6-28-2023 2:00 PM

Framework for Assessing Information System Security Posture Risks

Syed Waqas Hamdani, *The University of Western Ontario*

Supervisor: Kontogiannis, Prof. Kostas, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

In today's data-driven world, Information Systems, particularly the ones operating in regulated industries, require comprehensive security frameworks to protect against loss of confidentiality, integrity, or availability of data, whether due to malice, accident or otherwise. Once such a security framework is in place, an organization must constantly monitor and assess the overall compliance of its systems to detect and rectify any issues found. This thesis presents a technique and a supporting toolkit to first model dependencies between security policies (referred to as controls) and, second, devise models that associate risk with policy violations. Third, devise algorithms that propagate risk when one or more policies are found to be non-compliant and fourth, propose a technique that evaluates the overall security posture risk of a system as a function of the non-compliant policies, the affected policies, and the time elapsed since these policy violations discovered but not have been mitigated yet. More specifically, the approach is based on modeling the dependencies between the different controls in the NIST 800.53 framework by compiling a dependency multi-graph, devising a fuzzy-reasoning-based risk assessment technique that traverses the dependency multi-graph and assigns an overall security exposure risk score when one or more controls fail, and finally a technique for identifying the strategies an attacker can use, given the failed controls, and for which an organization should defend itself. This approach allows organizations to obtain a bird's-eye view of their Information Systems' cyber security posture and help triage the security control checks by focusing on the most vulnerable parts of their Information System ecosystem.

Keywords

System Security, Risk Analysis, Compliance, Security Controls, NIST 800.53, MITRE ATT&CK, Software engineering, Security services, Software security engineering, Security Intelligence

Summary for Lay Audience

The thesis is about designing and developing a cyber security methodology to assess a system's compliance against specific standard security frameworks, especially when the system operates in heavily regulated industries. These frameworks, such as the NIST 800.53, provide a collection of prescribed policies a system must comply with and are known as security controls. The collective states of the controls at any given time, whether they are violated or not, define what we refer to as the security posture of the Information System. However, having a snapshot of the security posture is insufficient to protect information and verify compliance. We must also constantly assess the system's posture and ensure potential risks are mitigated within the time allowed. The work performed in this research focuses on developing a novel method for assessing a failed security control's impact on other controls and evaluating the overall cybersecurity risk in the presence of such failures. This way, organizations can determine whether there are any weak spots in their cyber security and fix them promptly. The research aims to help organizations protect their information and prevent attacks that could harm their business.

Co-Authorship Statement

I hereby declare that this thesis incorporates material that is the result of joint research, as follows:

- Chapters 4 and 5 of the thesis include the outcome of the publication, which have the following other co-authors: Prof. Kostas Kontogiannis, Chris Brealey (IBM Canada), and Alberto Giammaria (IBM USA). In all cases, only my primary contributions towards the earlier publication are included in this thesis, and the contribution of co-authors Prof. Kostas Kontogiannis, Chris Brealey (IBM Canada), and Alberto Giammaria (IBM USA) was primarily through Prof. Kostas Kontogiannis provided research supervision, assistance in theoretical framework development, conceptualizing control dimensions, EICRS age dimension formula, and experiments design & analysis; Chris Brealey contributed expert feedback on theoretical model development; Alberto Giammaria provided necessary feedback and resources for this research.

I am aware of the University of Western Ontario Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to the co-authors of this research work. I want to thank my supervisor Prof. Kostas Kontogiannis for his invaluable guidance, support, and generosity throughout the completion of this research. I also extend my gratitude to the industry experts at IBM Canada, Chris Brealey, and IBM USA, Alberto Giammaria, for their valuable comments, technical discussions, and resource contributions to this work. Your expertise and insight have been instrumental in shaping the direction and quality of this thesis.

Additionally, I would like to extend my heartfelt appreciation to those who have been with me throughout this journey. Firstly, my parents, Zaitoon Bibi (Api), Meh Jabeen Iftikhar (Mother) & Iftikhar Ali Shah (Father) and fiancé Rubina Ahmad have been my inexhaustible sources of support and inspiration. Without your love, patience, and encouragement, I could not have achieved what I have today. Furthermore, I am forever grateful for your unwavering faith in me and willingness to stand by me through thick and thin. Your selflessness and dedication are the reasons for my success, and I promise to continue to work hard and make you proud.

Finally, I would like to express my appreciation to the University of Western Ontario, thesis examiners, and everyone else who has played a part in my journey, from my friends and colleagues to my mentors and educators in Canada and back home in Pakistan. Thank you for the lessons, memories, and countless ways you have contributed to my growth and development.

Special love to:

[+] Nasir Bashir / Iftikhar Ahmad – (My High School Teachers)

[+] Dua Fatima / Adiva Anees (Nieces)

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

IS – Information System(s)

CRSA – Cyber Risk Situational Awareness

AI – Artificial Intelligence

GRC – Governance, Risk, and Compliance

SIEM – Security Information and Event Management

CISA – Cybersecurity & Infrastructure Security Agency

AWARE – Agency-Wide Adaptive Risk Enumeration

NIST – National Institute of Standards and Technology

PCI-DSS – Payment Card Industry Data Security Standard

HIPPA – Healthcare Information Portability and Accountability Act

TTP – Tactics, Techniques and Procedures

IoT – Internet of Things

POC – Proof of Concept

CIA – Confidentiality, Integrity, or Availability

CVE – Common Vulnerabilities and Exposures

RM – Risk Management

RMF – Risk Management Framework

ISO – International Organization for Standardization

RV_C – Control Risk Value

JSON – JavaScript Object Notation

POJO – Plain Old Java Object

SCDG – Security Control Dependency Graph

SME – Small and Medium-sized Enterprises

CISSP – Certified Information Systems Security Professional

EICRS – Endpoint Independent Cumulative Risk Score

XML – Extensible Markup Language

SC – Security Category

FL – Fuzzy Logic

NLP – Natural Language Processing

COG – Center of Gravity

DAG – Dependency Attack Graph

IDS – Intrusion Detection System

DFS – Depth First Search

XSS – Cross-Site Scripting

IDE – Integrated development environment

OOP – Object-Oriented Programming

API – Application Programming Interface

WIP – Work in Progress

UML – Unified Modeling Language

AUC-ROC – Area Under the Receiver Operating Characteristic Curve

AT – Attack Technique Strategy

U – Union Strategy

AT-U – Attack Technique Union Strategy

CSP – Constraint Satisfaction Problem

Chapter 1

1 Introduction

Information is a precious commodity. Organizations and businesses realize that non-compliance against security regulatory requirements of their Information Systems (IS) can be detrimental to business processes, public image and relations, incur financial loss, and create legal issues [1]. Many recent cyber security incidents around the world have raised serious questions regarding organizations' ability to understand and prevent such incidents. Although governments and organizations invest heavily in commercial and state-owned cybersecurity products and technologies, security incidents are still rising. This paradox identifies the gap in our understanding and handling of cyber security risks. An organization's strong cyber security posture requires the constant application of novel methods and techniques to identify and assess risks. While we can never anticipate the precise time when a security incident may occur, it is crucial to remain prepared to act swiftly when the situation arises [2]. This thesis focuses on designing and developing a risk assessment methodology that supports the Cyber Risk Situational Awareness (CRSA) process and assesses system-wide exposure by predicting potential vulnerabilities in an Information System. In our research, we have incorporated NIST 800-53 rev4, MITRE ATT&CK, and other software engineering and Artificial Intelligence (AI) techniques to conduct the control-based cyber security posture analysis.

Compliance with security policies against threats has been a focal topic within the information security community. The software development market is currently experiencing a growing need for software systems that conform to international Governance, Risk, and Compliance (GRC) regimes [3]. Furthermore, organization-wide Information System (IS) security is directly dependent upon the effective implementation

of Information System policies inside the organization [4]. To measure the effectiveness of these policies, organizations should have risk management and assessment frameworks in place, which can support both qualitative and quantitative risk assessment techniques and determine the cumulative security risk for an IS.

To date, many Security Information and Event Management (SIEM) software frameworks are available commercially, which are regularly used by security professionals to assess the security compliance of an IS [5]. These SIEM frameworks support many risk assessment algorithms, such as the Agency-Wide Adaptive Risk Enumeration (AWARE), proposed by Cybersecurity & Infrastructure Security Agency (CISA), which calculates an IS's numerical cyber risk score. One implementation example of this SIEM software is IBM's QRadar, supported by the National Institute of Standards and Technology (NIST) Content Pack. This content pack add-on implements the NIST 800-53 protocol and aims to continuously monitor, analyze, and act upon the alerts generated. It also supports the AWARE algorithm, widely used by U.S. government agencies and departments to assess their cyber risk. As of September 2019, all United States federal organizations are being assessed under this new algorithm. [7]

Our work in this thesis can be used in many ways, such as it can be a risk assessment and compliance verification module for SIEM software or can be used as a standalone application to assess and gather threat intelligence. For example, imagine a regulated industry such as a power generation plant, bank, or state-owned facility whose cyber security is protected by NIST 800-53 framework. When all required controls are active, and the security posture is in place, these industries need to assess that posture and ensure security is continuously intact. If, for any reason, an auditor or the security analyst finds a control failing, our assessment method not only predicts security risks in the system but can also perform attack surface analysis to find any potential Tactics, Techniques and Procedures (TTP) as well as real-world Common Vulnerability Exposure (CVE) before any malicious actors do. In short, our method's objective is to perform a security scan of

the system using a custom-made non-linear multi-dimensional algorithm, predict risk due to control failure, quantify that risk, find system exposure, and based on the assessment, risk prediction, and attack surface analysis, the module generates a detailed security intelligence report. The report includes information about the identified vulnerabilities, potential attack methods, associated risk scores, and recommended actions to mitigate the risks.

1.1 Continuous Compliance

“Continuous Compliance” is a direct manifestation and result of “Continuous Testing and Continuous Integration” and aims at ensuring uninterrupted compliance against regulatory security standards. This process is proposed by Kellogg et al. [6], based on building and running a "verification tool on every commit for ensuring compliance is maintained at the source-code level.” This technique's advantage is reducing reliance on manual audits and letting the software do most of the job. Continuous compliance is essential in cyber security because it helps ensure that an organization's systems and processes constantly meet security standards and regulations. This is important because the threat landscape continually evolves, and new vulnerabilities and attacks always emerge. As a result, organizations can proactively address potential security issues by continuously monitoring and assessing compliance before attackers exploit them. Additionally, continuous compliance can help organizations demonstrate due diligence in a security incident or data breach. [6]

Continuous compliance typically works by implementing automated tools and processes that continuously monitor and assess an organization's systems and processes for compliance with security standards and regulations. This can include: *(i)* Vulnerability scanning: Regularly scanning software and network systems for known vulnerabilities and patching or mitigating them as necessary, *(ii)* Compliance checking: Continuously

monitoring systems and processes for compliance with specific security standards, such as PCI-DSS or HIPAA, *(iii)* Security testing: Regularly performing penetration testing, social engineering testing, and other types of security testing to identify and address vulnerabilities, *(iv)* Log monitoring: Continuously monitoring log files for suspicious activity and using analytics to identify potential security threats, *(v)* Risk assessment: Continuously assessing the organization's overall risk posture and identifying areas that need improvement, and *(vi)* Reporting: Generating regular reports on the organization's compliance status and identifying areas that need attention [6]. These steps are integrated and automated in single or multiple platforms that can provide insight into the current security posture, identify areas of non-compliance, and trigger appropriate remediation. By utilizing continuous monitoring and continuous analysis, continuous compliance can help organizations identify and address compliance issues before these become a problem. It also allows organizations to demonstrate to regulatory bodies that they are conforming to the appropriate regulatory policies. Additionally, as part of a continuous compliance program, organizations may conduct regular self-audits, which can be used to identify areas of non-compliance and take corrective actions before being audited by the regulatory body [6].

For this thesis, we take the stance of considering the process of Continuous Compliance at the security controls level. More specifically, let’s consider an organization or an IS which is monitored utilizing a security framework such as NIST 800-53 or ISO 27001/5. In this case, any change in the security policies, such as implementing a new control or removing current security control, or a failed control, can trigger an enterprise-wide compliance audit using the concept we have built up. The objective is to continuously monitor the system compliance against industry regulatory requirements, perform cyber risk assessment, and detect any weakness in the system which may arise due to this change.

1.2 Thesis Objectives and Contributions

This thesis has four objectives. The first objective is to analyze one of the most prevalent and complete security framework protocols, the NIST 800.53, and extract dependencies between its various security policies, which are referred to as security controls. This analysis results in a security controls dependency graph that models the associations and weighted impact between the various controls. The nodes in the graph are the security controls, and the labeled directed edges denote the impact one control (the source) has on another control (the target) if the source control node fails. The second objective is to devise a technique and an algorithm to propagate impact and weighted risk from one control (the source) to another (the target) when one or more source controls are found to be non-compliant within the scope of the NIST 800.53 standard. The third objective is to design a technique by which the overall security posture of an information system can be assessed as a function of the failed controls, the affected controls, their importance, and the time elapsed since their discovery and non-mitigation. The algorithm provides a score that directly associates with the security risk the system faces. The fourth objective is to analyze the system's security posture, superimpose it to security attack models an intruder could have utilized to violate the failed and the affected controls and alert the system administrators on checking any security holes related to these possible identified attack strategies. For this thesis, we have utilized MITRE's ATT&CK framework and matrix, which associates security controls with possible attack strategies that intruders can utilize to fail the aforementioned controls.

In this respect, the thesis contributions can be summarized as follows:

- The design of a novel domain model (schema) to denote dependencies between NIST 800.53 security controls. The domain model implements a custom-labeled, typed, directed multigraph.

- The design of a novel algorithm to propagate impact from one control to another when one or more controls fail or become at risk due to propagation. The algorithm takes into account the risk level of each control, its scope, the time since it was discovered as non-compliant, and the breadth of its applicability (i.e., low, medium, or high-security systems)

- The design of a novel technique and algorithm to evaluate an overall security posture risk score as a function of the failed controls, the affected controls, their risk level, and time elapsed since their non-compliance and non-mitigation.

- The design of a technique to associate failed or affected risks with possible attack strategies as these are identified and proposed by existing security frameworks. For this work, we have used MITRE's ATT&CK framework, which associates failed controls with possible attack strategies an intruder can utilize to fail these controls.

- The design and implementation of a toolkit that implements the aforementioned techniques and algorithms.

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 provides background information about the scope, structure, and content of the NIST 800.53 and ATT&CK frameworks. These frameworks have been proposed by the National Institute of Standards and Technology and the MITRE Corporation in the United States. These frameworks constitute the de-facto standard of security system assessment. Chapter 3 discussed related work in security risk and security posture assessment. Chapter 4 discusses theoretical aspects related to modeling risk dimensions, security controls dependency modeling, and fuzzy reasoning fundamentals. Chapter 5 presents the risk assessment process and the technical

framework. Chapter 6 presents experimental results by applying the proposed technique. Finally, chapter 7 concludes the thesis and provides pointers for future research.

Chapter 2

2 Background

This section will cover the background material used for this research and discuss the frameworks and relevant risk assessment methodology. We then briefly discuss our process for conducting the risk assessment. Let us first discuss our core frameworks.

2.1 NIST 800-53 Framework

NIST SP 800-53, also known as the "Recommendation for Security and Privacy Controls for Federal Information Systems and Organizations, is a comprehensive document published by the National Institute of Standards and Technology (NIST) that provides guidelines for securing information systems and organizations" [9]. This framework is intended to be used as a reference for organizations that handle sensitive or confidential information. It is widely used in the government and private sectors as a standard for information security management. NIST 800-53 includes security controls and assessment procedures for federal information systems and covers a wide range of topics, including access control, incident response, and system and communications protection. The controls in SP 800-53 are divided into three classes: management, operational, and technical. Management controls focus on establishing the policies, procedures, and responsibilities that organizations must have to protect their information systems. The operational controls focus on the day-to-day activities organizations must conduct to safeguard their information systems. The technical controls focus on the technologies and configurations organizations must implement to protect their information systems. [9]

One of the critical aspects of SP 800-53 is its focus on risk management. It provides guidance on identifying and assessing risks to information systems and organizations and recommends specific controls that organizations can implement to mitigate them. This emphasis on risk management is in line with the broader trend in information security by replacing the "compliance-based" approach with more of a "risk-based" methodology [9]. In addition to providing specific guidelines for securing information systems, SP 800-53 also guides how to conduct security assessments and evaluations. It recommends a "tiered" approach to security evaluations, where organizations start with a self-assessment and then move on to more in-depth evaluations as necessary. This approach is intended to make security evaluations more efficient and effective by focusing on the most significant risk areas. [9] It's worth mentioning that NIST SP 800-53 is updated periodically to adapt to new security threats and to reflect the latest technologies and best practices. The current version is NIST SP 800-53 Rev5, published on January 25, 2022; however, we have used NIST 800-53 Rev4 in our research. Knowing the latest version and updates is essential to have the best security controls and practices in place.

To mitigate security risks to “organizational operations and assets, individuals, other organizations, and the Nation,” organizations must develop comprehensive information security programs for Risk Management (RM) [9]. RM does not just minimize the loss of information assets due to unwanted events, but it also “provides a mechanism to the organizations to ensure that executive management knows the current risks” so that they can make informed decisions. The fundamental objective of risk management is to strengthen and protect the security principles of Confidentiality, Integrity, and Availability, the CIA triad. [8]

The NIST has developed a three-tiered RM approach, as shown in Figure 1, “to integrate the risk management process throughout the organization and more effectively address mission/business concerns” [9].

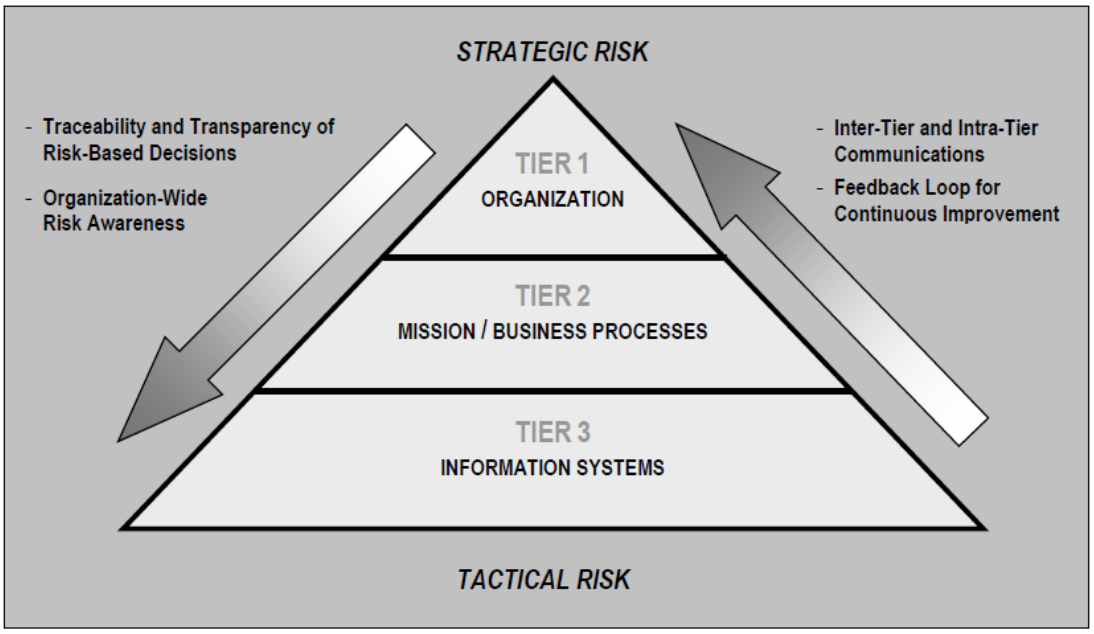


Figure 1: NIST Three-Tiered Risk Management Approach

In this tiered approach, Tiers 1 and 2 focus on organizations' strategic goals and mission/business processes; however, for this research, Tier 3 is our primary focus of study. To manage risk at this tier, we "require a risk management framework to ensure key risks are effectively identified and addressed." [8] The Australian Standard defines a risk management framework as follows:

"A set of components that provide the foundations and organizational arrangements for designing, implementing, monitoring, reviewing and continually improving risk management throughout the organization." [8]

NIST has developed a comprehensive security life cycle known as Risk Management Framework (RMF), as shown in Figure 2, for "addressing risks at Tier 3". "The framework addresses the security concerns of organizations related to design, development, implementation, operations, and disposal of information systems and the environments in which those systems operate." [9] It offers a structured and replicable

methodology for managing the security of IS and the related environment they operate in. We have incorporated Step 1, Step 2, Step 3, and Step 4 of the RMF in our research to assist in demonstrating our risk assessment process. These steps are "(i) Categorize Information System, (ii) Select Security Controls, (iii) Implement Security Controls, and (iv) Assess Security Controls." [9] The first step involves determining the system's security categorization, which is a characterization of the system's overall level of risk based on its "impact on organizational operations, assets, individuals, and the overall mission." In the second step, the organization selects a "set of security controls appropriate for the system's security categorization" and the potential risks associated with the system [8]. These controls are designed to mitigate the identified risks and ensure adequate system security. Furthermore, the third step involves "implementing the selected security controls into the system." The security controls must be implemented correctly, effectively, and efficiently. Finally, the fourth step involves evaluating the security controls to determine whether they function as intended and effectively mitigate the identified risks. The assessment should also identify any weaknesses in the controls and any areas for improvement. [9]

In this thesis, the RMF model shown in Figure 2 is incorporated from steps 1-4 in the manner that: (i) Step 1 in our Proof of Concept (POC) requires a user input against the prompt "Enter System Criticality (between 0 and 10): ", which must be filled in by a security professional. This number, which must be between zero to ten, designates the IS system criticality. If the number is higher, the system is considered more sensitive. This sensitivity is then reflected in the deployment of the framework and each control's RV_C. NIST 800-53 recommends computing the criticality of the system based on the required level of Confidentiality, Integrity, and Availability (CIA) [9]. A security professional meticulously calculates this input number and then uses it to assess system risk. For this reason, we have manually taken this input from a user and used it in our application for assessing security risk. (ii) Step 2 is about selecting controls for an IS. Once we have a system criticality from Step 1, we normalize the value between 0-4 using Equation 1. Then, we use the normalized value to select required controls from NIST 800-53

framework for that particular security category IS. The selection process takes place against the criteria discussed in later sections. (iii) In Step 3, we deploy the framework by activating the selected controls, depicting the security posture. (iv) Step 4 in our POC assesses deployed controls and ensures the security posture has reduced attack surface. At this stage, we perform a risk assessment and attack surface analysis and present our output as a system security intelligence report.

Risk, as per the definition in American Heritage Dictionary, is “the possibility of loss.” Cyber risk is a function of threat, vulnerability, and consequence. This function must reliably compute the “probability of a threat exploiting a vulnerability and the associated impact.” To evaluate this type of function, a comprehensive risk assessment process must be followed to examine the risks. [8]

Security risk assessment is a necessary process to manage risk in an IS because it helps (i) Organizations quantify risk, (ii) Decision-makers quantify potential risks to their organizations, and (iii) Evaluate control effectiveness [11]. Therefore, an organization must “conduct risk assessment to evaluate (i) Threats to its assets, (ii) Vulnerabilities present in the environment, (iii) The likelihood that a threat will be realized by taking advantage of an exposure, (iv) The impact that the exposure being realized will have on the organization, (v) Countermeasures available that can reduce the threat’s ability to exploit the exposure, (vi) The residual risk (e.g., the amount of risk that is left over when appropriate controls are properly applied to lessen or remove the vulnerability)” [8].

The risk assessment process can differ between organizations; however, the fundamental principle behind this process remains the same. It can be “qualitative, quantitative, or a hybrid of both” [8]. Qualitative assessments, based on traditional questionnaires, interviews, brainstorming, etc., define the risk in descriptive estimates such as Low, Medium, or High. These assessment methods are primarily employed when organizations

cannot furnish the probability and consequences of potential threats. Hence, they have "some drawbacks when being applied to huge complex network security environments." [11] Quantitative risk assessments are used when organizations need to provide specific measurements and impact in the numerical figure, usually in terms of cost representing the expected loss. These methods use formulas and mathematical expressions to calculate a numerical risk score. They are the preferred method of choice for organizations that can provide "estimates in numeric numbers for the probability and loss associated with each attack." [11]

We implemented a hybrid risk assessment model for our research, where both qualitative and quantitative features were brought into action. For example, in the later sections, we will introduce the concept of Control Risk Value (RV_C), and its value is a combination of both qualitative and quantitative features. We say that RV_C can be low, medium, or high, but then we use AI techniques to compute a solid number to define these subjective terms using fuzzy logic.

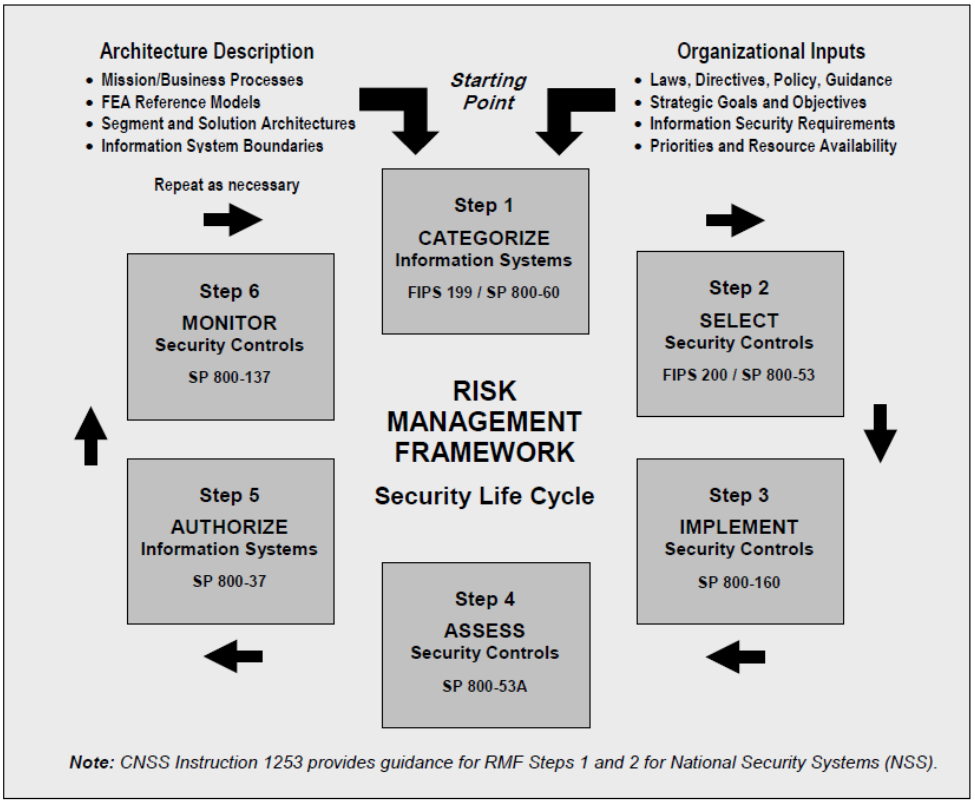


Figure 2: NIST Risk Management Framework (RMF)

2.2 MITRE ATT&CK Framework

The MITRE ATT&CK Framework is a comprehensive knowledge base for cyber security threat management developed by the MITRE Corporation [25]. The framework is organized into a matrix of "pre-attack" and "post-attack" phases, with each phase broken down into specific tactics and techniques. The framework also includes detailed information on the tools and malware used by attackers, as well as best practices for detection, defense, and response. The framework aims to help organizations improve their understanding of cyber threats and enhance their ability to defend against them. [22]

As discussed above, MITRE ATT&CK's core components are TTP which stands for Tactics, Techniques, and Procedures. The framework organizes the various methods used by malicious actors in cyber-attacks into these three categories. Tactics refer to the overall goals and objectives of an attack. For example, an attacker may have a tactic of gaining initial access to a network or establishing persistence within a network. The MITRE ATT&CK framework currently defines 15 tactics, which are further organized into the pre-attack and post-attack phases of a cyber-attack. Techniques refer to the specific methods or tools an attacker uses to achieve their tactics. For example, an attacker may use a spear-phishing campaign as a technique for gaining initial access to a network. This framework currently defines over 250 techniques. Procedures refer to the detailed steps an attacker takes to execute a technique. For example, the procedure for a spear-phishing campaign might include crafting a targeted email message, sending the message to specific individuals, and then using the email to deliver malware or trick the recipient into providing sensitive information. [22]

TTP in the ATT&CK framework can help organizations identify the potential threats facing an organization and develop and implement effective mitigation and response strategies. Knowing the TTP can also aid organizations in developing a comprehensive security plan and improve their understanding of cyber threats, thus enhancing their ability to defend against them. Overall, the ATT&CK framework offers a structured approach to understanding and defending against the various TTP malicious actors use in cyber-attacks. [22] Hence, we are interested in determining the TTP when one or more NIST 800-53 security control fails, which would further help us determine the system exposure against potential adversary attacks.

