	2	1,824	4,070	6,326	909	11,305	0.22	0.30	0.26
50.50.50.8	Win2012	39	43	4,780	2,957	888	8,625	0.11	0.23	0.17
50.50.50.9	Win2016	17	42	6,347	3,657	588	10,592	0.00	0.28	0.14
50.50.50.24	Debian10	1	1,618	965	2,043	423	3,431	0.11	0.09	0.10
50.50.50.5	Win8	15	25	572	570	253	1,395	0.12	0.04	0.08
50.50.50.6	Win10	19	30	329	679	92	1,100	0.10	0.03	0.06
50.50.50.4	Win7	10	23	113	56	6	175	0.10	0.00	0.05
50.50.50.29	Ubuntu18	4	16	106	182	28	316	0.08	0.01	0.04
50.50.50.25	Fedora33	2	1,740	3	9	4	16	0.06	0.00	0.03
50.50.50.30	Ubuntu20	2	12	146	239	47	432	0.04	0.01	0.02
50.50.50.23	Centos	2	1,403	6	3	0	9	0.05	0.00	0.02
50.50.50.7	Win11	17	14	2	7	9	18	0.00	0.00	0.00
50.50.50.28	Ubuntu16	2	15	86	162	21	269	0.00	0.01	0.00
50.50.50.1	Router1	1	1	8	10	9	27	0.00	0.00	0.00
50.50.50.26	openSUSE	5	2,320	25	23	3	51	0.00	0.00	0.00

In the above example, workstation 50.50.50.7 has all controls applied, workstation 50.50.50.6 is missing 4 must have controls and 4 optional controls, and workstation

50.50.50.27 has no controls applied. Must have controls are highlighted whereas the optional controls are ignored because they are optional. Table 5.2 lists the network nodes, applied controls, number of dependents, associated node-based, and infrastructure-based control (CR). We can see that node 50.50.50.27 (Raspbian) has no security control applied (M_Ap, G_Ap, O_Ap) and it is at a high risk of 100% (NR), followed by workstation 50.50.50.5 (Win8) at 54%. We can also see that nodes 50.50.50.1 and 50.50.50.9 have the highest number of dependents (Deps.). Please note that nodes with risk 0 do not mean they are 100% secure. This process also takes a few seconds to execute.

(ii) **Vulnerability-Based Risk**: CyVIA flags nodes based on the number of vulnerabilities found in each. There may be a case where on one hand a node has a higher number of reported vulnerabilities most medium or low severities. And on the other hand, a node with a high number of high severity vulnerabilities. CyVIA is not only capable of highlighting both cases, but also the applications with the highest numbers of reported vulnerabilities and their classifications. Table 5.3 provides a summary of node-based vulnerabilities (Total), the number of applications (Apps.), open ports (O.P.), control-risk (CR), vulnerability-risk (VR), and the aggregated risk (TR). We can see that node 50.50.50.27 (Raspbian) has the highest number of vulnerabilities (30%), highest control risk (22%), and the highest risk portion within the infrastructure (26%). This process takes $\approx$ 1 minute, and depending on the number of applications installed on a node it can take up to 4 minutes. For our network, it took $\approx$ 20 minutes to complete the analysis.

d) Additional Analysis

CyVIA produces various analyses that play a significant role in securing the cyber infrastructure. Table 7.5 provides information about the top 10 most vulnerable products with the highest number of vulnerabilities and their associated weakness types found by CyVIA.

Table 5.4: CyVIA Infrastructure-based Top 10 Most Vulnerable Products

S#	Product	CVEs	CWEs
1	Microsoft MPI ...	6,377	97
2	jackd 5+nmu1	3,070	87
3	chromium 90.0.4430...	1,468	59
4	Windows 8.1 Enterprise	1,107	62
5	Windows Server 2012 R2	949	42
6	ssh 1:7.9p1-10	748	73
7	SQL Server 2017	640	37
8	zip 3.0-11+b1	584	54
9	SQL Server 2017	516	14
10	SQL Server 2017	516	14

Table 5.5 on the other hand provides information on which product has the highest observed mean, max, and mode scores. Although Microsoft MPI has the highest number of reported vulnerabilities (6,377), however, simple-scan has the highest vulnerability scores, meaning it is more vulnerable as compared with Microsoft MPI. Furthermore, Table 5.6 spotlights the top 10 weakness types, their percentage and count. For example, 12.20% vulnerabilities fall under SQL injection type and 11.15% are related to buffer overflow.

Figure 5.5 provides information on the open ports found on each node versus the actual number of dependents. For example, node Win2012 has 39 ports open whereas the actual number of dependents is only 6. This raises a red flag for the administrator. Figure 5.6 illustrates an overview of control and vulnerability risk. Node Raspbian has the highest control and vulnerability risk as compared with all other nodes, whereas nodes Win11, Router1, and OpenSUSE15 have very low risks. Figure 5.7 provides the percentage of vulnerability severities and access vector. Among the found vulnerabilities, 46.5% are high severity and 83.5% can be exploited through network access. Table 5.7 provides further statistics related to the found vulnerability severities. We can observe a low standard

Table 5.5: CyVIA Infrastructure-based Top 10 Mean, Max, and Mode Scores

Product	**Mean**	**Max**	**Mode**
simple-scan 3.30.1.1-1+b1	10.0	6.40	10.0
gpicview 0.2.5-2+b1	10.0	6.39	10.0
tcl8.6 8.6.9+dfsg-2	10.0	10.00	10.0
lp-solve 5.5.0.15-4+b1	10.0	6.40	10.0
SolarWinds Collector	10.0	10.00	10.0
mscompress 0.4-3+b1	10.0	6.40	10.0
eog 3.28.4-2+b1	10.0	6.38	10.0
enchant 1.6.0-11.1+b1	10.0	6.42	10.0
user-setup 1.81	10.0	7.03	10.0
whiptail 0.52.20-8	10.0	10.00	10.0

deviation for the high severity vulnerabilities meaning most high severity vulnerabilities are closer to the mean value i.e. 8.29, which can also be noticed by the percentile values. Figure 5.8 highlights the top 10 CVEs found among the current network nodes. Similarly, CyVIA is capable of highlighting common CVEs across different products or the vulnerabilities that are present in multiple products. This is very helpful for generating relational analysis.

```
CVE-2010-1444:  ['vlc 3.0...', 'zip 3.0..']
CVE-2018-6559:  ['Ubuntu16...', 'Ubuntu18...', 'Ubuntu20...']
CVE-2015-0095:  ['Microsoft MPI...', 'Windows 8.1...', 'Server2012...']
CVE-2017-9383:  ['curl 7.47...', 'curl 7.64...', 'wget 1.20...', 'curl 7.58...',
'curl 7.68...']
```

Furthermore, CyVIA provides detailed information about each network node. For example, CVE-2019-12068 is the most common vulnerability among the 11,305 found vulnerabilities on the high-risk node (Raspbian). This vulnerability is basically a software

Table 5.6: CyVIA Infrastructure-based Top 10 Weakness Types

Description	**%**	**Count**
Other	14.73	5,563
SQL Injection	12.20	4,607
Buffer Overflow	11.15	4,211
Insufficient Information	8.97	3,388
Improper Input Validation	6.17	2,330
Cross-site Scripting	5.55	2,095
Unauthorized Access	5.45	2,057
Access Controls	4.70	1,774
Resource Management Errors	3.18	1,200
Code Injection	3.11	1,176

Table 5.7: CyVIA Infrastructure-Based Vulnerability Severity Analysis

Sv.	**Count**	**Mean**	**Std.**	**Min**	**50%**	**75%**	**Max**
H	17558	8.29	1.0	7.1	7.6	9.3	10.0
M	16923	5.30	1.0	4.0	5.0	6.5	6.9
L	3281	2.58	0.7	1.0	2.1	3.5	3.8

bug (an infinite loop) that can lead to a successful denial-of-service attack. 36% of these vulnerabilities are high severity, 53.5% can be exploited via the network, and the majority of vulnerabilities belong to the class "Other," followed by "Cross-site Scripting". On the given network cluster, there are 37,761 vulnerabilities found in total and for 156 vulnerabilities, no information is found within the CyVIA dataset. These are newly discovered vulnerabilities for which relational information within the CyVIA dataset was not present at the time of the scan. The server-side scheduler is responsible to update vulnerability information and is currently set to update once a week.

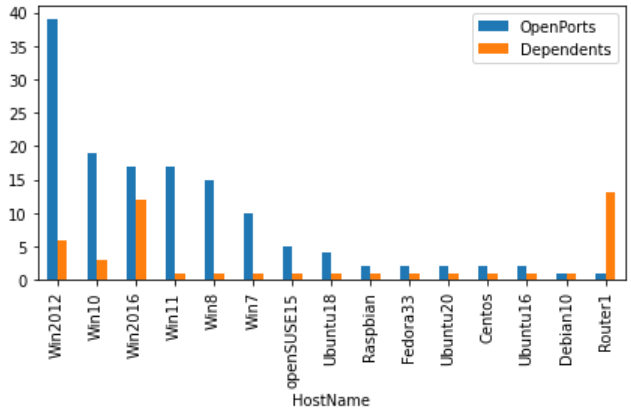

Figure 5.5: CyVIA Infrastructure-based Open Ports vs Dependents

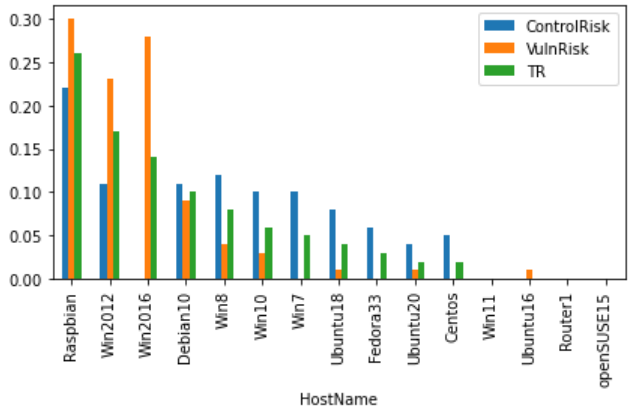

Figure 5.6: CyVIA Infrastructure-based Control and Vulnerability Risks

e) Potential Consequences and Mitigation

After identifying the found weaknesses in the infrastructure, CyVIA is capable of educating the cyber defender about the common consequences caused by the found weaknesses and at the same time how to mitigate them. The sample output below provides the information about CWE-200 i.e. unauthorized access.

```
CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor...
CWE-200 - Common Consequences:
.  Confidentiality:
..  IMPACT:
```

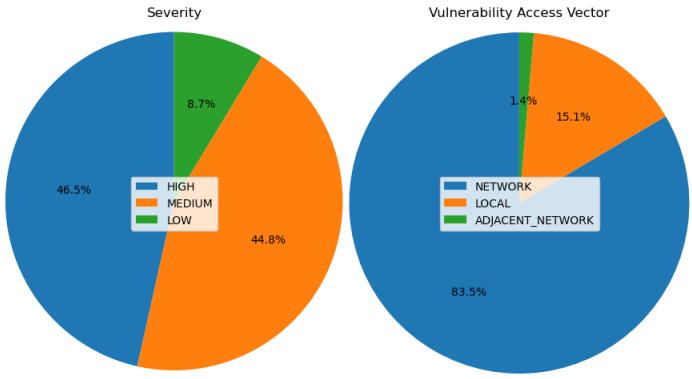

Figure 5.7: CyVIA Infrastructure-based Severity and Vulnerability Access Vector

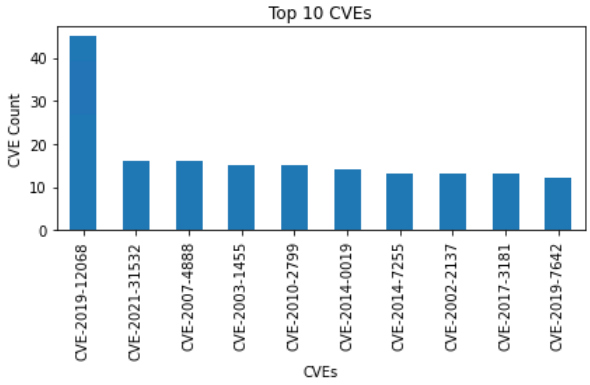

Figure 5.8: CyVIA Infrastructure-based Top 10 CVEs

```
..  - Read Application Data

CWE-200 - Potential Mitigations:

.  Architecture and Design:

..  DESCRIPTION:

..  - Compartmentalize the system to have safe areas where trust boundaries can
be ...  more ...

..  STRATEGY:

..  - Separation of Privilege.
```

5.8.2 Analysis by other Tools

The trial version of Nessus Essentials allows scanning of up to 16 nodes on the network. Nessus results showed asset classification based on vulnerability severity as seen in Table 5.8. On the other hand, Nessus also provides remediation information for the found vulnerabilities. As per the results, node 50.50.50.8 i.e. a Windows Server 2012 R2 has the highest number of found vulnerabilities followed by 50.50.50.26 (OpenSuse 15.2.1). Time taken by Nessus to scan the network was $\approx$ 33 minutes.

Table 5.8: Nessus Results

Node	**Critical**	**High**	**Med**	**Low**	**T.**
Server2012	**25**	**238**	**83**	8	**354**
OpenSUSE	29	106	63	5	**203**
Debian10	36	85	18	1	**140**
Server2016	16	51	23	0	**90**
Fedora33	5	49	22	1	**77**
Ubuntu16	7	13	6	0	**26**
Win81	0	2	8	1	**11**
Ubuntu18	3	5	3	0	**11**
Centos831	1	5	3	1	**10**
Raspbian	1	6	1	0	**8**
Win7	1	1	1	0	**3**
Win10	0	0	2	0	**2**
Win11	0	0	2	0	**2**
Ubuntu20	0	1	1	0	**2**

InsightVM by Rapid7 allows the creation of sites and asset assignments to each site making asset management much easier. InsightVM keeps track of asset risk over time, providing a classification of assets by OS (Windows, Linux, etc.), exploitability (by adver-

sary skill e.g. novice, intermediate, expert, etc.), vulnerabilities, exploits, malware, and risk scores. InsightVM also keeps track of software packages and services. The results generated are shown in Table 5.9. It was observed that node 50.50.50.9 (Server2016) has the highest number of vulnerabilities, however, node 50.50.50.8 (Server2012) has the highest risk score value compared with node 50.50.50.9. Time taken by InsightVM to scan the network was $\approx$ 10 minutes.

