SECURITY MANAGEMENT OF SMART HOME INTERNET-OF-THINGS: A FRAMEWORK, FINITE-STATE ATTACK MODELING, AND WORST ATTACK VULNERABILITY ANALYSIS

A Dissertation
IN
Telecommunications and Computer Networking
and
Computer Science

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
FATHIMA JAMES

B.E., Anna University, India, 2005
M.S., University of Tennessee Chattanooga, USA, 2013

Kansas City, Missouri
2023

SECURITY MANAGEMENT OF SMART HOME INTERNET-OF-THINGS: A FRAMEWORK, FINITE-STATE ATTACK MODELING, AND WORST ATTACK VULNERABILITY ANALYSIS

Fathima James, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2023

ABSTRACT

Smart Home Internet of Things (SHIoT) provides a rich compendium of innovative, ubiquitous, and interactive services to users using a variety of smart sensors, devices and applications. However, owing to the strongly internet-facing, dynamic, and heterogeneous and low capability nature of these devices, and existence of vulnerabilities in them, in their controlling applications and their configurations, there are security threats in SHIoT that affect the safe and secure functioning of these systems. Because of the complexity of the SHIoT system, it is difficult to effectively determine the security posture. We consider attack vulnerabilities and how to identify those vulnerabilities to prevent attacks from spreading for Smart Home Internet of Things (SHIoT). We then address the problem of assessing the worst vulnerability, that is the one that has the potential to cause maximum damage, in the SHIoT.

The resource-constrained nature of many of the IoT devices present in a smart home environment does not permit the implementation of standard security solutions. Therefore, the special purpose SHIoT devices and their services with rich human interactions are more vulnerable to cyberattacks. To understand the vulnerability of the threat and attacker motive in SHIoT environment, we introduce a graph-based framework for attacks in IoT security. In this framework, an attack graph is first represented through Finite-state automata for three different SHIoT based cyberattacks - a confidentiality attack, an authentication attack and an access control attack. we then present vulnerability analysis for different SHIoT based attack graphs, followed by a fortification process to enhance the overall system security.

For the problem on the worst path vulnerability in the attack graph for SHIoT, we needed to address the probabilistic nature of arcs of the attack graph. In particular, the attack path has non-additive property. We showed how the problem can be transformed to an equivalent problem with additive property so that a short path based approach can be applied to determine the worst path vulnerability. We also present an approach to iteratively fortify the environment to reduce impact from vulnerability. Finally, we apply Common Vulnerability Scoring System (CVSS) to determine attack probabilities on arcs in the attack graph and present an analysis on representative attack graphs.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled "Security Management of Smart Home Internet-Of-Things: A Framework, Finite-State Attack Modeling, And Worst Attack Vulnerability Analysis," presented by Fathima James, candidate for the Doctor of Philosophy degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Deep Medhi, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Yugyung Lee, Ph.D.
Department of Computer Science & Electrical Engineering

Cory Beard, Ph.D.
Department of Computer Science & Electrical Engineering

Farid Nait-Abesselam, Ph.D.
Department of Computer Science & Electrical Engineering

Indrajit Ray, Ph.D.
Department of Computer Science
Colorado State University

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to sincerely thank my professor and research advisor, **Dr. Deep Medhi** whose insight and knowledge into the subject matter steered me through this research, for his enthusiasm towards this research work, and also for his support, encouragement, and patience. Without him, I would not have done this research or had the opportunity to write a dissertation, and for that, I am forever grateful to my professor.

Next, I would like to thank my parents, **James Rayappan** and **Rita James**, for their support during my prolonged studies. They were the most loving parents a person could imagine. I want to thank my brothers, **Albert James** and **Robert James**, for supporting me and taking care of my mother after the sad loss of my father.

I want to give a special thanks to my loving daughter, **Giselle Marie**, for being such a good girl and for being endlessly patient with me during the many days, nights, weekends, and months for the last two years. I am very sorry for being even grumpier than normal, especially for the last couple of months. I love you more than you imagine.

I would like to thank my committee members, **Dr. Yugyang Lee, Dr. Cory Beard, Dr. Faird Nait-Abesselam** and **Dr. Indrajit Ray** for serving as my committee members and for their encouragement, support, helpful input, and valuable feedback throughout this long research journey. I want to give a special thanks and appreciation to **Dr. Indrajit Ray**, Colorado State University, for accepting my request to serve as my committee member and for providing his brilliant comments and suggestions to help me

write up my research papers.

I would like to thank my colleague, **Dr. Marvin J. Loiseau** for his tremendous support and encouragement and for always ensuring that I am getting proper help and support in the working environment. A special thanks to my work-place college president, **Dr. Aisha Francis**, for her endless encouragement and for providing me with ample time to complete my dissertation and defense than I had promised while I was interviewing. I would also like to thank my labmates and friends who have supported me throughout this process, especially my friends, **Bhanu Prakash Panchakarla** and **Dr. Rohit Abhishek** for their timely help and support whenever I needed it.