2.3 Risk Assessment Process

The process for the proposed system is depicted in Figure 3. The process has six main steps. In the first step, the JSON representation of the NIST 800.53 model is analyzed, and its structure is mapped onto Plain Old Java Object (POJO), which then gives us the ability to treat individual controls as Java objects. Using industry-standard graph management technologies, we then create a Security Control Dependency Graph (SCDG). In the second step, the Importance (I_{RVC}), Coverage (C_{RVC}), and Connectivity (A_{RVC}) values for each control are calculated using AI techniques such as fuzzy logic and then normalized in the [0, 4] range. In the third step, the normalized Importance (I_{RVC}), Coverage (C_{RVC}), and Connectivity (A_{RVC}) values are fuzzified. In the fourth step, fuzzy rules are applied, and an overall risk score for each control is calculated. We have devised these rules based on the subjective understanding of security-related information. In the real-world scenario, such rules should be researched and developed by a team of security professionals who may have to analyze each rule from many different dimensions before crafting them. It's an art!

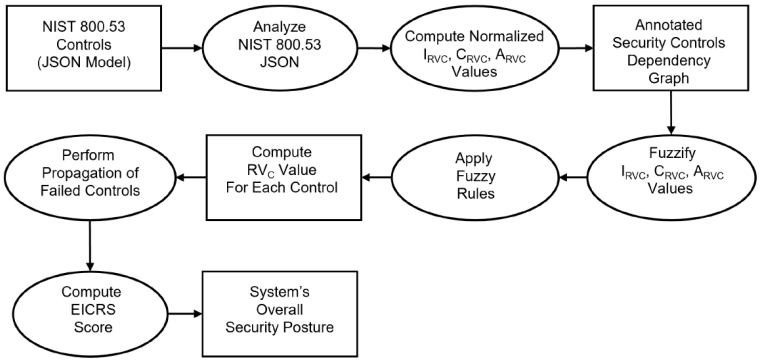


Figure 3: Our Six-Step Process

In the fifth step, once control is found to be failing, a propagation process is applied. Next, all affected controls are identified, forming a subgraph of the dependency graph we refer to as the affected subgraph. In the sixth step, based on the affected subgraph, security posture analysis takes place, where our methodology tries to predict the system risk and exposure by leveraging NIST 800-53 security control dependencies violations and MITRE ATT&CK mappings. Finally, a comprehensive cyber security intelligence report is generated by our tool, which lists components of risk assessment, system exposure, and other model performance-related details computed using our custom-made algorithms. This custom report represents the overall security posture analysis of the system.

Chapter 3

3 Related Works

This section presents pointers for related research work and security standards that motivated our work. With respect to related literature, Kreicberga (2010) [1] in her work "Internal threat to information security - countermeasures and human factor within SME" highlights the importance of understanding the internal threats within SMEs and the countermeasures that can be put in place to mitigate them. Pfleeger (2015) [2] in his book "Security in Computing" (5th ed.), delves into the concepts and principles of security in computing and provides a complete overview of the field. Jorshari and Tawil (2015) [3] propose a high-level scheme for an ontology-based compliance framework in software development in their work. Straub and Welke (1998) [4] in their work proposed a security planning model for management decision-making to cope with systems risk.

In the other related literature, Jha et al. (2002) [10] presents a formal analysis of their proposed data structure based on a graph in their paper "Two formal analyses of Attack Graphs" presented at the 15th IEEE Computer Security Foundations Workshop. Their research aims to assist analysts in determining the minimum security measures required to ensure the safety of a system. In Alhomidi and Reed (2014) [11], the authors propose an attack graph-based risk assessment approach in their paper "Attack graph-based risk assessment and optimization approach," published in the Journal of Internet Technology and Secured Transaction. They apply genetic algorithms to analyze and explore different paths in attack graphs to identify potential vulnerabilities in a system. Derbyshire et al. (2020) [23] present a method for cyber security risk assessment that considers the moves of “intelligent adversaries, who make decisions based on various factors, including the cost of their attacks.” The paper also provides a small set of control mappings to attack

techniques based on their own mapping methodology. This knowledge became the basis for further investigation into this methodology, leading us to active NIST – MITRE mapping research. Their work is significant because it addresses the gap in traditional cyber security risk assessment methods that do not consider the cost of attacks or predict attack techniques from intelligent adversaries. The authors propose a new approach that considers the cost-benefit analysis of an attacker, which can help organizations to better anticipate and prepare for potential cyber-attacks.

In their publication titled "A situation awareness model for information security risk management," which was published in Computers & Security, Webb et al. (2014) [12] introduce a model for situational awareness. The model is based on Endsley's Cyber Risk Situational Awareness (CRSA) model and involves the "collection, analysis, and reporting of risk-related information across an entire organization." Pereira and Berlin (2009) [21] in their paper "Wave propagation and deep propagation for Pointer Analysis" propose methods for pointer analysis in software systems. The authors propose wave propagation and deep propagation methods to analyze the behavior of pointers in software systems to detect potential vulnerabilities.

The related standard, the CISSP program, also influences our research. We have thoroughly reviewed Domain 1 of the CISSP, also known as Security and Risk Management, to build the industrial knowledge foundation for our thesis. It covers the concepts, principles, and practices of security and risk management, including the topics of "Security Governance, Compliance, Legal & regulations & investigations & compliance, Business continuity and disaster recovery planning, Risk Management, Security Assessment, testing, and Security Operations." [8] The knowledge gathered in Domain 1 has greatly benefited our cybersecurity research in several ways. Firstly, by understanding the concepts, principles, and practices of security and risk management, we gained a comprehensive understanding of the security landscape techniques, including the threats, vulnerabilities, and impacts associated with various security incidents. Secondly,

a deep understanding of the risk assessment process provided us with necessary information about the steps organizations should take to recognize and tackle security threats. With this knowledge, we could devise improved risk assessment strategies. Furthermore, the knowledge of security governance, compliance, legal and regulatory requirements was crucial for ensuring that POC is aligned with the needs and requirements of prospective organizations and complies with relevant regulations and standards such as NIST 800-53. Finally, the knowledge of security assessment and testing enhanced our risk assessment process by guiding the design and implementation of comprehensive and effective security testing strategies and ensuring that security solutions are tested and validated close to real-world environments. In summary, by thoroughly understanding Domain 1 of the CISSP program, we have developed a comprehensive and well-informed understanding of the security landscape, processes, and technologies used to manage and mitigate security risks.

The above literature heavily focuses on the technical aspect of information security, specifically identifying system vulnerabilities and assessing the associated risk. However, none of the literature reviewed presents a methodology for predicting and calculating system exposure due to security control failure. Additionally, no discussion was found on the dependencies between security controls and the potential impact of violating those dependencies on the system's overall security posture. This gap in the research motivated our current work, in which we propose a methodology to not only predict system exposure but also help meet legal and regulatory compliance requirements, which is a crucial aspect of information security.

Chapter 4

4 Theoretical Framework

This chapter will discuss the proposed theoretical model for our research. First, Section 4.1 will describe the security controls modeling, which includes design and development, fuzzification and reasoning of its dimensions. Next, Section 4.2 discusses dependency graph modeling, known as Security Controls Dependency Graph (SCDG). We will now explain them in detail in the following sections.

4.1 Security Controls Model

To create our prototype's theoretical framework, our first step is to model the overall NIST 800-53 framework as per our requirements. There are two main components to this process: (i) NIST 800-53 Controls` Modeling and (ii) Computing Control Risk Value (RV_C). We first begin with modeling the NIST 800-53 Framework in the section below.

4.1.1 NIST 800-53 Controls and Modeling

The most basic unit in our whole framework is security control. NIST 800-53 offers a comprehensive model of interrelated security controls, available in related NIST publication and XML format. "Controls are the countermeasures prescribed for an IS or organization that are designed to: (i) Protect the Confidentiality, Integrity, and Availability of information that is processed, stored, and transmitted by those

systems/organizations; and (ii) satisfy a set of defined security requirements." [9] For ease of selecting and using security controls, they are organized into eighteen families, as shown in Table 1.

ID	FAMILY	ID	FAMILY
AC	Access Control	MP	Media Protection
AT	Awareness and Training	PE	Physical and Environmental Protection
AU	Audit and Accountability	PL	Planning
CA	Security Assessment and Authorization	PS	Personnel Security
CM	Configuration Management	RA	Risk Assessment
CP	Contingency Planning	SA	System and Services Acquisition
IA	Identification and Authentication	SC	System and Communications Protection
IR	Incident Response	SI	System and Information Integrity
MA	Maintenance	PM	Program Management

Table 1: NIST 800-53 Security Control Families (Courtesy of NIST)

Every security control in NIST 800-53 is assigned to a particular family as per the general security topic. Each family has a unique identifier that helps identify that family; for example, AC represents the Access Control family. The control structure, as shown in Figure 4, is composed of five components: "(i) A control, (ii) Supplemental guidance, (iii) Control enhancements, (iv) References, (v) Priority and Baseline allocation." [9]

```
"number": "AC-1",
"baseline-impact": [
 "LOW",
 "MODERATE",
 "HIGH"
],
"supplemental-guidance": {
 "related": "PM-9", procedures for the effective implementation
 "description": "This control addresses the establishment of … },
"references": {"reference": [
"family": "ACCESS CONTROL",
"title": "ACCESS CONTROL POLICY AND PROCEDURES",
"priority": "P1"
}
```


Figure 4: Sample NIST 800-53 Rev4 Security Control

The control section defines a set of security activities or actions that an organization must carry out. The supplementary guidance section presents additional control information. This information can be used when defining, developing, or implementing a control [25]. The control enhancement section defines "statements of security-related capabilities, i.e., to add functionality and increase the strength of control" [9]. These enhancements are not intended to be implemented in isolation; they must be used in conjunction with corresponding baseline control. In the reference section of control, we can find relevant "federal laws, executive orders, policies, regulations, and guidelines." On the other hand, the priority and control baseline section recommends priority codes that aid in decision-making regarding control implementation and allocation, as well as the implementation of enhancements specific to the control. [9]

We have modeled the NIST 800-53 framework, as shown in Figure 5. This model helps us easily access control-related information by mapping JSON data onto Java POJO

objects. We extract the relevant information from the control object to create a dependency graph representing the cyber security posture.

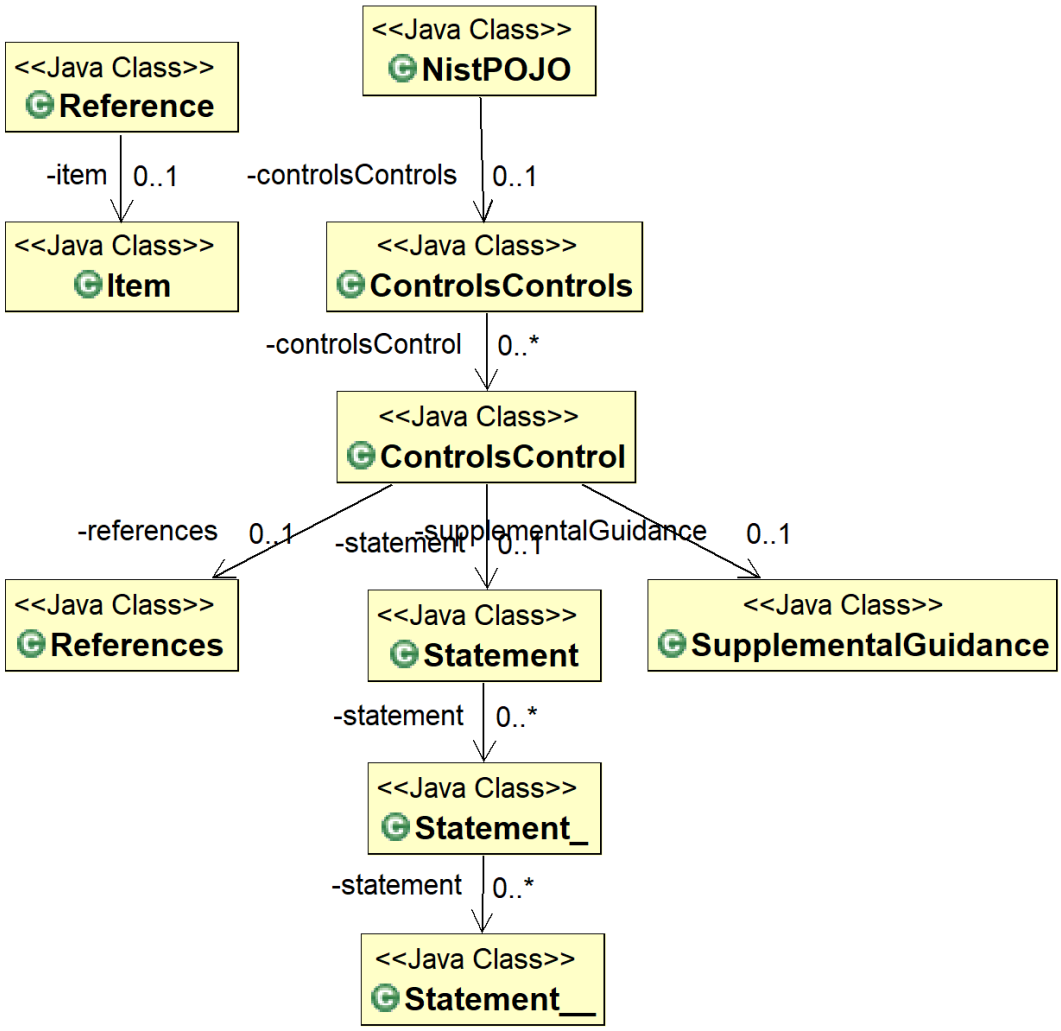


Figure 5: NIST 800-53 Security Controls Model

4.1.2 Control Risk Value (RV_C)

Control Risk Value (RV_C) is associated with each individual control, representing its criticality. RV_C is made up of three dimensions, Importance, Coverage, and Connectivity, as shown in Figure 6. We formally define RV_C as:

Definition 1 [Control Risk Value]: Control Risk Value or RV_C is associated with each control and represents a potential impact and damage to the Information System if the control is compromised. The RV_C is a triplet of pairs < (Importance, I_{RVC}), (Coverage, C_{RVC}), (Connectivity, A_{RVC}) > where each pair corresponds to a dimension (i.e., Importance, Coverage, Connectivity). The RV_C is computed based on these dimensions and can have low, medium, or high values.

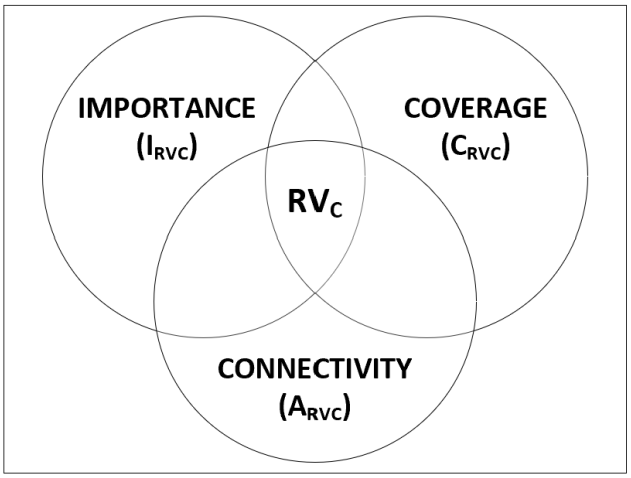


Figure 6: Control Risk Value (RVC) Dimensions

4.1.2.1 RV_C Dimensions

We now start our discussion with the RV_C dimensions, which are essential in determining a control's criticalness based on numerous factors. Instead of looking at control in the default format, we have taken a multi-dimensional view of individual control to determine its general criticality for a particular IS. This allows us to categorize the control according to the environment, relatability, and criticality. In our case, the selected dimensions reflect a particular control's critical aspects. The Importance dimension tells us why and how much a particular control is essential for a particular IS. In contrast, the Coverage dimension addresses the control aspect of being recommended for multiple systems. Connectivity discusses a relationship of control with other controls. We have only chosen three dimensions because they are sufficient to address NIST controls in a multidimensional view and help determine the importance and relevance of control about security posture. Any other dimension which may come into the light can easily be incorporated into our model for future iterations.

4.1.2.1.1 Importance (I_{RVC})

Importance (I_{RVC}) is a RV_C dimension that relates to the IS Security Category (SC) for which a specific control is applicable. It is based on NIST 800-53 Framework's security baseline allocation values: Low, Moderate, and High. The NIST 800-53 Rev4 publication outlines a "format for expressing the SC of an information system:

$SC_{\text{Information System}}$ = {(Confidentiality, Impact), (Integrity, Impact), (Availability, Impact)}, where the acceptable values for potential impact are low, moderate, or high." [9]

The NIST defines a low-risk system as a non-mission critical IS with low-security objectives for all three categories. The system is classified as moderate-risk if no security objectives exceed a moderate impact and at least one objective is rated moderate. Lastly, a high-risk system is classified as a mission-critical IS with at least one security objective set to high. Each control can be recommended for IS in any of these three categories. [9]

In the proposed framework, the security category of a system is provided as input from the user, classifying their systems with a risk value in the range of [0, 10], with 0 being a no-risk SC IS and 10 being the highest risk classification of SC for an IS. This input value is normalized to the [0, 4] interval to yield I_{RVC} using the following Equation 1:

$$\text{Normalized Output} = \text{outMin} + \left(\frac{(\text{outMax} - \text{outMin})}{(\text{inMax} - \text{inMin})} \times (\text{in} - \text{inMin})\right) \ldots\ldots\ldots 1$$

The normalization formula shown above has five parameters: (i) Input Minimum (inMin), (ii) Input Maximum (max), (iii) Output Minimum (outMin), (iv) Output Maximum (outMax), (v) User Input (in). "Normalized Output" is the value we get after normalization. "outMin" and "outMax" are the minimum and maximum values of the desired output range. "in" is the input value that we want to normalize, and "inMin" & "inMax" are the minimum and maximum values of the input range. We first calculate the scaling factor that maps the range of the input values to the range of the desired output values. It determines how much the input values must be stretched or compressed to fit the desired output range. We then multiply the scaling factor by the difference between the input value and the minimum of the input range. This scales and shifts the input value to fit the desired output range. Finally, we add the scaled and shifted value to the minimum of the desired output range. This ensures that the normalized value is within the desired output range. For example, an input value of 2.5 will be mapped to an output value of 1, and an input value of 6 will be mapped to an output value of 2.4. This

normalized value constitutes the I_{RVC} value and is then fuzzified by applying a fuzzification process discussed in later sections (see Section 4.1.2.2.1 and Figure 7). For example, as depicted in Figure 7(a), an I_{RVC} value of 3.1 is associated with different degrees of confidence as being High (0.4), Medium (0.25), or Low (0). The fuzzified I_{RVC} values will be used by a fuzzy reasoner and a collection of fuzzy rules to compute the overall RV_C value for a control, as discussed in Section 4.1.2.2.2.

4.1.2.1.2 Coverage (C_{RVC})

Coverage (C_{RVC}) is a dimension of RV_C that addresses the question of how many different types of systems a control is recommended for. If control is recommended for more than one NIST 800.53 Security Category (i.e., High, Medium, Low), then its importance is elevated. For example, if the control is recommended for low-risk systems, its value will be 1, while if it is applied to two categories, say Moderate (entails Low), the coverage value will be 2, while if it is applied to three categories, (i.e., High which (entails Low, and Medium) the coverage value will be 3. If the control is not recommended for any SC, its value would be Null. The membership function is simple and is depicted in Figure 7(b).

4.1.2.1.3 Connectivity (A_{RVC})

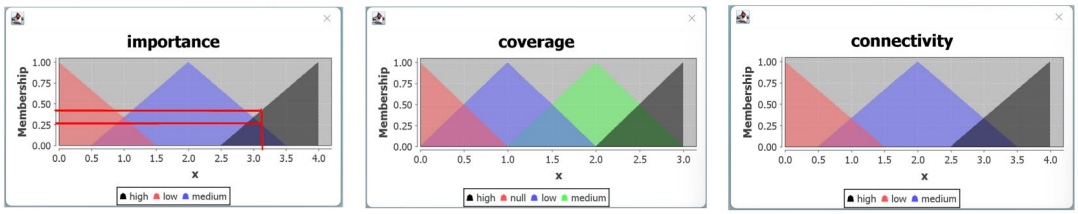
Connectivity is a dimension of RV_C that addresses the question of how many other controls a control relates to. More specifically, according to the NIST 800.53 framework, each control in a family of controls (e.g., the Access Control Policy and Procedures family) may relate to zero or more other controls. Let us call this number *rcSize*. For example, if a control is related to 6 other controls, then its *rcSize* will be 6. After normalizing the *rcSize* of each control to a value in the range of [0, 4] (see Equation 1),

we yield the A_{RVC} value. This normalized A_{RVC} value is fed to a fuzzification membership function (see Section 4.1.2.2.1 and Figure 7(c) to determine the level of confidence this value being Low, High, or Medium. This fuzzified A_{RVC} values will be used by a fuzzy reasoner and a collection of fuzzy rules to compute the RV_C value for a control, as discussed in Section 4.1.2.2.2.

4.1.2.2 Fuzzy Logic (FL)

Fuzzy Logic (FL) is a computing paradigm that allows for a more flexible approach to reasoning and decision-making using degrees of truth rather than strict true/false values. It is a mathematical framework for dealing with uncertainty and imprecision and allows for reasoning with incomplete or ambiguous information. Its ability to handle ambiguity and uncertainty makes it a valuable tool in Artificial Intelligence (AI). In FL, a statement can be partially true and partially false, with a degree of membership between 0 and 1, rather than entirely true or false. It can replicate human thought and reasoning using degrees of truth instead of strict binary values, making it well-suited for decision-making and control systems involving imprecise data. For instance, it is helpful in Natural Language Processing (NLP) technologies and in regulating and controlling machine outputs based on multiple input variables, such as adaptive cruise control technology in automobiles. [26]

FL has four components in its architecture: “(i) Fuzzification, (ii) Knowledge Base, (iii) Inference Engine, and (iv) Defuzzification” [28]. The fuzzification process transforms the dimensions input, which are numbers, into fuzzy sets, while Knowledge Base supplies IF-THEN rules [27]. After obtaining the fuzzy sets, the Inference process emulates human reasoning using Inference Engine on the inputs by applying IF-THEN rules. Finally, when the inference is made, the Defuzzification process converts the fuzzy set from the Inference Engine to a real-number value [26].



(a) Importance Membership **(b) Coverage Membership** **(c) Connectivity Membership**

Figure 7: Membership Functions for I_{RVC}, C_{RVC}, A_{RVC}

4.1.2.2.1 Fuzzification Process

In the Fuzzification process, we associate the I_{RVC}, C_{RVC}, and A_{RVC} values with a confidence value of a Fuzzy logic linguistic variable of the form "The <Dimension> of <Control> is <DiscreteValue>" (e.g., "The Importance of AC-2 is High"). The fuzzification process then uses a membership function to associate a Fuzzy linguistic variable with a number in the range of [0, 1], as shown in Figure 7.

We have chosen triangular membership functions to help model our linguistic variables. When the range of a linguistic variable is limited and well-defined, the triangular membership function can be a good choice. It allows you to easily specify the lower and upper bounds of the linguistic term. These functions are relatively simple to define and work with. It has only a few parameters, making interpreting and adjusting easy. The linguistic terms that might be modeled using triangular membership functions include "low," "medium," and "high" for variables like speed, distance, humidity, etc. This type of function is beneficial when the linguistic variable being modeled has a precise and symmetric "center" point. The triangular membership function can effectively capture this symmetry when data is symmetrically distributed around a central value. For example,

when modeling the concept "The Importance of AC-2 is High", the desired dimension has a precise midpoint; a triangular membership function could be appropriate. Also, these functions are intuitive and easy to understand for humans, which is vital in fuzzy logic systems where the goal is to model human-like reasoning. In the future, we may experiment with other available membership functions to further enhance the capabilities of fuzzy logic predictions.

Our first input variable for FL is Importance (I_{RVC}) which is a normalized value in the range of [0, 4], as discussed in Section 4.1.2.1.1. Using the membership function shown in Figure 7(a), we associate I_{RVC} with a confidence score in the range of [0, 1]. There can be three confidence scores denoting that the I_{RVC} is considered Low, Medium, or High. For example, using the membership function as shown in Figure 7(a), an I_{RVC} value of 3.1 will have a confidence score of 0 for being Low, a score of 0.25 for being considered Medium, and a score of 0.4 for being considered High. These confidence scores are fed to the FL reasoning engine and the defuzzification process to assign a fuzzy score to the overall risk value RV_C for a given control. For our PoC, we exclude any controls for which their Impact linguistic variable is 0. For example, for control AC-2, a value of I_{RVC} = 2.5 indicates a confidence score of 0 for the linguistic variable "The Importance of AC-2 is Low", and non-zero scores for the linguistic variables "The Importance of AC-2 is Medium", "The Importance of AC-2 is High". In this case, we only include controls that are applied for Medium and High-risk systems.

In our prototype, the C_{RVC} value does not need to be fuzzified, and the value of C_{RVC} = 1 is associated with confidence 1 for being considered Low, a confidence 0 for being considered Medium and High. Similarly, a value of C_{RVC} = 2 is associated with confidence 0 for being considered Low, confidence 1 for being considered Medium, and confidence 0 for being considered High. Finally, the C_{RVC} = 3 is associated with confidence 0 for being considered Low, confidence 0 for being considered Medium, and

confidence 1 for being considered High. As noted, the membership function for C_{RVC} is simple, as shown in Figure 7(b).

Like I_{RVC}, the A_{RVC} value is also normalized in the range of [0, 4]. Using the membership function depicted in Figure 7(c), we associate the value of $A_{RVC} = 2$ with a confidence score of 0, denoting that the A_{RVC} is considered Low, a second confidence score of 1, denoting that the A_{RVC} is considered Medium, and a third confidence score of 0, denoting that the A_{RVC} is considered High. For our PoC, we use the same membership function shown in Figure 7 for A_{RVC}.

4.1.2.2.2 Reasoning Process

The FL's reasoning engine uses the fuzzy rules, as shown in Table 2, and the Fuzzy values for I_{RVC}, $C_{RVC,}$ and A_{RVC} (see Figure 7) to calculate the overall risk value RV_C for every NIST 800.53 control. Our prototype uses the Center-of-Gravity (COG) method for defuzzification. Figure 8 shows an example of using the COG method. Although serving as examples of Fuzzy Logic in our model, these figures serve two different purposes. Figure 7 shows the membership functions example for our linguistic variables known as dimensions. In contrast, Figure 8 shows how these variables work together in the fuzzy logic to determine a control's RV_C. We see two rules, both concluding a fuzzy score for RV_C. The first rule has two premises linked by a conjunction and concludes that the linguistic variable, The RV_C is High. The second rule has one premise and concludes the linguistic variable The RV_C is Medium. The numbers shown on the x-axis of each graph in Figure 8, such as 3.1, 3.3, etc., are the normalized inputs the user provides for a particular dimension or gathered while deploying controls. Then these numbers are used to ascertain if a linguistic variable is low, medium, or high with a degree of truth, which helps determine the RV_C of control. The membership functions for the linguistic variables The Importance is High, The Importance is Medium, and The Connectivity is High are

shown above the corresponding linguistic variable in Figure 8. The rule “If the Importance is High” corresponds to the value 3.3, which can be classified as medium or high. However, the probability value shown on the y-axis further solidifies the belief that it is high. The same can be said for the second rule, “Connectivity is high”, where the input value is 3.7. Since the antecedents of the first rule are conjuncted (AND), the minimum y-axis value from the two membership variables, The Importance is High, and The Connectivity is High, is propagated to the membership function of the conclusion of the rule pertaining to the linguistic variable The RV_C is High.