Table 5.9: InsightVM Results

Node	**Exploits**	**Malw.**	**Vulns.**	**Risk**
Server2016	**134**	**0**	**1,946**	**670,047**
Server2012	342	10	**1,798**	700,461
OpenSUSE	11	0	**751**	159,590
Debian10	9	0	**595**	162,486
Win7	16	0	**567**	170,516
Win10	4	0	**142**	30,080
Raspbian	0	0	**63**	11,682
Ubuntu16	0	0	**61**	17,308
Centos831	1	0	**56**	11,411
Win81	3	0	**37**	16,718
Ubuntu18	0	0	**17**	8,508
Win11	0	0	**4**	845
Ubuntu20	0	0	**4**	1,495
Fedora33	0	0	**3**	742

Open Vulnerability Assessment System (OpenVAS) has become a part of Greenbone Vulnerability Manager (GVM) which is still available to the community. GSM, on the other hand, is the professional edition and is only available under multiple licensing options similar to InsightVM and Nessus. GSM classifies the nodes by severity of nodes and OS severity based on the found vulnerabilities. GSM also generates network topology based

Table 5.10: GSM Results

Node	**Sev.**	**Score**	**H.**	**M.**	**L.**	**T.**
OpenSUSE	High	9.3	39	110	10	**159**
Debian10	High	10	48	80	7	**135**
Fedora33	High	7.2	9	13	5	**27**
Win10	High	7.7	2	14	0	**16**
Win81	High	7.8	3	7	1	**11**
Server2012	High	10	1	9	1	**11**
Win7	High	7.8	3	4	1	**8**
Raspbian	High	7.5	1	3	2	**6**
Ubuntu16	Med.	4.9	0	2	2	**4**
Ubuntu18	High	10	1	2	1	**4**
Win11	Med.	5	0	1	1	**2**
Server2016	Med.	5	0	1	1	**2**
Centos831	Med.	4.3	0	1	1	**2**
Ubuntu 20	Low	2.6	0	0	2	**2**

on the found network nodes and keeps track of open ports and installed packages. Results generated by GSM are shown in Table 5.10. We can see that node 50.50.50.26 (OpenSUSE 15.2 1) has the highest number of vulnerabilities found, followed by 50.50.50.24 (Debian 10). Time taken by GSM to scan the network was $\approx$ 48 minutes.

5.8.3 Comparison of CyVIA with Other Tools

Each tool has some strengths that make the tool better than the other, for example, Greenbone tools are open-sourced and still available to the community whereas Tenable and Rapid7 products are not. Tenable provides customize-able reports options whereas Greenbone products do not offer such rich reporting options. Rapid7 on the other hand provides a very informative interface and customize-able reports as well. Among the three tools,

Greenbone is very stable and ran without any issues, whereas Rapid7 took the minimum time for scanning the network and generating analysis. One main difference between these tools and CyVIA is that all three generate on-demand analysis whereas CyVIA provides dynamic risk assessment and keeps the administrator informed at all times for any changes in node configurations or risk. Table 5.11 lists the vulnerability counts by all four tools, however, CyVIA provides further details of contextual cyber risk assessment that is very useful for the administrator.

Table 5.11: Tool Comparison in Terms of Detected Vulnerabilities

Node	**CyVIA**	**O.VAS**	**Nessus**	**Nexpose**
Win7	175	8	3	567
Win81	1,395	11	11	37
Win10	1,100	16	2	142
Windows11	18	2	2	4
Server2012	8,625	11	354	1,798
Server2016	10,592	2	90	1,946
Centos831	9	2	10	56
Debian10	3,431	135	140	595
Fedora33	16	27	77	3
OpenSUSE	51	159	203	751
Raspbian	11,305	6	8	63
Ubuntu16	269	4	26	61
Ubuntu18	316	4	11	17
Ubuntu20	432	2	2	4

The number of observed vulnerabilities in CyVIA is higher compared to other tools due to CyVIA's comprehensive consideration of vulnerabilities present in both the operating system (OS) and each user-installed application. Table 5.12 presents the specific numbers of reported vulnerabilities for individual products installed on the Win81 node. It is evident

that CyVIA reported 1395 Common Vulnerabilities and Exposures (CVEs) for this node, whereas other tools reported significantly lower numbers: 11, 11, and 37 vulnerabilities, as indicated in Table 5.11. However, Table 5.12 reports a total of 2672 vulnerabilities for the Win81 node [101]. This disparity arises because a single vulnerability may exist in multiple products, and CyVIA takes this into account by reporting only unique vulnerabilities for each node. MITRE's CVE search list can be accessed at: `https://cve.mitre.org/cve/search_cve_list.html`.

Additionally, CyVIA goes beyond other tools by providing more in-depth insights into infrastructure-based risk and node-based risk, highlighting critical areas across the entire system. Conversely, other tools primarily focus on specific aspects of individual nodes, offering a narrower perspective.

5.9 Summary

Heterogeneity in cyberspace has introduced a wide spectrum of weaknesses and uncertainties for cyber defenders to defend against. In such a scenario, keeping the organizational infrastructure safe is a major challenge. To address the research question Q2 under the objective B, we present a threat intelligence system CyVIA, that provides contextual cyber situational awareness to a cyber defender. CyVIA considers various key elements that play a significant role in evaluating organizational cybersecurity. We evaluate CyVIA on a network cluster and compare the results with the state-of-the-art. Our results indicate that CyVIA provides an extensive amount of analyses indicating infrastructure-based loopholes as compared with other tools. This work is published at the EAI Endorsed Transactions on Security and Safety 8.30 (2022).

The upcoming chapter of this dissertation will showcase our efforts on developing an AI-based prediction engine. This prediction engine is aimed at enhancing the analysis process by efficiently predicting potential attack types based on identified vulnerabilities and loopholes within the cyber infrastructure.

Table 5.12: Reported Vulnerabilities in Products by MITRE

Product	**Reported CVEs**
Windows 8.1 Enterprise Evaluation Build 9600	1184
Python 3.7.7 Core Interpreter (64bit) 3.7.7150.0	3
Python 3.7.7 Test Suite (64bit) 3.7.7150.0	143
Python 3.7.7 pip Bootstrap (64bit) 3.7.7150.0	7
Python 3.7.7 Executables (64bit) 3.7.7150.0	2
Python 3.7.7 Utility Scripts (64bit) 3.7.7150.0	25
Python 3.7.7 Tcl/Tk Support (64bit) 3.7.7150.0	27
Python 3.7.7 Development Libraries 3.7.7150.0	27
Microsoft Visual C++ 2013 x86 12.0.21005	9
Python Launcher 3.7.7008.0	834
TunnelBear 4.2.10.0	1
SolarWinds Agent 2020.2.2593.5 120.2.2593.5	5
Adobe Acrobat Reader DC 21.005.20048	340
Microsoft Visual C++ 2008 9.0.30729.6161	5
Java(TM) SE Development Kit 16.0.1 16.0.1.0	15
Microsoft Visual C++ 2015 x86 14.0.24215	9
PuTTY release 0.75 (64bit) 0.75.0.0	2
GlobalProtect 5.2.4	1
Microsoft Silverlight 5.1.50918.0	33
Total CVEs	**2672**

Chapter 6

Dynamic Vulnerability Classification

Cyber-threat landscape and adversarial capabilities have strengthened significantly due to the digital transformation and increased computational capacity of individuals. To stay ahead in the game, a cyber defender must have full situational awareness of any existing infrastructural vulnerabilities. Leveraging vulnerability reports from NVD, MITRE, Twitter, etc., is an uphill task as one must find the existing vulnerabilities first, find vulnerability reports for the same, and then prepare a mitigation plan by going through each report individually. Moreover, human attention is needed to understand the context and decide whether the risk is acceptable or actionable. In this Chapter, we present the architecture and implementation of the AI-based prediction engine for our CyVIA framework to classify vulnerability reports based on inferred attack types to to address research questions Q1 under the objective C. This AI-engine speeds up the vulnerability analysis process for cyber defenders by providing the applicable attack types on the evaluated infrastructure. We test various unsupervised and supervised machine learning models to classify vulnerability reports. Furthermore, we compare the results, tune the best-observed models, and propose a final fully trained model with the highest accuracy for classifying new vulnerability reports.

6.1 AI-Based Prediction Engine

This Section describes the detailed approach to infer vulnerability reports in order to determine the various types of attacks the cyber infrastructure is exposed to. We discuss the vulnerability dataset, implementation and evaluation strategies, and the different ML models we train to find the best model to identify the trends and patterns for vulnerability

data. Fig. 7.1 presents the overall AI-based prediction engine (AI-engine) architecture and the different phases. We utilize CyVIA's automatically curated vulnerability dataset from multiple online sources. This work primarily focuses on the AI-engine, which has two sub-modules, classification, and inferencing, as seen in Fig. 7.1. We introduce the classification component here, which has three different stages, 1) Analysis: responsible for finding hidden groups of data within the vulnerability reports, 2) Labeling: responsible for preparing the ground truth dataset, and 3) Classification: classifying vulnerability documents. We discuss each in the following subsections.

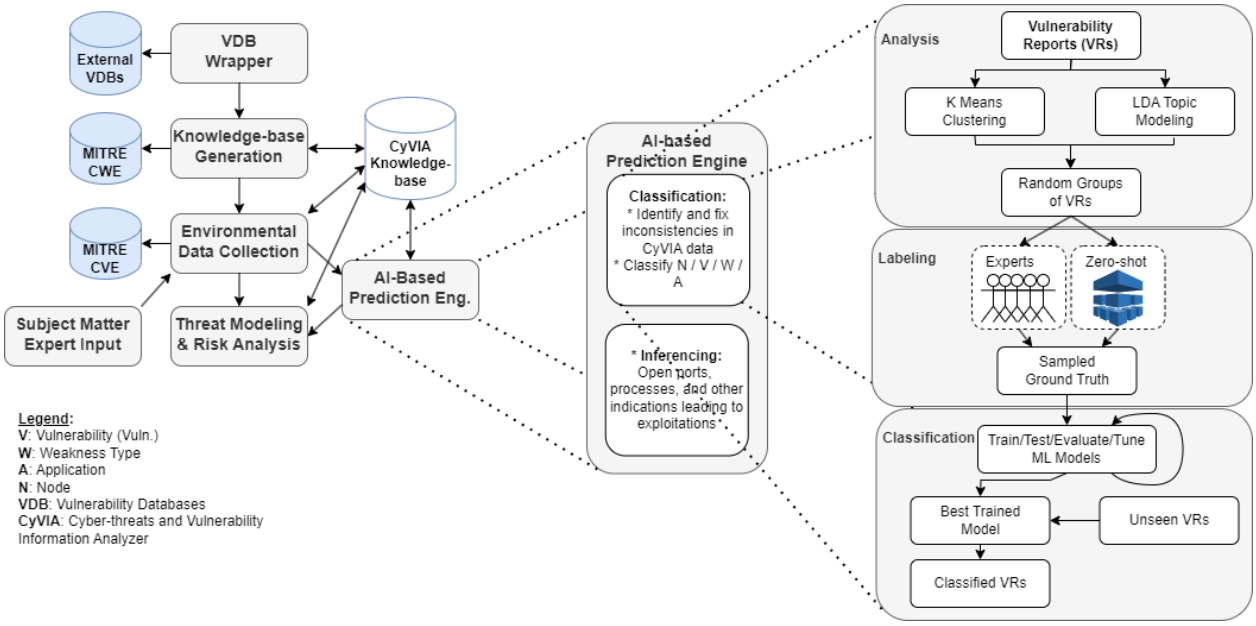

Figure 6.1: CyVIA AI-based Prediction Engine Architecture v1

6.1.1 Dataset Overview

The VDB Wrapper module of CyVIA is responsible for curating vulnerability data from external vulnerability databases. This data is then organized in the CyVIA knowledge-base based on the relationships found between data elements such as vulnerability ID and weakness types. It includes vulnerability reports from the year 2001 to the present. This data is obtained, processed, and updated regularly by our continuous risk assessment

framework CyVIA [99]. The dataset currently contains 201,416 vulnerability documents with 20 different features. The vulnerability reports are available under each document's 'description' feature. Our main goal is to extract attack information (kind of attack) from each report description and classify new vulnerability reports according to the attack class inferred by each description. This data, however, is not labeled; we experiment with various ways to find distinct groups within the dataset. We also intend to manually label random documents from the discovered groups and train multiple ML algorithms to find the best classifier. Our strategy for dealing with this issue is as follows:

6.1.2 Implementation Strategy

In this Section, we discuss and present the step-by-step process that we have followed to implement the AI-based prediction engine.

Data Analysis

In the first step, we analyze data to find the hidden groups of vulnerabilities. The input of this stage is unlabelled vulnerability documents collected by CyVIA from the data collection stage. We use K-Means clustering to identify vulnerability clusters and the Elbow method [102] to find the optimal number of clusters. On the other hand, we also implement the Latent Dirichlet Allocation (LDA) [103] model to see the different topics within the vulnerability documents. And to find the best number of topics, we use GridSearch with LDA. This stage results different clusters and Topics representing different groups in the vulnerability dataset. We analyze both results and take random samples from the best-observed method to start labeling the vulnerability documents.

Labeling

In this stage, we select random samples from each cluster or topics group and label the kind of attack as observed within the vulnerability descriptions. There can be several types of

attacks, including buffer overflow, code injection, cross-site scripting, and so on. Our list contains 36 different types of attacks collected from MITRE and other sources. We take two different approaches for labeling: first, we ask a set of experts to label the randomly selected subset of vulnerability documents based on the results from the previous stage and then feed the same documents to a set of zero-shot learning models. Zero-shot learning is developed for scenarios where the ML algorithms have seen few or no examples. This type of classification is known as also known as classification on the fly. We implement zero-shot learning algorithms BERT, Flair, and Transformer models (Bart-large-mnli, Cross-Encoder, Bart-large-nli) [104,105]. Both the experts and zero-shot algorithms are provided with the same set of attack type labels for labeling. We then combine the labeled data from experts and zero-shot models and take the max count of assigned labels for each document. After carefully observing the final labels, we feed the labeled data to different classification algorithms in the next stage.

Classification

Once the labels are ready, we start processing the data for training. We use different classification algorithms, such as K-Nearest Neighbors, Decision Trees, Support Vector Machines, etc., to find the best classifier with the highest accuracy. These algrithms are non-parametric models, that do not rely on specific parameter settings, and hence, produce more accurate results. We also use pre-trained and learned word embedding models for classifying our labeled dataset to see if we can achieve better results. The outcome of this stage is the best resulting algorithm which is then tuned further to improve the performance and results. We plan to classify the remaining and newly received vulnerability documents with the best-observed ML model.