Finally, I thank **God**, my good father, for letting me through all the difficulties. I have experienced your guidance day by day and have seen you through the people I mentioned above, especially through my Professor, **Dr. Deep Medhi**. I am nothing, and you are the one who let me finish my doctoral degree. I will keep on trusting you for my future. **Thank you, Lord**.

CHAPTER 1

INTRODUCTION

1.1 Background Context

IoT security management is a challenging issue. The devices may be compromised for a variety of reasons. A typical IoT device comes with no security features beyond a default password. This security oversight allows remote attackers to control an entire system by exploiting unpatched vulnerabilities. The more ways IoT devices can connect, the more opportunities there are for cybercriminals to exploit due to lack of security software, lack of cybersecurity awareness and large attack surface [2]. In this context, it is useful to cast an IoT environment as a complex network because the theoretical framework and computational tools for complex networks derive new approaches in analyzing IoT architectures, networks, and services and also pursues new ways to represent, characterize, and analyze the connections and interactions between a huge volume of sensors, actuators, and processors. Most importantly, it develops explicit approaches to analyze the collective behaviors in IoT architecture [3]. However, the security and survivability are dependent not only on the security of the underlying infrastructure but also on the ability to ensure that unforeseen circumstances, such as, changes to the mission requirements, zero-day attacks, and unpredictable human errors in interactions with the mission, are adequately addressed and managed. Additionally, in the worst case, there

needs to be provisions for the graceful degradation of mission services by avoiding cascading catastrophic failures, when all defensive measures have failed.

To ensure that such a complex network continues to operate in a survivable manner, it is important to be proactive in understanding and reasoning about evolving threats to the service availability, their potential effects on the mission survivability, and identify ways to best defend against these threats, instead of being reactive. A graph-theoretic analysis of networks has the potential to help with such proactive analysis. Instead of covering the entire spectrum of all types of IoT devices and environments, we illustrate here the problem for Smart Home IoT (SHIoT). An attack on an SHIoT system can take place either by initiating an attack from within the smart environment (that is, an insider or local network attack) or by initiating the attack from an external source (i.e., an outsider or public network attack) [4]. SHIoT devices are more vulnerable to cyber-attacks because they are special purpose internet-connected devices and run tiny operating systems such as INTEGRITY, Contiki, FreeRTOS, and VxWorks, whose security solutions are not entirely robust and once deployed, may not be easily upgradable to ensure security capability against evolving cyberattacks [5].

On the other hand, Smart Home IoT (SHIoT) devices enable increased collaboration among distributed smart objects through diverse communication technologies and applications. This, in turn, allows smart homes to interact and leverage diverse service providers, such as utility suppliers, infrastructure providers and third-party software or hardware vendors [6], to provide a rich and novel living experience to their occupants.

Unfortunately, such rich functionality comes with a security and privacy cost. The security of SHIoT systems is a serious issue due to the increasing numbers of services and users in IoT networks. In SHIoT-based smart environments without robust security systems, applications and services will be at risk. Confidentiality, integrity, and availability are the most important security aspects of applications and services in SHIoT-based smart environments and that span all layers of the IoT architecture [7].

Security vulnerabilities in SHIoT can be exploited to create large, distributed bots that can then be leveraged to launch large scale attacks. Because of the large number of IoT devices involved and their diversity, the potential attack surface of a smart home is significant and complex. Moreover, the data exchanged between these IoT devices, the supporting applications and the service providers are often sensitive in nature and, if leaked, can potentially cause harm to the end user. Therefore, when building an SHIoT system, it is important that the end user have a comprehensive view of how the network of devices (including the corresponding applications) can be attacked, how easy or difficult it is to launch those attacks (under some metrics), what the consequences of those attacks are and how can those attacks be defeated.

Smart home technologies export large attack surfaces. An attack on the system can take place either by the attacker initiating an attack from within the smart environment (that is, an insider or local network attack) or by initiating the attack from an external source i.e., outsider or public network attack [4]. Thus, to generate a parameterized attack procedures and functions, there is a need for an attack model which will predict all possible ways an attacker can breach a system and potentially assign chances to each path

according to some metric (e.g., time-to compromise via the local/public network) [8].