Rule 1	IF Importance IS High AND Coverage IS High AND Connectivity IS High THEN rvcValue IS High
Rule 2	IF Importance IS High AND Coverage IS High AND Connectivity IS Medium THEN rvcValue IS High
Rule 3	IF Importance IS Medium AND Coverage IS Medium AND Connectivity IS High THEN rvcValue IS High
Rule 4	IF Importance IS Medium AND Coverage IS Medium AND Connectivity IS Medium THEN rvcValue IS Medium
Rule 5	IF Importance IS Medium AND Coverage IS Medium AND Connectivity IS Low THEN rvcValue IS Medium
Rule 6	IF Importance IS Medium AND Coverage IS Medium AND Connectivity IS High THEN rvcValue IS Medium
Rule 7	IF Importance IS Low AND Coverage IS Low AND Connectivity IS Medium THEN rvcValue IS Low

Rule 8	IF Importance IS Low AND Coverage IS Low AND Connectivity IS Low THEN rvcValue IS Low

Table 2: PoC Knowledge Base for Fuzzy Inference

In this respect, an area is formed. The same applies to the second rule. To aggregate the result of applying both rules, the two areas are joined yielding the shaded area in the bottom right in Figure 8. By computing the COG point of this area and obtaining its x-coordinate, we obtain the final RV_C value for a given control. In our POC, we want to defuzzify the RV_C value by applying the COG method to the confidence values of the linguistic variables "The RV_C of < Control X> is Low, "The RV_C of < Control X> is Medium, and "The RV_C of < Control X> is High, along with the rules depicted in Table 2. In this respect, there are three steps that need to be followed.

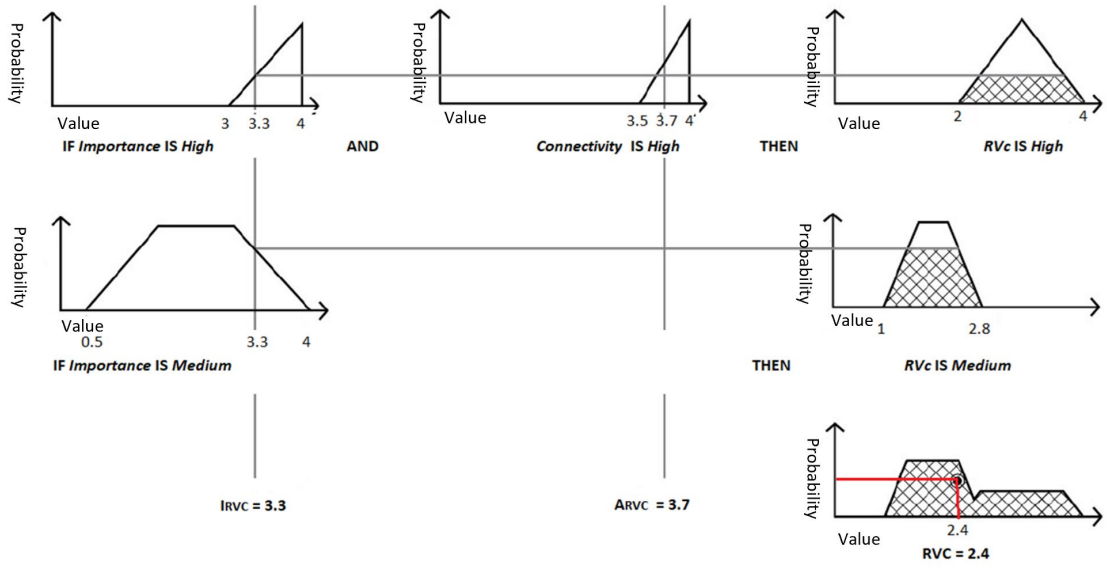


Figure 8: An Example of Fuzzy Reasoning using Two Rules and the COG Technique

The first step is to assign the individual values I_{RVC}, C_{RVC}, and $A_{RVC,}$ as discussed above. The second step is to fuzzify the linguistic variables for all Low, Medium, and High classifications of the Important, Coverage, and Connectivity dimensions by using the membership functions depicted in Figures 7(a), 7(b), and 7(c). The third step is to apply the fuzzy rules, aggregate the results and compute the COG point for the RV_C variable. The x-coordinate of the COG point indicates the final RV_C value to be assigned in each graph node. Once the RV_C for each control has been calculated, these are assigned to each control. Finally, we classify the calculated RV_C value into three categories, as shown in Table 3, depending on its range.

Risk Category	**RV_C Range**	**Assigned Color**
Low	$0 < RV_C < 1.5$	Green
Medium	$1.5 <= RV_C < 2.5$	Yellow
High	$RV_C >= 2.5$	Red

Table 3: RVC Risk Category Classification

For example, if a risk value for control is calculated as $RV_C = 1.75$, as shown in Figure 9, we can see that the probability of this control being low risk is 0.24, whereas the probability of it being medium risk is 0.75. Hence, based on the RV_C defuzzifier membership function, we can classify this control as medium risk. These classifications help us display the cyber security posture in easily identifiable controls in visual representation using data structure discussed in the next section.

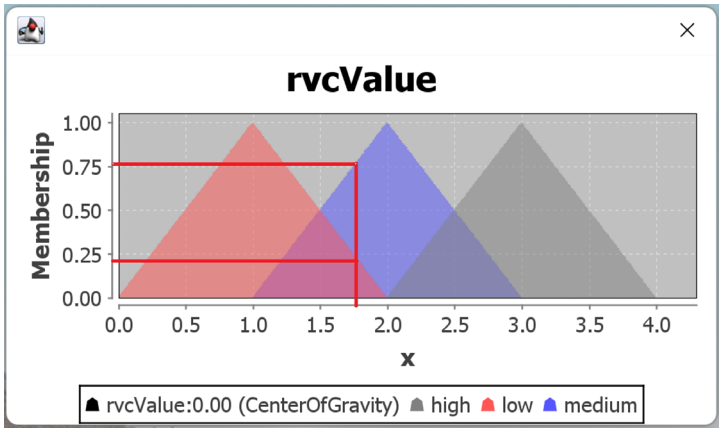


Figure 9: RV_C Membership Function

4.2 Security Controls Dependency Graph (SCDG)

The most critical component in our theoretical framework is the underlying data structure, a special kind of graph known as the Security Control Dependency Graph (SCDG), which supports the NIST 800-53 Framework implementation and risk assessment process. It can serve as an effective tool for offense, defense, forensic and detection analysis [10]. SCDG used in our research is somewhat synonymous with the Dependency Attack Graph (DAG), as proposed by Alhomidi and Reed, which has the capability to show attack scenarios by evaluating affected controls and dependencies between them [11]. However, the nomenclature, level of granularity, and other technical details of our SCDG differ from the DAG. Let's start by formally defining our SCDG:

Definition 2 [Dependency Graph]: A Security Controls Dependency Graph or SCDG is a directed weighted graph $G = (V, E, R)$ where V is a set of vertices $\{v_1, v_2, \ldots, v_n\}$ that represent NIST 800-53 security controls and associated attributes, E is a set of edges $e_{i,j}$ such that $i, j \in \{1, 2, \ldots n\}$ and $i \neq j$ that represents intra-control dependencies as these are extracted by the Related tag of each control in the NIST 800-53 between nodes v_i and v_j, and R is a set of control risk values r_i for each individual vertex v_i (see RV_C values).

The SCDG is essentially a graph-based custom data structure comprising NIST 800-53 controls and intra-control dependencies as defined in Definition 2. It contains all the NIST controls recommended for a particular security category IS and their related controls. Controls are shown in graph nodes, whereas edges between controls show their inter-relatability and dependencies. This data structure depicts an IS security posture when all controls are deployed. An example SCDG is depicted in Figure 10, where each node has four values I_{RVC} (Importance), C_{RVC} (Coverage), A_{RVC} (Connectivity), and overall Risk Value RV_C.

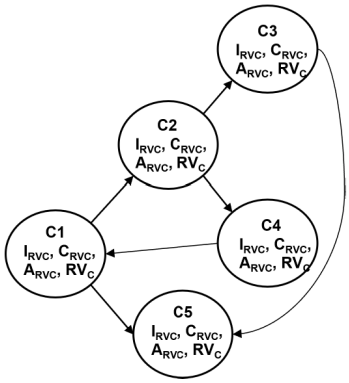


Figure 10: Security Control Dependency Graph (SCDG)

Definition 3 [Affected Graph]: An affected graph with respect to a SCDG $G = (V, E, R)$ is a graph $J = (H, I, L)$, where $H \subset V$ and $I \subset E$. H represents a set of all the related failed controls vertices including the failed source control. I is a set of all the affected edges between the vertices of H, and $L \subset R$.

The proposed system aims to constantly monitor the IS under analysis for any failing controls. Once a control is found to be failing, the propagation process is initiated. The propagation process involves two steps.

In the first step, we identify one or more controls $F = \{F_1, F_2, \dots, F_k\}$ which are flagged as failing by some system auditor, the fan-out from each of the failing nodes $F_i \in F$ are identified and a set of target nodes $T = \{T_{1j}, T_{1n}, \dots, T_{kw}\}$ are collected forming a new set $F_i' = \{F_i \cup T\}$ consisting of the failed principle control and its related controls, where T_{ij} indicates the control 'j' for which there is dependency edge from control 'i'. In the second step, all nodes in F_i' are flagged as activated, and then we start the propagation from node F_i and check if its RV_C is greater or equal to the RV_C of its target node T_{ij} and along the way flagging all the positively identified affected nodes T_{ij} as failing. Once all the nodes in F_i' have been propagated, we repeat step 1 for each F_i' and add further affected controls in the set as well. In other words, the propagation occurs from any failing node to any other connected to it node as long as the RV_C value of the failing node is greater or equal to the RV_C value of its target node. The important point to note here is that each F_i' corresponds to an affected graph $J_i \in J$, where Multiple F_i' generates multiple J_i. These are used to perform security analysis for identifying failed controls, determine potential risks, and predict potential TTPs and CVEs. We create assessment reports for each F_i which outputs valuable cyber security intelligence for a desired IS upon failure of particular security controls. Figure 11 illustrates this proposed propagation concept, where when control C1 fails, the propagation spreads to nodes C2 and C3 only as they have RV_C score less than the RV_C score of C1.

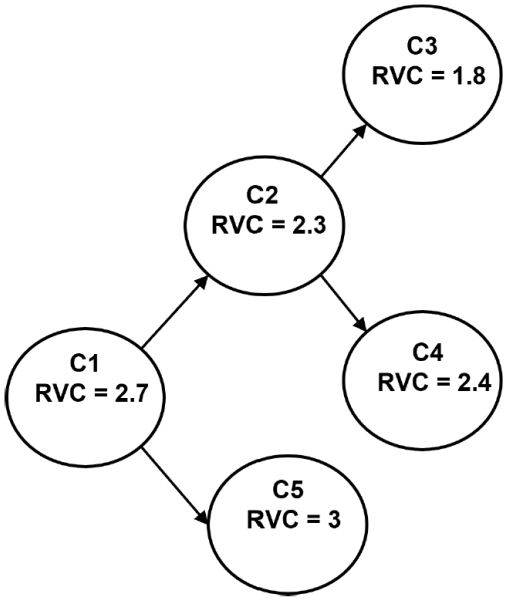


Figure 11: Control Propagation Example

Chapter 5

5 Cyber Risk Assessment

In this section, we will divulge deeply into assessing risk in an IS. First, section 5.1 will discuss the fundamentals of our risk assessment process, which includes the generic risk score concept, situation awareness, attack surface analysis and our custom-made algorithm EICRS. Next, in section 5.2, we will discuss the technical environment used to develop our prototype and related incorporated technologies.

5.1 Risk Assessment Process

The risk assessment procedure starts with risk analysis, which is composed of two parts: (i) Risk Identification & Estimation and (ii) Risk Evaluation. To perform a risk analysis, we must first identify and estimate risk in the security posture. [9]

As per the RMF Step 4 shown in Figure 2, we assume that when a hypothetical system security audit takes place, all security controls must be in place and active. When NIST 800-53 security controls fail for whatever reason, it represents a potential risk in the security posture. Once the control failure is identified, our software will estimate the risk in the security posture by evaluating the impact of the failed control on other related controls. Once we have successfully estimated the risk in the system, we will move into the Risk Evaluation stage, where our designed algorithm will evaluate a cumulative risk to the system. This process is necessary for achieving comprehensive situational awareness as defined by Endsley's model to achieve the desired cyber security posture. Our next section will define Cyber Risk Situational Awareness (CRSA) in more detail.

5.1.1 Cyber Risk Situational Awareness (CRSA)

Our increasing dependency on IS has greatly amplified the need for situation awareness. It is an essential component towards understanding the cyber environment, accurately predicting the potential problems, and strengthening the cyber security posture. Cyber systems with vulnerabilities may present significant risks to commercial and national security organizations. By anticipating risks to these systems, organizations can develop effective countermeasures to protect their critical business processes. “Situation Awareness is being aware of what is happening around you and understanding what that information means to you now and in the future. This awareness is usually defined in terms of what information is important for a particular job or goal. The concept of Situation awareness is usually applied to operational situations, where people must have situation awareness for a specific reason, for example, to drive a car, treat a patient, or separate traffic as an air traffic controller” [12]. For this research, we have designed a novel risk assessment algorithm based on the situational awareness descriptive model, as shown in Figure 12, and was initially conceived by Endsley in 1995 [12]. Our algorithm will be targeted toward designing and building Level 1 of this model. We will go over the algorithm details in a later section.

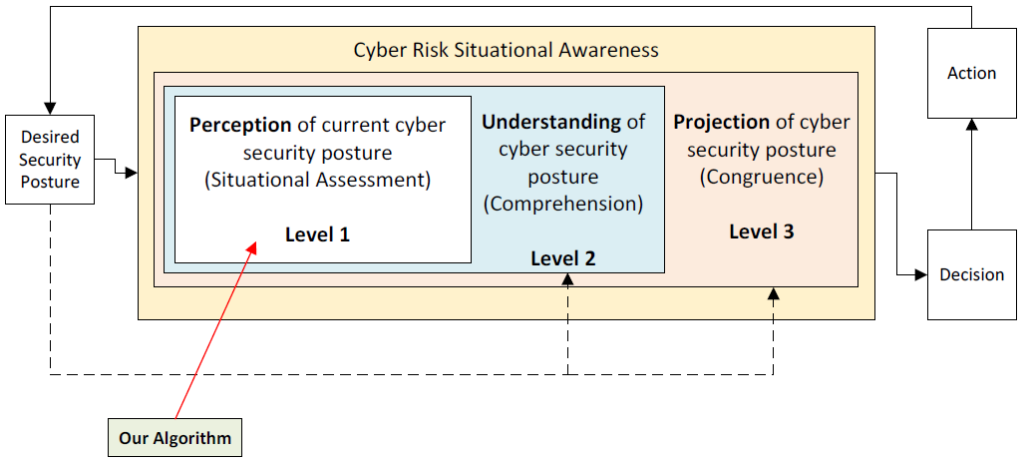


Figure 12: Endsley's CRSA Model

The CRSA model comprises three levels: (i) Perceptual Awareness, (ii) Comprehension Awareness, and (iii) Projection Awareness. The first level refers to the ability to gather and process information about the cyber environment, while the second level refers to understanding the meaning and significance of the information gathered in Level 1. Finally, the third level refers to the ability to predict future events and potential risks based on the information understood in level 2, such as predicting the likelihood of a cyber attack. We will now go into detail about each level to provide a quick overview of this model. [12]

Level 1 of awareness is crucial for detecting the presence of cyber threats and vulnerabilities in an organization's IS. The main goal of this level is to perceive the cyber environment by collecting and processing data from various sources, such as network logs, Intrusion Detection Systems (IDS), security alerts, etc. Achieving this level of awareness can be done through a combination of technical methods, such as using automated tools for data collection and analysis, and human methods, such as having trained personnel monitor and analyze the data. It also involves the capability to detect cyber threats and vulnerabilities within an organization's systems. It involves the

capability to identify unusual activity, including unusual network traffic, unauthorized access attempts, and other indicators of potential security violations. CRSA Level 1 is to obtain information about the cyber environment and identify cyber threats, which is the foundation for advanced comprehension and projection awareness. This level lays the groundwork for comprehending the cyber environment and detecting potential risks, which will then be utilized to establish protective measures to safeguard critical business operations. [12]

CRSA Level 2, or Comprehension Awareness, refers to the ability to understand the meaning and significance of the information gathered in Level 1. This level of awareness is vital for understanding the likely impact of a cyber threat and determining the appropriate response to it. It involves analyzing and interpreting the collected data in Level 1 and identifying patterns and trends. Awareness can be achieved through both technical and human means. Technical means include using data analytics and machine learning algorithms to identify patterns and trends in the data, while human means include having trained personnel interpret and analyze the data. This includes identifying the type of threat and determining the potential damage it could cause, such as data loss, system downtime, or financial loss. It also includes evaluating the threat's likelihood and determining its risk level. In a nutshell, Level 2 of CRSA is focused on understanding the meaning and significance of the information gathered in Level 1 and identifying patterns, trends, and potential impact, which will be used to develop effective countermeasures to protect critical business processes. [12]

CRSA Level 3, or Projection Awareness, refers to the ability to anticipate future events and potential risks based on the information understood in Level 2. This level of awareness is essential for predicting the likelihood of a cyber-attack and planning countermeasures to protect against it. It involves projecting future events based on the analysis and interpretation of data in Level 2. Security professionals may utilize machine learning, artificial intelligence, and simulation tools to anticipate future events and

identify vulnerabilities and risks. An essential technical capability an organization must possess is predicting the likelihood of a cyber-attack and preparing countermeasures to defend against it. Identifying possible attack scenarios, estimating the impact of each scenario, and evaluating the probability of each scenario occurring will enhance the security posture of the IS. Organizations can implement measures to protect against identified risks and vulnerabilities, such as deploying security controls, developing incident response plans, and preparing disaster recovery plans. [12]

Our novel algorithm focuses on CRSA Level 1 because it forms the basis for understanding the cyber security environment. This is essential for identifying potential risks and developing effective countermeasures to protect against them. It will also provide the foundation for our future research to build toward more advanced levels.

5.1.2 Endpoint Independent Cumulative Risk Score (EICRS)

We have designed and developed a non-linear, multi-dimensional algorithm based on the Generic Risk Scoring Concept, as shown in Figure 13, which also supports the design of the AWARE algorithm used by the Department of Homeland Security [13]. Our algorithm's dimensions are synonymous with those used in the AWARE algorithm; however, technical and implementation details differ. We now define the generic cyber risk score formula in the next section and then will review the algorithm dimensions and the actual algorithm itself.

5.1.2.1 Cyber Risk Score

Qualitative and Quantitative analysis are two kinds of risk assessments an organization can perform to assess the cyber security posture. Qualitative is a more subjective assessment of the posture than quantitative analysis, where you can define risk numerically. Every aspect of quantitative risk assessment, such as frequency, probability, impact, countermeasure effectiveness, and others, has a discrete mathematical value assigned to them. In most cases, organizations perform a mix of qualitative and quantitative approaches since comprehensive quantitative analysis might not be possible given resource constraints and required subjective input by security experts [8]. In this thesis, we have taken a hybrid risk assessment approach and quantified the risk using a generic cyber risk formula. Cyber risk is formally defined as a "function of three variables: threats, vulnerabilities, and impact" [29], as shown in Figure 13.

NIST 800-30 Rev1 defines threat as "any circumstance or event with the potential to adversely impact organizational operations and assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service" [14]. Threats can be human, natural, technical, physical, environmental, or operational [8] and have two parameters - capability and intent. We must only classify something or someone as a threat if both the parameters are validated. If something or someone has the capability to harm but does not have the intention to do so cannot be labeled as a threat. Also, if someone or something intends to cause harm but does not possess the required capabilities to do so cannot be categorized as a threat. Hence to be considered a threat, it must have both the capability and malicious intention to cause harm. The available cyber trend indicators usually analyze a threat's capability, whereas the threat's intention can be measured by undertaking cyber threat intelligence analysis.

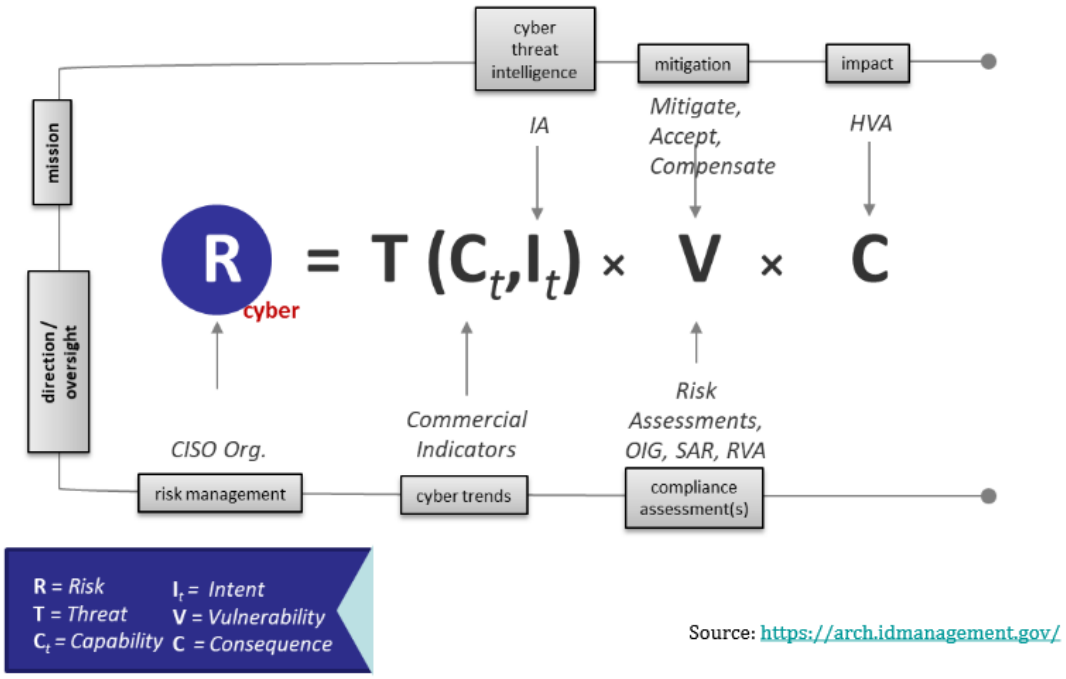


Figure 13: Generic Cyber Risk Scoring Concept

NIST 800-30 Rev1 defines a vulnerability as "an inherent weakness in an information system, security procedures, internal controls, or implementation that could be exploited by a threat source" [14]. Vulnerabilities can exist in "people, processes, data, technology, and facilities" [8]. We can address this risk score variable in several ways, such as endpoint scanning, code reviews (both manual and automatic), human psychological assessments and security interviews, physical and electromagnetic data security, physical and IS access control security, etc. Although addressing every aspect of this variable is desired, it may not be feasible due to many constraints. We have developed a novel concept of quantifying the vulnerability variable of a cyber risk function by assigning it a discrete mathematical value, which represents a potential weakness in a system upon failure of a particular control. Security controls are designed to prevent and detect weaknesses in processes, data, technology, and facilities and, upon failure, may precisely expose those same weaknesses.

NIST 800-30 Rev1 defines impact as “consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability” [14]. This variable of the cyber risk function is very subjective to the nature of an organization’s business. These organizations must appropriately define and assign impact definitions, including “loss of life, loss of dollars, loss of prestige, loss of market shares, or other factors” [8]. For this research, we will quantify the impact variable in a way that will reflect damage in the security posture as a consequence of control failure. We will go over the details in the following sections.

5.1.2.2 Our Algorithm – EICRS

Our algorithm, Endpoint Independent Cumulative Risk Score (EICRS), defines risk in terms of control decay (age), posture vulnerability, and impact on posture, representing Threat, Vulnerability, and Consequence in generic cyber risk function as explained in Section 5.1.2.1. These risk score variables will become our algorithm’s dimensions, as shown in Figure 14, and we will design our cumulative risk score around these dimensions. Let’s first formally define EICRS:

Definition 4 [EICRS]: Endpoint Independent Cumulative Risk Score (EICRS) = Age x Vulnerability x Impact is a multi-dimensional risk assessment algorithm, where Age represents the threat in terms of control decay, Vulnerability represents an attacker’s ability to breach security controls showing extend of the problem, and Impact represents the proportion of failed controls versus the total number of security control in the system. Each of these dimensions is scaled between 1 – 5. EICRS score is scaled between 1 – 125.

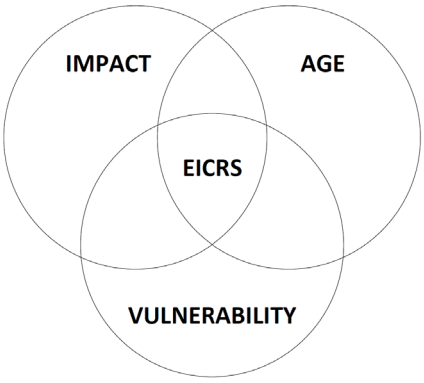


Figure 14: EICRS Dimensions

We have named this algorithm EICRS because it can evaluate the IS risk without considering each physical endpoint in the system and its related vulnerabilities. As per Definition 4, EICRS has three dimensions: Age, Vulnerability, and Impact, and we will now discuss these dimensions in detail. The EICRS score is scaled between 1-125, which we plan to normalize in future iterations between 1-100 using the normalization equation shown in Equation 1. This will help us set the standard scale and easily compare and analyze data. It will also help in improving the accuracy of our theoretical model.

The Age Factor denotes the elevated risk due to time elapsed without control validation from when it was first discovered as being failed. This dimension represents a threat in the EICRS risk score. When a security control fails, it may make the whole system vulnerable to many kinds of attack techniques, and with the passage of time, there is an increased likelihood that those techniques are more susceptible to successful exploitation. Hence, any security control decay is a direct risk to the cyber security posture of an organization. We will represent the passage of time in the number of days because it would help us define the decay in security controls as a natural decay, represented by a natural logarithmic formula.

If a control failure happens, and that failure is identified, we must also consider the grace period to fix the issue. The grace period is necessary because security teams in an industry environment sometimes require time to prepare the mitigation plan, techniques, and technologies upon identification of control failure. During this grace period, the risk score of an IS won't be affected. However, after the grace period, every passing day increases the likelihood of exploitation, which would also gradually increase the cyber risk of an IS. To connect decay and grace period to the aging threat, we have designed the logarithmic formula shown in Equation 2.

$$\text{Age Factor} = \text{Max Risk Score} + (\text{Max Risk Score} \times \text{Grace Factor}) \ln\left(\frac{\text{Age Decay}}{\text{Max Grace Period}}\right) \ldots \ldots 2$$

There are four variables forming the age factor formula: (i) Max Risk Score, (ii) Grace Factor, (iii) Age Decay, and (iv) Max Grace Period. Max Risk Score is the maximum score of the age dimension. Grace Factor is a decay curve's slope that can be set to control zero-risk days. Age Decay is the number of days the control failure has happened since. Max Grace Period is the maximum number of zero-risk grace days allowed. Every dimension is scaled between 1 – 5. The variables of this formula and their default values used in our POC are defined in Table 4, and we have also visualized the Age Factor formula in a scatter chart, as shown in Figure 15.

Formula Variables	**Description**	**Default Value**
Max Risk Score	Maximum scale value of the aging dimension	5
Grace Factor	Number of zero-risk grace days.	0.56 (10 days)
Age Decay	Number of days a control is failed since	User-defined
Max Grace Period	Maximum number of grace days after which	60+ Days

	threat to the IS would be catastrophic	

Table 4: Age decay formula components

Our next dimension of interest is the Vulnerability Factor. As discussed in Section 5.1.2.1, vulnerability represents the weakness in a system from a technological point of view. In our research, we have used this dimension in a novel way to compute the extent of the problem without leveraging endpoint scanning or code reviews. Instead, we see the system's vulnerability in the context of the weakness in an overall security posture. When we scan the endpoint or do code audits to assess the potential real-world vulnerabilities, that may not truly represent the actual state of security lapse. For example, even when the software system is up to date, a security control failure, such as an access control failure or an unnecessary open port, may lead to potential vulnerabilities which an intelligent attacker can exploit for their benefit. Therefore, we should complement technological vulnerability assessment alongside security posture vulnerability assessment to comprehensively understand an organization's security readiness.