6.2 Evaluation Strategy

Because we have a multi-stage implementation strategy, we compare each model's performance and results with other models implemented at the same stage. We use hyperparameter tuning to improve the model's performance and accuracy, producing the best results before proceeding to the next step.

6.3 Baseline and Evaluation Metrics

As we have more than one ML algorithms in play, we plan to use the first-level classification as our baseline. Over time, we tune our models to improve their performance and results. We plan to evaluate the performance of the trained models based on various metrics such as accuracies, F1 scores, precision, and recall. This entire process is repeated periodically when CyVIA's VDB Wrapper module curates new vulnerability data from external sources.

In the next section, we discuss the performance of each algorithm in detail.

6.4 Results

We ran our Python code on a 16 Core Intel(R) Xeon(R) 2.10GHz CPU with 32GB RAM. The datasets, labels, notebooks, and trained models are available for the research community with other information pieces on our Git repository[1]. This section discusses our evaluation strategy for each stage, as shown in Fig. 7.1.

6.4.1 Vulnerability Data Analysis

To segment the unlabeled vulnerability dataset, we used two approaches: K-Means clustering and LDA topic modeling. Finding the optimal number of clusters took $\approx$ 15 minutes and returned $k = 16$ as the most promising number of groups residing within the dataset

[1]https://github.com/trucyber/Risk-Assessment-Framework

we provided. GridSearch, on the other hand, took ≈ 5 hours to complete and returned $n\_components = 10$ as the best number of topics. After analyzing the clusters and topics by random sampling, we noticed multiple subgroups of data present inside each topic group. In contrast, the K-Means returned comparatively cleaner and well-organized individual groups of data. Additionally, K-Means ran much faster in terms of time than LDA; therefore, we decide to proceed further with K-Means results.

Table 6.1: Labeling Summary of Domain Experts

Experts	Found Attack Types
Expert 1	13
Expert 2	14
Expert 3	16

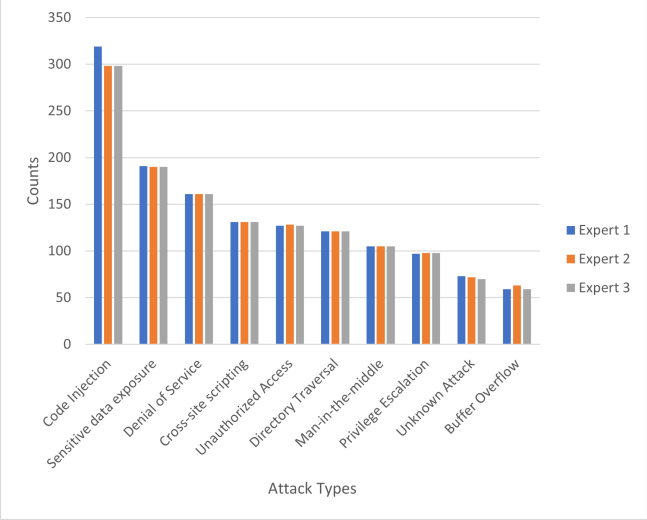

Figure 6.2: Top 10 Experts Labels

6.4.2 Labeling

We took 100 random samples from each cluster and generated a new dataset containing 1,400 random documents for labeling, removing vulnerability documents containing re-

jected, and disputed entries [106]. Although the K-Means provide 16 different clusters, we engineer 36 different classes to represent the types of attacks upon analyzing data. However, these types are limited to the 172,288 vulnerability documents within the collected vulnerability dataset. We provide these 1,400 random documents with 36 attack labels to a set of cybersecurity researchers to label them based on the observed attack type found in each vulnerability description. Table 6.1 summarizes the time taken and the number of attacks found by each expert. Fig. 6.2, on the other hand, highlights the top 10 attack types as labeled by the experts. It is apparent that the experts are on the same page for most types of attacks.

Table 6.2: Labeling Summary of Zero-Shot Models

Models	Labeling Time	Found Attack Types
BERT	20 min	25
Flair	1 hour	15
Bart-large-mnli	3 hours	25
Cross-Encoder	21 min	32
Bart-large-nli	2 hours	25

Furthermore, we used five zero-shot classification algorithms to label the same data. Table 6.2 shows that BERT took the minimum time to label the provided data and returned 25 attack types. On the other hand, Cross-Encoder took around a similar time but returned the widest range of attack types compared to all other algorithms. Zero-shot algorithms took 20 minutes to 3 hours to label the 1,400 documents. Fig. 6.3 illustrates the count of correctly classified top 10 attack types as labeled by the zero-shot models. This is useful in illustrating which zero-shot models accurately predicted the labels. Figure 6.4 illustrates the count of predicted attack types by the zero-shot models and the expert predictions for each type of predicted attack. Table 6.3 presents a detailed analysis of these zero-shot labels, revealing that BERT has the highest number of incorrect label predictions, while CE has the lowest compared to the final expert labels. Additionally, Figure 6.5 provides

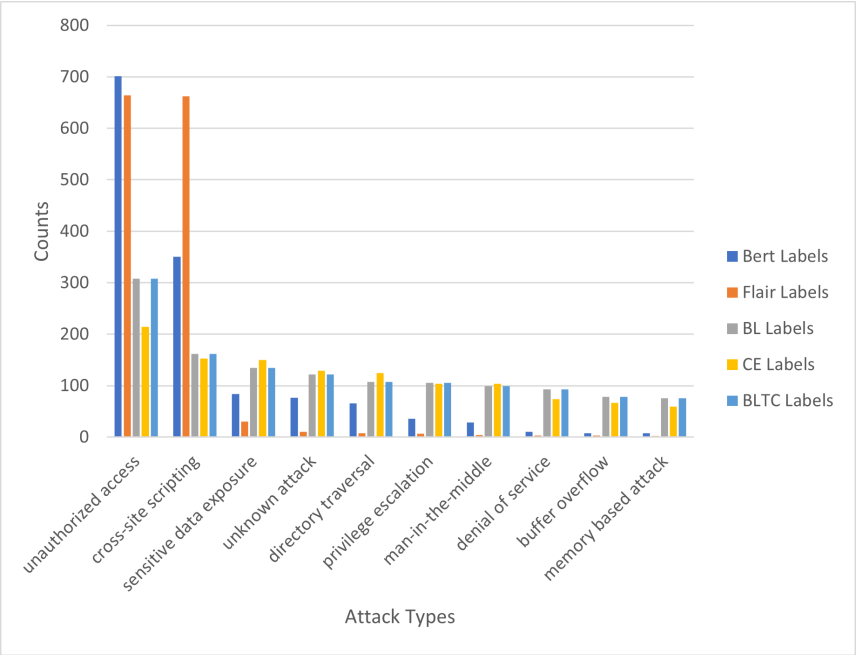

Figure 6.3: Top 10 Zero-shot Labels

Table 6.3: Misclassification Rate of Zero-shot Models

	BERT	**Flair**	**BL**	**CE**	**BLTC**
# of Misclassified Labels	1340	1206	610	515	610
Percentage	95.71	86.14	43.57	36.79	43.57

valuable insights into the found attack types by both the experts and zero-shot models. It is evident from the figure that the final labels encompass the majority of observed votes.

As discussed next, we feed this labeled ground truth to different classification algorithms to find a potential model best suited for our situation.

6.4.3 Multi-class Classification of the Vulnerability Data

The aim of this study was to compare the performance of various classification algorithms in the multi-class classification of vulnerability data and to determine how to optimize their accuracy. The preprocessing stage involved using the TF-IDF Vectorizer with stop words

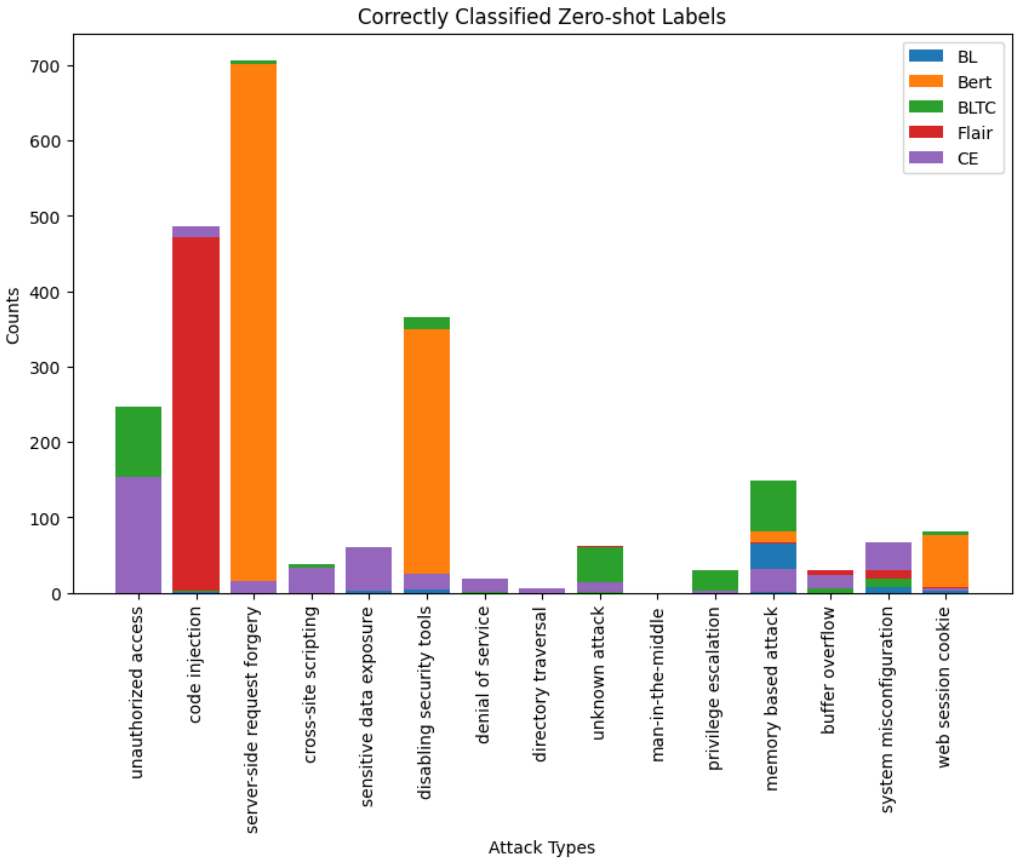

Figure 6.4: Top 10 Correctly Classified Labels

removal, resulting in 5891 features from 1400 labeled vulnerability documents. An 80/20 training/test split was used to evaluate the following classification algorithms:

a) Support Vector Machine (SVM)

SVM is suitable for datasets with small number of features and a large number of samples. We ran SVM with default parameters and the model returned 80% accuracy in $\approx$ 1.9 seconds.

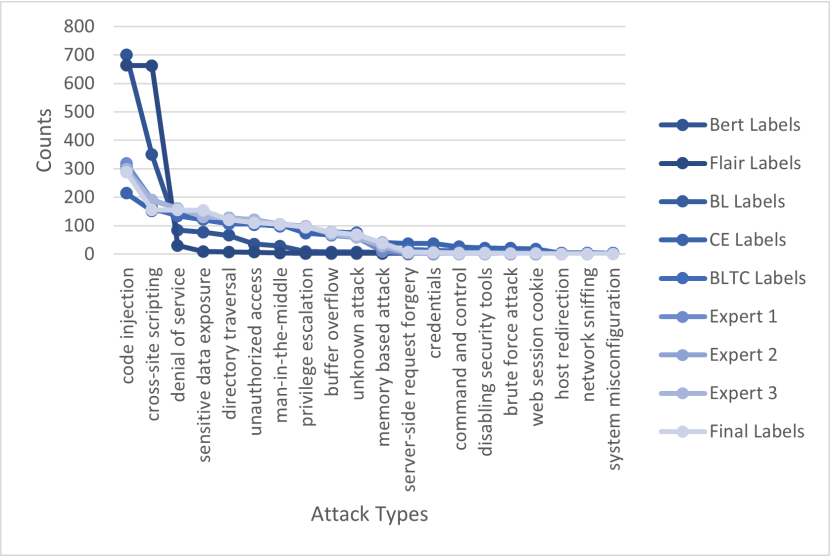

Figure 6.5: Found Attack Types

b) Linear Support Vector Machine (LinearSVM)

LinearSVM took comparatively less time than SVC (0.4 seconds), and returned higher accuracy (85%). This is mainly because we have high-dimensional text data where the number of features is larger than the number of samples. This data is linearly separable LinearSVM can handle this situation more efficiently and can be trained faster than the SVM.

c) KNeighbors

We ran K-Nearest Neighbors algorithm with k ranging from 1 to 12. It took $\approx$ 1.38 seconds to find the best k, i.e., 9, with an accuracy of 78%. Similar to SVM, KNeighbors's performance can also be impacted by the high dimensionality and sparsity of the text data.

d) Random Forest

This algorithm may not be the best choice for text classification but it has the ability to handle high-dimensional feature spaces and can handle noisy and missing data. With the

default parameters, this model returned the accuracy of 85% in $\approx$ 0.9 seconds, which is as good as LinearSVM.

e) Naive Bayes

We implemented Multinomial, Gaussian, and Bernoulli Naive Bayes models. These models are commonly used for text classification tasks and have shown to perform well in various scenarios. Among the three, Bernoulli returned the highest classification accuracy of 81% in $\approx$ 0.06 seconds.

f) Decision Tree

With a depth ranging from 1 to 14, Decision Tree took $\approx$ 2.4 seconds and returned 82% accuracy on depth 14. As this algorithm is prone to overfitting in case of high-dimensional data, but in our case, it ran as good as Naive Bayes Bernoulli.

g) Artificial Neural Network (ANN)

To configure an ANN on our dataset, we set up a Multilayer Perceptron with default parameters and observed 82% accuracy in $\approx$ 29.17 seconds. ANNs are considered good for text classification, however, their performance can be affected by the size of the dataset. The labeled dataset in our case included 1600 examples, which is relatively on the lower side.

h) Dense Network

Dense Neural Networks (DNN) are considered very good for text classification as they can learn complex relationships between features and classes in the data and can handle high-dimensional input spaces. However, similar to ANN, the size of the dataset is one main reason for performance issues in case of a DNN. A network with two hidden layers, batch size 16, and 10 epochs produced the best results (83%) among the other configurations we have tried.

i) Deep Neural Network

Although these models are suitable for classifying image data, we wanted to test how well they perform for text classification. The results were not significantly different than ANN. We configured VGG1 - VGG5 with variating batch sizes from 16 - 256 at ten epochs each. VGG1 with batch size 32 returned the highest accuracy of 82%.

j) Word Embedding

We trained a word embedding model with our dataset and compared it against a pre-trained model (glove [106]). The pre-trained model outperformed the other model by providing 5% higher accuracy because it is trained on a large Wikipedia corpus.

6.4.4 Improvements and Model Tuning

We use the initial results discussed above as our baseline. Before tuning the best-observed model, we take one step back to TF-IDF vectorizer to re-engineer the number of features to reduce the noise in the data, keeping as much variation as needed to keep the originality of the dataset. We removed multiple spaces, single characters, special characters, and terms appearing in less than three documents and obtained term contexts using lemmatization. This process reduced the number of features from 5891 to 1210 features, allowing the models to run faster than before. Furthermore, we tune the following algorithms to improve the results.

a) LinearSVM with Principal Component Analysis (PCA)

Dimensionality reduction is often used at the preprocessing stage; in our case, we apply PCA on LinearSVM to outperform other algorithms. Using PCA, we reduced the number of inputs from 1210 to 50 on LinearSVM. With Principal Component Analysis (PCA) applied on top of LinearSVM, n_components=300, the algorithm improved the accuracy from 85% to 89%. To evaluate the model, we observe the f1-score, which considers both

false positives and false negatives. We can observe in Fig. 6.6 that the model works very well for all the classes except 8, 11, 14, and 16, as there was not enough data available for these. The weighted average of f1 demonstrates the efficiency of our model.

Classification Report:

	precision	recall	f1-score	support
0	0.92	0.86	0.89	14
1	0.89	0.94	0.92	18
2	0.93	0.90	0.92	30
3	0.56	0.67	0.61	15
4	0.89	0.94	0.92	54
5	1.00	0.94	0.97	31
6	1.00	0.50	0.67	2
7	0.86	0.75	0.80	8
8	0.00	0.00	0.00	2
10	0.96	0.96	0.96	28
11	0.00	0.00	0.00	1
12	1.00	1.00	1.00	23
13	0.88	1.00	0.93	21
14	0.00	0.00	0.00	1
15	0.78	0.81	0.79	31
16	0.00	0.00	0.00	1
accuracy			0.89	280
macro avg	0.67	0.64	0.65	280
weighted avg	0.88	0.89	0.88	280

Figure 6.6: Classification Report of LinearSVM with PCA

b) Dense Network with PCA

As PCA improved the results for LinearSVM, we designed a dense network with PCA with two hidden layers (500,500), batch_size=16. This improved the results of the Dense Network from 83% to 85%.

c) Word Embedding

Word Embedding is widely used for text analysis; we trained our own Word Embedding model using the vulnerability data and compared the results with a pre-trained model glove. The pre-trained model ran slower but produced better results as compared to the newly trained model.

6.4.5 Overall Results

Among the ML algorithms we have trained and evaluated on the vulnerability dataset, we found Linear SVM performing better than other algorithms in classification accuracy and performance. This is also because our data is highly dimensional, and SVM works better with mapping high-dimensional feature space to categorize data points. We observed that reducing the number of features has improved the results even further for most algorithms. This shows that most of the text features in the vulnerability data are optional to classify attacks from the vulnerability documents. Furthermore, we got even better results by applying PCA's dimensionality reduction technique. As an example, if we consider the following vulnerability description:

CVE-2018-21215: Certain NETGEAR devices are affected by a ***buffer overflow*** *by an unauthenticated attacker. This affects D3600 before 1.0.0.67, D6000 before 1.0.0.67, D6100 before 1.0.0.56, EX2700 before 1.0.1.28, R7500v2 before 1.0.3.24, R9000 before 1.0.2.52, WN2000RPTv3 before 1.0.1.20, WN3000RPv3 before 1.0.2.50, and WN3100RPv2 before 1.0.0.56.*

One can identify the type of attack highlighted in CVE-2018-21215 is a buffer overflow. The first part of the vulnerability description highlights the vulnerability type, followed by impact and vendor information, which is not essential to extract the attack information at this point in our case. Similarly, the following two examples highlight code injection and cross-site scripting.

CVE-2020-7623: jscover through 1.0.0 is vulnerable to Command Injection. It allows execution of arbitrary command via the source argument.

CVE-2012-2644: Cross-site scripting (XSS) vulnerability in the MT4i plugin 3.1 beta 4 and earlier for Movable Type allows remote attackers to inject arbitrary web script or HTML

via unspecified vectors, a different vulnerability than CVE-2012-2642.

Table 6.4: Final Experimental Results

Algorithm	**Accuracy**	**Execution Time**
Support Vector Machine	0.8047	1.9834 secs
Linear SVM	0.8542	0.4531 secs
K-Nearest Neighbors	0.7828	0.0045 secs
Random Forest	0.8516	0.7722 secs
Bernouli Naive Bayes	0.8081	0.0630 secs
Decision Tree	0.8187	0.2226 secs
Multilayer Perceptron	0.8203	26.9247 secs
Linear SVM with PCA	0.8928	8.1363 secs
Dense Network	0.8338	1 sec / epoch (10 Total)
Dense Network with PCA	0.8469	1 sec / epoch (10 Total)
Deep Neural Network (VGG1)	0.8222	1 sec / epoch / batch
Word Embedding Learned	0.7525	2 sec / epoch (20 Total)
Word Embedding Pre-Trained	0.8015	12 sec / epoch (20 Total)

Table 6.4 lists the algorithms, their accuracies, and the execution time. We can see that the Linear SVM with PCA has outperformed all other algorithms; Fig. 6.6 shows the confusion matrix and classification report. We can see that four classes (8, 11, 14, 16) have zero precision and recall because all four classes have very few examples in the training and test data (4, 1, 3, 0, and 2, 1, 1, 1, respectively). Although we have taken an equal number of random examples from each cluster, the final labels produced by experts and ML models included some classes with very few examples as seen in Table 6.5. To overcome this, we removed the classes with few examples and re-ran the experiment. However, this did improve the performance or accuracy of the ML algorithms. On the other hand, we were hopeful that the Word Embedding model would outperform other models, but the

results show the classifier accuracy on the lower side as compared to other algorithms (Table 6.4). Between the Learned and Pre-Trained Word Embedding models, Pre-Trained produced better results with the trainable parameter set to True. Another observation is that keeping smaller batch sizes in deep neural network models (batch_size = 16-32) produced the best results.

The outcome of the AI-engine is to determine the attack type from any vulnerability description. The average vulnerability description is 283 characters; however, the descriptions can be longer than this, and the max length we have noticed is 3819 characters long among the labeled data. This output is fed back to the CyVIA framework providing the types of attacks the infrastructure is prone to, and the mitigation plan is generated from the CyVIA's Threat Modeling and Risk Analysis module. A few examples of vulnerability descriptions and related classification are as follows:

CVE-2021-20349: IBM Tivoli Workload Scheduler 9.4 and 9.5 is vulnerable to a stack-based buffer overflow, caused by improper bounds checking. A local attacker could overflow a buffer and gain lower level privileges. IBM X-Force ID: 194599. $\longrightarrow$ Attack type: Buffer Overflow.

CVE-2021-35504: Afian FileRun 2021.03.26 allows Remote Code Execution (by administrators) via the Check Path value for the ffmpeg binary. $\longrightarrow$ Attack type: Code Injection.

CVE-2021-1230: A vulnerability with the Border Gateway Protocol (BGP) for Cisco Nexus 9000 Series Fabric Switches in Application Centric Infrastructure (ACI) mode could allow an unauthenticated, remote attacker to cause a routing process to crash, which could lead to a denial of service (DoS) condition. This vulnerability is due to an issue with the installation of routes upon receipt of a BGP update. An attacker could exploit this vulnerability by sending a crafted BGP update to an affected device. A successful exploit could allow the attacker to cause the routing process to crash, which could cause the device to reload. This

vulnerability applies to both Internal BGP (IBGP) and External BGP (EBGP). Note: The Cisco implementation of BGP accepts incoming BGP traffic from explicitly configured peers only. To exploit this vulnerability, an attacker would need to send a specific BGP update message over an established TCP connection that appears to come from a trusted BGP peer. $\longrightarrow$ Attack type: Denial of Service.

6.5 Summary

Staying abreast of the latest trends in cybersecurity poses a significant challenge for cyber defenders. In response, we present a cyber situational awareness framework designed to keep defenders and administrators informed about relevant threats. This work addresses the research questions Q1 under the objective C, which is *"Identify and classify the risks associated with the cyber infrastructure into multiple uniform groups by leveraging the unstructured and unlabeled threat data"*. We provide a comprehensive architecture and step-by-step process flow of the AI engine for our threat-centric real-time analytics framework, CyVIA. The objective of the AI engine is to accelerate the analysis and decision-making process for defenders, allowing them to quickly identify the type of attacks to which an infrastructure may be vulnerable and suggest immediate mitigation plans. This work has been published in the proceedings of IEEE SYSCON 2023 conference.

Thus far, we have focused on identifying the risks associated with cyber infrastructure. However, it is equally important to provide mitigation techniques to cyber defenders to effectively manage and reduce these risks. In the next chapter, we will present the AI-based inferencing engine, which is designed to propose mitigation strategies for the risks identified through the CyVIA framework. This engine will leverage the power of AI to provide defenders with valuable insights and recommendations for effective risk mitigation strategies.

Table 6.5: Final labels grouped by count

Attack Type	**Count**
code injection	288
cross-site scripting	156
denial of service	155
sensitive data exposure	154
directory traversal	121
unauthorized access	111
man-in-the-middle	105
privilege escalation	95
buffer overflow	77
unknown attack	69
memory based attack	39
server-side request forgery	8
credentials	6
command and control	4
disabling security tools	4
brute force attack	2
web session cookie	2
host redirection	1
network sniffing	1
system misconfiguration	1

Chapter 7

Scalable Cyber Risk Assessment and Mitigation Framework

The increasing dependence on digital technology and the internet has made cybersecurity a critical issue for organizations, with cyber-attacks becoming more frequent and sophisticated. In this context, cyber risk evaluation and mitigation have become essential components of modern cyber infrastructures to ensure the security and resilience of digital assets and services in the face of ever-evolving cyber threats. To address the research question Q2 under objective C, which is: *"Design a mitigation recommendation subsystem to assist in resolving anomaly alerts in real-time."*, in this Chapter we emphasize the significance of the Cyber-threats and Vulnerability Information Analysis to proactively understand the cyber risks and abnormalities in real-time and provide appropriate mitigation strategies. Our work incorporates an inferencing layer to our AI-engine focusing on cyber risk assessment and mitigation. This inferencing layer prioritizes significant risks and presents a mitigation plan to address them. We discuss the key steps and processes implemented as part of the cyber risk and mitigation (CRAM) framework including use of machine learning algorithms for risk assessment and mitigation. Furthermore, we evaluate and compare the effectiveness of the mitigation plan using strategies provided by the MITRE Corporation, a trusted source in cybersecurity. Overall, this Chapter highlights the importance of incorporating a real-time risk assessment and mitigation system in organizations' cybersecurity infrastructure. Our framework provides a practical and efficient solution to identify and address potential cyber threats, minimizing the risk of data breaches and financial loss.

7.1 AI-Based Inferencing Engine

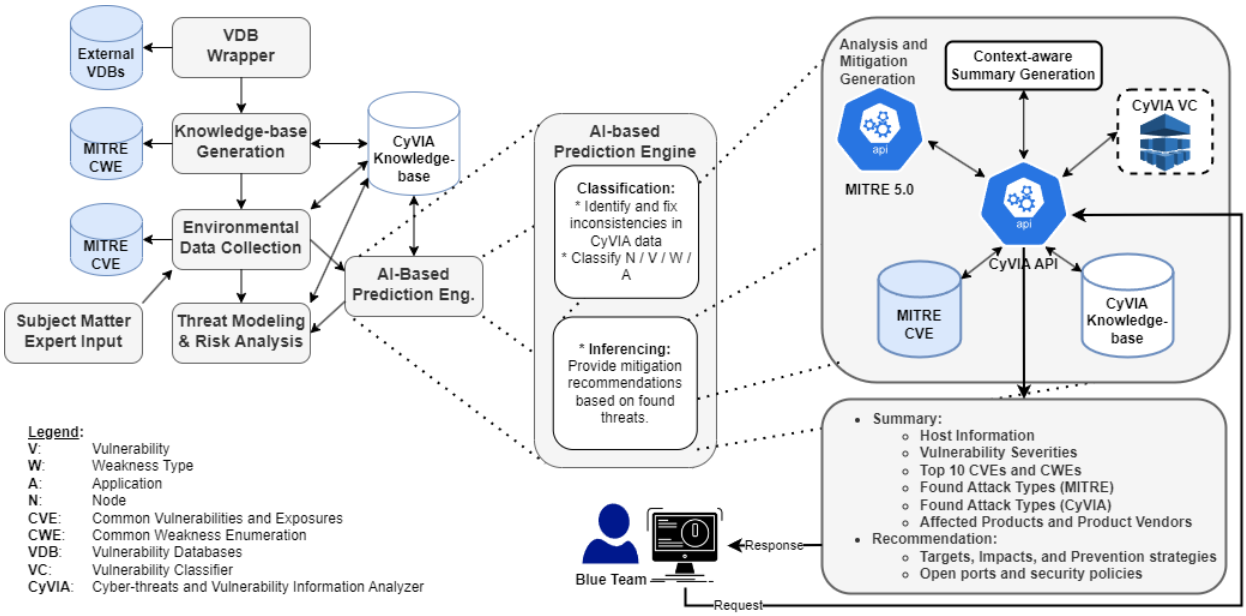

Figure 7.1: CyVIA AI-based Inferencing Engine

This section of the dissertation provides a formal overview of the core components and objectives of the CyVIA inferencing engine, its integrated components, along with their interaction within the overall framework of the CyVIA. CyVIA collects and stores vulnerability data from online vulnerability databases and the computing environment into the CyVIA knowledgebase, which is then used by the threat modeling and risk analysis module to produce risk analytics based on the identified relationships between vulnerabilities.