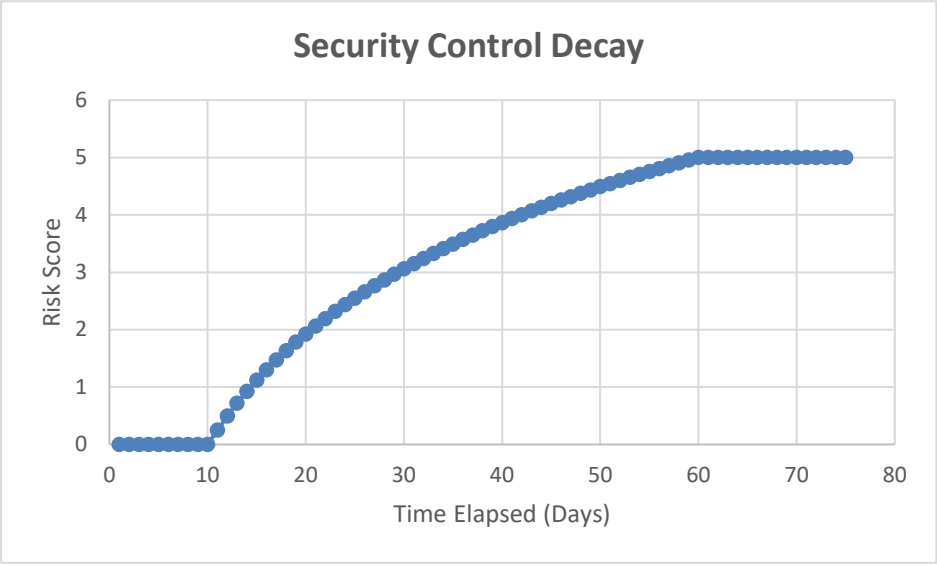


Figure 15: Security control decay graph

To determine the extent of the problem, we have devised a formula to assess the control-based vulnerabilities in a cyber security posture, which reflects an attacker's ability to compromise controls due to the weaknesses upon a particular control failure. We represent this using Equation 3 below:

$$\text{Vulnerability Factor} = \frac{\text{Failed Edges}}{\text{Total Number of Edges}} \quad \ldots\ldots\ldots\ldots\ldots\ldots 3$$

One metaphorical example of control-based vulnerabilities is house locks. Imagine that your house's front door malfunctioned for some reason. This malfunction can provide access to an attacker, and he/she may freely enter the house and navigate to different places. Even if there is a lock on each room door inside the house, once the attacker is inside, it would be relatively safer and easier for him/her to comprise those room locks as well. If the initial main door lock hadn't malfunctioned, or the lock failure was quickly identified and mitigated, this opportunity would have been denied to the adversary, and the likelihood of further successful exploitation would have been drastically reduced.

Our final dimension of EICRS is Impact, which denotes the proportion of failed controls versus the total number of active security controls. The formula for our impact factor is shown in Equation 4 below:

$$\text{Impact Factor} = \frac{\text{Failed Controls}}{\text{Total Number of Controls}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4$$

As discussed in Section 5.1.2.1, the impact dimension is very subjective to the nature of an organization's business, and the responsibility to define security lapse impact falls on the shoulders of corporate security professionals. When we look at the impact that may have occurred after a successful breach, we quantify it with respect to how many controls failed due to initial control failure, given our security posture. Since we define security posture is the total number of implemented security controls, the impact factor is proportional to the posture itself. Greater the damage in the posture, the higher the impact of a breach.

Our EICRS algorithm is shown in Figure 16. When discussing algorithms, it is also necessary to discuss their performance. We will now explore the EICRS's time and space complexities to better understand its efficiency.

Input: G = (V, E, R)

Output: Information System Risk

Pseudocode:

1. For Each Failed Control Found in G
    a. Compute Affected Sub-Graphs F_i
    b. For Each F_i
        i. Compute Age Factor
        ii. Compute Impact Factor
        iii. Compute Vulnerability Factor
        iv. Compute Risk Score
        v. Compute System Risk
        vi. Perform Attack Surface Analysis
        vii. Log Results
    c. End For
2. End For

Figure 16: EICRS Algorithm

Our program initially starts by deploying the cyber security posture for an IS using a SCDG by leveraging the JGraphT-based Java library. This library is developed around the Graph $< V, E >$ interface, as shown in Figure 17. The V and E generic parameters determine the types of Java objects utilized as graph nodes and edges, respectively. It allows for using any entity as a vertex or edge, and the interface enables basic graph operations and element access. All pre-defined graph classes implement this interface, and the algorithms require a Graph instance as input. [15]

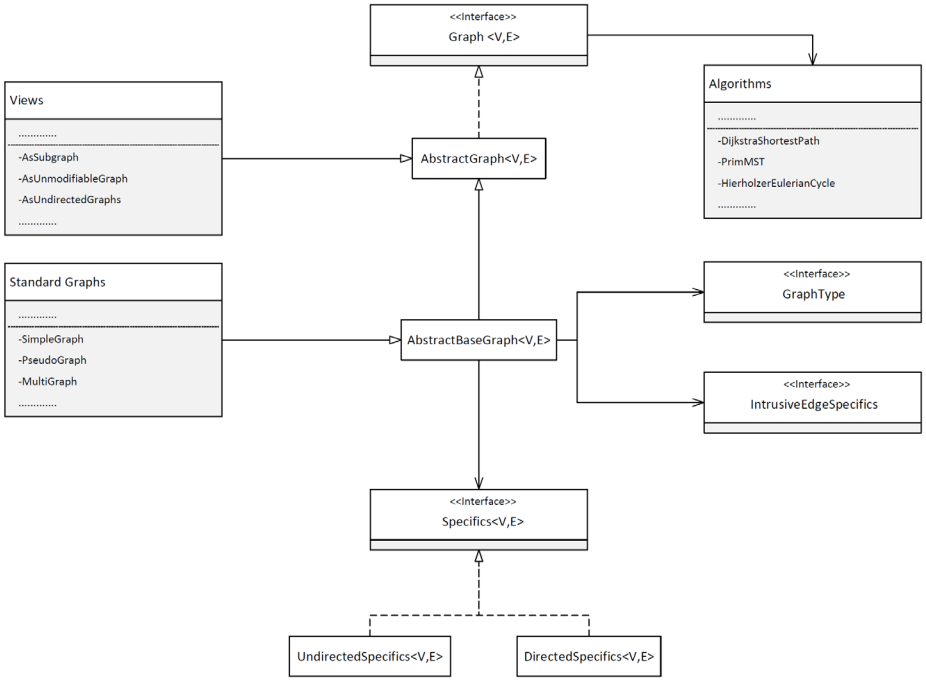


Figure 17: JGraphT Core Structure

The data structures used in JGraphT are highly customizable, providing a good balance between time and space complexities for common uses. The library enables users to create graphs utilizing the Builder design pattern and automatically identifies the most appropriate graph implementation. As shown in Figure 17, AbstractBaseGraph class extends AbstractGraph that "provides various methods to customize a graph and its storage mechanisms" using a customized implementation of Specifics and

IntrusiveEdgesSpecifics interfaces. Fundamental actions, such as adding or removing vertices or edges, can be executed in constant $O(1)$ time. When building our SCDG, we have used SimpleDirectedWeightedGraph $< V, E >$ class, where vertices are type String representing NIST 800-53 security controls, and edges are type DefaultWeightedEdge representing dependencies between controls. We also maintain another array of type ControlNode, which takes $O(V)$ space to store the corresponding control node object. This object consists of control related information, as shown in Figure 10. NIST 800-53 SCDG graph size (number of edges and vertices) takes $O(V + E)$ space to implement, where V is the number of security controls in NIST 800-53 and E is the set of edges representing dependencies between controls. [15]

After we fully deploy the SCDG, hypothetical auditing first identifies any root failed controls in the security posture, and then our algorithm EICRS will spin into the action. We iterate through each root failed control and perform a Depth-First Search (DFS) to identify other affected controls forming an affected sub-graph. DFS goes deeper into the graph whenever possible. We arbitrarily cut off the search after three vertexes to control propagation. It searches edges out of the most recently discovered nodes with unexplored edges fanning out from them. Once all edges have been properly explored, DFS backtracks to explore the other edges from the parent node. We visit each node $v \in V$ only once and during the execution of DFS, the loop executes $|Adj[v]|$ times. We know that $\sum_{v \in V} |Adj[v]| = \ominus (E)$ and since DFS is called once per vertex, the total cost of executing DFS is $\ominus (V + E)$, which is also our runtime of computing an affected subgraph. After all the affected sub-graphs have been identified, we perform the security posture analysis by conducting risk and attack surface assessments and logging all the required results. In the grand scheme of things, we consider the amount of work done by the security posture analysis shown between lines 1(b)(i) – 1(b)(vii) in Figure 16 to be constant $O(1)$ time. This algorithm repeats this process until all the root failed controls have been explored. Hence, based on the above analysis, we conclude that EICRS runtime is $\ominus (V + E)$. [30]

The Security Control Dependency Graph (SCDG) is a powerful tool for visualizing the cyber security posture of an information system. It can be used to identify the interdependencies between different security controls and to understand how one failed control can affect the effectiveness of other controls. An example of a fully deployed SCDG representing information system cyber security posture is shown in Figure 18, where we have selected NIST 800-53 controls for a system of criticality 7. This graph was generated by our tool and represented nodes in three colors: Green, Yellow, and Red. As per Table 3, these colors show the severity of the security control based on the RV_C value.

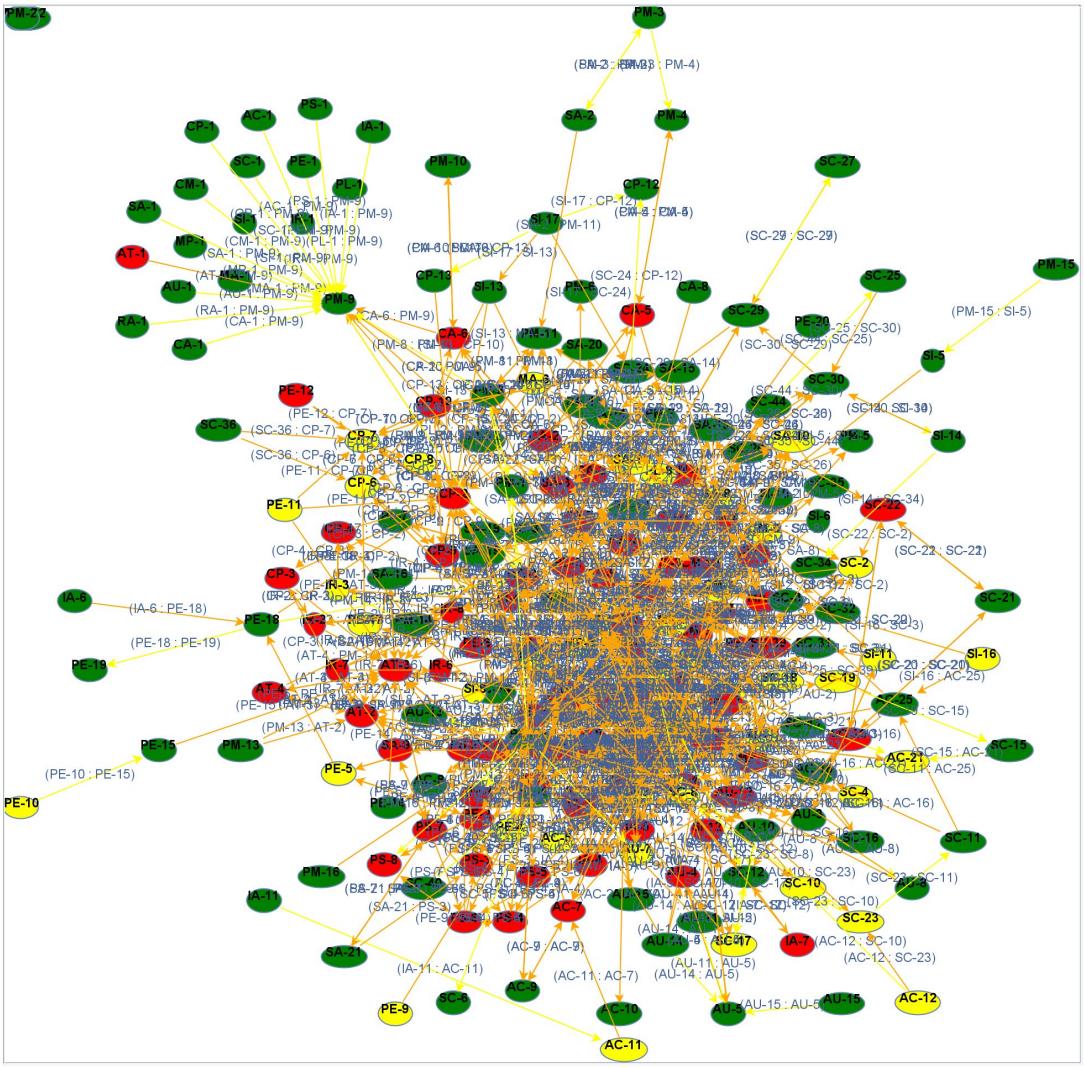


Figure 18: Cyber Security Posture shown in SCDG – System Criticality: 7

Once the SCDG is in place, EICRS would start propagation from the root failed control and list all the affected controls. This list is composed of controls whose RV_C value is less than the predecessor, as we have earlier discussed in detail in Section 4.2. We convert this list into a subgraph to visualize and perform our related actions, such as risk assessment and attack surface analysis. For example, Figure 19 shows an affected subgraph for a system whose criticality is 7, when NIST 800-53 security control AC-2 initially failed. EICRS would use the DFS-based propagation and builds the subgraph from this root failed control (AC-2). By forming the subgraph, EICRS can quickly and easily identify

all the controls affected by a single security control failure rather than manually identifying these dependencies. Once this subgraph is built up, EICRS performs risk assessment, system exposure analysis, and model performance-related operations. When there are multiple root failed controls, multiple subgraphs are formed. We have yet to devise a way to merge these subgraphs to build up the multi-control failure audit capability. This may be an area for future research to enhance the EICRS algorithm's capabilities.

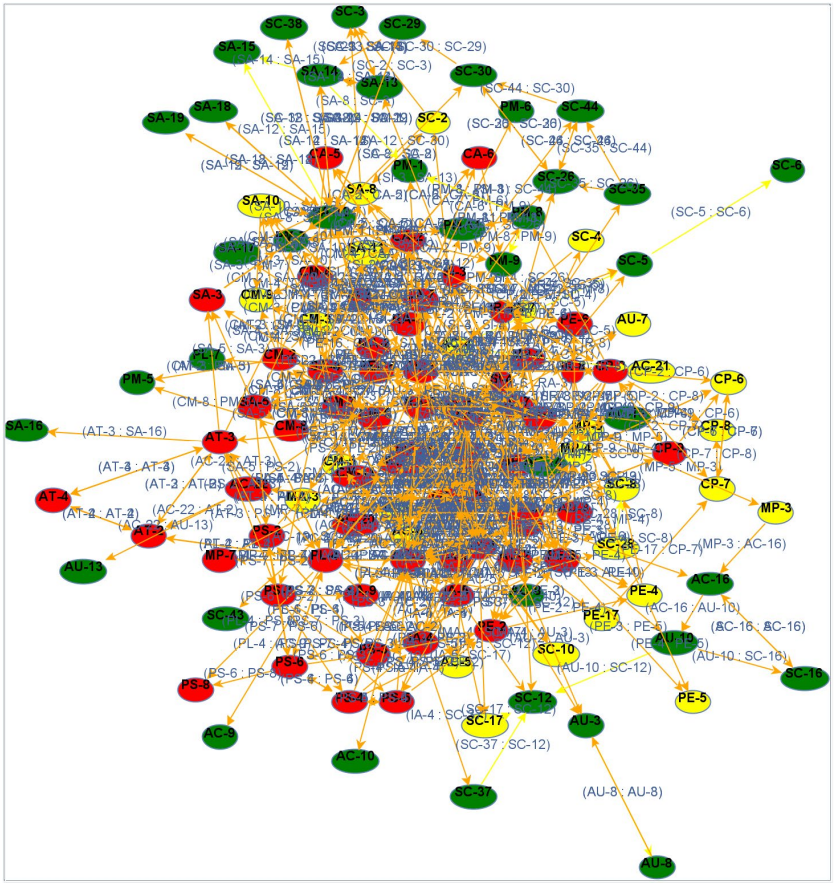


Figure 19: Control AC-2 Affected SCDG Sub-Graph – System Criticality: 7

When we represent the security posture of an IS as a graph, it has many useful reasons such as: (i) Visualization: A graph provides an ability to have a visual representation of

the interdependencies between different security controls, which makes it easier to understand the relationships between controls and the impact of a control's failure on other controls., (ii) Scalability: Graphs can be easily extended to include more nodes and edges as the security posture of an IS changes over time to include more or less controls. This makes it a scalable representation of the security posture that can grow as the information system evolves, and (iii) Efficient analysis: The algorithms used to traverse and analyze graphs are often more efficient than other methods for analyzing complex systems. For example, in the EICRS, a DFS algorithm is used to identify affected subgraphs, which have a time complexity of O(V+E). Overall, the graph provides a clear and efficient way to visualize and analyze the interdependencies between different security controls, making it a valuable tool for improving the overall cybersecurity posture of an information system.

5.1.2.3 Attack Surface Analysis

The attack surface is composed of the “components available to be used by an attacker against the product” or IS itself [8]. It refers to the total number of vulnerabilities and entry points an attacker can exploit. The attack surface includes all the different ways a malicious actor can interact with the system, such as (i) Network services and protocols: This includes all the network services and protocols that are exposed to the internet or other untrusted networks, such as the web servers, email servers, and file transfer protocols, (ii) Software vulnerabilities: This includes all the vulnerabilities in the operating system, application software, and firmware that can be exploited by an attacker, (iii) Hardware interfaces: This includes all the physical and logical interfaces that can be accessed by an attacker, such as USB ports, Ethernet ports, and wireless interfaces, and (iv) User interactions: This includes all the ways users interact with the system, including login interfaces, web forms, and command-line interfaces. The attack surface is an important consideration when assessing the security of a system or network. If the attack surface is large, the more potential vulnerabilities and entry points an attacker has that

he/she can potentially exploit. [31] Therefore, reducing the attack surface by disabling or limiting unnecessary services, applying software patches, and removing unnecessary hardware interfaces can help to improve security. [8]

MITRE Center for Threat-Informed Defense has been actively doing research on predicting IS attack surface by mapping NIST 800-53 security controls onto MITRE Tactics, Techniques and Procedures (TTP). As per MITRE, “These mappings provide a critically important resource for organizations to assess their security control coverage against real-world threats as described in the ATT&CK knowledge base and provide a foundation for integrating ATT&CK-based threat information into the risk management process.” [32] When it comes to practical security measures, frameworks like NIST 800-53 may not directly correspond to actionable TTP. To address this issue, MITRE has created a well-organized and publicly accessible set of mappings between the NIST controls and ATT&CK TTP. These mappings enable professionals to swiftly recognize the correlation between the security controls they have in place and the TTP utilized by attackers, which allows for a more efficient and focused approach to threat management. [16]

5.1.2.3.1 Security Control Mappings

Security controls to TTP mappings offer organizations a vital resource for evaluating the effectiveness of their existing security controls against actual threats listed in the ATT&CK knowledge base. The mappings also form the basis for “integrating ATT&CK-based threat data into the risk management process.” [32]

It is a time-consuming and often subjective process to map NIST to ATT&CK. Managing these mappings can be difficult and prone to errors due to the dynamic nature of security

threats and the large number of controls in a framework. MITRE has realized a need for mapping and “an opportunity to collaborate cooperatively and enhance threat-informed defense with the larger community by partnering with AttackIQ, the Center for Internet Security, and JPMorgan Chase.” This work has significantly lessened the load on the IT Security community by “providing over 6,300 unique mappings between NIST 800-53 and ATT&CK, allowing organizations to focus their limited time and resources on comprehending” how these mappings would help to strengthen their cyber security posture. All MITRE ATT&CK TTP are represented in Figure 20 with the NIST 800-53 Rev4 mapping coverage; “the darker the technique, the more NIST 800-53 controls map to it.” [32]

To understand the security control coverage and selection process behind the development of this mapping repository, it is crucial to consider MITRE's scoping decisions. The ATT&CK Scope, Controls vs. Control Enhancements, Policy & Procedure Controls, and Technical in Focus are among the scoping decisions considered. The scope of MITRE was limited to ATT&CK techniques within the Enterprise domain, with no coverage of Mobile techniques. Mappings are created at the security control level rather than the control enhancements. Since the focus is on NIST’s technical and operational aspects, policy and procedure controls are not considered within scope. Unlike non-technical mitigation methods, mappings are made for technical safeguards and countermeasures specific to the system. [32]

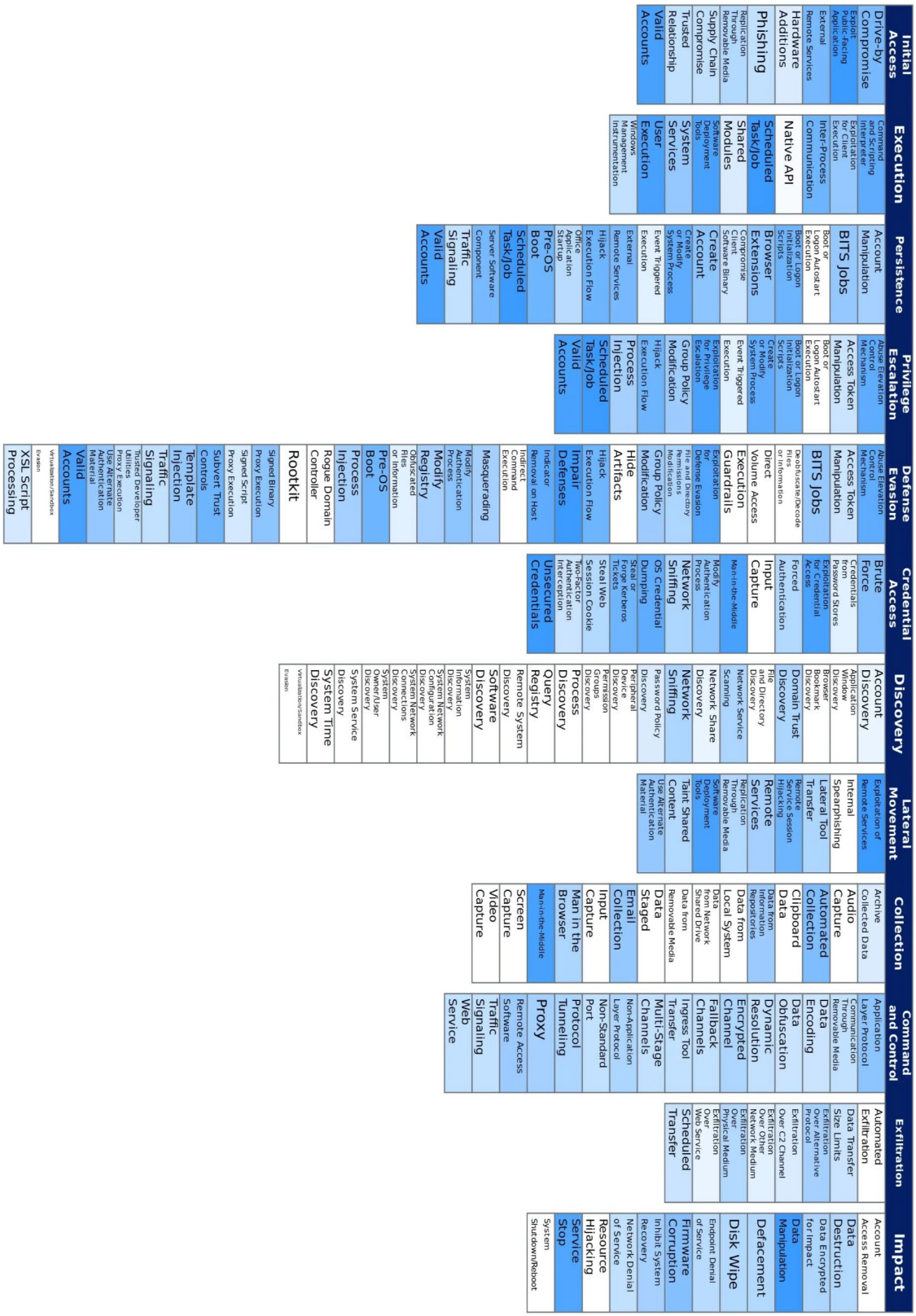


Figure 20: MITRE ATT&CK Framework – Navigator Layer snapshot

5.1.2.3.1.1 Controls – TTP Mapping Methodology

We will now discuss the methodology to map NIST onto the MITRE framework. MITRE developed this approach to translate NIST 800-53 to ATT&CK, but it was created to be easily customized and applied to various security control frameworks. Mapping the NIST onto the ATT&CK framework enables “the integration of ATT&CK-based threat intelligence into the risk management process” [32]. This gives organizations a robust approach to assessing their security control coverage in relation to related ATT&CK techniques.

The ATT&CK knowledge base represents adversary goals as tactics and the behaviors to achieve those goals (how) as techniques [16]. The ATT&CK Mitigation structure depicts security principles and categories of tools that may thwart the successful use of a group of techniques or sub-techniques. By utilizing the data in the ATT&CK knowledge base and its underlying data model, the process described below creates the context that is then utilized to choose which security controls to map to a particular technique or sub-technique [32].