The AI-based prediction engine plays a critical role in the CyVIA system, with two primary objectives: 1) to accelerate the vulnerability analysis process for cyber defenders by providing applicable attack types for the evaluated infrastructure, and 2) to prioritize risks based on their significance and provide a comprehensive mitigation plan to address them. Fig. 7.1 outlines the system architecture for the CyVIA AI-based inferencing engine, with each component discussed in detail in the following sections.

7.1.1 Vulnerability Classifier (VC)

In order to facilitate human judgment of risk, CyVIA employs a vulnerability classifier that has been trained on a dataset of vulnerability data spanning two decades. Given the size and dimensionality of the vulnerability data, a Linear Support Vector Machine (Linear SVC) model has been trained and tuned for classification purposes. To further enhance computational efficiency, Principal Component Analysis (PCA) has been applied to the Linear SVC model to reduce noise and features in the dataset. This approach focuses the classifier on the most important features, leading to better accuracy and efficiency. The trained classifier is capable of predicting attack types associated with text-based vulnerability descriptions, thereby streamlining the analysis process. The CyVIA API utilizes the vulnerability classifier to generate a list of applicable attack types for a given computing infrastructure, providing a comprehensive assessment of associated risks.

7.1.2 Context-Aware Summary Generator

A key component of CyVIA that significantly contributes to its capacity to offer precise and practical guidance to cyber defenders concerning identified risks is the ability to extract specific feedback and actions from a diverse range of mitigation strategies present in its knowledgebase. This component plays a pivotal role in enabling efficient and timely risk mitigation and management. Through the extraction of targeted feedback and actions, CyVIA can provide cyber defenders with guidance that is aligned with the specific context of the identified risk. This level of specificity enhances the effectiveness of the guidance provided, as it enables cyber defenders to address the risk in a manner that is tailored to the organization's unique security needs and priorities. Moreover, the ability to extract targeted feedback and actions allows for streamlined decision-making and more efficient allocation of resources, as cyber defenders can focus their efforts on the most critical risks. The summary generator's methodology involves the utilization of an abstractive text summarization approach employing a pre-trained natural language processing (NLP) pipeline,

specifically SpaCy. The approach involves the extraction of the most salient sentences from the given text based on their respective rankings.

7.1.3 CyVIA Knowledgebase

The CyVIA knowledgebase is a NoSQL database that is organized in a document-oriented manner. Its primary purpose is to serve as a repository for security controls, policies, procedures, reported vulnerabilities, network nodes, and other relevant data. Information in the knowledgebase is stored based on identified relationships between vulnerabilities, weakness types, network nodes, operating systems, applications, and other relevant factors. The goal of the knowledgebase is to make information available to cyber defenders in the most useful form possible, allowing them to quickly identify threats and understand how to mitigate them effectively. Additional information about the CyVIA knowledgebase, including its architecture, functionality, and other components as illustrated in Figure 7.1, can be found in [107].

7.1.4 CyVIA API

The CyVIA API is a fundamental component of the CyVIA AI-based prediction engine, serving as the core of the system's risk analytics and mitigation capabilities. The API is developed using the Flask REST API framework and is primarily responsible for managing communication and interaction between all internal and external components in a timely and sequential manner. When a request for analysis is made for a specific network or node, the CyVIA API initiates the necessary functions to collect and process the relevant information from all other components. During this process, the API interacts with external sources, including the MITRE repository and API, to gather the pertinent data [108]. The API provides a JSON-formatted response that contains details about the requested host/node or the entire infrastructure, such as the severity of discovered vulnerabilities, top 10 vulnerabilities and weakness types, identified attack types,

and affected products. In addition, the mitigation plan includes information related to targets, potential impacts, and preventive measures. Moreover, the API has additional embedded functionality, including the ability to check the status of the knowledge base (http://server_ip: port/CyVIA), find vulnerabilities for a specific operating system or application (http://server_ip: port/CVEs/Win7), provide analysis for a given network node (http://server_ip: port/CyVIA_analysis/node_name), offer information on a given CVE (http://server_ip: port/Describe_CVE/CVE-ID), and infer attack type from the vulnerability text description.

7.1.5 MITRE API and CVE Repository

The external components referenced, including MITRE's Common Vulnerabilities and Exposures (CVE) system, MITRE's API for the CVE system, and MITRE's Common Attack Pattern Enumeration and Classification (CAPEC) system, are all maintained by MITRE. The CyVIA API interacts with these external components as needed to obtain relevant information [84, 108, 109].

In the next section, we provide an evaluation of how risk analytics generated by CyVIA can facilitate the comprehension and expedient mitigation of risk for any cyber infrastructure by cyber defenders.

7.2 Results

This section presents the analytical summary generated by CyVIA's inferencing engine and its potential to provide valuable insights to cyber defenders. Through the comparison of the summarized information provided by CyVIA, we demonstrate the advantages of using CyVIA's analytical summary. Specifically, we provide examples of how the information provided by CyVIA can enhance a cyber defender's understanding of the evaluated cyber infrastructure. We first compare the text descriptions, attack types, and mitigation techniques provided by MITRE and CyVIA in the following subsection. Later, we provide

additional insights that CyVIA provides to increase cyber situational awareness for the defenders. By examining these aspects, we can highlight the strengths and limitations of each source of information and provide insights into the effectiveness of their respective approaches.

7.2.1 Comparing MITRE Data with CyVIA's Summarized Information

We present Table 7.1 as a means to facilitate the comparison between two sources of information related to various CVEs found within the evaluated cyber infrastructure. Table 7.1 displays a subset of information, including vulnerability descriptions that have been shortened by approximately 20-30%, while keeping the most useful features of the text in place. This has been achieved by reducing the length of the texts, as is evident from the lengths of the vulnerability descriptions. In addition, CyVIA attack types, which are derived from a collection of 36 most commonly used types of cyber attacks gathered from MITRE, NVD, and other sources, are also compared to MITRE attack types in Table 7.1. We have observed that CyVIA attack types use more commonly used terminologies by a zero to intermediate level of cyber defenders. Moreover, we have utilized a context-aware summary generator to extract the most relevant actions from the available prevention techniques using CyVIA's inferencing engine. In this regard, we have been able to reduce the length of the text by approximately 55-80%.

7.2.2 CyVIA Vulnerability Classifier (VC)

To provide a classification example, let us consider the following vulnerability description of CVE-2022-31177:

Description: Flask-AppBuilder is an application development framework built on top of Flask python framework. In versions prior to 4.1.3 an authenticated Admin

Table 7.1: Comparisons

CVE	MITRE Description	CyVIA Description	MITRE Attack-Type	CyVIA Attack Type	MITRE Prevention	CyVIA Prevention
CVE-2022-29216	TensorFlow is an open source platform for machine... (length: 638)	TensorFlow open source platform machine learning... (Length: 450)	CWE-94 Improper Control of Generation of Code ('Code Injection')	Code Injection	Run your code in a jail or similar... (Length: 538)	Run your code... (Length: 133)
CVE-2016-7914	The assoc array insert into terminal node function... (Length: 412)	The assoc array insert into terminal node... (Length: 300)	CWE-125 Out-of-bounds Read	Sensitive Data Exposure	Assume all input is malicious... (Length: 1409)	When performing input validation... (Length: 380)
CVE-2013-1229	TMSSNMP Service in TelePresence... (Length: 216)	TMSSNMP Service TelePresence Manager... (Length: 180)	CWE-20 Improper Input Validation	Denial of Service	For any security checks... (Length: 914)	Understand all the potential ... (Length: 412)
CVE-2022-21668	pipenv is a Python development... (Length: 1143)	pipenv Python development workflow tool... (Length: 892)	CWE-20 Improper Input Validation	Code Injection	Inputs should be decoded and... (Length: 661)	Avoid double-decoding and... (Length: 159)
CVE-2019-9854	LibreOffice has a feature where... (Length: 902)	LibreOffice feature documents... (Length: 706)	CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	Directory Traversal	Ensure that error messages only... (Length: 1031)	In the context of path traversal... (Length: 328)

user could query other users by their salted and hashed passwords strings. These filters could be made by using partial hashed password strings. The response would not include the hashed passwords, but an attacker could infer partial password hashes and their respective users. This issue has been fixed in version 4.1.3. Users are advised to upgrade. There are no known workarounds for this issue.

MITRE Attack Type: CWE-916 - Use of Password Hash With Insufficient Computational Effort.

CyVIA Attack Type: Sensitive Data Exposure.

Based on our analysis, it can be inferred that MITRE and CyVIA utilize distinct terminologies for identifying attack types. Specifically, our analysis indicates that MITRE relies on more technical terminologies compared to CyVIA, which tends to use more commonly understood terms. The findings are presented in Table 7.2. The results underscore the significance of employing appropriate vocabularies that are more effective in aiding an

average cyber defender's case.

Table 7.2: MITRE and CyVIA Vulnerability Attack Types

CVE	MITRE AT	CyVIA AT
CVE-2022-23594	Out-of-bounds Read	Unauthorized Access
CVE-2021-29614	Out-of-bounds Write	Code Injection
CVE-2022-32151	Improper Certificate Validation	Man-in-the-middle
CVE-2022-27237	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	Cross-site Scripting
CVE-2016-7914	Out-of-bounds Read	Sensitive Data Exposure
CVE-2014-9090	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	Denial of Service
CVE-2003-0819	Improper Restriction of Operations within the Bounds of a Memory Buffer	Buffer Overflow
CVE-2019-20916	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	Directory Traversal
CVE-2019-9850	Improper Input Validation	Code Injection
CVE-2022-24761	Inconsistent Interpretation of HTTP Requests ('HTTP Request /Response Smuggling')	Server-side Request Forgery

7.2.3 CyVIA Context-Aware Summary Generator

To illustrate, let us consider the following example of a mitigation strategy:

Full Text: Assume all input is malicious. Use an accept known good input validation strategy, i.e., use a list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and

conformance to business rules. As an example of business rule logic, boat may be syntactically valid because it only contains alphanumeric characters, but it is not valid if the input is only expected to contain colors such as red or blue. Do not rely exclusively on looking for malicious or malformed inputs. This is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However, denylists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. (Length: 1124)

Summary: When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. Use an accept known good input validation strategy, i.e., use a list of acceptable inputs that strictly conform to specifications. (Length: 384)

Upon investigating, we see the effectiveness of the CyVIA summary generator in extracting key actions that can be performed to mitigate risk. Our analysis shows that the CyVIA summary generator significantly reduces the amount of text and accurately identifies and can extract the key actions to mitigate risk.

7.2.4 Mitigation Strategies

Table 7.3 displays a list of anticipated attacks that are expected to affect the entire infrastructure. The list is prioritized from high to low priority, and it requires immediate attention. Here, we present the mitigation plan for the highest-priority risk, which is Code Injection:

1. Terminate the client session after each request.

2. Use only SSL communication.

3. Turn all pages to non-cacheable.

4. Use a web server that employs a strict HTTP parsing procedure, such as Apache [REF-433].

5. Run your code in a jail or similar sandbox environment that enforces strict boundaries between the process and the operating system.

6. With Struts, write all data from form beans with the bean's filter attribute set to true.

7. Refactor your program so that you do not have to dynamically generate code.

8. Be especially careful to validate all input when invoking code that crosses language boundaries, such as from an interpreted language to native code.

9. For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.

10. Use an input validation framework such as Struts or the OWASP ESAPI Validation API. Note that using a framework does not automatically address all input validation problems; be mindful of weaknesses that could arise from misusing the framework itself (CWE-1173).

11. ...more...

Through the implementation of the proposed mitigation plan for code injection, developers can incorporate the recommended techniques to improve the quality of their code and prevent any potential exploitation. Furthermore, tailored mitigation plans are available for each type of attack, providing cyber defenders with targeted strategies to mitigate the risk of security breaches.

7.2.5 Overall Risk Analytics

Table 7.4 depicts an evaluation of the cyber infrastructure consisting of 15 nodes equipped with commonly used operating systems and applications, outlining the infrastructure-wide risks. The analysis identifies Raspbian node as the most vulnerable, with the highest number of detected vulnerabilities. These vulnerabilities are categorized into 215 MITRE-defined attack types and 18 CyVIA-defined attack types. Table 7.3 presents a prioritized list of attacks against the entire network, ranked from most to least vulnerable, along with a corresponding mitigation plan for each type of attack.

Additionally, these attack types can be analyzed in detail, allowing cyber defenders to focus on individual vulnerabilities. For each vulnerability, relevant information such as affected products, versions, prevention stages, and strategies are available, enabling defenders to take appropriate measures as part of the process.

CVE: CVE-2022-29216

MITRE Attack Type: CWE-94 * Improper Control of Generation of Code ('Code Injection')

CyVIA Attack Type: Code Injection

Affected Product: tensorflow

Affected Product Version(s): <2.6.4, >=2.7.0rc0, <2.7.2, >=2.8.0rc0, <2.8.1, >=2.9.0rc0, < 2.9.0

Target(T): Access Control, Non-Repudiation, Integrity.

Prevent(P) at Stage: Architecture and Design, Implementation, Operation, Testing.

P_Strategy(PS): Environment Hardening, Input Validation, Compilation or Build Hardening.