If a security control is mapped to a TTP, it may prevent its successful execution, similar to an ATT&CK mitigation process. The mapping methodology does not indicate levels of mapping or the efficacy of the controls. Instead, controls are either assigned to a particular technique or sub-technique, or they are not. In this sense, the mappings offer an essential resource that is easy to understand and is meant to guide risk management choices. [32]

This methodology relies on ATT&CK's mitigations, which link TTP to the controls that counter them. The methodology consists of four key components that follow an iterative

process, gradually increasing an analyst's understanding of TTP in relation to mitigation. The analyst can then choose the appropriate security controls to map. This methodology consists of four steps, as shown in Figure 21: (i) ATT&CK Mitigation Review, (ii) ATT&CK Technique Review, (iii) Security Control Review, and (iv) Create a Mapping. [32]

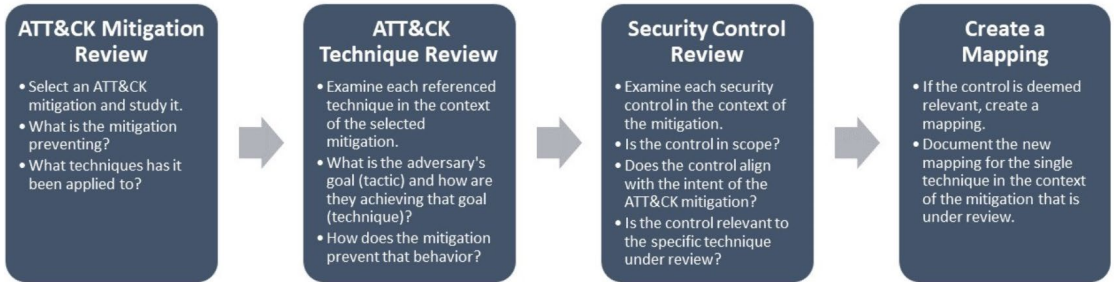


Figure 21: Security Control Mapping Methodology

Step 1 is about reviewing and analyzing each mitigation. The mitigations described by ATT&CK are classes of tools and security ideas that may stop a set of techniques or sub-techniques from being successfully executed. Studying these mitigations offers ideas and innovations to consider while identifying pertinent and useful security policies. Examining potential security controls is guided by a thorough understanding of the unique context of what a given mitigation is intended to avoid and how mitigation relates to or applies to a technique or sub-technique. Take the ATT&CK mitigation for Credential Access Protection ID: M1043 as an example. This mitigation covers a wide range of capabilities that prevent credential access and credential dumping. Next, M1043 lists several techniques and sub-techniques that this class of security capabilities could be able to thwart. Since ATT&CK mitigations are typically abstract, it is typical to discover more detailed guidance for each technique or sub-technique to which the mitigation is applied. [32]

Step 2 assists in comprehending the adversary's goals that a TTP is intended to achieve. It is vital to consider each technique and sub-technique independently because there could be significant differences between them that may lead to different security control mappings. Understanding the adversary's goal (tactic) and how (technique) they achieve that goal helps to improve our understanding of the mitigation and builds context as we examine relevant security controls. Relevant information is provided through ATT&CK techniques and sub-techniques, including domain- and platform-specific knowledge, configuration concepts, and tools. For instance, OS Credential Dumping ID: T1003 is a specific technique addressed by M1043 Credential Access Protection. This method is employed by adversaries trying to “dump account login and credential information from the operating system and applications, typically in the form of a hash or a clear-text password” [16]. Additionally, this technique's sub-technique LSASS Memory ID: T1003.001, which is related to M1043, exhibits a more specific behavior. This sub-technique involves an adversary attempting to access credentials stored in the Local Security Authority Subsystem Service's process memory (LSASS). [32]

Step 3 examines the controls from the perspective of the mitigation and specific TTP. It is essential to comprehend and recognize the security concepts and tools that can be employed to prevent the effective execution of a specific action. The context of the mitigation under review is applied as security controls are assessed for each technique or sub-technique examined in the previous stage. MITRE ascertains the relevance of each security control to the technique or sub-technique under consideration and whether it aligns with the mitigation's intended goals. For example, technique T1003 OS Credential Dumping and sub-technique T1003.001 LSASS Memory can be mapped to NIST 800-53 security control families of Access Control (AC), Configuration Management (CM), Risk Assessment (RA), and System and Information Integrity (SI). Additional contextual information provided classifies specific controls mapped to T1003 and T1003.01: AC-3 and AC-4 for Access Flow Enforcement and Information Flow Enforcement, CM-2 Baseline Configuration and CM-6 Configuration Settings, RA-5 for Vulnerability Scanning, and SI-4 for System Monitoring. [32]

Finally, step 4 identifies and creates security control mappings to TTP. The first three steps of the methodology provide the analytical context needed to identify a list of potential security controls. Following the identification of this potential list of security controls, it is further examined, evaluated, and modified in accordance with the control mapping scoping decisions to completely ascertain matches to techniques and/or sub-techniques. After completing this step, the security control selection is made, and mappings are constructed and linked to the chosen technique or sub-technique. Using the previous example as a guide, more examination and analysis reinforce the identified control choice, and mappings can be made for the techniques T1003 OS Credential Dumping and T1003.001 LSASS Memory. The subsequent resulting mappings are listed in Table 5. [32]

Technique	**Control(s)**
T1003	AC-3, AC-4
T1003	CM-2, CM-6
T1003	RA-5
T1003	SI-4
T1003.001	AC-3, AC-4
T1003.001	CM-2, CM-6
T1003.001	RA-5
T1003.001	SI-4

Table 5: Example Mappings

5.1.2.3.1.2 TTP – CVE Mapping Methodology

Once we obtain the control mappings for TTP, we try to characterize the impact of a vulnerability as described in the Common Vulnerabilities and Exposures (CVE) list, as shown in Appendix G. ATT&CK TTP explains the attack vectors that attackers could use to exploit a vulnerability and the potential outcomes they could achieve through such exploitation. [33]. These TTP "can be used as a set of standard terms to describe the exploitation process of a vulnerability." For instance, in the CVE to TTP mapping example shown in Figure 22, MITRE analyzed the following three techniques that could be performed to attack a flaw in the event of credentials being supplied in clear text: (i) Exploit Public-Facing Application (T1190), (ii) which gets an attacker unsecured credentials (T1552), and (iii) that leads to a valid account (T1078). [34] Once an attacker gains control of a valid account, there exist numerous potential routes that they could take [33]. It is simpler for defenders to include vulnerabilities in their threat modeling when a vulnerability is defined using TTP. MITRE aims to facilitate CVE standardization to support vendors, researchers, vulnerability databases, etc., communicating the vulnerabilities' impact [33]. This ATT&CK-based impact data will be useful to defenders in improving their risk models. CVEs with TTP references support security analysts in understanding their controls for a specific CVE better when used with security control frameworks that are mapped to ATT&CK. This methodology's ultimate goal is to provide a vital link between vulnerability management and threat modeling. [17]

When creating this methodology, MITRE discovered that three steps in the attack are typically the maximum that can be reasonably articulated. These three steps are: (i) Exploitation Technique, (ii) Primary Impact, and (iii) Secondary Impact, as shown in Figure 23 [17]. This model primarily shows that a vulnerability can be exploited using a particular technique, such as [EXPLOITATION TECHNIQUE], to achieve [Primary Impact], resulting in [Secondary Impact]. Exploitation Techniques refer to the methods utilized for vulnerability exploitation. Primary Impact pertains to the initial advantage

achieved by exploiting a vulnerability. Lastly, the Secondary Impact aims to ascertain the potential actions that the adversary could take by leveraging the benefits of the Primary Impact. [34]

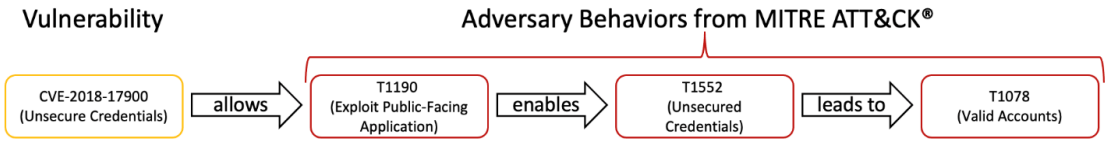


Figure 22: MITRE CVE to TTP Mapping Example

MITRE has established three techniques for linking CVE to TTP: (i) Vulnerability Type, (ii) Functionality, and (iii) Exploit Technique. The Vulnerability Type entails categorizing vulnerabilities that share similar vulnerability types (for example, XSS and Buffer Overflow), which also have common mappings. Functionality category groups together mappings "based on the specific type of functionality" that an adversary can gain by exploiting the vulnerability. Lastly, the Exploit Technique involves mappings based on the specific TTP utilized for vulnerability exploitation. [34] Of these three techniques, "only the Vulnerability Type method incorporates mappings for all three categories. The Functionality method has mappings for both Primary and Secondary impacts, whereas the Exploit Techniques method solely includes mappings for exploitation technique categories." [17]

In some cases, MITRE may not have exhaustively mapped every possible category for each approach, namely Exploitation Technique, Primary Impact, and Secondary Impact, as shown in Figure 23. TTP mapping is only deemed necessary if it is expected that multiple vulnerabilities within the category would use the same technique. For example, memory modification vulnerabilities (such as buffer overflows) have a shared Primary Impact. However, due to the significant variability of attack paths at the Secondary

Impacts and Exploitation Techniques stages, these two categories are not included in the methodology's mapping. [34] There may be more than one technique for mapping category in some groupings due to the common variations within those groupings. By leveraging MITRE's research of mapping NIST 800-53 to the ATT&CK framework, we have assessed the system exposure upon failure of security controls. [17]

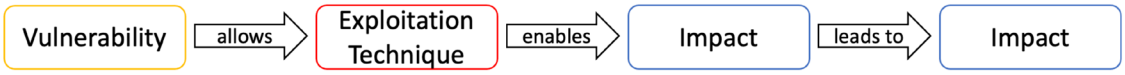


Figure 23: MITRE CVE to TTP Three-Step Mapping Model

Due to time constraints, our current POC has not yet implemented the technique of mapping TTP to CVEs. However, we plan to incorporate this technique in future studies as it would provide valuable insights into the relationship between specific attack strategies and known vulnerabilities in the information system. The ability to map TTPs to CVEs would allow us to better understand the potential vulnerabilities an attacker may exploit and help organizations better prioritize their efforts in addressing those vulnerabilities. This methodology would be a valuable addition to our current research, and we look forward to implementing it in future studies.

5.2 Technical Framework

Establishing a supportive technical environment was an essential step in conducting this research. As a part of this process, we established the required technical environment to support the development work in the language of our choice. This involved setting up the necessary software, hardware, and network to carry out the coding tasks efficiently. It was important to ensure the technical environment was robust enough to support our research activities, as the outcomes largely depended on the results' accuracy. The technical setup

was also critical to ensure the coding process was consistent and reproducible. After configuring the environment, the subsequent step was to commence coding. Section 5.2 is dedicated to delving into these technical aspects.

In Section 5.2.1, the focus is on the programming environment and tools used in the project. The language selection is discussed, and the use of an Integrated Development Environment (IDE) is highlighted. This section provides a clear understanding of the technical setup used in the project, which plays a critical role in making sure the coding process is streamlined and consistent. Section 5.2.2 focuses on the development methodology and project management approach used in the project. This section provides insight into the processes used to ensure the project is executed effectively and efficiently. This section also helps readers understand the steps taken to manage the project from start to finish. Lastly, Section 5.2.3 focuses on the third-party technologies incorporated into the POC. This section provides a comprehensive list of the technologies used, which helps readers understand the full scope of the project and the resources used to execute it. Incorporating third-party technologies is also critical in ensuring the project is executed with the most up-to-date and relevant technologies.

5.2.1 Programming Environment

Our first task was to select an appropriate programming language to build our Proof of Concept (POC). We opted for Java. The decision was based on many factors. Apart from personal expertise in this language, we have chosen Java because it has emerged as a preferred language for developing solutions due to its robustness, usability, cross-platform capabilities, and security features. It is an object-oriented language, which means it adheres to programming principles that incorporate ideas like class, inheritance, etc. Creating applications using Object Oriented Programming (OOP) concepts is

recommended because it makes the system extendable and flexible. Java encourages using Design Patterns to write code in a more structured and organized way.

Java is an open-source programming ecosystem, so you don't have to start from scratch. The developers are given access to the source code for dependencies so they can reuse it and redistribute it as needed. Java's fantastic programming tools make it much simpler for developers to work on projects. Java benefits from the assistance of a large community that aids in problem-solving as soon as they arise. The developer community welcomes the curiosity and support of other like-minded developers to promote the expansion of their network. The growing community exchanges knowledge and relevant data to aid aspiring developers in honing their coding abilities.

Java has a rich API repository that can help create software and apps by allowing the usage of various tools. The best thing about Java APIs is that programmers can use them even if they are unfamiliar with how internal coding structures are implemented. Java APIs are sufficiently compatible, which makes them a good fit for use with other codes. Additionally, more than 4500 APIs are available for use, so developers can pick whichever best suits their unique needs. These APIs provide practically everything, including DB connectivity, utilities, XML parsing, etc.

Java has multi-threading abilities. As a result, you can create incredibly responsive and dynamic apps by using multiple concurrent threads of activity. Java's support for multi-threaded environments also offers users quick reaction times, fewer problems, multi-operations, higher performance, and quicker concurrent access. Java also provides memory management to allocate or de-allocate objects. It helps in optimizing the effectiveness and speed of the applications. Java has an automatic memory management system that is called Garbage Collector. The Garbage Collector handles the allocation and release of memory to make room for an object. This ensures that developers using

managed code do not need to write code for memory management functions. The garbage collector helps identify and delete objects that cannot be reached in memory. It ensures that the heap has as much free space as possible.

We have used the Eclipse Java Integrated Development Environment (IDE) to manage our POC coding project. “An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining software editing, building, testing, and packaging capabilities in an easy-to-use application. Just as writers use text editors and accountants use spreadsheets, software developers use IDEs to make their job easier.” [18]. By providing a range of features such as the ability to set breakpoints, automatically validate syntax, a robust debugger, readymade code templates, and a powerful Java editor that supports code refactoring and syntax coloring, this IDE streamlined our productivity by consolidating the software development activities into one application, including editing source code, building executables, and debugging. As a result, this IDE effectively structured the code and proficiently handled all the related dependencies for our POC. [35]

5.2.2 Software Development Methodology

We have embraced the recommended project management guidelines and principles when developing our POC. We must first establish what a project is before delving into the specifics of our project management details. The Project Management Institute (PMI) defines a project as “a temporary endeavor undertaken to create a unique product, service or result” [19]. Every project must have the Goal, Timeline, Budget, Stakeholders, and Project Manager. In our case, the goal was to create a POC to show that we can predict the system-wide vulnerability exposure upon a particular control failure. Our timeline was

approximately one year, and stakeholders included my supervisor, IBM Canada (who has sponsored this work), and myself.

We have chosen the Agile Development methodology to complete this project. Agile project management approaches are becoming increasingly popular due to an intensely competitive corporate climate and more innovation. Agile techniques generally place a high priority on flexibility and shorter, iterative cycles. The Agile methodology has four central values: “(i) Individuals and interactions over processes and tools, (ii) Working software over comprehensive documentation, (iii) Customer collaboration over contract negotiation, and (iv) Responding to change over following a plan” [19].

One of the most popular methodologies based on the Agile Manifesto is Scrum, an iterative, incremental framework for managing and completing complex projects. Development teams use it to plan, execute, and deliver software incrementally. Scrum includes roles such as “the Product Owner, Scrum Master, and Development Team,” each with specific responsibilities and goals [36]. The basic process of Scrum is called a Sprint that generally lasts three to four weeks. When Sprint starts, the development team and the product owner create a Sprint backlog that needs to be completed during the Sprint. The team then works to complete the tasks on the backlog, with daily meetings to discuss progress, identify and resolve issues, and plan for the next day's work. Finally, at Sprint’s end, the team conducts a review to demonstrate what has been completed and plan for the next Sprint. [37]

Another Agile methodology is Kanban, a visual workflow management system that emphasizes workflow visualization, work-in-progress (WIP) limits, and continuous delivery. Instead of sprints and iterations, Kanban teams work in a continuous flow, with work items being pulled into the process as they become ready and progress through

various stages, such as "To-Do," "In Progress," and "Done." Kanban helps with reducing wait and improving the flow of work.

In both Agile methodologies, the teams work closely with the customer to gather and prioritize requirements and deliver working software at the end of each iteration. This allows for rapid feedback and helps the team to adjust course if necessary. For our project, we elected to use the Scrum methodology. In the first Sprint of our POC development, we gathered basic technical and topic-related requirements and developed the initial project scope. In the second iteration, we revised our requirements and developed the UML 2.0 based software design. In addition, we made sure that appropriate Design Patterns were implemented wherever required. In the third iteration, we started developing the prototype in Eclipse Java IDE along with requirements revision. In the remaining few Sprints, we integrated third-party technologies and developed new in-house technologies to address project requirements such as report management, graph visualizations, etc.

5.2.3 Technologies Developed and Incorporated

When developing a technical project, it is common to incorporate various 3rd party technologies to enhance its functionality or usability. The technical community always recommends never to reinvent the wheel. Some examples of 3rd party technologies that may be incorporated into a project include: (i) Frameworks, (ii) Libraries, (iii) Tools, and (iv) APIs, etc. It's always essential to ensure that any 3rd party technologies used are well-maintained, secure, and have significant community support. The technologies we have developed or incorporated for our research are listed in Table 6.

Technology Name	Technology Type	Technology Description
Commons-Lang3-3.12.0	3rd Party	"Apache Commons Lang, a package of Java utility classes for the classes that are in java.lang's hierarchy or are considered to be so standard as to justify existence in java.lang." [20]
Report Management	In-House	Our report management technology helps in gathering and logging required information to generate security intelligence reports regarding the system.
commons-math3-3.6.1.jar	3rd Party	"The Apache Commons Math project is a library of lightweight, self-contained mathematics and statistics components addressing the most common practical problems not immediately available in the Java programming language or commons-lang." [20]
Jackson-Annotations-2.11.3	3rd Party	"Core annotations used for value types by Jackson data binding package." [20]
Jackson-Core-2.11.3	3rd Party	"Core Jackson processing abstractions (aka Streaming API), implementation for JSON" [20]
Experiments	In-House	Manage the different experiment

Management		iterations and their results in an appropriate format and files
Jackson-Databind-2.11.3	3rd Party	"General data-binding functionality for Jackson: works on core streaming API" [20]
jFuzzyLogic	3rd Party	"jFuzzyLogic is an open-source Java library for fuzzy logic. It allows you to define fuzzy inference systems using a set of linguistic variables, fuzzy sets, and fuzzy rules. It provides a simple API for creating and executing fuzzy systems, as well as a set of tools for visualizing and debugging the systems." [20]
JGrapht-Core-1.5.0	3rd Party	"JGraphT is a free and open-source Java library for working with graphs. It provides a wide range of graph-theoretic algorithms and data structures, including support for directed and undirected graphs, weighted and unweighted graphs, and various types of graph traversals and shortest path algorithms. JGraphT Core is the core package of the JGraphT library. It provides the basic functionality for working with graphs, such as creating and manipulating vertices and edges, and performing graph traversals. The

		JGraphT core package also includes a number of utility classes for working with graph data structures and algorithms.” [20]
JGrapht-Ext-1.5.0	3rd Party	“JGraphT-Ext is an extension package for JGraphT library that provides additional functionality for working with graphs, such as graph generators, statistical analysis, graph visualization, and support for various file formats.” [20]
JGraphX	3rd Party	“JGraphX is a third-party library built on top of JGraphT, it's a Java Swing diagramming (graph visualisation) library for creating, editing, and automatically layout diagrams. It provides a wide range of features for creating diagrams and graphs, such as support for various graph layouts, automatic edge routing, and support for custom cell editors and renderers.” [20]
JHeaps-0.14	3rd Party	“JHeaps is a Java library for working with heaps data structure, it provides a generic implementation of several heap data structures, including binary heaps, Fibonacci heaps, pairing heaps, and d-ary heaps. It also provides efficient

		algorithms for common heap operations such as inserting, deleting and merging elements.” [20]
OpenCSV-5.3	3rd Party	“A simple library for reading and writing CSV in Java” [20]
XML JSON	3rd Party	“Conversion of XML to JSON and vice versa” [20]

Table 6: POC Technologies Incorporated

Chapter 6

6 Experiments & Results

In a research project, experiments and results refer to the process of testing and evaluating the project's performance and functionality. The results of these experiments can provide valuable insights into the strengths and weaknesses of the project and can be used to guide further development and optimization. The experiments conducted in a research project may include testing the project's model performance, such as accuracy. They may also include testing the model functionality, such as ensuring it works correctly under different conditions or inputs. The results of these experiments are typically recorded & analyzed and are used to identify areas where the project can be improved. Therefore, the experiments must be performed in a controlled and repeatable environment, allowing us to compare results with others and be able to replicate the experiments by others. Our experiments aim to determine the model's accuracy for NIST 800-53 to MITRE ATT&CK mappings by using the F1 Score.

This section will explain our experiments, their results, and the analysis & discussion. The key points of a porotype tool are presented in Section 6.1. The initial designs of our model accuracy experiments are addressed in Section 6.2. In the upcoming section 6.3, the design of F1 algorithms for our proposed model will be examined. Additionally, the results of our model's experiments will be analyzed and discussed in Section 6.4.

6.1 Prototype Tool: Purple Eye

To assess the proposed technique, we have designed and implemented a prototype tool that assesses security posture at a control level. Security in general can broadly be categorized into two dimensions: Defense and Offense, and cyber security is no different. Security professionals in the defensive domain are called blue teams, and professionals working legally to undermine security and perform offensive penetration testing are known as red teams.

In order to demonstrate our technique as a Proof of Concept (POC), we designed and developed the aforementioned Java-based tool, which we call Purple Eye. The motivation for the name emerges from the fact that it supports both types of security professionals: Blue and Red. This tool incorporates advanced technologies such as AIOps, Graph Management, Report Management, etc. The prototype tool provides valuable insights for both teams by creating a coherent security intelligence report to evaluate the desired system from both vantage points.

The prototype tool first deploys the NIST 800-53 security framework, then based on an audit, if a security control is found to be failing, a comprehensive security assessment takes place. Next, it determines the security risk in the system. It computes all the potential MITRE TTP that an attacker might leverage to undermine an IS's Confidentiality, Integrity, or Availability (CIA). It then tries to determine the potential Common Vulnerabilities and Exposures (CVE) by a methodology that is under active research by MITRE.

When building this tool, we have kept two most essential questions in mind, as recommended by [8]: (i) Will our tool “enhance any of the core security principles?” and

(ii) Will our tool “impact any of the core security principles?”. Security principles in these questions are Confidentiality, Integrity, and Availability. The first question interests us because we want our tool to scan the IS from the defensive point of view. In an organization’s security posture, security controls ensure that the security principles are adequately addressed. If any control violation is found, it may also violate those principles. Our software enhances all three security principles by performing a comprehensive risk assessment. It assesses the overall damage in a security posture upon a control failure, detects other affected controls, and quantifies the cyber risk using a non-linear multi-dimensional algorithm. All these actions output valuable system intelligence, which may be used to enhance the core security principles by mitigating any potential security issues.

The second question is also very interesting because it concerns the offensive security realm. This tool we have developed also examines the impact of control violations from an adversary’s point of view, i.e., control failure leading to potential vulnerabilities in the system. Once the security posture damage is assessed, we use this information to determine potential MITRE TTPs and real-world CVEs. An adversary armed with these CVEs and malicious intentions can easily become a threat to the system and may impact the core security principles by which that system is protected. If an organization’s security professionals have this information in advance, they can mitigate the issues by patching up the system from those potential vulnerabilities.

6.2 F1 Score

The F1 score evaluates a model's accuracy that considers both precision and recall. It is commonly used in binary classification problems where the goal is to balance precision and recall. “The F1 score is the harmonic mean of precision and recall, where an F1 score

reaches its best value at 1 (perfect precision and recall) and worst at 0" [38]. It is defined as:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \ldots\ldots\ldots\ldots\ldots\ldots 5$$

Precision is a metric used in conjunction with a recall to evaluate the model performance in a binary classification problem. It measures the model's ability to correctly identify positive instances while minimizing the number of false positives. In other words, precision measures the model's ability to avoid false alarms. Precision is calculated as the ratio of true positive instances to the total number of positive instances predicted by the model. The formula for precision is:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \ldots\ldots\ldots\ldots\ldots\ldots 6$$

For example, if a model predicts 100 instances as positive, 80 are actually positive, and 20 are false positives, the precision would be 80/100 = 0.8. A high precision means a model with a low false positive rate is less likely to classify a negative instance as positive. A low precision means that a model has a high false positive rate and is more likely to classify a negative instance as positive. High precision is desired in cases where false positives are costly and should be minimized, such as in medical diagnoses, spam filtering, and fraud detection. High precision means the model can identify the relevant positive instances and avoid the irrelevant ones, thus reducing the number of false alarms. However, precision alone is not enough to assess the performance of a model. It should be used in conjunction with recall, another metric that measures the model's ability to identify all the positive instances, regardless of the number of false positives.

Recall is a metric used to determine how well a model performs in a binary classification problem. It measures the amount of true positive results out of all the actual positive results in the dataset. In simpler terms, recall evaluates the model's ability to identify all relevant positive instances. The recall value is computed by dividing the number of true positive instances by the total number of actual positive instances in the dataset. We can calculate recall by the formula:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \ldots\ldots\ldots\ldots 7$$

For example, there are 120 positive cases in a dataset, and a model can correctly identify 90 as positive and 30 as false negatives. The recall in this scenario would be calculated as 90/120 = 0.75. When a model has a high recall, it has a low rate of false negatives and can identify a large proportion of positive cases. On the other hand, a low recall indicates that the model has a high rate of false negatives and cannot identify many positive cases. The high recall is desirable when false negatives are costly and must be avoided, for example, in medical diagnosis, search engines, and security systems. This is because a high recall indicates that the model can identify a large proportion of relevant positive cases and thus increases the chances of catching positive results.

For our model, precision tells us that out of MITRE TTP, which are supposed to be the positive match, how many TTP were actually identified as a positive match, which reduces the overall false alarms. Whereas recall tells that upon identification of a NIST 800-53 failed control, out of candidate TTP, how many TTP were successfully mapped and predicted, leading to the determination of true positives. F1 Score then would measure how well the model can correctly identify TTP and map them to NIST 800-53

controls, considering the number of false alarms and the number of missed TTP. It's worth noting that in some cases, precision is prioritized over recall, as in the case of Fraud Detection or Spam Filtering. However, recall is prioritized over precision in other cases, such as Disease Diagnosis or Search Engines, where missing a positive instance is more costly than a false positive. In practice, models are trained and tested to maximize the F1 score. The optimal threshold for a model is the point where precision and recall are balanced, which is when the F1 score is at its highest. The F1 score is commonly used along with other metrics like accuracy, precision, recall and AUC-ROC (Area Under the Receiver Operating Characteristics) to evaluate the performance of a model.

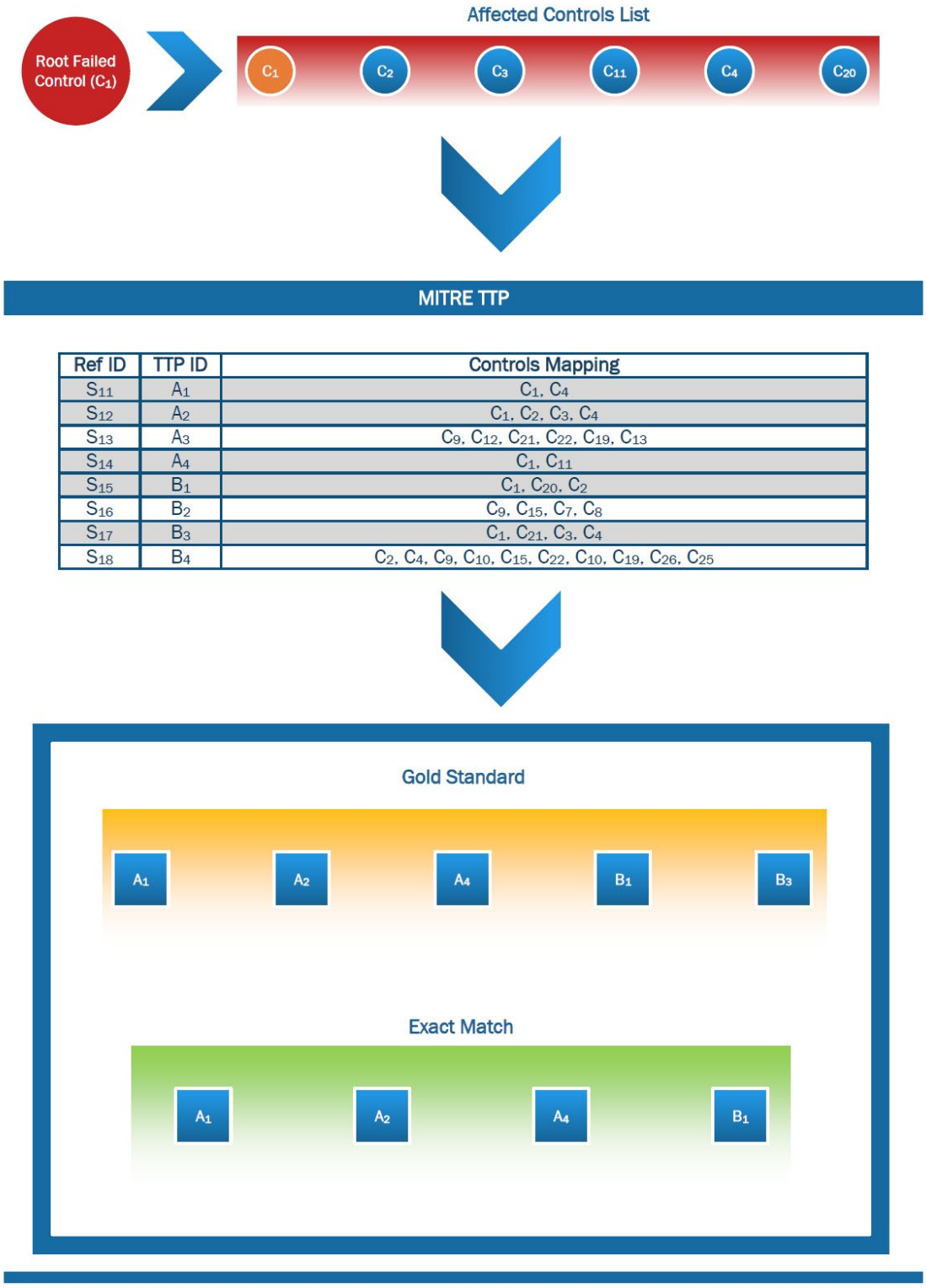
6.3 F1 Strategies & Algorithms

We have developed three different F1 evaluation strategies based on the traditional binary classification approach. The first technique we developed considers a TTP as correctly identified or not. This approach allows us to calculate precision and recall for the model and use them to compute the F1 Score. We call this technique the Attack Technique (AT) strategy. The second technique takes the union of control mappings for the predicted TTP and calculates precision and recall. This allows us to look at the model from another vantage point. This technique is known as Union (U) strategy. Finally, the third technique we developed combines the previous two techniques. We first predict potential TTPs from all the affected controls in parallel, converting those potential techniques into control mappings and then taking the union of those mappings to calculate precision and recall. This technique, known as Attack Technique Union (AT-U), allows us to evaluate the model performance from a more balanced view regarding expected trade-offs between precision and recall. Overall, these three F1 Score techniques allow us to evaluate our model from different perspectives and identify areas for improvement. By combining these techniques, we can ensure that our model is robust and accurate in identifying TTPs and mapping them to NIST 800-53 controls.

6.3.1 Attack Technique (AT) Strategy

The first F1 model we will discuss is Attack Technique (AT) Strategy. In this strategy, once a full security scan is completed, we start with the failed control C_1 and search the affected list of controls. We start iterating over the affected control list and determine TTP for each control. All those potential TTP that contain the relevant control in their mapping form the gold standard for matching purposes. After that, we will determine the exact TTP matches corresponding to the affected control list and calculate F1 Score. We will now go into detail about the AT strategy process.