7.3 SWOT Analysis of the Proposed Framework

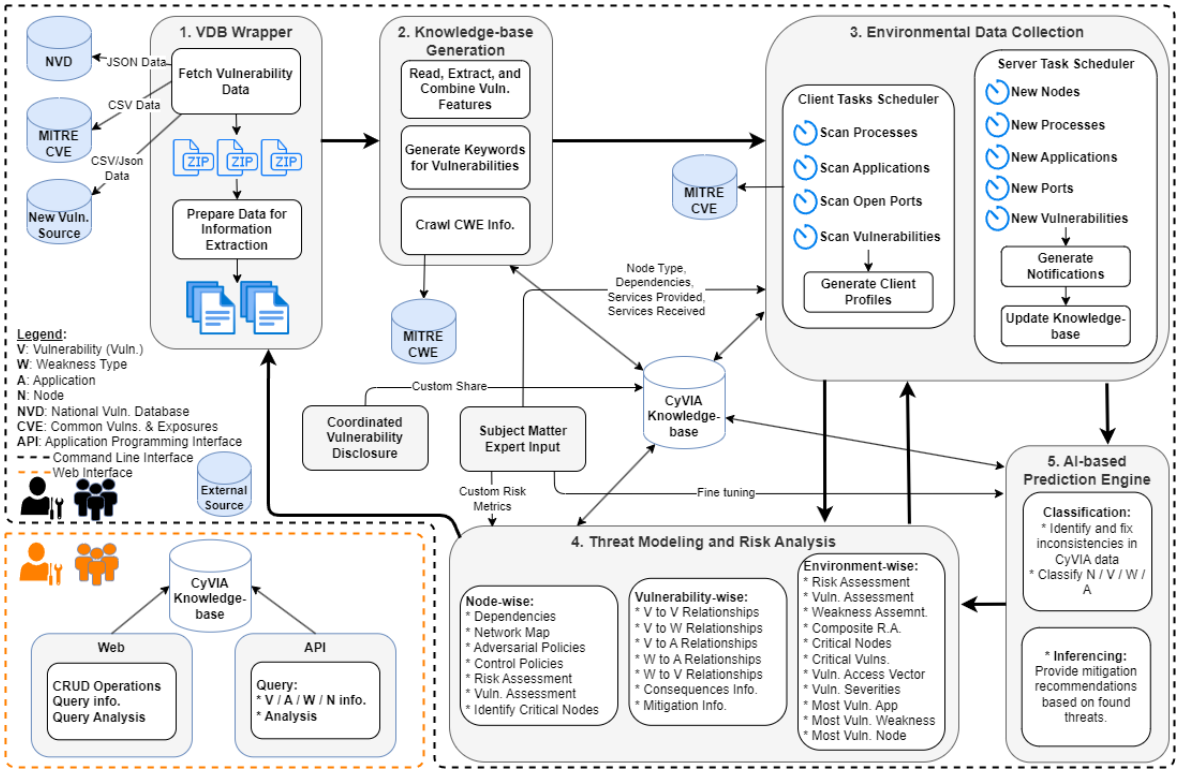

Figure 7.2: CyVIA Architecture

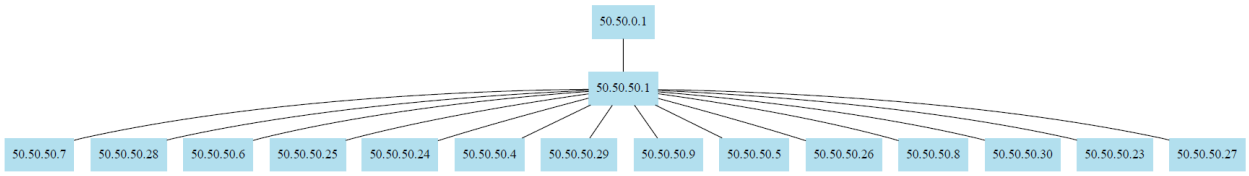

Figure 7.3: CyVIA Network Dependencies Map

This section aims to conduct a SWOT analysis of our proposed framework, CyVIA. Specifically, we aim to evaluate the strengths, weaknesses, opportunities, and threats of CyVIA as a whole, instead of solely focusing on its AI engine. The subsequent subsections will elaborate on each of these aspects. Fig. 7.2 is presented here to provide an overview of the overall architecture of CyVIA, which is crucial for contextualizing the SWOT analysis and identifying areas where improvements may be required.

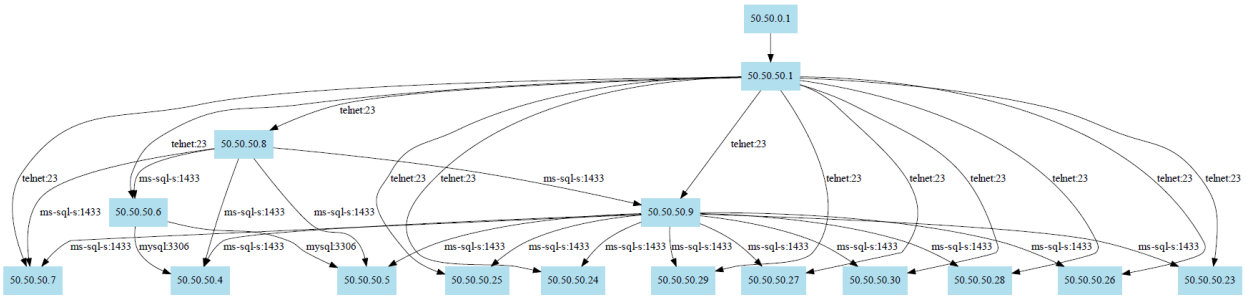

Figure 7.4: CyVIA Node Service Dependencies Map

7.3.1 Strengths

A standard risk assessment framework typically involves the use of one or more tools or frameworks to gather data, after which cyber defenders evaluate the results to determine the severity of the risk and whether it is actionable or acceptable. However, with CyVIA, the entire process, from data gathering to analysis generation, is fully automated. CyVIA provides continuous risk monitoring and threat-centric analytics that capture changing network configurations, regardless of time or space constraints. CyVIA offers the following key advantages:

(a) Identify network and service dependencies within cyber infrastructures (Fig. 7.3 and Fig. 7.4).

(b) Evaluate individual nodes and the infrastructure as a whole for risk, taking into account implemented security controls and the risk from internal and external adversaries.

(c) Identify vulnerabilities within the operating systems and running applications of network nodes, and provide information on associated consequences and mitigation strategies.

(d) Classify the vulnerabilities based on type of weakness, severity, and access vectors.

(e) Infrastructure-based top 10 most vulnerable products (Table 7.5).

(f) Highlight products based on mean severity, vulnerability scores, and number of vulnerabilities.

(g) Identify high-priority vulnerabilities and weakness types that defenders should prioritize for remediation.

(h) Generate relational analyses between the found vulnerabilities, products, and weakness types. Fig. 7.5 illustrates the relationships between product and weakness types identified in the evaluated infrastructure.

(i) Monitor for anomalous user activities based on recent adversarial trends.

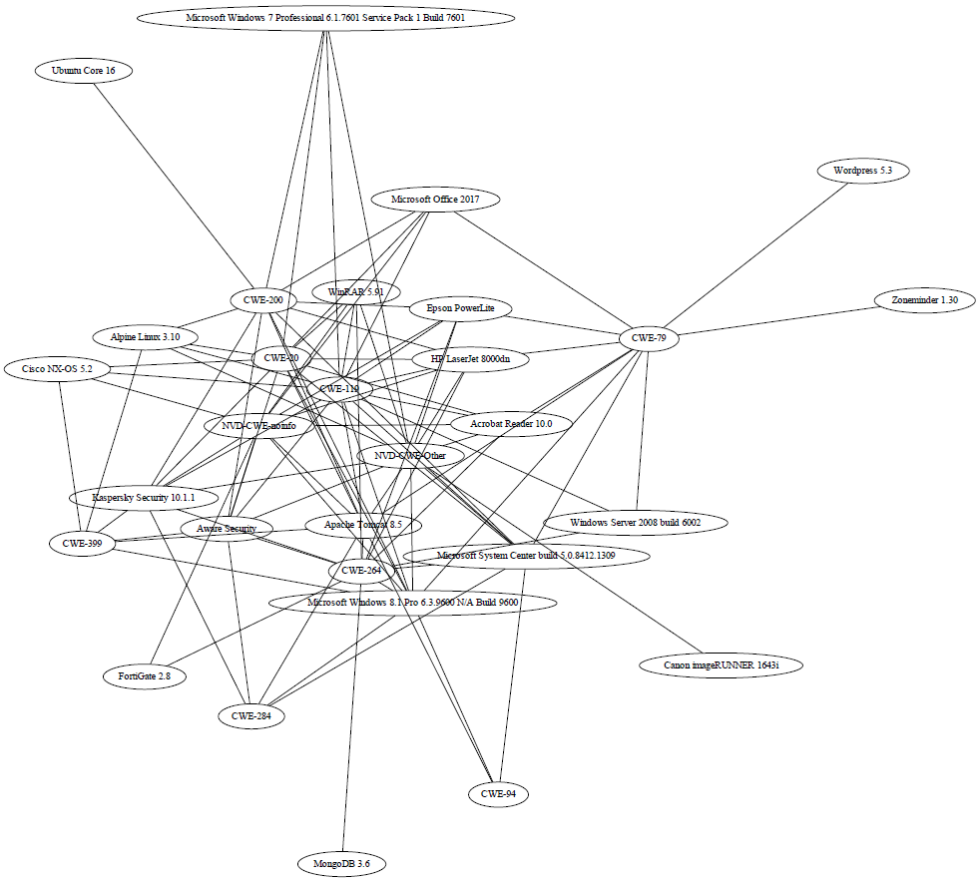

Figure 7.5: Weakness Types to Products Relationships

7.3.2 Weaknesses

We aim to identify and discuss the weaknesses of our proposed system by exploring its underlying assumptions and limitations.

Assumptions

In the development of CyVIA, we have made the following assumptions:

(a) We assume that the CVE features, such as CVSS scores, CWE IDs, Severity values, etc., stored in the National Vulnerability Database (NVD) are accurately assigned.

(b) As the NVD is populated with data from MITRE, and the Common Weakness Enumeration (CWE) is managed by MITRE, we rely on MITRE for the final CWE features.

(c) We match the final list of possible vulnerabilities with MITRE's CVE search engine.

(d) We use a Raspberry Pi device as a representative of IoT devices that communicate with different sensors for data collection.

(e) Due to resource limitations, we were unable to deploy CyVIA on a live large network. However, we have conducted multiple trials of CyVIA on various network clusters containing different versions of Microsoft Windows and Linux. We are confident that it can be deployed on any large network.

Limitations

At this stage, CyVIA has the following limitations:

(a) The local agent can capture information from nodes running Windows 7 onwards, with PowerShell script execution enabled. For Linux, we have tested agents on Ubuntu, Kali, Debian, and Fedora.

(b) The services offered by nodes are captured through remote scanning, but the administrator must identify the nodes that are utilizing these services.

We acknowledge these limitations and aim to address them in the future research.

7.3.3 Opportunities

At this time, we have the API, and how someone can take advantage and expand on this. the outcome can feed to a new directions or research problem. dont use in the future...

In our future work, we aim to develop a website that facilitates global interaction between cyber defenders and CyVIA. This website will serve as a platform for coordinated vulnerability disclosure processes and for obtaining information related to specific threats. We believe that this will enable a more comprehensive and collaborative approach to threat and vulnerability management. Furthermore, we intend to focus on enhancing the following two key components and an optional upgrade of the CyVIA framework to make it more robust:

Abnormal User Behaviour

CyVIA maintains data for running processes for each user and identifies any abnormal user behavior. This data includes CPU, memory, and disk usage for each process. Upon linking this real-time data with a trained machine learning model, we can highlight users running unfamiliar processes with 1) unusual names, 2) high resource usage, and 3) connections to suspicious outside domains or remote servers. These anomalous processes can be flagged for further investigation or terminated to prevent potential harm. The use of machine learning algorithms will allows CyVIA to adapt to new threats and improve its detection capabilities over time.

Enriching Threat Database

The utilization of online forums, social media, conferences and events, chat groups and instant messaging, etc., are common platforms for threat and vulnerability discussion, in addition to vulnerability databases. By augmenting the capability of CyVIA AI-engine's context-aware summary generator, it can collect and collate information from these sources to provide comprehensive contextual information about the risks and how they can be addressed. This process may also entail validating the authenticity of the sources and the data captured to ensure the accuracy and reliability of the insights provided by CyVIA.

Optional Upgrade

CyVIA's prototype is developed using Python, a programming language that is typically regarded as slower than other languages due to its interpreted nature. To improve performance, it is suggested that CyVIA be translated into C or C++, which are widely known as very fast programming languages, or Java, a programming language recognized for its high performance capabilities. By doing so, the overall speed of CyVIA can be significantly enhanced.

7.3.4 Threats

CyVIA can be exploited in a number of ways:

(a) It utilizes an open-source, NoSQL document-oriented database, namely Apache CouchDB. By gaining access to the database, the stored information can be manipulated, leading to a change in the analytical results.

(b) The communication between client agents and server agent is not encrypted, which means that this communication can be altered with misleading information.

(c) There is no authentication process for any new nodes on the network. Any new node having access to a client agent can add false information to the database, thus

generating a change in the overall risk for the entire network.

(d) The current version of CyVIA API can be secured by using a combination of authentication, authorization, encryption, rate limiting, input validation, and other security best practices to ensure the safety and privacy of data being transmitted through the API.

A combination of strong authentication and encryption mechanisms can make CyVIA more resilient against these threats.

7.4 Summary

Risk mitigation in cyber infrastructures is imperative due to the escalating frequency and complexity of cyberattacks, which can have severe consequences for organizations, including damage to reputation, operations, finances, and even human lives. To assist in this effort, we present an AI-based prediction engine to identify and infer detected risks within a given cyber infrastructure. This work satisfies the research question Q2 under objective C. The engine's primary responsibility is to provide cyber defenders with information on the risks' severity and mitigation strategies. Additionally, with the aid of the CyVIA API, defenders can engage with the engine to obtain additional insights on the risks. Going forward, we aim to make the API publicly accessible to enable individuals to interact with and learn more about the latest trends in cybersecurity. We also plan to utilize this framework as a coordinated vulnerability disclosure process through a website that will allow external cyber defenders to interact with CyVIA. This work is submitted to a conference and is under review.