AT strategy consists of three steps. First, we consider a set of initially failed controls. We will call this set I, for example, control C_1 in Figure 24 is a member of this set. We will then start the propagation phase and collect all the failed controls starting from the root failed controls in I. Let's denote this set as β, consisting of all the affected controls including the root failed control. In the second step, we select a control $C_i \in \beta$ and collect all the MITRE's TTP associated with this failed control, which means we go from a failed control in the propagation list to potential TTP. Let's call this set φ, which becomes our gold standard and contains true positive plus false negative matches. This set will consist of TTP whose control mappings at least contain control C_i. For example, in Figure 24, if the table lists all TTPs in MITRE ATT&CK, the TTP shown in grey color rows are the ones that will be part of φ. Third, we collect all the TTP whose mappings are the subset of β, and we will call this set ψ which will contain a list of TTP which are the exact match. This list will be composed of true positive plus false positive.



Ref ID	TTP ID	Controls Mapping
S_{11}	A_1	C_1, C_4
S_{12}	A_2	C_1, C_2, C_3, C_4
S_{13}	A_3	C_9, C_{12}, C_{21}, C_{22}, C_{19}, C_{13}
S_{14}	A_4	C_1, C_{11}
S_{15}	B_1	C_1, C_{20}, C_2
S_{16}	B_2	C_9, C_{15}, C_7, C_8
S_{17}	B_3	C_1, C_{21}, C_3, C_4
S_{18}	B_4	C_2, C_4, C_9, C_{10}, C_{15}, C_{22}, C_{10}, C_{19}, C_{26}, C_{25}

Figure 24: F1 Model – Attack Technique (AT) Strategy

When we have these two TTP lists, φ and ψ, we will work our way through the precision, recall and F1 score calculations. To keep the list simple, the F1 score formula has been left out from the list of formulas below, as it has already been discussed in the previous section in Equation 5. We calculate precision and recall by using the formulas in Equations 8 and 9 below:

$$Precision = \frac{Number\ of\ true\ positive\ matches}{Number\ of\ Exact\ Match\ elements} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 8$$

$$Recall = \frac{Number\ of\ true\ positive\ matches}{Number\ of\ Gold\ Standard\ elements} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ 9$$

Our algorithm for this F1 model is shown in Figure 25. This algorithm takes a SCDG and returns precision, recall and F1 score as per the discussion above. We know from Section 5.1.2.2 discussion that SCDG's space complexity is $\ominus (V + E)$. In this algorithm, we consider the work done between lines 1(a)(i)(1) – 1(a)(i)(3) to be constant for simplicity; hence, AT strategy algorithm will have space and time complexity of $\ominus (V + E)$.

```
Input: G = <V, E, R>

Output: Precision, Recall, F1 Score

Pseudocode:
List T1 = Gold Standard
List T2 = Exact Match

  1. For every control in base graph (NIST 800-53 Rev4):
       a. Check if control is present in the affected controls
          list
            i. If Yes:
                 1. Find all MITRE TTP this control belongs to
                    (T1).
                 2. Find all MITRE TTP which are subsets of
                    "Affected Controls" (T2)
                 3. Calculate Precision, Recall and F1 score
  2. Loop
```

Figure 25: AT Strategy Algorithm

6.3.2 Union (U) Strategy

The second strategy for our F1 Model is Union (U) strategy, as shown in Figure 26. In this strategy, we consider TTP mappings instead of TTP themselves in contrast to the previously discussed AT strategy. Like the previous strategy, upon completing a system security scan, we again consider each failed control in the affected control list separately. Let's discuss this strategy in detail now.

Union strategy consists of three steps. First, we consider a set of initially failed controls. We will call this set I, for example, control C_1 in Figure 26 is a member of this set. We will then start the propagation phase and collect all the failed controls starting from the root failed controls in I. Let's denote this set as β, consisting of all the affected controls including the root failed control. In the second step, we select a control $C_i \in \beta$ and collect all the MITRE's TTP associated with this failed control. We then convert the

collected TTP into a set of control mappings and union all the elements in this set. Let's call this set φ, which becomes our gold standard and contains true positive plus false negative matches. This set will consist of TTP whose control mappings at least contain control C_i. For example, in Figure 26, if the table lists all TTPs in MITRE ATT&CK, the TTP shown in grey color rows are the ones that will be part of φ. Third, we create another set $\psi \subseteq \beta$.

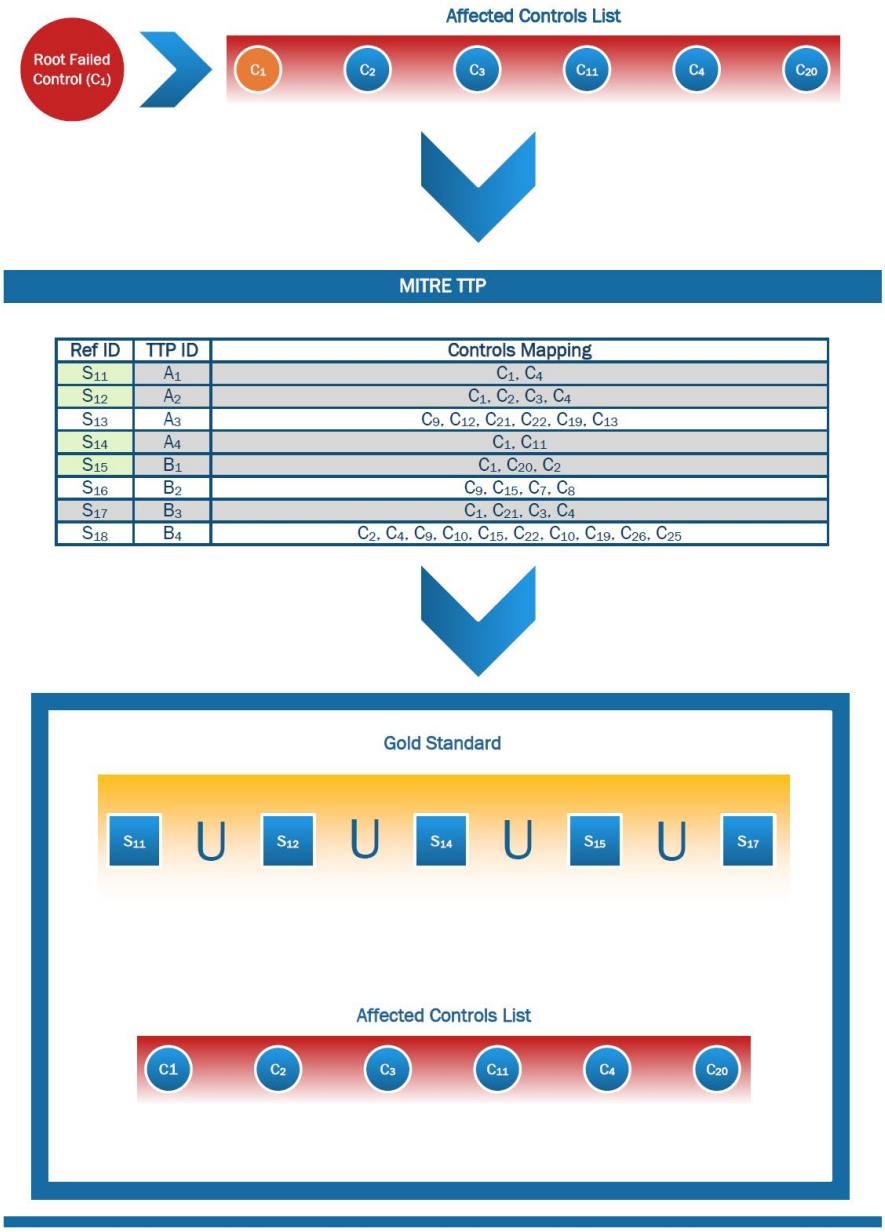


Figure 26: F1 Model – Union (U) Strategy

When we have these two lists composed of controls, φ and ψ, we will work our way through the precision, recall and F1 score calculations. We calculate precision and recall by using the formulas in Equations 10 and 11 below:

$$Precision = \frac{|\ Gold\ Standard\ \cap\ Affected\ control\ List\ |}{|\ Affected\ Control\ List\ |} \ldots\ldots\ldots\ldots\ldots\ldots\ldots 10$$

$$Recall = \frac{|\ Gold\ Standard\ \cap\ Affected\ control\ List\ |}{|\ Gold\ Standard\ |} \ldots\ldots\ldots\ldots\ldots\ldots\ldots 11$$

Our algorithm for the U Strategy is shown in Figure 27. Like the AT strategy, this algorithm will also take a SCDG and return precision, recall and F1 score. The space and time complexities for this algorithm will be similar to the AT strategy, which is ⊖ (V + E).

```
Input: G = <V, E, R>

Output: Precision, Recall, F1 Score

Pseudocode:
List T1 = Gold Standard
List T2 = Affected Control List

  1. For every control in base graph (NIST 800-53 Rev4):
       a. Check if control is present in the affected controls
          list
            i. If Yes:
                  1. Find all MITRE TTP this control belongs to
                  2. Union the found MITRE TTP controls mapping
                     (T1)
                  3. Find intersection between T1 and T2
                  4. Calculate Precision, Recall and F1 score
  2. Loop
```

Figure 27: Union (U) Strategy Algorithm

6.3.3 Attack Technique Union (AT-U) Strategy

The first two strategies for our F1 Model, the Attack Technique (AT) strategy and the Union (U) strategy involve conducting a full security scan and using the results to calculate the precision, recall, and F1 score. The AT strategy identifies the exact TTP matches corresponding to the affected control list, while the U strategy considers TTP mappings instead of TTP. Building upon these previous strategies, the third strategy for our F1 assessment, called Attack Technique Union (AT-U) strategy shown in Figure 28, is modeled by considering TTP and their mappings to provide a comprehensive and accurate model assessment.

The evaluation of the proposed approach consists of three steps. First, we consider a set of initially failed controls. Let's call this set I, and we collect all the failed controls after the propagation phase terminates, starting from the failed controls in I. Let's denote this set as β. Second, we collect all the MITRE's TTP and associate it with each failed control $C_i \in \beta$ (i.e., we go from all failed controls in the propagation list to TTP). Let's call this set τ. Third, we take the union of all controls associated with each TTP in τ using the MITRE's list (i.e., we go from attack strategies in τ to controls - note a TTP may have more than one control associated so the reverse process may yield fewer or more controls than β). Let's call this set Υ, and we consider it the ground truth. We then compute precision, recall, and F1 scores using the sets β and Υ similar to our Union (U) strategy using Equations 10 and 11.

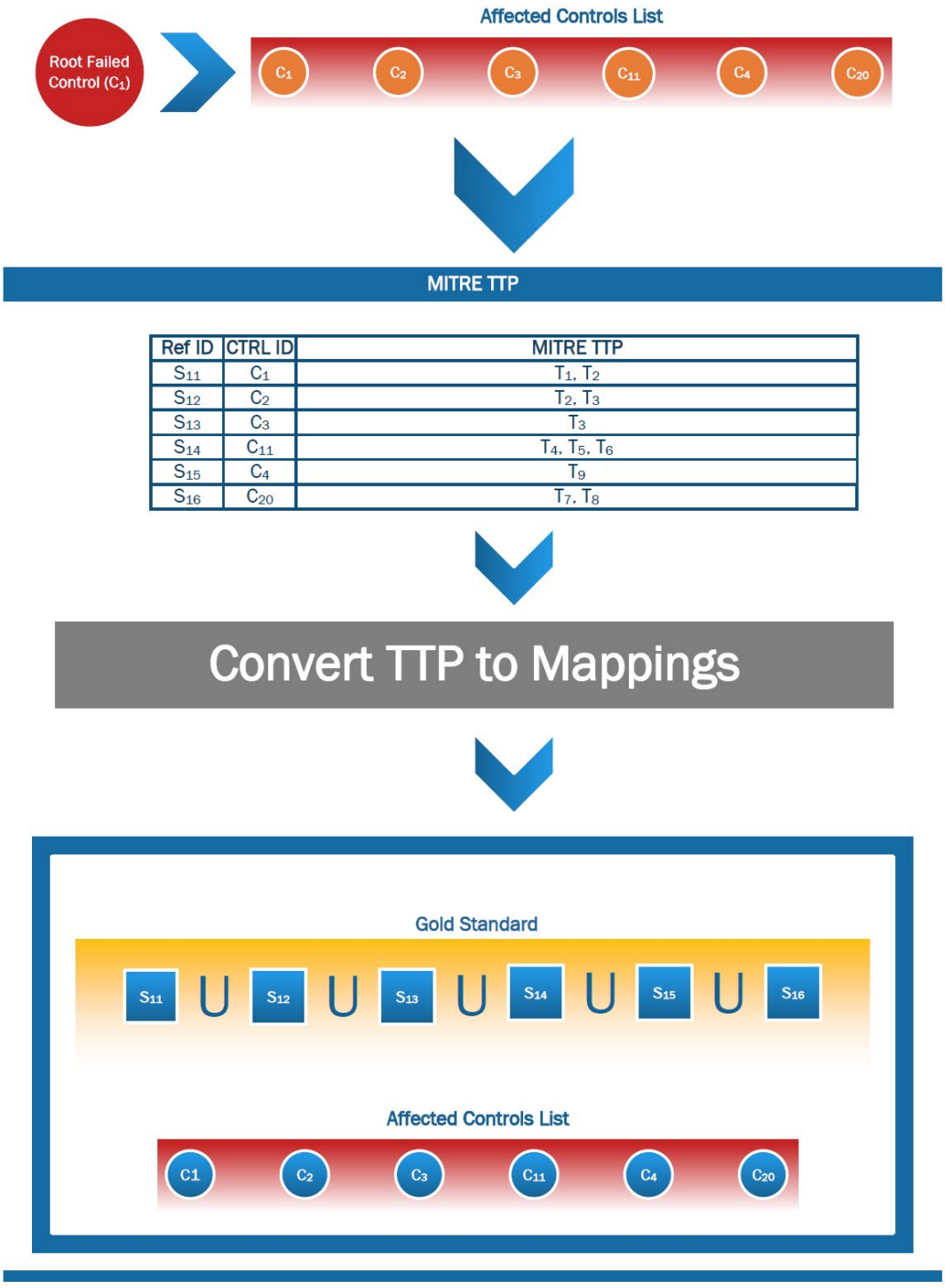


Figure 28: F1 Model – Attack Technique Union (AT-U) Strategy

We describe the algorithm for our AT-U algorithm in Figure 29. Similar to the AT and U strategies, this algorithm will also have the space and time complexity of $\ominus (V + E)$. In conclusion, the third strategy AT-U builds upon the previous two strategies by taking into account the specific TTP affecting the system and the relationship between TTP and their mappings, providing a more comprehensive and accurate assessment of the system's security. The combination of AT and U strategies, parallel computation of TTP and the union of all TTP mappings for each failed control enhance model accuracy predictability.

```
Input: G = <V, E, R>

Output: Precision, Recall, F1 Score

Pseudocode:
List S1 = Affected Controls
List T = Union of all potential for each control in S1
List S2 = Gold Standard
List S3 = S1 intersection S2

   1. For every control in base graph (NIST 800-53 Rev4):
        a. Check if it is member of S1
             i. If Yes:
                  1. Find all MITRE TTP this control belongs to
                  2. Add them into T
        b. Loop
   2. Union the TTP in T
   3. Generate S2 & S3
   4. Calculate Precision, Recall and F1 Score
```

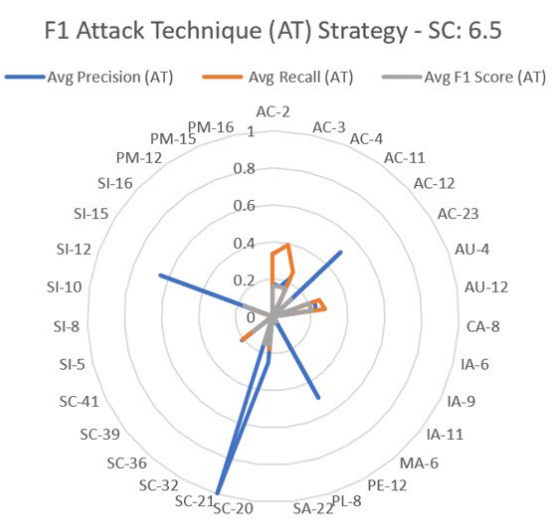
Figure 29: AT-U Strategy Algorithm

6.4 Experiments Result & Discussion

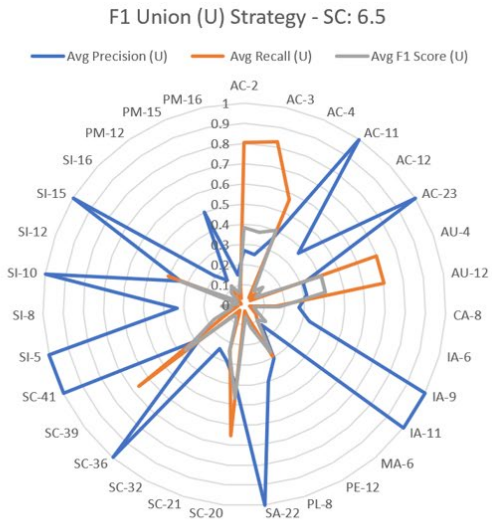
This section will provide details regarding the experiments performed, their results, and our analysis. We aim to evaluate whether the proposed propagation mechanism can assist in identifying a list of TTP that an attacker may exploit. Our basis is MITRE's ATT&CK framework which associates security controls to TTP and vice-versa. To measure the model accuracy and to understand the effect of control failure on security posture, we

decided to comprehensively scan NIST 800-53 for each of the five arbitrarily selected IS Security Categories (SC), which are 3, 4.5, 6.5, 8.5, and 10. We failed every control separately in that framework, which is recommended for the particular SC, and recorded average recall, average precision and average F1 score for all three F1 strategies (AT, U and AT-U, as discussed in Section 6.2). One such experiment was performed for IS category 6.5, as shown in Figure 30. The results suggest that the AT strategy did not perform well in terms of model accuracy, while the U strategy performed better than the AT strategy. The AT-U strategy performed slightly better than the other two strategies. The number of controls shown are less than the total number of controls in NIST 800-53 for a SC of 6.5 due to the condition imposed by us where we only recorded the control's average recall, precision and F1 scores if the data existed for at least one F1 strategy. Finally, once all the data had been collected, we took the comprehensive average of all the previously calculated averaged values for each F1 strategy.

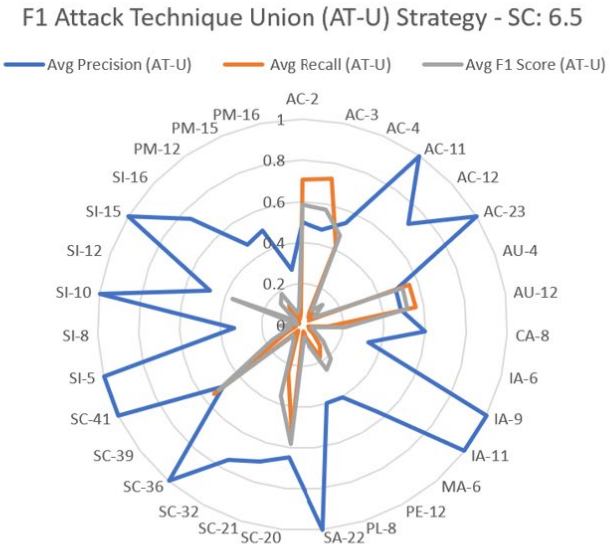
Our experiments found that the Attack Technique Union (AT-U) strategy best provided stable and consistent results. As described earlier, the AT strategy identifies the exact TTP matches corresponding to the affected control list. Furthermore, the U strategy considers TTP mappings instead of TTP themselves to provide a more accurate F1 assessment. Our AT-U strategy builds upon the previous two strategies by considering the specific TTP affecting the system and the relationship between TTP and their control mappings. The stable and consistent result obtained from the AT-U strategy in our experiments indicates that this approach is the most effective in providing accurate model assessment. Therefore, we will now shift our focus to the results of the AT-U strategy, as presented in Table 7, for the five SC we previously discussed.



(a) AT Strategy Results – SC: 6.5



(b) U Strategy Results – SC: 6.5



(c) AT-U Strategy Results – SC: 6.5

Figure 30: Experiment Results for System Security Category 6.5

System Criticallity (0-10)	Precision (Avg, StDev, Min, Max)	Recall (Avg, StDev, Min, Max)	F1-Score (Avg, StDev, Min, Max)
3 out of 10	(0.5, 0.2, 0.13, 1)	(0.23, 0.13, 0.02, 0.45)	(0.28, 0.13, 0.03, 0.5)
4.5 out of 10	(0.51, 0.21, 0.13, 1)	(0.22, 0.13, 0.02, 0.43)	(0.27, 0.13, 0.03, 0.48)
6.5 out of 10	(0.68, 0.25, 0.27, 1)	(0.2, 0.22, 0.02, 0.73)	(0.22, 0.2, 0.03, 0.59)
8.5 out of 10	(0.65, 0.23, 0.28, 1)	(0.2, 0.21, 0.02, 0.68)	(0.24, 0.19, 0.03, 0.6)
10 out of 10	(0.6, 0.21, 0.1, 1)	(0.24, 0.2, 0.02, 0.59)	(0.3, 0.19, 0.03, 0.57)

Table 7: AT-U Model Performance Results

The evaluation in its forward phase maps all the failed controls (i.e., after the propagation has been completed) to TTP, forming an expanded set of possible attack strategies an organization should defend against. In the backward phase, this expanded set of attack strategies is mapped back to an expanded set of possibly affected controls. Since a TTP is associated with more than one control and vice-versa, the control strategies identified by the proposed approach and the control strategies identified by the backward phase of the evaluation process may differ. Table 7 depicts the average, min, max, and standard deviation values for the comprehensive average of Precision, Recall, and F1 scores for AT-U strategy by experimenting when each NIST 800.53 control is failing separately for systems of mission criticality 3, 4.5, 6.5, 8.5, and 10, and for an elapsed period of 45 days. The results indicate that the comprehensive average recall and precision of our approach, assuming that only one control fails at a time, is low, but the standard deviation is relatively high. Also, we have observed that in many failed controls the recall is very high (e.g., above 0.8) while in others very low. The approach is performing very well for some controls while poorly for others. We are currently investigating which security control features affect this behavior.

Chapter 7

7 Conclusion & Future Work

In this thesis, we have presented a system for evaluating how failed security controls may trigger the investigation of other security controls and will affect the overall security posture of an Information System. Initial results indicate that the proposed approach and system create a comprehensive list of possible attack techniques an attacker can exploit and forewarn analysts of pending threats.

We have followed the industry standard risk assessment process proposed by NIST to create our proof of concept prototype. Specifically, we modeled the NIST 800-53 security controls and associated a control risk value (RV_C) with each control denoting its criticality. The multi-dimensional risk value for a control RV_C was determined by leveraging Fuzzy Logic's fuzzification and reasoning engines. We also designed and used a security control dependency graph to support the NIST 800-53 implementation and risk assessment process. Our work was built upon Endsley's CRSA model and quantified the risk using the cyber risk formula. Furthermore, we have designed and developed a custom-made non-linear multi-dimensional algorithm EICRS to evaluate cyber risk and system exposure by incorporating the MITRE ATT&CK framework. Finally, we have designed three strategies (AT, U and AT-U) to assess our model accuracy using the F1 score.

Future extensions of our research may include an extended ruleset and different membership functions. We also plan to transform the control dependency problem into a Constraint Satisfaction Problem (CSP), which has been proven helpful in many AI-style problems. "A CSP is defined as a triplet $< X, D, C >$, where X is a set of variables, D is a

domain of values, and C is a set of constraints $C_1(S_1) \ldots C_n(S_n)$ where each S_i is a set of variables. A constraint C_i is a combination of valid values for the variables S_i. A solution to the CSP is an assignment of values to $S_1 \ldots S_n$ that satisfies all constraints." [39] The required domains and constraints will be worked out with the industry specialists, and we may also elicit them from framework documentations and the internet by leveraging AI, Natural Language Processing (NLP) and other text parsing techniques. We also intend to use the Wave Propagation algorithm to improve our propagation method based on Pereira & Berlin's work. This technique improves overall algorithm running time, predictability, and scalability [21].

Another possible extension is to change some of our risk dimensions and their evaluation techniques. We calculated Vulnerability and Impact factors in the current thesis by considering controls. We haven't yet considered the endpoint or application vulnerabilities and how their exploitation would impact the security posture. In the future, we will add another vulnerability dimension which will be evaluated by scanning endpoints, networks and applications using industry-standard tools and software. The code-level compliance verification will be performed as proposed by [6] and other penetration testing methodologies. We will then compare the system exposure found by industry-standard tools to the exposure predicted by our methodology and improve upon the results.

This thesis researched one defensive framework, i.e., NIST 800-53. Hence there will always be a risk to the results. It would be interesting to see how ISO 27001/5 and other security frameworks would deliver in our proposed theoretical framework.

In this thesis, we have opted for an applied and experimental methodology to conduct our research. Still, other methodologies, such as surveys and questionnaires, case studies, etc., may improve the overall model and shed light upon other dimensions. For example, they may address human factors with Information Systems and can also help reduce human bias [1].

References

1. Kreicberga, L. (2010). Internal threat to information security - countermeasures and human factor within SME. Retrieved October 15, 2020, from
2. Pfleeger, C. P. (2015). Security in computing. In Security in computing (5th ed.). Westford, Massachusetts: Prentice Hall.
3. Jorshari, F. Z., & Tawil, R. H. (2015). A High-Level Scheme for an Ontology-Based Compliance Framework in Software Development. 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
4. Straub, D. W., & Welke, R. J. (1998). Coping with Systems Risk: Security Planning Models for Management Decision Making. MIS Quarterly, 22(4), 441. doi:10.2307/249551
5. Gupta, S., Chaudhari, B. S., & Chakrabarty, B. (2016). Vulnerable network analysis using war driving and security intelligence. 2016 International Conference on Inventive Computation Technologies (ICICT). doi:10.1109/inventive.2016.7830165
6. Martin Kellogg, Martin Schäf, Serdar Tasiran, and Michael D. Ernst. 2020. Continuous Compliance. In 35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20), September 21–25, 2020, Virtual Event, Australia. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3324884.3416593
7. Security, I. (n.d.). IBM QRadar SIEM Support of NIST 800-53 Security Controls. Retrieved October 15, 2020, from https://www.ibm.com/downloads/cas/ODQ9P07O
8. Gordon, A. (2015). Official (ISC)2 guide to the CISSP CBK (Fourth ed.). Boca Raton, FL: CRC Press.
9. United States of America, National Institute of Standards and Technology, Department of Commerce. (2013, April). NIST Special Publication 800-53 Revision 4 - Security and Privacy Controls for Information Systems and

Organizations. Retrieved September 17, 2020, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

10. Jha, S., Sheyner, O., & Wing, J. (2002). Two formal analyses of Attack graphs. Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15. doi:10.1109/csfw.2002.1021806
11. Alhomidi, M., & Reed, M. (2014). Attack graph-based risk assessment and optimisation approach. Journal of Internet Technology and Secured Transaction, 6(3). doi:10.20533/jitst.2046.3723.2014.0029
12. Webb, J., Ahmad, A., Maynard, S. B., & Shanks, G. (2014). A situation awareness model for information security risk management. Computers & Security, 44, 1-15. doi:10.1016/j.cose.2014.04.005
13. Homeland Security. (2018, November 7). AWARE Scoring (United States of America, Department of Homeland Security, CDM PMO). Retrieved February 26, 2021, from https://www.fbcinc.com/e/FITSC/presentations/Otto-FITSC2018.pdf
14. Initiative, J. T. F. T. (2012, September 17). *Guide for Conducting Risk Assessments*. CSRC. Retrieved August 11, 2022, from https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final
15. Michail, D., Kinable, J., Naveh, B., & Sichi, J. V. (2020). JGraphT—a Java library for graph data structures and algorithms. *ACM Transactions on Mathematical Software*, *46*(2), 1–29. https://doi.org/10.1145/3381449
16. Engenuity, M. I. T. R. E. (2022, September 1). *NIST 800-53 control mappings*. CTID. Retrieved September 27, 2022, from https://ctid.mitre-engenuity.org/our-work/nist-800-53-control-mappings/
17. Engenuity, M. I. T. R. E. (2022, November 3). *Mapping ATT&CK to CVE for impact*. CTID. Retrieved January 5, 2023, from https://ctid.mitre-engenuity.org/our-work/attck-to-cve/
18. *What is an IDE? - Integrated Development Environment Explained - AWS*. Amazon. (n.d.). Retrieved February 21, 2023, from https://aws.amazon.com/what-is/ide/
19. *Project Management Institute | PMI*. (n.d.). Retrieved January 7, 2023, from https://www.pmi.org/

20. *Maven Repository: Repositories*. (n.d.). Retrieved January 14, 2023, from https://mvnrepository.com/repos
21. Pereira, F. M., & Berlin, D. (2009). Wave propagation and deep propagation for Pointer Analysis. *2009 International Symposium on Code Generation and Optimization*. https://doi.org/10.1109/cgo.2009.9
22. *www.mitre.org*. MITRE ATT&CK Framework. (n.d.). Retrieved January 23, 2023, from https://www.mitre.org/sites/default/files/2021-11/getting-started-with-attack-october-2019.pdf
23. Derbyshire, R., Green, B., & Hutchison, D. (2021). "Talking a different language": Anticipating adversary attack cost for Cyber Risk Assessment. *Computers & Security*, *103*, 102163. https://doi.org/10.1016/j.cose.2020.102163
24. *What is the mitre ATT&CK framework?: Get the 101 guide*. Trellix. (n.d.). Retrieved February 16, 2023, from https://www.trellix.com/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html
25. Department of the Interior - Office of the Chief Information Officer, Cyber Security Division. (2008, March 17). *Information Technology Security policy handbook version 3*. Retrieved February 17, 2023, from https://cdn.ymaws.com/www.acil.org/resource/resmgr/imported/Dept_of_Interior_ITSecurityPolicy.pdf
26. McAllister, M. L., Ovchinnikov, S. A., Dockery, J. T., & Adlassnig, K.-P. (1985). Tutorial on fuzzy logic in simulation. Proceedings of the 17th Conference on Winter Simulation; - WSC '85. https://doi.org/10.1145/21850.253069
27. Talha, M., Asghar, F., & Kim, S. H. (2016). Fuzzy logic based energy management for wind turbine, photo voltaic and diesel hybrid system. *Journal of Korean Institute of Intelligent Systems*, *26*(5), 351–360. https://doi.org/10.5391/jkiis.2016.26.5.351
28. G, S., & Kumar, T. A. (2015). Design of Fuzzy Based Improved Disturbance Observer for Non-Minimum Phase Time Delay System. *International Journal & Magazine of Engineering, Technology, Management and Research*, *2*(12).

29. *Federal ICAM architecture introduction*. Federal ICAM Architecture Introduction. (n.d.). Retrieved February 17, 2023, from https://arch.idmanagement.gov/
30. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). The MIT Press.
31. Rieke, R. (2014). *Security Analysis of System Behaviour* (dissertation). Philipps University, Marburg, Bundesland.
32. Baker, J. (2021, June 17). *Security control mappings: A bridge to threat-informed Defense*. Medium. Retrieved February 17, 2023, from https://medium.com/mitre-engenuity/security-control-mappings-a-bridge-to-threat-informed-defense-2e42a074f64a
33. Aghaei, E. (2022). *Automated Classification And Mitigation Of Cybersecurity Vulnerabilities* (dissertation). The University of North Carolina, Charlotte.
34. Center-For-Threat-Informed-Defense, M. I. T. R. E. (2021, November 1). *Using MITRE ATT&CK® to Describe Vulnerabilities*. GitHub. Retrieved February 17, 2023, from https://github.com/center-for-threat-informed-defense/attack_to_cve/blob/master/methodology.md
35. GeeksforGeeks. (2022, May 20). *How to use mongodb in Eclipse?* GeeksforGeeks. Retrieved February 18, 2023, from https://www.geeksforgeeks.org/how-to-use-mongodb-in-eclipse/
36. Beerbaum Dr., D. O. (2022). Towards an agile organization in the financial service industry - regsafe2©. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4230454
37. UIKEY, N. I. T. I. N. (2016). *Limitations And Solutions For Agile Software Development Methodologies* (dissertation). ProQuest, Ann Arbor, MI.
38. Akinwamide, S. O. (2022). Prediction of fraudulent or genuine transactions on credit card fraud detection dataset using Machine Learning Techniques. *International Journal for Research in Applied Science and Engineering Technology*, *10*(6), 5061–5071. https://doi.org/10.22214/ijraset.2022.44962
39. Tlijani, H., Hatem, T., Jilani, K., & NacerKouider, M. (2014). A hybrid algorithm combining genetic algorithms and CSP approach to plane an optimized path for a

mobile robot moving under Time window. *IOSR Journal of Computer Engineering*, *16*(3), 74–83. https://doi.org/10.9790/0661-16337483

Appendices

Appendix A: Experiment Results for Information System of Security Category 3

Control ID	Avg Precision (U)	Avg Recall (U)	Avg F1 Score (U)	Avg Precision (AT)	Avg Recall (AT)	Avg F1 Score (AT)	Avg Precision (AT-U)	Avg Recall (AT-U)	Avg F1 Score (AT-U)
AC-2	0.416666667	0.351260987	0.361223976	0.347222222	0.048832032	0.073133074	0.5	0.252631579	0.335664336
	0.181818182	0.040906018	0.064852608	0	0	0	0.181818182	0.022988506	0.040816327
AC-3	0.306958474	0.382581655	0.318213934	0.253588517	0.033897046	0.055364015	0.409090909	0.284210526	0.335403727
AC-4	0.276719577	0.550516082	0.338051373	0.283157895	0.103595809	0.129625132	0.466666667	0.442105263	0.454054054
AC-5	0.3098659	0.41223093	0.330279777	0.325396825	0.046412653	0.074908249	0.402777778	0.305263158	0.347305389
AC-6	0.289772727	0.542692998	0.347603846	0.203463203	0.064547226	0.075943546	0.454545455	0.421052632	0.43715847
AC-11	0.484375	0.24232209	0.297828059	0	0	0	0.666666667	0.168421053	0.268907563
AC-12	0.458333333	0.098071941	0.158120379	0.5	0.077380952	0.133928571	0.75	0.088235294	0.157894737
AC-14	0.276851852	0.419830471	0.302149167	0.304812834	0.032548901	0.05503016	0.416666667	0.315789474	0.359281437
AC-23	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
AU-4	0.29382716	0.47564714	0.339218089	0.250666667	0.050406761	0.072147882	0.432098765	0.368421053	0.397727273
AU-5	0.291836735	0.276071619	0.262787538	1	0.024821144	0.047916031	0.408163265	0.210526316	0.277777778
AU-7	0.315068493	0.44862423	0.34533077	0.250666667	0.050406761	0.072147882	0.438356164	0.336842105	0.380952381
AU-14	0.351648352	0.419920915	0.358722559	0.325396825	0.046412653	0.074908249	0.430769231	0.294736842	0.35
AU-15	0.3	0.048401598	0.082829275	0	0	0	0.3	0.032967033	0.059405941
CM-3	0.325439267	0.481359758	0.361850759	0.258139535	0.082963615	0.112398115	0.441558442	0.357894737	0.395348837
CM-4	0.271428571	0.20617813	0.214840538	0.453125	0.025803692	0.047913604	0.35	0.147368421	0.207407407
CP-6	0.290148448	0.363767979	0.296531788	1	0.05	0.095238095	0.456140351	0.273684211	0.342105263
CP-7	0.27587564	0.460188029	0.319146511	0.6	0.020392151	0.038875963	0.428571429	0.347368421	0.38372093
CP-13	0.272727273	0.20227405	0.202446955	0	0	0	0.363636364	0.126315789	0.1875
IA-3	0.294256757	0.492634163	0.340124144	0.245192308	0.062540169	0.07573641	0.4625	0.389473684	0.422857143
IA-6	1	0.026315789	0.051282051	0	0	0	1	0.026315789	0.051282051
IA-9	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
IA-10	0.299465241	0.464166196	0.338463419	0.250666667	0.050406761	0.072147882	0.441558442	0.357894737	0.395348837
IA-11	1	0.071428571	0.133333333	0	0	0	1	0.071428571	0.133333333
IR-3	0.183333333	0.144290954	0.138274381	0	0	0	0.3	0.066666667	0.109090909
MA-3	0.381395349	0.284946729	0.311001469	1	0.05	0.095238095	0.465116279	0.210526316	0.289855072
MA-6	0.316358025	0.375408267	0.308450062	0.385416667	0.022740682	0.042198256	0.444444444	0.252631579	0.322147651
MP-3	0.299283154	0.366276162	0.307333064	0.5	0.031115858	0.057473165	0.435483871	0.284210526	0.343949045
MP-5	0.274725275	0.559089778	0.340215907	0.212301587	0.062708403	0.071947492	0.472527473	0.452631579	0.462365591
MP-6	0.277777778	0.536467765	0.340093299	0.215686275	0.052012351	0.068927524	0.466666667	0.442105263	0.454054054
PE-4	0.314652015	0.505476222	0.362872024	0.260416667	0.064623503	0.076274657	0.448717949	0.368421053	0.404624277
PE-5	0.362637363	0.197470636	0.249615884	1	0.024821144	0.047916031	0.371428571	0.138297872	0.201550388
PE-10	0.125	0.04	0.060606061	0	0	0	0.125	0.04	0.060606061
PE-11	0.251875	0.374662839	0.272609156	1	0.024821144	0.047916031	0.390625	0.263157895	0.314465409
PE-12	0.272727273	0.20227405	0.202446955	0	0	0	0.363636364	0.126315789	0.1875
PE-17	0.308426733	0.473379025	0.345300525	0.375	0.032654658	0.058347044	0.452054795	0.347368421	0.392857143
PE-20	0.666666667	0.077798731	0.138465	0.6	0.026647911	0.05075333	0.666666667	0.063157895	0.115384615
PL-8	0.284313725	0.465964951	0.324931318	0.265714286	0.060820937	0.077391256	0.435897436	0.357894737	0.393063584
SA-2	0.111111111	0.030235344	0.045064386	0	0	0	0.2	0.032967033	0.056603774
SA-11	0.260759494	0.420523897	0.294405545	0.265714286	0.060820937	0.077391256	0.379746835	0.315789474	0.344827586
SA-18	0.362637363	0.197470636	0.249615884	1	0.024821144	0.047916031	0.371428571	0.138297872	0.201550388
SA-19	0.33877551	0.193569672	0.235187539	1	0.024821144	0.047916031	0.4	0.147368421	0.215384615
SA-22	0.293040293	0.359569878	0.303558478	0.5	0.031115858	0.057473165	0.412698413	0.273684211	0.329113924
SC-4	0.388585608	0.449722329	0.396764609	0.328125	0.098414344	0.142447996	0.476923077	0.326315789	0.3875
SC-17	0.418181818	0.137622967	0.200143459	0	0	0	0.733333333	0.120879121	0.20754717
SC-18	0.336904762	0.482730619	0.371415151	0.272435897	0.046688603	0.07302959	0.486111111	0.368421053	0.419161677
SC-19	0.360449735	0.389375938	0.346371978	0.347222222	0.048832032	0.073133074	0.482142857	0.284210526	0.357615894
SC-20	0.408163265	0.410055112	0.37694726	0.333333333	0.032195143	0.054711706	0.612244898	0.315789474	0.416666667
SC-21	0.34841629	0.150806867	0.198162494	0	0	0	0.764705882	0.142857143	0.240740741
SC-31	0.303209459	0.442686037	0.334308191	0.272435897	0.046688603	0.07302959	0.432432432	0.336842105	0.378698225
SC-32	0.425531915	0.351260987	0.364682437	0.347222222	0.048832032	0.073133074	0.510638298	0.252631579	0.338028169
SC-36	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
SC-38	0.414835165	0.176849821	0.240417285	0.333333333	0.030301083	0.053558542	0.464285714	0.136842105	0.211382114
SC-40	0.359844271	0.42488481	0.359624523	0.265714286	0.060820937	0.077391256	0.467741935	0.305263158	0.369426752
SC-41	1	0.090909091	0.166666667	0	0	0	1	0.090909091	0.166666667
SI-5	0.349462366	0.195358245	0.229843882	0.453125	0.025803692	0.047913604	0.387096774	0.126315789	0.19047619
SI-7	0.379545455	0.521882578	0.406401528	0.236111111	0.066209016	0.087786975	0.606060606	0.421052632	0.49689441
SI-8	0.293137908	0.507790628	0.341353127	0.245192308	0.062540169	0.07573641	0.481012658	0.4	0.436781609
SI-10	1	0.016393443	0.032258065	0	0	0	1	0.016393443	0.032258065
SI-11	0.338171263	0.350010313	0.316330435	1	0.024821144	0.047916031	0.490566038	0.273684211	0.351351351
SI-13	0.263736264	0.210008956	0.207534401	1	0.05	0.095238095	0.358974359	0.147368421	0.208955224
SI-15	0.299242424	0.444849523	0.331405324	0.3125	0.037473316	0.061783454	0.458333333	0.347368421	0.395209581
SI-16	0.443223443	0.181758731	0.246703335	0	0	0	0.619047619	0.136842105	0.224137931
SI-17	0.1875	0.128005274	0.13358701	0	0	0	0.375	0.06741573	0.114285714
PM-12	0.229457731	0.518804758	0.295121835	0.245192308	0.062540169	0.07573641	0.398058252	0.431578947	0.414141414
PM-13	0.25	0.161443736	0.182585136	0	0	0	0.4	0.144578313	0.212389381
PM-15	0.666666667	0.116257275	0.193190016	1	0.05	0.095238095	0.666666667	0.065934066	0.12
PM-16	0.256571429	0.372566991	0.277164309	0.342857143	0.034827302	0.06044712	0.357142857	0.263157895	0.303030303

Comp. Avg Precision (U)	Comp. Avg Recall (U)	Comp. Avg F1 Score(U)	Comp. Avg Precision (AT)	Comp. Avg Recall (AT)	Comp. Avg F1 Score(AT)	Comp. Avg Precision (AT-U)	Comp. Avg Recall (AT-U)	Comp. Avg F1 Score (AT-U)
0.391150962	0.298861518	0.257263142	0.329949927	0.033230458	0.050734395	0.504815113	0.225988721	0.277108538
Standard Deviation	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**
0.225228601	0.170719588	0.102942995	0.322359151	0.026895147	0.037954891	0.202514398	0.132283573	0.129857547
Minimum	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**
0.111111111	0.016393443	0.032258065	0	0	0	0.125	0.016393443	0.032258065
Maximum	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**
1	0.559089778	0.406401528	1	0.103595809	0.142447996	1	0.452631579	0.49689441

Appendix B: Experiment Results for Information System of Security Category 4.5

Control ID	Avg Precision (U)	Avg Recall (U)	Avg F1 Score (U)	Avg Precision (AT)	Avg Recall (AT)	Avg F1 Score (AT)	Avg Precision (AT-U)	Avg Recall (AT-U)	Avg F1 Score (AT-U)
AC-2	0.416666667	0.351260987	0.361223976	0.347222222	0.048832032	0.073133074	0.5	0.252631579	0.335664336
	0.181818182	0.040906018	0.064852608	0	0	0	0.181818182	0.022988506	0.040816327
AC-3	0.306958474	0.382581655	0.318213934	0.253588517	0.033897046	0.055364015	0.409090909	0.284210526	0.335403727
AC-4	0.279651163	0.530618662	0.334305461	0.283157895	0.103595809	0.129625132	0.465116279	0.421052632	0.44198895
AC-5	0.3098659	0.41223093	0.330279777	0.325396825	0.046412653	0.074908249	0.402777778	0.305263158	0.347305389
AC-6	0.289772727	0.542692998	0.347603846	0.203463203	0.064547226	0.075943546	0.454545455	0.421052632	0.43715847
AC-11	0.484375	0.24232209	0.297828059	0	0	0	0.666666667	0.168421053	0.268907563
AC-12	1	0.034482759	0.066666667	0	0	0	1	0.034482759	0.066666667
AC-23	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
AC-25	0.381818182	0.368589348	0.356172532	0.304812834	0.032548901	0.05503016	0.454545455	0.263157895	0.333333333
AU-4	0.29382716	0.47564714	0.339218089	0.250666667	0.050406761	0.072147882	0.432098765	0.368421053	0.397727273
AU-5	0.291836735	0.276071619	0.262787538	1	0.024821144	0.047916031	0.408163265	0.210526316	0.277777778
AU-7	0.315068493	0.44862423	0.34533077	0.250666667	0.050406761	0.072147882	0.438356164	0.336842105	0.380952381
AU-14	0.351648352	0.419920915	0.358722559	0.325396825	0.046412653	0.074908249	0.430769231	0.294736842	0.35
AU-15	0.3	0.048401598	0.082829275	0	0	0	0.3	0.032967033	0.059405941
CM-3	0.325439267	0.481359758	0.361850759	0.258139535	0.082963615	0.112398115	0.441558442	0.357894737	0.395348837
CM-4	0.271428571	0.20617813	0.214840538	0.453125	0.025803692	0.047913604	0.35	0.147368421	0.207407407
CP-6	0.290148448	0.363767979	0.296531788	1	0.05	0.095238095	0.456140351	0.273684211	0.342105263
CP-7	0.306956522	0.319143456	0.288115266	1	0.05	0.095238095	0.46	0.242105263	0.317241379
CP-13	0.272727273	0.20227405	0.202446955	0	0	0	0.363636364	0.126315789	0.1875
IA-3	0.294256757	0.492634163	0.340124144	0.245192308	0.062540169	0.07573641	0.4625	0.389473684	0.422857143
IA-6	1	0.026315789	0.051282051	0	0	0	1	0.026315789	0.051282051
IA-9	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
IA-10	0.299465241	0.464166196	0.338463419	0.250666667	0.050406761	0.072147882	0.441558442	0.357894737	0.395348837
IA-11	1	0.071428571	0.133333333	0	0	0	1	0.071428571	0.133333333
IR-3	0.183333333	0.144290954	0.138274381	0	0	0	0.3	0.066666667	0.109090909
MA-3	0.381395349	0.284946729	0.311001469	1	0.05	0.095238095	0.465116279	0.210526316	0.289855072
MA-6	0.313545151	0.349555419	0.295621419	0.385416667	0.022740682	0.042198256	0.442307692	0.242105263	0.31292517
MP-3	0.299283154	0.366276162	0.307333064	0.5	0.031115858	0.057473165	0.435483871	0.284210526	0.343949045
MP-5	0.272677446	0.532942237	0.332266266	0.212301587	0.062708403	0.071947492	0.460674157	0.431578947	0.445652174
MP-6	0.275417923	0.525155172	0.334930768	0.235576923	0.05005479	0.070646604	0.460674157	0.431578947	0.445652174
PE-4	0.3125	0.458426751	0.348101783	0.260416667	0.064623503	0.076274657	0.426666667	0.336842105	0.376470588
PE-5	0.362637363	0.197470636	0.249615884	1	0.024821144	0.047916031	0.371428571	0.138297872	0.201550388
PE-10	0.125	0.04	0.060606061	0	0	0	0.125	0.04	0.060606061
PE-11	0.25	0.353333308	0.26379766	1	0.024821144	0.047916031	0.387096774	0.252631579	0.305732484
PE-12	0.272727273	0.20227405	0.202446955	0	0	0	0.363636364	0.126315789	0.1875
PE-17	0.310739437	0.455907411	0.34126211	0.375	0.032654658	0.058347044	0.450704225	0.336842105	0.385542169
PE-20	0.666666667	0.077798731	0.138465	0.6	0.026647911	0.05075333	0.666666667	0.063157895	0.115384615
PL-8	0.284313725	0.465964951	0.324931318	0.265714286	0.060820937	0.077391256	0.435897436	0.357894737	0.393063584
SA-2	0.111111111	0.030235344	0.045064386	0	0	0	0.2	0.032967033	0.056603774
SA-8	0.271390374	0.244420042	0.228559509	0.484848485	0.026549573	0.049424044	0.386363636	0.178947368	0.244604317
SA-11	0.260759494	0.420523897	0.294405545	0.265714286	0.060820937	0.077391256	0.379746835	0.315789474	0.344827586
SA-18	0.362637363	0.197470636	0.249615884	1	0.024821144	0.047916031	0.371428571	0.138297872	0.201550388
SA-19	0.33877551	0.193569672	0.235187539	1	0.024821144	0.047916031	0.4	0.147368421	0.215384615
SA-22	0.293040293	0.359569878	0.303558478	0.5	0.031115858	0.057473165	0.412698413	0.273684211	0.329113924
SC-2	0.282639885	0.249771287	0.235778112	0.484848485	0.026549573	0.049424044	0.414634146	0.178947368	0.25
SC-4	0.408024691	0.412246966	0.391714632	0.328125	0.098414344	0.142447996	0.45	0.284210526	0.348387097
SC-10	1	0.055555556	0.105263158	0	0	0	1	0.055555556	0.105263158
SC-11	0.390769231	0.184640309	0.236075572	0	0	0	0.52	0.136842105	0.216666667
SC-17	0.418181818	0.137622967	0.200143459	0	0	0	0.733333333	0.120879121	0.20754717
SC-18	0.336904762	0.482730619	0.371415151	0.272435897	0.046688603	0.07302959	0.486111111	0.368421053	0.419161677
SC-19	0.360449735	0.389375938	0.346371978	0.347222222	0.048832032	0.073133074	0.482142857	0.284210526	0.357615894
SC-20	0.408163265	0.410055112	0.37694726	0.333333333	0.032195143	0.054711706	0.612244898	0.315789474	0.416666667
SC-21	0.34841629	0.150806867	0.198162494	0	0	0	0.764705882	0.142857143	0.240740741
SC-23	0.58056266	0.257279048	0.336953755	0.5	0.077380952	0.133928571	0.739130435	0.182795699	0.293103448
SC-31	0.303209459	0.442686037	0.334308191	0.272435897	0.046688603	0.07302959	0.432432432	0.336842105	0.378698225
SC-32	0.425531915	0.351260987	0.364682437	0.347222222	0.048832032	0.073133074	0.510638298	0.252631579	0.338028169
SC-36	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
SC-38	0.414835165	0.176849821	0.240417285	0.333333333	0.030301083	0.053558542	0.464285714	0.136842105	0.211382114
SC-40	0.359844271	0.42488481	0.359624523	0.265714286	0.060820937	0.077391256	0.467741935	0.305263158	0.369426752
SC-41	1	0.090909091	0.166666667	0	0	0	1	0.090909091	0.166666667
SI-5	0.349462366	0.195358245	0.229843882	0.453125	0.025803692	0.047913604	0.387096774	0.126315789	0.19047619
SI-7	0.378303748	0.511210943	0.401110859	0.255462185	0.065201703	0.091119465	0.6	0.410526316	0.4875
SI-8	0.293137908	0.507790628	0.341353127	0.245192308	0.062540169	0.07573641	0.481012658	0.4	0.436781609
SI-10	1	0.016393443	0.032258065	0	0	0	1	0.016393443	0.032258065
SI-11	0.338171263	0.350010313	0.316330435	1	0.024821144	0.047916031	0.490566038	0.273684211	0.351351351
SI-13	0.263736264	0.210008956	0.207534401	1	0.05	0.095238095	0.358974359	0.147368421	0.208955224
SI-15	0.299242424	0.444849523	0.331405324	0.3125	0.037473316	0.061783454	0.458333333	0.347368421	0.395209581
SI-16	0.443223443	0.181758731	0.246703335	0	0	0	0.619047619	0.136842105	0.224137931
SI-17	0.1875	0.128005274	0.13358701	0	0	0	0.375	0.06741573	0.114285714
PM-12	0.229457731	0.518804758	0.295121835	0.245192308	0.062540169	0.07573641	0.398058252	0.431578947	0.414141414
PM-13	0.25	0.161443736	0.182585136	0	0	0	0.4	0.144578313	0.212389381
PM-15	0.666666667	0.116257275	0.193190016	1	0.05	0.095238095	0.666666667	0.065934066	0.12
PM-16	0.256571429	0.372566991	0.277164309	0.342857143	0.034827302	0.06044712	0.357142857	0.263157895	0.303030303

Comp. Avg Precision (U)	Comp. Avg Recall (U)	Comp. Avg F1 Score(U)	Comp. Avg Precision (AT)	Comp. Avg Recall (AT)	Comp. Avg F1 Score(AT)	Comp. Avg Precision (AT-U)	Comp. Avg Recall (AT-U)	Comp. Avg F1 Score (AT-U)
0.408171439	0.285676056	0.253482132	0.326695725	0.032062753	0.049472095	0.515273401	0.215489751	0.269018367
Standard Deviation	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**
0.240129482	0.164321644	0.102808921	0.327060086	0.026809872	0.038415536	0.212164947	0.12698659	0.126076819
Minimum	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**
0.111111111	0.016393443	0.032258065	0	0	0	0.125	0.016393443	0.032258065
Maximum	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**
1	0.542692998	0.401110859	1	0.103595809	0.142447996	1	0.431578947	0.4875

Appendix C: Experiment Results for Information System of Security Category 6.5

Control ID	Avg Precision (U)	Avg Recall (U)	Avg F1 Score (U)	Avg Precision (AT)	Avg Recall (AT)	Avg F1 Score (AT)	Avg Precision (AT-U)	Avg Recall (AT-U)	Avg F1 Score (AT-U)
AC-2	0.270327467	0.807089753	0.382368878	0.179195962	0.336319545	0.164075462	0.5	0.705263158	0.585152838
AC-3	0.252630534	0.826680611	0.366781377	0.16130719	0.395669574	0.165266652	0.47260274	0.726315789	0.572614108
AC-4	0.356756757	0.569688331	0.401862496	0.229209979	0.259419267	0.158711831	0.540540541	0.421052632	0.473372781
AC-11	1	0.043478261	0.083333333	0	0	0	1	0.043478261	0.083333333
AC-12	0.371428571	0.078792344	0.127401015	0.5	0.077380952	0.133928571	0.714285714	0.076923077	0.138888889
AC-23	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
AU-4	0.311061104	0.699590816	0.406779313	0.232664465	0.26215845	0.203364825	0.486238532	0.557894737	0.519607843
AU-12	0.313138307	0.702667875	0.409100483	0.231770833	0.278952773	0.208738632	0.486238532	0.557894737	0.519607843
CA-8	0.274074074	0.160049588	0.176075233	0	0	0	0.6	0.126760563	0.209302326
IA-6	0.333333333	0.026315789	0.048780488	0	0	0	0.333333333	0.026315789	0.048780488
IA-9	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
IA-11	1	0.071428571	0.133333333	0	0	0	1	0.071428571	0.133333333
MA-6	0.133333333	0.098924518	0.102040884	0	0	0	0.555555556	0.135135135	0.217391304
PE-12	0.306722689	0.292297837	0.270234436	0.5	0.020751635	0.03665512	0.404761905	0.178947368	0.248175182
PL-8	0.4	0.069295102	0.118055556	0	0	0	0.4	0.064516129	0.111111111
SA-22	1	0.03125	0.060606061	0	0	0	1	0.03125	0.060606061
SC-20	0.401558442	0.655313934	0.464533047	0.247732426	0.179201885	0.150289623	0.649350649	0.526315789	0.581395349
SC-21	0.288537549	0.233801048	0.242209121	1	0.083333333	0.153846154	0.696969697	0.242105263	0.359375
SC-32	0.25	0.052345679	0.081470103	0	0	0	0.75	0.051724138	0.096774194
SC-36	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
SC-39	0.281930334	0.661790214	0.367866915	0.210185185	0.204232842	0.132055728	0.490566038	0.547368421	0.517412935
SC-41	1	0.090909091	0.166666667	0	0	0	1	0.090909091	0.166666667
SI-5	1	0.033333333	0.064516129	0	0	0	1	0.033333333	0.064516129
SI-8	0.333333333	0.066666667	0.111111111	0	0	0	0.333333333	0.066666667	0.111111111
SI-10	1	0.016393443	0.032258065	0	0	0	1	0.016393443	0.032258065
SI-12	0.337438424	0.403367204	0.340575718	0.645	0.109534959	0.165108937	0.482758621	0.294736842	0.366013072
SI-15	1	0.020408163	0.04	0	0	0	1	0.020408163	0.04
SI-16	0.208333333	0.062295606	0.087137854	0	0	0	0.75	0.082191781	0.148148148
PM-12	0.14619883	0.112675585	0.112482795	0	0	0	0.473684211	0.113924051	0.183673469
PM-15	0.5	0.033333333	0.0625	0	0	0	0.5	0.033333333	0.0625
PM-16	0.151515152	0.086472008	0.100878169	0	0	0	0.272727273	0.055555556	0.092307692

Comp. Avg Precision (U)	Comp. Avg Recall (U)	Comp. Avg F1 Score(U)	Comp. Avg Precision (AT)	Comp. Avg Recall (AT)	Comp. Avg F1 Score(AT)	Comp. Avg Precision (AT-U)	Comp. Avg Recall (AT-U)	Comp. Avg F1 Score (AT-U)
0.523279083	0.229477341	0.17960823	0.133453743	0.071192104	0.053936824	0.673966022	0.19371886	0.224204059
Standard Deviation	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**
0.342334095	0.275776036	0.138846089	0.237204982	0.119103972	0.078401419	0.254102894	0.224611462	0.191368554
Minimum	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**
0.133333333	0.016393443	0.032258065	0	0	0	0.272727273	0.016393443	0.032258065
Maximum	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**
1	0.826680611	0.464533047	1	0.395669574	0.208738632	1	0.726315789	0.585152838