Table 7.3: Predicted Attack Types

	Attacks	**Count**
1	Code Injection	34,511
2	Sensitive Data Exposure	15,000
3	Directory Traversal	12,882
4	Cross-site Scripting	12,362
5	Unauthorized Access	10,835
6	Buffer Overflow	8,943
7	Denial of Service	8,103
8	Memory Based Attack	5,567
9	Privilege Escalation	4,129
10	Man-in-the-middle	1,708
11	Unknown Attack	1,145
12	Server-side Request Forgery	418
13	Credentials	300
14	Web Session Cookie	41
15	Command and Control	32
16	Host Redirection	29
17	Disabling Security Tools	22
18	Brute Force Attack	6

Table 7.4: Node-wise Vulnerabilities, Affected Products, MITRE and CyVIA Attack Types

Node	**CVEs**	**APs**	**MITRE**	**CyVIA**
Raspbian	133,298	7,266	215	18
Debian10	48,037	3,978	211	18
Win2016	12,211	663	113	13
Win2012	10,102	663	115	14
Win8	4,360	306	105	14
CentOS	2,967	239	63	13
Win10	2,609	293	98	13
Ubuntu16	2,244	122	93	12
Win7	1,731	162	90	12
Fedora33	1,482	170	44	12
Win11	1,030	151	89	12
Ubuntu20	490	64	64	12
Ubuntu18	259	41	61	12
openSUSE15	103	7	18	10
Router1	27	6	10	6

Table 7.5: CyVIA Infrastructure-based Top 10 Most Vulnerable Products

S#	**Product**	**CVEs**	**CWEs**
1	Microsoft MPI ...	6,377	97
2	jackd 5+nmu1	3,070	87
3	chromium 90.0.4430...	1,468	59
4	Windows 8.1 Enterprise	1,107	62
5	Windows Server 2012 R2	949	42
6	ssh 1:7.9p1-10	748	73
7	SQL Server 2017	640	37
8	zip 3.0-11+b1	584	54
9	SQL Server 2017	516	14
10	SQL Server 2017	516	14

Chapter 8

Conclusion and Future Work

8.1 Conclusion

With the increasing heterogeneity in numbers and types of devices in modern cyber infrastructures, cyber defenders face numerous known and unknown challenges. To assist cyber defenders in keeping up with such scenarios, we present a three-dimensional security framework called Cyber-threats and Vulnerability Information Analyzer (CyVIA). CyVIA provides continuous risk evaluation of cyber infrastructures, incorporating an embedded quantitative risk assessment model that considers asset-wise and overall risks, as well as risk propagation among dependent nodes. Additionally, CyVIA integrates major Vulnerability Databases (VDBs) for comprehensive cyber-threat analysis, utilizing a multi-formatted knowledge-base generated from vulnerability reports to identify critical vulnerabilities within a target cyber infrastructure. CyVIA also features an AI-based prediction engine that concludes found vulnerabilities into applicable attack types and suggests mitigation strategies. My future research plan includes adding security and encryption to the CyVIA framework, evaluating CyVIA on a live network to understand user patterns, and conducting extensive research in other areas as discussed in the following section.

8.2 Future Research Plan

In the future, I want to perform an extensive study on the following topics to make CyVIA more robust and adaptable.

8.2.1 AI-powered Anomaly Detection

CyVIA has the capability to capture running processes and monitor memory utilization of each running process in real-time. As part of my research plan, I intend to deploy CyVIA on a live network to capture live user activities and gain insights into various user behaviors during normal and peak hours. By analyzing the captured data, we can label malicious user processes and leverage it to train an AI-powered Anomaly Detection model that can flag potentially malicious processes in real-time. This will enable cyber defenders to quickly identify and isolate potentially infected or compromised nodes on the network, enhancing the overall cybersecurity posture of the system.

8.2.2 Addressing the Inconsistencies within Vulnerability Databases

As discussed in Chapter 4, several studies have emphasized the need to address inconsistencies within vulnerability data available on various online vulnerability databases. CyVIA's AI-based prediction engine is trained to predict attack types based on the given vulnerability descriptions. In Chapter 7, Table 7.2, we compare CyVIA's predicted attack types with the attack types defined by MITRE. This model can be further refined to label the missing vulnerability types in MITRE data that are originally classified as NVD-CWE-noinfo and NVD-CWE-other. Table 8.1 highlights the inconsistencies within the NVD data that can be addressed, and the corrected dataset can be made available for others to use via the CyVIA API.

8.2.3 Collaborative Cyber Threat Intelligence

The original idea of including CTI (Cyber Threat Intelligence) and CyVIA API as part of the CyVIA framework was to spread awareness not only among local cyber defenders but also on a global scale, where cyber defenders from different regions can interact, gain information, discuss, and collaborate on cybersecurity issues to keep the CyVIA knowledge base updated with the most recent trends. This may also include a coordinated vulnerabil-

Table 8.1: NVD Inconsistencies

Description	**CVE Count**	**Replacement**
CWE ID classified as NVD-CWE-noinfo	16,534	-
CWE ID classified as NVD-CWE-Other	27,178	-
Missing Severity value	9,215	N/A
Missing CVSS Score V2	9,215	-1
Missing CVSS Score V3	82,591	-1
Missing Vulnerability Access Vector	9,215	N/A
Missing User Interaction Required	10,242	N/A
Missing Language information	9,172	N/A
Total CVE records	158,450	-

ity disclosure process, where organizations can contribute to the knowledge base through a reporting facility that allows them to reach out to specific hardware/software vendors to address the identified threats.

These future directions and contributions are expected to facilitate the widespread adoption of the CyVIA framework, enabling cyber defenders to make informed decisions based on the derived analysis and promptly mitigate identified threats. CyVIA framework aims to enhance cyber situational awareness, promote collaboration among cyber defenders, and enable effective risk mitigation strategies. These efforts are anticipated to improve the overall cybersecurity posture of cyber infrastructures and empower defenders in addressing emerging cyber threats efficiently.

Chapter 9

Publications Resulted from this Dissertation

1) Malik, Adeel A., and Deepak K. Tosh. "Quantitative risk modeling and analysis for large-scale cyber-physical systems." 2020 29th International Conference on Computer Communications and Networks (ICCCN). IEEE, 2020.

2) Malik, Adeel A., and Deepak K. Tosh. "Robust Cyber-threat and Vulnerability Information Analyzer for Dynamic Risk Assessment." 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). IEEE, 2021.

3) Malik, Adeel A., and Deepak K. Tosh. "Dynamic Risk Assessment and Analysis Framework for Large-Scale Cyber-Physical Systems." EAI Endorsed Transactions on Security and Safety 8.30 (2022).

4) Malik, Adeel A., and Deepak K. Tosh. "Dynamic Vulnerability Classification for Enhanced Cyber Situational Awareness." (SYSCON). IEEE, 2023.

5) Malik, Adeel A., and Deepak K. Tosh. "Towards Developing a Scalable Cyber Risk Assessment and Mitigation Framework.", Under submission.

References

[1] NIST, *CVSS Severity Distribution Over Time*, 2022 (last retrieved Jan 4 2019).

[2] S. Musman and A. Turner, "A game theoretic approach to cyber security risk management," *The Journal of Defense Modeling and Simulation*, vol. 15, no. 2, pp. 127–146, 2018.

[3] PwC, *The Global State of Information Security Survey 2018*, 2018 (last retrieved Jan 2 2022).

[4] S. Musman and A. Turner, "A game theoretic approach to cyber security risk management," *The Journal of Defense Modeling and Simulation*, vol. 15, pp. 127–146, Apr. 2018.

[5] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[6] NIST, *National Vulnerability Database*, 2022 (last retrieved Nov 22 2022).

[7] C. Coalition, *Policy Priorities for Coordinated Vulnerability Disclosure and Handling*, 2019 (last retrieved Jan 2 2022).

[8] NCSC, *CVD Guideline*, 2018 (last retrieved Jan 2 2022).

[9] GFCE, *GFCE Global Good Practices - CVD*, 2017 (last retrieved Jan 2 2022).

[10] ENISA, *Economics of Vulnerability Disclosure*, 2018 (last retrieved Jan 2 2022).

[11] CMU, *Software Engineering Institute*, 2022 (last retrieved Jan 2 2022).

[12] Cisco, *Cisco Cybersecurity Report*, 2019 (last retrieved Jan 2 2022).

[13] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212–223, 2018.

[14] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: Analysis, challenges and solutions," *Computers & Security*, vol. 68, pp. 81–97, 2017.

[15] NIST, *Federal Information Security Modernization Act*, 2022 (last retrieved Jan 2 2022).

[16] HIPAA, *HIPAA Risk Assessment*, 2022 (last retrieved Jan 2 2022).

[17] ISMS, *Information Security Risk Management*, 2022 (last retrieved Jan 2 2022).

[18] NIST, *NIST's Guide for Conducting Risk Assessments*, 2012 (last retrieved Jan 2 2022).

[19] NIST, *NVD JSON Feeds*, 2021 (Online; accessed May 20, 2021). `https://nvd.nist.gov/vuln/data-feeds#JSON_FEED`.

[20] MITRE, *MITRE CWE List Version 4.2*, 2021 (Online; accessed May 20, 2021). `https://cwe.mitre.org/data/downloads.html`.

[21] S. Weerawardhana, S. Mukherjee, I. Ray, and A. Howe, "Automated extraction of vulnerability information for home computer security," in *International Symposium on Foundations and Practice of Security*, pp. 356–366, Springer, 2014.

[22] D. Mu, A. Cuevas, L. Yang, H. Hu, X. Xing, B. Mao, and G. Wang, "Understanding the reproducibility of crowd-reported security vulnerabilities," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 919–936, 2018.

[23] K. Kritikos, K. Magoutis, M. Papoutsakis, and S. Ioannidis, "A survey on vulnerability assessment tools and databases for cloud-based web applications," *Array*, vol. 3, p. 100011, 2019.

[24] I. Chalvatzis, D. A. Karras, and R. C. Papademetriou, "Evaluation of security vulnerability scanners for small and medium enterprises business networks resilience towards risk assessment," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pp. 52–58, IEEE, 2019.

[25] B. Mburano and W. Si, "Evaluation of web vulnerability scanners based on owasp benchmark," in *2018 26th International Conference on Systems Engineering (ICSEng)*, pp. 1–6, IEEE, 2018.

[26] M. El, E. McMahon, S. Samtani, M. Patton, and H. Chen, "Benchmarking vulnerability scanners: An experiment on scada devices and scientific instruments," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 83–88, IEEE, 2017.

[27] K. Umezawa, Y. Mishina, and K. Takaragi, "Threat analyses using vulnerability databases—possibility of utilizing past analysis results—," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, IEEE, 2019.

[28] H. Ghani, J. Luna, A. Khelil, N. Alkadri, and N. Suri, "Predictive vulnerability scoring in the context of insufficient information availability," in *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*, pp. 1–8, IEEE, 2013.

[29] A. Anwar, A. Abusnaina, S. Chen, F. Li, and D. Mohaisen, "Cleaning the NVD: Comprehensive Quality Assessment, Improvements, and Analyses," vol. 13, 2020.

[30] V. H. Nguyen and F. Massacci, "The (un) reliability of nvd vulnerable versions data: An empirical experiment on google chrome vulnerabilities," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 493–498, 2013.

[31] Y. Dong, W. Guo, Y. Chen, X. Xing, Y. Zhang, and G. Wang, "Towards the detection of inconsistencies in public security vulnerability reports," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 869–885, 2019.

[32] S. Christey and B. Martin, "Buying into the bias: Why vulnerability statistics suck," *BlackHat, Las Vegas, USA, Tech. Rep*, vol. 1, 2013.

[33] A. Tripathi and U. K. Singh, "Taxonomic analysis of classification schemes in vulnerability databases," in *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pp. 686–691, IEEE, 2011.

[34] P. Johnson, R. Lagerström, M. Ekstedt, and U. Franke, "Can the common vulnerability scoring system be trusted? a bayesian analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 1002–1015, 2016.

[35] R. Wang, L. Gao, Q. Sun, and D. Sun, "An improved cvss-based vulnerability scoring mechanism," in *2011 Third International Conference on Multimedia Information Networking and Security*, pp. 352–355, IEEE, 2011.

[36] Redscan, *Redscan Report 2021*, 2021 (Online; accessed Aug 1, 2021). `https://bit.ly/3gtr0a1`.

[37] R. Syed, "Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system," *Information & Management*, vol. 57, no. 6, p. 103334, 2020.

[38] G. Roldán-Molina, M. Almache-Cueva, C. Silva-Rabadão, I. Yevseyeva, and V. Basto-Fernandes, "A comparison of cybersecurity risk analysis tools," *Procedia computer science*, vol. 121, pp. 568–575, 2017.

[39] Webroot, *Social Engineering, examples & prevention*, 2021 (Online; accessed Aug 1, 2021). `https://bit.ly/3tIJirG`.

[40] P. Patil, "Artificial intelligence in cyber security," *Int. J. Res. Comput. Appl. Robot*, vol. 4, no. 5, pp. 1–5, 2016.

[41] N. Kaloudi and J. Li, "The ai-based cyber threat landscape: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–34, 2020.

[42] R. Prasad and V. Rohokale, "Artificial intelligence and machine learning in cyber security," in *Cyber Security: The Lifeline of Information and Communication Technology*, pp. 231–247, Springer, 2020.

[43] R. Xiao, H. Zhu, C. Song, X. Liu, J. Dong, and H. Li, "Attacking network isolation in software-defined networks: New attacks and countermeasures," in *2018 27th international conference on computer communication and networks (ICCCN)*, pp. 1–9, IEEE, 2018.

[44] H. Sedjelmaci, F. Guenab, S.-M. Senouci, H. Moustafa, J. Liu, and S. Han, "Cyber security based on artificial intelligence for cyber-physical systems," *IEEE Network*, vol. 34, no. 3, pp. 6–7, 2020.

[45] Capgemini, *AI in Cybersecurity*, 2019 (last retrieved Jan 2 2022).

[46] D. Bisson, *5 Social Engineering Attacks to Watch Out For*, 2019 (last retrieved Jan 2 2022).

[47] G. Jakobson, "Mission cyber security situation assessment using impact dependency graphs," in *14th International Conference on Information Fusion*, pp. 1–8, IEEE, 2011.

[48] M. U. Aksu, M. H. Dilek, E. İ. Tatlı, K. Bicakci, H. I. Dirik, M. U. Demirezen, and T. Aykır, "A quantitative cvss-based cyber security risk assessment methodology for it systems," in *2017 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8, IEEE, 2017.

[49] B. M. Ayyub, W. L. McGill, and M. Kaminskiy, "Critical Asset and Portfolio Risk Analysis: An All-Hazards Framework," *Risk Analysis*, vol. 27, no. 4, pp. 789–801, 2007.

[50] S. Musman, A. Temin, M. Tanner, D. Fox, and B. Pridemore, "Evaluating the impact of cyber attacks on missions," *MITRE Corporation*, 2010.

[51] R. Anupindi, S. Chopra, S. D. Deshmukh, J. A. Van Mieghem, and E. Zemel, *Managing business process flows: principles of operations management*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[52] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "A systems engineering approach for crown jewels estimation and mission assurance decision making," in *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pp. 210–216, IEEE, 2011.

[53] S. Musman and A. Temin, "A cyber mission impact assessment tool," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, pp. 1–7, IEEE, 2015.

[54] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams, "Cauldron mission-centric cyber situational awareness with defense in depth," in *2011-MILCOM 2011 Military Communications Conference*, pp. 1339–1344, IEEE, 2011.

[55] R. Garrido-Pelaz, L. González-Manzano, and S. Pastrana, "Shall we collaborate?: A model to analyse the benefits of information sharing," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, pp. 15–24, ACM, 2016.