Appendix D: Experiment Results for Information System of Security Category 8.5

Control ID	Avg Precision (U)	Avg Recall (U)	Avg F1 Score (U)	Avg Precision (AT)	Avg Recall (AT)	Avg F1 Score (AT)	Avg Precision (AT-U)	Avg Recall (AT-U)	Avg F1 Score (AT-U)
AC-2	0.305737939	0.753801352	0.411688606	0.245322245	0.195533172	0.167606718	0.548672566	0.652631579	0.596153846
AC-3	0.257339015	0.764570197	0.364444034	0.200534759	0.264013683	0.150765236	0.484848485	0.673684211	0.563876652
AC-4	0.427083333	0.458886276	0.415467233	0.21031746	0.033544308	0.04949738	0.592592593	0.336842105	0.429530201
AC-11	1	0.043478261	0.083333333	0	0	0	1	0.043478261	0.083333333
AC-12	0.371428571	0.078792344	0.127401015	0.5	0.077380952	0.133928571	0.714285714	0.076923077	0.138888889
AC-23	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
AU-4	0.3125	0.103888322	0.150579186	0	0	0	0.4375	0.077777778	0.132075472
AU-9	0.315133038	0.564012341	0.378370911	0.304713805	0.13364326	0.163463094	0.465909091	0.431578947	0.448087432
AU-10	0.300862689	0.675565621	0.393981319	0.254794521	0.162000963	0.172277775	0.485981308	0.547368421	0.514851485
AU-12	0.31122449	0.647080111	0.395559451	0.245322245	0.195533172	0.167606718	0.49	0.515789474	0.502564103
CA-8	0.274074074	0.160049588	0.176075233	0	0	0	0.6	0.126760563	0.209302326
CA-9	0.279371585	0.740328944	0.382858982	0.245322245	0.195533172	0.167606718	0.508333333	0.642105263	0.56744186
CM-2	0.488122605	0.439557017	0.421457059	0.372340426	0.156553764	0.149442499	0.644444444	0.305263158	0.414285714
CM-3	0.54954955	0.367189076	0.405524219	0.372340426	0.156553764	0.149442499	0.648648649	0.252631579	0.363636364
CM-10	0.428509154	0.523512	0.444372371	0.288679245	0.099111448	0.132557207	0.596774194	0.389473684	0.47133758
CM-11	0.341337907	0.707172704	0.431566193	0.217954346	0.310053663	0.198543468	0.535353535	0.557894737	0.546391753
IA-6	0.333333333	0.026315789	0.048780488	0	0	0	0.333333333	0.026315789	0.048780488
IA-9	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
IA-11	1	0.071428571	0.133333333	0	0	0	1	0.071428571	0.133333333
IR-7	0.320634921	0.511384541	0.357131387	0.190789474	0.089297682	0.047081439	0.514285714	0.378947368	0.436363636
PE-11	0.1875	0.049297573	0.077773731	0	0	0	0.5	0.070175439	0.123076923
PE-12	0.291101056	0.273063837	0.255394017	0.5	0.020751635	0.03665512	0.435897436	0.178947368	0.253731343
PL-8	0.4	0.069295102	0.118055556	0	0	0	0.4	0.064516129	0.111111111
RA-3	0.357142857	0.08704142	0.134924719	0.75	0.015853799	0.030781027	0.357142857	0.055555556	0.096153846
SA-22	1	0.03125	0.060606061	0	0	0	1	0.03125	0.060606061
SC-20	0.246153846	0.116119268	0.14899211	1	0.083333333	0.153846154	0.769230769	0.12195122	0.210526316
SC-21	0.328571429	0.134063234	0.177483386	0	0	0	0.7	0.107692308	0.186666667
SC-32	0.25	0.052345679	0.081470103	0	0	0	0.75	0.051724138	0.096774194
SC-36	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
SC-39	0.30141844	0.590834097	0.373931005	0.295321637	0.090253978	0.121626116	0.522222222	0.494736842	0.508108108
SC-41	1	0.090909091	0.166666667	0	0	0	1	0.090909091	0.166666667
SI-5	1	0.033333333	0.064516129	0	0	0	1	0.033333333	0.064516129
SI-8	0.333333333	0.066666667	0.111111111	0	0	0	0.333333333	0.066666667	0.111111111
SI-10	1	0.016393443	0.032258065	0	0	0	1	0.016393443	0.032258065
SI-12	0.4	0.124551215	0.179855731	0	0	0	0.6	0.101694915	0.173913043
SI-15	1	0.020408163	0.04	0	0	0	1	0.020408163	0.04
SI-16	0.208333333	0.062295606	0.087137854	0	0	0	0.75	0.082191781	0.148148148
PM-12	0.14619883	0.112675585	0.112482795	0	0	0	0.473684211	0.113924051	0.183673469
PM-15	0.5	0.033333333	0.0625	0	0	0	0.5	0.033333333	0.0625
PM-16	0.151515152	0.086472008	0.100878169	0	0	0	0.272727273	0.055555556	0.092307692

Comp. Avg Precision (U)	Comp. Avg Recall (U)	Comp. Avg F1 Score(U)	Comp. Avg Precision (AT)	Comp. Avg Recall (AT)	Comp. Avg F1 Score(AT)	Comp. Avg Precision (AT-U)	Comp. Avg Recall (AT-U)	Comp. Avg F1 Score (AT-U)
0.492937762	0.244862614	0.202871453	0.154843821	0.056973644	0.054818193	0.649130027	0.200124919	0.238224498
Standard Deviation	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**
0.307620143	0.260817428	0.144607966	0.22971181	0.085410277	0.073067902	0.233540821	0.210396737	0.186870539
Minimum	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**
0.14619883	0.016393443	0.032258065	0	0	0	0.272727273	0.016393443	0.032258065
Maximum	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**
1	0.764570197	0.444372371	1	0.310053663	0.198543468	1	0.673684211	0.596153846

Appendix E: Experiment Results for Information System of Security Category 10

Control ID	Avg Precision (U)	Avg Recall (U)	Avg F1 Score (U)	Avg Precision (AT)	Avg Recall (AT)	Avg F1 Score (AT)	Avg Precision (AT-U)	Avg Recall (AT-U)	Avg F1 Score (AT-U)
AC-2	0.3121833	0.621228868	0.388476608	0.212962963	0.048112872	0.059255093	0.573033708	0.536842105	0.554347826
AC-3	0.266127583	0.639401054	0.352361436	0.2625	0.03452116	0.052058819	0.504761905	0.557894737	0.53
AC-4	0.431818182	0.118324702	0.180317112	0	0	0	0.727272727	0.08988764	0.16
AC-11	1	0.043478261	0.083333333	0	0	0	1	0.043478261	0.083333333
AC-12	0.325	0.078792344	0.124090193	0.5	0.077380952	0.133928571	0.625	0.076923077	0.136986301
AC-18	0.30248896	0.630225162	0.382117485	0.305813953	0.110843878	0.13999446	0.563829787	0.557894737	0.560846561
AC-19	0.429309708	0.561188599	0.456065107	0.305813953	0.110843878	0.13999446	0.682539683	0.452631579	0.544303797
AC-23	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
AT-2	0.273333333	0.221888843	0.228986532	0	0	0	0.48	0.146341463	0.224299065
AU-4	0.308333333	0.513810777	0.360975586	0.4	0.028384058	0.047068458	0.512820513	0.421052632	0.462427746
AU-9	0.304637971	0.492319357	0.354844105	0.434535104	0.116822052	0.160449643	0.456790123	0.389473684	0.420454545
AU-10	0.308777429	0.564787668	0.376137039	0.434535104	0.116822052	0.160449643	0.505747126	0.463157895	0.483516484
AU-12	0.307594937	0.51458853	0.361154918	0.4	0.028384058	0.047068458	0.506329114	0.421052632	0.459770115
CA-2	0.311895425	0.585734886	0.380084832	0.226315789	0.037225769	0.055058449	0.529411765	0.473684211	0.5
CA-3	0.308150183	0.607789586	0.383153359	0.193333333	0.098981424	0.054514459	0.527472527	0.505263158	0.516129032
CA-8	0.333333333	0.263445219	0.27068559	1	0.083333333	0.153846154	0.692307692	0.21686747	0.330275229
CA-9	0.315192744	0.669966663	0.402617055	0.293683347	0.170418654	0.157118923	0.551020408	0.568421053	0.559585492
CM-2	0.471111111	0.446015451	0.419779945	0.251282051	0.019833449	0.03607153	0.666666667	0.315789474	0.428571429
CM-3	0.530434783	0.350135893	0.387914657	0.251282051	0.019833449	0.03607153	0.657142857	0.242105263	0.353846154
CM-10	0.450340136	0.43288737	0.414530725	0.293333333	0.033132157	0.050948388	0.612244898	0.315789474	0.416666667
CM-11	0.348636364	0.659241485	0.425424789	0.251984127	0.169732711	0.141738579	0.568181818	0.526315789	0.546448087
CP-3	0.197959184	0.281232127	0.211555936	1	0.083333333	0.153846154	0.408163265	0.210526316	0.277777778
CP-9	0.338709677	0.467150613	0.36224974	0.35	0.065596754	0.101647371	0.532258065	0.347368421	0.420382166
CP-10	0.271517997	0.448415365	0.31272893	1	0.083333333	0.153846154	0.507042254	0.378947368	0.43373494
IA-6	0.333333333	0.026315789	0.048780488	0	0	0	0.333333333	0.026315789	0.048780488
IA-7	0.42962963	0.421903623	0.392997052	0.35	0.065596754	0.101647371	0.666666667	0.315789474	0.428571429
IA-9	1	0.035714286	0.068965517	0	0	0	1	0.035714286	0.068965517
IA-11	0.5	0.057453416	0.1025	0	0	0	1	0.08	0.148148148
IR-5	0.31162465	0.705318599	0.406779895	0.131623932	0.120792517	0.072023597	0.549019608	0.589473684	0.568527919
IR-6	0.287749288	0.668999378	0.376782568	0.184210526	0.118711613	0.088555869	0.519230769	0.568421053	0.542713568
IR-7	0.33353151	0.424726936	0.337630974	0.2625	0.03452116	0.052058819	0.5	0.305263158	0.379084967
MA-2	0.309185606	0.603647305	0.3838734	0.26984127	0.034589379	0.054180018	0.545454545	0.505263158	0.524590164
MA-3	0.350246305	0.445249109	0.371325742	0.333333333	0.041915555	0.066601935	0.603448276	0.368421053	0.45751634
MA-5	0.362804878	0.542737613	0.407824645	0.2625	0.03452116	0.052058819	0.569444444	0.431578947	0.491017964
PE-9	0.142857143	0.031527094	0.051587302	1	0.083333333	0.153846154	0.285714286	0.047619048	0.081632653
PE-11	0.1875	0.049297573	0.077773731	0	0	0	0.5	0.070175439	0.123076923
PE-12	0.228840125	0.181456355	0.178291755	1	0.083333333	0.153846154	0.379310345	0.115789474	0.177419355
RA-3	0.2	0.034482759	0.058823529	0	0	0	0.2	0.034482759	0.058823529
SA-2	0.111111111	0.037037037	0.055555556	0	0	0	0.111111111	0.037037037	0.055555556
SA-22	1	0.03125	0.060606061	0	0	0	1	0.03125	0.060606061
SC-20	0.292517007	0.212363611	0.227888069	0.5	0.077380952	0.133928571	0.666666667	0.157303371	0.254545455
SC-21	0.363636364	0.13797292	0.184113698	0	0	0	0.727272727	0.087912088	0.156862745
SC-32	0.460784314	0.209583126	0.264221961	0	0	0	0.705882353	0.127659574	0.216216216
SC-36	0.275	0.074659734	0.117173939	0	0	0	0.625	0.087719298	0.153846154
SC-39	0.330127104	0.531048608	0.376652751	0.424242424	0.02377327	0.044245036	0.577464789	0.431578947	0.493975904
SC-41	1	0.090909091	0.166666667	0	0	0	1	0.090909091	0.166666667
SI-5	1	0.033333333	0.064516129	0	0	0	1	0.033333333	0.064516129
SI-8	0.333333333	0.066666667	0.111111111	0	0	0	0.333333333	0.066666667	0.111111111
SI-10	1	0.016393443	0.032258065	0	0	0	1	0.016393443	0.032258065
SI-12	0.403409091	0.215905543	0.263936663	0.5	0.077380952	0.133928571	0.727272727	0.170212766	0.275862069
SI-13	0.269230769	0.12924215	0.164240831	0	0	0	0.615384615	0.117647059	0.197530864
SI-15	1	0.020408163	0.04	0	0	0	1	0.020408163	0.04
SI-16	0.411764706	0.182353037	0.228989012	0	0	0	0.764705882	0.138297872	0.234234234
SI-17	0.111111111	0.031247339	0.048408057	0	0	0	0.333333333	0.052631579	0.090909091
PM-12	0.285294118	0.250789275	0.244416777	0.555555556	0.10515873	0.171218897	0.588235294	0.210526316	0.310077519
PM-15	0.5	0.033333333	0.0625	0	0	0	0.5	0.033333333	0.0625
PM-16	0.277777778	0.142107972	0.172097397	0.5	0.077380952	0.133928571	0.444444444	0.086956522	0.145454545

Comp. Avg Precision (U)	Comp. Avg Recall (U)	Comp. Avg F1 Score(U)	Comp. Avg Precision (AT)	Comp. Avg Recall (AT)	Comp. Avg F1 Score(AT)	Comp. Avg Precision (AT-U)	Comp. Avg Recall (AT-U)	Comp. Avg F1 Score (AT-U)
0.418408508	0.296898199	0.242769121	0.256862845	0.044127351	0.06047445	0.609855552	0.241850991	0.292881853
Standard Deviation	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**	**Standard Deviation**
0.252363797	0.235869919	0.139414784	0.29263159	0.048150774	0.062207355	0.21203257	0.192843194	0.185502195
Minimum	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**	**Minimum**
0.111111111	0.016393443	0.032258065	0	0	0	0.111111111	0.016393443	0.032258065
Maximum	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**	**Maximum**
1	0.705318599	0.456065107	1	0.170418654	0.171218897	1	0.589473684	0.568527919

Appendix F: NIST 800-53 – MITRE ATT&CK TTP Mappings Sample (Courtesy of MITRE)

	A	B	C	D	E	F
1	**Control ID**	**Control Name**	**Mapping Type**	**Technique ID**	**Technique Name**	
47	AC-16	Security Attributes	mitigates	T1550.001	Application Access Token	
113	AC-17	Remote Access	mitigates	T1550.001	Application Access Token	
172	AC-19	Access Control For Mobile Devices	mitigates	T1550.001	Application Access Token	
409	AC-20	Use Of External Information Systems	mitigates	T1550.001	Application Access Token	
1488	CA-8	Penetration Testing	mitigates	T1550.001	Application Access Token	
1519	CM-10	Software Usage Restrictions	mitigates	T1550.001	Application Access Token	
1554	CM-11	User-Installed Software	mitigates	T1550.001	Application Access Token	
1736	CM-2	Baseline Configuration	mitigates	T1550.001	Application Access Token	
2203	CM-6	Configuration Settings	mitigates	T1550.001	Application Access Token	
2759	IA-2	Identification And Authentication (Organizational Users)	mitigates	T1550.001	Application Access Token	
2844	IA-4	Identifier Management	mitigates	T1550.001	Application Access Token	
3393	SC-28	Protection Of Information At Rest	mitigates	T1550.001	Application Access Token	
3698	SC-8	Transmission Confidentiality And Integrity	mitigates	T1550.001	Application Access Token	
3815	SI-12	Information Handling And Retention	mitigates	T1550.001	Application Access Token	
4387	SI-4	Information System Monitoring	mitigates	T1550.001	Application Access Token	
4615	SI-7	Software, Firmware, And Information Integrity	mitigates	T1550.001	Application Access Token	
4694						

Appendix G: MITRE TTP – CVE Mappings Sample (Courtesy of MITRE)

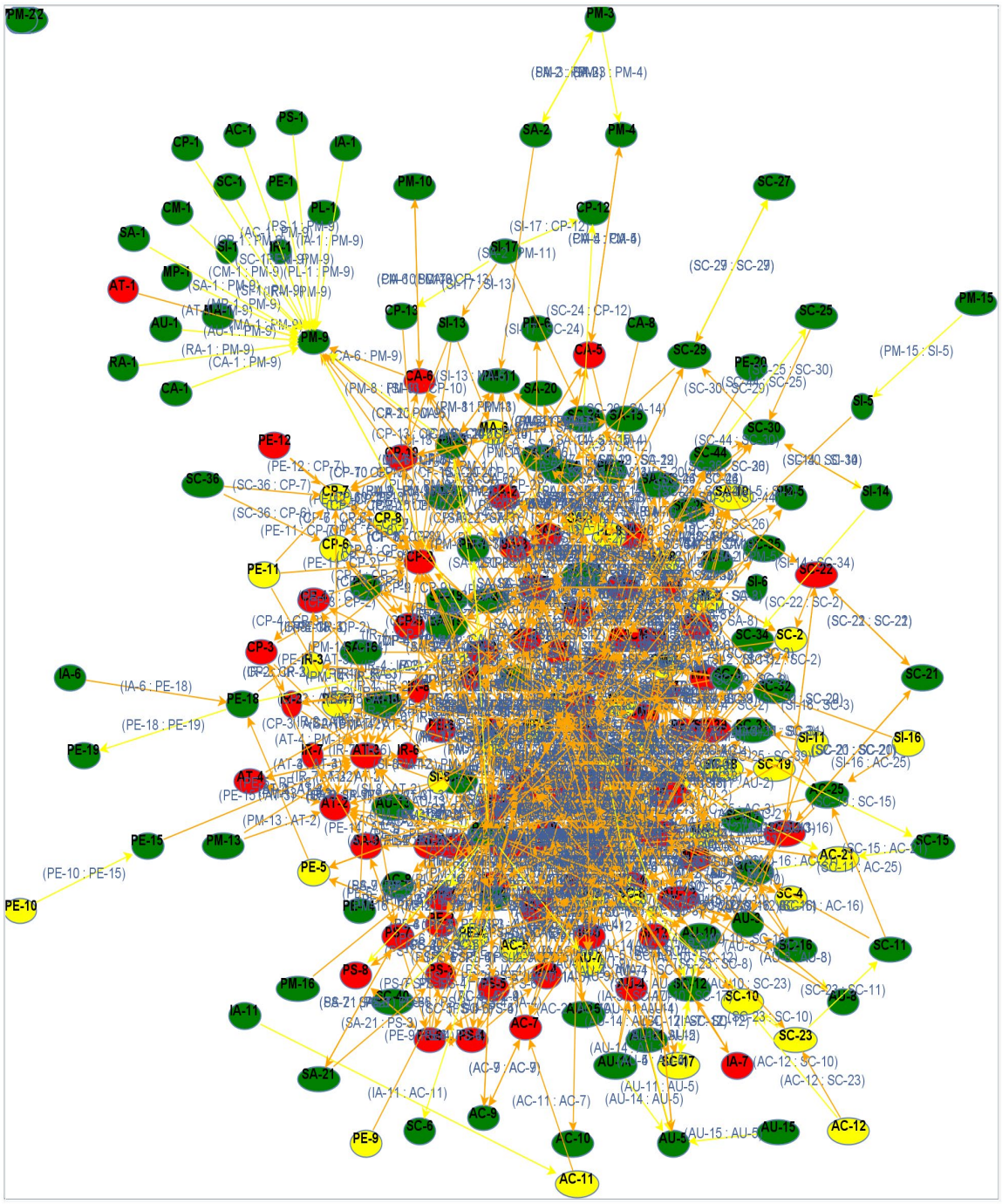
	A	B	C	D	E	F
1	**CVE ID**	**Primary Impact**	**Secondary Impact**	**Exploitation Technique**	**Uncategorized**	**Phase**
2	CVE-2019-15243	T1059		T1190; T1078		Phase 2
3	CVE-2019-15976	T1068	T1059	T1190		Phase 2
4	CVE-2019-15956	T1499; T1098		T1190; T1078		Phase 2
5	CVE-2019-15958	T1059		T1190		Phase 2
6	CVE-2019-12660	T1574	T1562	T1078		Phase 2
7	CVE-2019-1753	T1068	T1059	T1190; T1078		Phase 2
8	CVE-2019-1860	T1557	T1005			Phase 2
9	CVE-2019-1831	T1036	T1566			Phase 2
10	CVE-2019-1942	T1059	T1005; T1565.001	T1133; T1078		Phase 2
11	CVE-2019-15972	T1059	T1005; T1565.001	T1133; T1078		Phase 2
12	CVE-2019-16009	T1608		T1204.001		Phase 2
13	CVE-2019-1879	T1068	T1059	T1078		Phase 2
14	CVE-2019-1863	T1068	T1565.001	T1190; T1078		Phase 2
15	CVE-2020-3403	T1068	T1059	T1078		Phase 2
16	CVE-2019-1941	T1059.007	T1557	T1204.001		Phase 2
17	CVE-2020-3292	T1499.004	T1059	T1190; T1078		Phase 2
18	CVE-2018-15397	T1529		T1190		Phase 2
19	CVE-2020-3253	T1059		T1078		Phase 2

Appendix H: Purple Eye Security Intelligence Report Upon Control "AC-2" Failure (Sample)

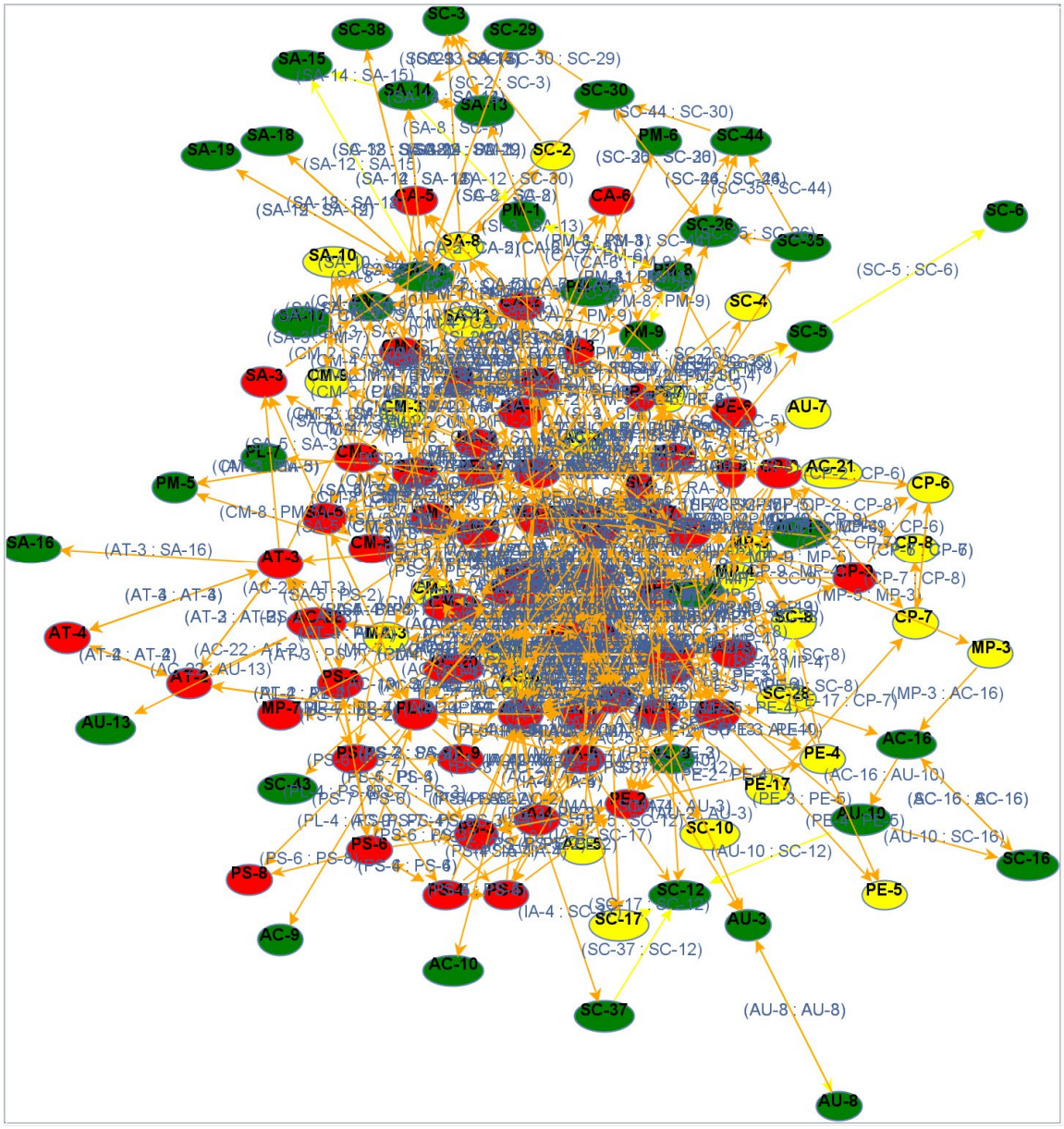
Basic Information	
Root Failed Control	AC-2
Number of Controls Affected	132
Affected Controls List	AC-2, AC-3, AC-4, AC-5, AC-6, AC-10, AC-17, AC-19, AC-20, AU-9, IA-2, IA-4, IA-5, IA-8, CM-5, CM-6, CM-11, MA-3, MA-4, MA-5, PL-4, SC-13, AC-21, SA-8, SC-2, SC-5, AC-16, AU-10, SC-16, AC-18, CA-3, CA-7, CM-8, IA-3, PE-17, SC-10, SI-4, CA-9, CM-2, MP-2, MP-4, MP-5, SC-7, SC-43, SI-3, SA-9, AC-22, AT-2, AT-3, AU-13, PE-2, PE-3, PE-6, CM-3, CM-7, AU-2, AU-3, MA-2, MP-6, PL-2, SC-17, PE-4, RA-3, AU-6, PE-5, PS-3, PM-7, SA-17, SC-3, SC-6, AU-16, CA-2, CA-5, CA-6, CM-4, PM-6, PM-9, RA-5, SA-11, SA-12, SI-2, PM-5, AC-14, SC-8, CP-7, AC-8, AC-9, MP-7, PS-6, PS-8, SA-5, AU-7, IR-4, SC-26, SC-35, SI-7, CM-9, SA-10, CP-6, MP-3, SC-28, CP-8, SA-4, SA-13, SC-44, IR-8, SC-37, PS-5, SC-12, PS-4, PS-2, CP-9, RA-2, SA-14, SA-15, SA-18, SA-19, SC-29, SC-30, SC-38, AU-8, PE-16, SC-4, CP-2, PL-7, PM-1, PM-8, PM-11, AT-4, PS-7, SA-3, SA-16

Age Decay	34 Days
Risk Assessment	
Age Factor	**3.41**
Vulnerability Factor	4.0
Impact Factor	4.0
EICRS Risk Score	54.56
System Risk	High
Attack Surface Analysis	
Total number of MITRE TTPs in DB	**371**
Number of exposed MITRE TTPs	123
Exposure to Attack Techniques	SID-History Injection, Pass the Ticket, Golden Ticket, GUI Input Capture, Two-Factor Authentication Interception, Man in the Browser, Steal Web Session Cookie, Plist Modification, Re-opened Applications, Windows Management Instrumentation, Systemd Timers, Network Device CLI, Cloud Account, Access Token Manipulation, Create Process with Token, Make and Impersonate Token, Token Impersonation/Theft, Service Stop, Cloud (PARTIAL LIST)

Appendix I: Enlarged copy of Figure 18



Appendix J: Enlarged copy of Figure 19



Curriculum Vitae

Name:	Syed Waqas Hamdani
Post-secondary Education and Degrees:	University of Waterloo Waterloo, Ontario, Canada 2016 B.A Liberal Studies. University of Western Ontario London, Ontario, Canada 2017-2021 BSc. (Hons) Specialization Computer Science
Honours and Awards:	Graduate Fellowship (Master) University of Western Ontario 2021 – 2022 President's Scholarship University of Waterloo 2007 International Student Excellence Scholarship University of Waterloo 2007
Related Work Experience	Graduate Teaching Assistant The University of Western Ontario 2021 – 2022

Publications:

Hamdani, S. W., Brealey, C., Kontogiannis, K., & Giammaria, A. (2023). Evaluating the Impact of NIST 800.53 Security Control Violations. *CASCON '22: Proceedings of the 32nd Annual International Conference on Computer Science and Software Engineering*, 187–192. https://doi.org/10.5555/3566055.3566078