[56] A. Rutkowski, Y. Kadobayashi, I. Furey, D. Rajnovic, R. Martin, T. Takahashi, C. Schultz, G. Reid, G. Schudel, M. Hird, *et al.*, "Cybex: The cybersecurity infor-

mation exchange framework (x. 1500)," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 5, pp. 59–64, 2010.

[57] D. Tosh, S. Sengupta, C. Kamhoua, K. Kwiat, and A. Martin, "An evolutionary game-theoretic framework for cyber-threat information sharing," in *2015 IEEE International Conference on Communications (ICC)*, pp. 7341–7346, IEEE, 2015.

[58] P. Naghizadeh and M. Liu, "Inter-temporal incentives in security information sharing agreements," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[59] H. Luiijf and A. Kernkamp, "Sharing cyber security information: Good practice stemming from the dutch public-private-participation approach," 2015.

[60] United States., *Guideline for automatic data processing risk analysis.* Federal information processing standards publication, FIPS pub ; 65, Washington: Dept. of Commerce, National Bureau of Standards, 1979.

[61] "How Much Is Enough? A Risk-Management Approach to Computer Security."

[62] D. L. Buckshaw, G. S. Parnell, W. L. Unkenholz, D. L. Parks, J. M. Wallner, and O. S. Saydjari, "Mission Oriented Risk and Design Analysis of Critical Information Systems," *Military Operations Research*, vol. 10, no. 2, pp. 19–38, 2005.

[63] R. Anderson, "Why information security is hard - an economic perspective," in *Seventeenth Annual Computer Security Applications Conference*, pp. 358–365, Dec. 2001. ISSN: null.

[64] R. Pal, L. Golubchik, K. Psounis, and P. Hui, "Will cyber-insurance improve network security? a market analysis," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 235–243, IEEE, 2014.

[65] R. Aliyev and L. Peñalver, "Analyzing vulnerability databases," in *Proc. 10th IEEE International Conference on Application of Information and Communication Technologies (AICI)*, 2016.

[66] G. Yun-hua and L. Pei, "Design and research on vulnerability database," in *2010 Third International Conference on Information and Computing*, vol. 2, pp. 209–212, IEEE, 2010.

[67] Y. Zhang, Y. Li, W. Deng, K. Huang, and C. Yang, "Complex networks identification using bayesian model with independent laplace prior," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 1, p. 013107, 2021.

[68] Y. Zhang, C. Yang, K. Huang, M. Jusup, Z. Wang, and X. Li, "Reconstructing heterogeneous networks via compressive sensing and clustering," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.

[69] P. Institute, *Ponemon Library*, 2021 (Online; accessed Aug 1, 2021). `https://www.ponemon.org/research/ponemon-library/`.

[70] P. Institute, *Ponemon Study*, 2021 (Online; accessed Aug 1, 2021). `https://bwnews.pr/3BOFOrR`.

[71] C. Ventures, *Cybersecurity Ventures' 2019 Cybersecurity Market Report*, 2021 (Online; accessed Aug 1, 2021). `https://bit.ly/3f1r3Iy`.

[72] M. Benz and D. Chatterjee, "Calculated risk? a cybersecurity evaluation tool for smes," *Business Horizons*, vol. 63, no. 4, pp. 531–540, 2020.

[73] R. Sabillon, J. Serra-Ruiz, V. Cavaller, and J. Cano, "A comprehensive cybersecurity audit model to improve cybersecurity assurance: The cybersecurity audit model (csam)," in *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, pp. 253–259, IEEE, 2017.

[74] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, S. Sastry, *et al.*, "Challenges for securing cyber physical systems," in *Workshop on future directions in cyber-physical systems security*, vol. 5, Citeseer, 2009.

[75] C. H. Liu and Y. Zhang, *Cyber physical systems: architectures, protocols and applications*, vol. 22. CRC Press, 2015.

[76] H. I. Kure, S. Islam, and M. A. Razzaque, "An integrated cyber security risk management approach for a cyber-physical system," *Applied Sciences*, vol. 8, no. 6, p. 898, 2018.

[77] C. Sauerwein, C. Sillaber, A. Mussmann, and R. Breu, "Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives," 2017.

[78] V. Mavroeidis and S. Bromander, "Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence," in *2017 European Intelligence and Security Informatics Conference (EISIC)*, pp. 91–98, IEEE, 2017.

[79] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, p. 101589, 2019.

[80] D. K. Tosh, M. Molloy, S. Sengupta, C. A. Kamhoua, and K. A. Kwiat, "Cyber-investment and cyber-information exchange decision modeling," in *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pp. 1219–1224, IEEE, 2015.

[81] M. Haustein, H. Sighart, D. Titze, and P. Schoo, "Collaboratively exchanging warning messages between peers while under attack," in *2013 International Conference on Availability, Reliability and Security*, pp. 726–731, IEEE, 2013.

[82] G. Fisk, C. Ardi, N. Pickett, J. Heidemann, M. Fisk, and C. Papadopoulos, "Privacy principles for sharing cyber security data," in *2015 IEEE Security and Privacy Workshops*, pp. 193–197, IEEE, 2015.

[83] T. Sander and J. Hailpern, "Ux aspects of threat information sharing platforms: An examination & lessons learned using personas," in *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*, pp. 51–59, 2015.

[84] MITRE, *Common Attack Pattern Enumeration and Classification*, 2021 (Online; accessed Aug 1, 2021). `https://capec.mitre.org/`.

[85] MITRE, *MITRE ATT&CK*, 2015 (Online; accessed May 20, 2021). `https://attack.mitre.org/`.

[86] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickles, A. G. Pennington, and C. B. Thomas, *MITRE ATT&CK™ : DESIGN AND PHILOSOPHY*, 2018 (Online; accessed Aug 1, 2021). `https://bit.ly/2SZtNy7`.

[87] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," *Technical report*, 2018.

[88] L. Chan, I. Morgan, H. Simon, F. Alshabanat, D. Ober, J. Gentry, D. Min, and R. Cao, "Survey of ai in cybersecurity for information technology management," in *2019 IEEE technology & engineering management conference (TEMSCON)*, pp. 1–8, IEEE, 2019.

[89] Z. Zhang, H. Ning, F. Shi, F. Farha, Y. Xu, J. Xu, F. Zhang, and K.-K. R. Choo, "Artificial intelligence in cyber security: research advances, challenges, and opportunities," *Artificial Intelligence Review*, pp. 1–25, 2021.

[90] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *arXiv preprint arXiv:1802.07228*, 2018.

[91] B. Sagar, S. Niranjan, N. Kashyap, and D. Sachin, "Providing cyber security using artificial intelligence–a survey," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 717–720, IEEE, 2019.

[92] Microsoft, *Protecting the protector: Hardening machine learning defenses against adversarial attacks*, 2018 (last retrieved Jan 2 2022).

[93] M. Taddeo, T. McCutcheon, and L. Floridi, "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Machine Intelligence*, vol. 1, no. 12, pp. 557–560, 2019.

[94] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35, IEEE, 2018.

[95] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*, pp. 284–293, PMLR, 2018.

[96] C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," *arXiv preprint arXiv:1808.10307*, 2018.

[97] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

[98] A. A. Malik and D. K. Tosh, "Quantitative risk modeling and analysis for large-scale cyber-physical systems," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, IEEE, 2020.

[99] A. A. Malik and D. K. Tosh, "Robust cyber-threat and vulnerability information analyzer for dynamic risk assessment," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pp. 168–173, IEEE, 2021.

[100] A. A. Malik and D. K. Tosh, "Dynamic risk assessment and analysis framework for large-scale cyber-physical systems," *EAI Endorsed Transactions on Security and Safety: Online First*, 1 2022.

[101] MITRE, *Search CVE List*, 2022 (last retrieved Nov 22 2022).

[102] Yellowbrick, *Elbow Method*, 2022 (last retrieved Nov 22 2022).

[103] scikit learn, *Latent Dirichlet Allocation*, 2022 (last retrieved Nov 22 2022).

[104] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[105] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "Flair: An easy-to-use framework for state-of-the-art nlp," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pp. 54–59, 2019.

[106] G. Stanford, *Global Vectors for Word Representation*, 2022 (last retrieved Nov 22 2022).

[107] A. A. Malik and D. K. Tosh, "Dynamic risk assessment and analysis framework for large-scale cyber-physical systems," *EAI Endorsed Transactions on Security and Safety*, vol. 8, no. 30, 2022.

[108] MITRE, *MITRE API*, 2023 (Online; accessed Mar 1, 2023). https://cveawg.mitre.org/api/cve/.

[109] MITRE, *MITRE CVEs*, 2023 (Online; accessed Mar 1, 2023). https://cve.mitre.org/.

Appendix A

CyVIA User Guide

The standard risk assessment framework usually requires the use of tools or frameworks to collect data, followed by manual evaluation by cyber defenders to assess risk severity and determine if action needs to be taken. In contrast, CyVIA introduces a fully automated process that covers the entire risk assessment workflow, from data gathering to analysis generation. It enables continuous risk monitoring and provides threat-centric analytics that can adapt to changing network configurations without being restricted by time or space limitations. The key advantages of CyVIA include:

- Identify network and service dependencies within cyber infrastructures.
- Evaluate individual nodes and the infrastructure as a whole for risk, taking into account implemented security controls and the risk from internal and external adversaries.
- Identify vulnerabilities within the operating systems and running applications of network nodes, and provide information on associated consequences and mitigation strategies.
- Classify the vulnerabilities based on the type of weakness, severity, and access vectors.
- Infrastructure-based top 10 most vulnerable products.
- Highlight products based on mean severity, vulnerability scores, and the number of vulnerabilities.

- Identify high-priority vulnerabilities and weakness types that defenders should prioritize for remediation.
- Generate relational analyses between the found vulnerabilities, products, and weakness types.
- Monitor for anomalous user activities based on recent adversarial trends.

Setup Instructions

You will need the following on your server machine:

- Graphviz 2.38 (should be available in the path environment variables)
- Flask 1.1.2 (for CyVIA API)
- Python 3
- CouchDB 3.1.1
- Jupyter Notebook

Requirements file

aniso8601==8.0.0

certifi==2022.9.24

charset-normalizer==2.1.1

click==7.1.2

colorama==0.4.6

CouchDB==1.2

CouchDB2==1.13.0

Flask==1.1.2

Flask-RESTful==0.3.8

Flask-SQLAlchemy==2.4.3

idna==3.4

itsdangerous==1.1.0

Jinja2==2.11.2

joblib==1.2.0

lxml==4.9.1

MarkupSafe==1.1.1

numpy==1.23.4

pandas==1.5.1

pynput==1.7.6

python-dateutil==2.8.2

pytz==2020.1

requests==2.28.1

scipy==1.9.3

six==1.15.0

SQLAlchemy==1.3.18

tqdm==4.64.1

urllib3==1.26.12

Werkzeug==1.0.1

Jupyter Notebooks

- 1_CWE_Master_Data_CouchDB.ipynb : Create Master Data for CWE referencing from MITRE.
- 2_Fetch_MITRE_CWE_CSV_Feeds.ipynb : Collect the latest CWE feeds from MITRE.
- 3_Fetch_NVD_JSON_Feeds.ipynb : Collect vulnerability data from NVD.
- 4_Prepare_Dataset.ipynb : Compile collected files and prepare CyVIA knowledgebase based on the found relationships between the data.
- 5_Network_Scanner.ipynb : Scans network for nodes and open ports.

- 6_Dependency_Mapper.ipynb : Maps service and network dependencies between the found network nodes.
- 7_Control_Mapper.ipynb : Evaluates network nodes for applied security controls.
- 8_Process_Monitor.ipynb : Monitors running processes on network nodes.
- 9_Scheduler.ipynb : Responsible for scheduling jobs to keep a check on updates and network activity.
- Node_Analysis.ipynb : Evaluates each network node and prepares the detailed report for each node.

Other Python files

- cyvia_api.py : API file
- agent_linux_v2.py : CyVIA agent for Linux nodes to collect node information and pass it to server agent.
- agent_windows_v2.py : CyVIA agent for Windows nodes.
- client_scheduler_v2.py : CyVIA scheduler to keep the agent timely running and communicating with the server.
- server_scheduler_v2 : Server side scheduler to interact with client scheduler and keep the server up to date.
- config.py : Server configuration.
- functions.py : Functions library for CyVIA.
- get-pip.py : If pip is not installed on your machine, you can use this file.
- process_scanner_v2.py : Process scanner for network nodes, works with the client scheduler file.

- Spinner.py : On Python notebooks, if it takes a long time, the spinner spins to let the user know there is a process working in the background.

Script files

- install_linux_req.sh : Installs required libraries on Linux network nodes for the agent to work.
- install_windows_req.bat : Installs required libraries on a Windows network node.
- linux_client_info.sh : Fetches Linux network node information for the Linux agent.
- win_client_info.psl : Fetches Windows network node information for the Windows agent. You may need to turn on the power shell execution on Windows nodes; see Turn on scripts on windows.txt.

Network setup

We tested CyVIA on a simulated network environment. We have used VMWare and VirtualBox. All network nodes should be reachable for the framework to generate analysis. The general process flow is as follows:

- Deploy the agents on network nodes, and ensure all nodes have Python and required libraries installed. Run the schedulers on nodes, and the scan process will start.
- On the server side, you may also run the server-side scheduler. Once the client and server-side schedulers communicate, the network node profiles will be created in the CyVIA knowledgebase.
- After this, the individual Jupyter Notebooks can be run on a need basis to see network analysis.

For further references and detailed information see CyVIA repository or go to the url: `https://github.com/trucyber/Risk-Assessment-Framework/blob/master/README.md`.

Appendix B

Curriculum Vitae

Adeel Ashraf Malik is currently a Ph.D. candidate in the Department of Computer Science at the University of Texas at El Paso (UTEP). He holds a Master of Science (M.S.) degree in Software Engineering from UTEP, which he obtained in 2018. His research interests encompass multiple areas, including cybersecurity, distributed systems, and artificial intelligence. During his tenure at UTEP, Adeel served as a Teaching and Research Assistant. Throughout his Ph.D., he has been involved in the design, development, implementation, and evaluation of a cyber risk assessment framework known as Cyber-threats and Vulnerability Information Analyzer (CyVIA). Additionally, Adeel has worked as a Software Engineering Intern at Lam Research Corporation, where he contributed to the development of an automation module for analyzing risk in various production tools. In the future, Adeel plans to remain in academia to further pursue his research interests.

Contact Information: adeelm@live.com