In this thesis work, we present a formalism for our attack model for SHIoT-based cyberattacks and show through several scenarios how the model enables one to obtain a better understanding of the security posture of the system. We then introduce a graph-based representation and analysis for attack modeling. An advantage of our approach is that depending on the types of cybersecurity-based attacks (such as confidentiality-based or authentication-based) and for the type of network environment, we can generate appropriate FSA-based attack graphs. We then address the problem of assessing vulnerability from an attack source to a compromised state by considering the attack graph of an SHIoT system. Towards this, we start with a broader framework for the graph-based approach for attack graphs of an SHIoT system. For this, we consider vulnerabilities of an arc in an attack graph. However, due to probabilistic values, the path vulnerabilities do not have additive cost properties. That's where we use the shortest path method in order to handle this problem. Furthermore, we address the problem of fortifying the systemic view of the SHIoT from vulnerabilities from a systems management perspective. For this, we build on the graph-theoretic analysis to tackle this problem through an iterative process. In particular, we identify the weakest arcs in the attack graph that can then help systems administrators to take actions to reduce vulnerability in the attack graph, and to then iteratively fortify the SHIoT system. For our study, we use probabilistic values based on Common Vulnerability Scoring System (CVSS) [1] for vulnerabilities to how this fortification can be assessed on representative graphs.

1.2 Problem and Challenges

Smart home attack surface vulnerabilities should identify with cybersecurity challenges in the different IoT layers in order to stop invading the privacy of smart home inhabitants, stealing personal sensitive information and prevent by building a botnet network through vulnerable smart home devices.

Problem 1: How effectively identify smart home attack surface vulnerabilities with different cyber security challenges?

Whether its a smart home or an organization, a malicious act of an attacker is not easy to predict, detect and analysis. Thus, to generate a parameterized attack procedures and functions, there is a need for an attack model which will predict all possible ways an attacker can breach a system.

Problem 2: How efficiently an attack model analyze the smart home security threats and attack paths?

Smart home technologies export large attack surfaces particularly legacy components that use old versions of software which can not be regularly patched and updated pose a particularly challenging problem.

Problem 3: How does the graph-based FSA attack model analysis help identify which vulnerabilities to patch?

Fortifiaction process enables the SHIoT system security by improving security functionalities through the use of multiple, diverse control systems that covers the technical details of system architecture and functionality. Some of fortification security procedures are quite simple while others are quite complex. In either case, these fortifications are ideally

implemented at the inception of a new system, and at every point of system alteration or expansion.

Problem 4: Why does the fortification process rely on iterations to secure the SHIoT based network system?

With each iteration of the fortification process, the SHToT system is preferably altered or updated with a resource in order to reduce the vulnerability and improve the SHIoT system′s security.

1.3 Contributions

We made the following contributions in this thesis work:

- We first present a formalism of finite state automata-based attack model (FSAA) in order to understand, and explore smart home-based security threats.
- We then develop a framework for modeling attacks in SHIoT based on finite state automata (FSA), which has a graph-based representation.
- We determine the attack probabilities on arcs in the attack graph using the Common Vulnerability Scoring System (CVSS) and present analysis on representative attack graphs.
- We then show an approach to determining the worst vulnerability on an arc in the attack graph using the shortest path-based algorithm due to the non-additive nature of an attack paths attack probability.

- We finally illustrate an approach to iteratively fortify the SHIoT environment to reduce impact from vulnerability with 9-node, 15-node, 26-node, 50-node and 60-node attack graphs using the what-if analysis.

1.4 Publications

The publications for this thesis are listed with a brief contribution summary. The papers in this thesis were presented at conferences and published in the corresponding conference proceedings.

[P1]: IoT Cybersecurity based Smart Home Intrusion Prevention System

This paper presents the intrusion prevention system methodology for three cyber security aspects: confidentiality, authentication, and access control, in order to detect the most critical attacks on smart home IoT end devices. It also shows, with different case studies, how to protect and prevent the affected system from future attacks [9].

[P2]: A Risk Management Framework and A Generalized Attack Automata for IoT based Smart Home Environment

This paper presents a finite state automata-based attack model that supports the modeling of SHIoT based single state and multistate complex attacks, and the proposed risk management framework is enforcing system security, enhancing user privacy, and helping to further realize the potential risks in IoT based smart home environments [10].

[P3]: Situational Awareness for Smart Home IoT Security via Finite State Automata Based Attack Modeling

This paper presents the power of the Finite State Attack Automata (FSAA) model to capture and represent a substantial amount of information needed for situational awareness in SHIoT in order to understand the vulnerability of the threat and attacker motive [11].

[P4]: Worst Attack Vulnerability and Fortification for IoT Security Management: An approach and An Illustration for Smart Home IoT

This paper presents a graph-based framework in order to represent the different attack graphs through the finite state automata to assess and determine the worst vulnerability and illustrates an approach to iteratively fortify the attack environment to reduce the impact of vulnerability and enhance the overall system security.

This paper is currently accepted for the NOMS 2023 workshop conference.

Other Publications:

[P5]: Demodulation of faded wireless signals using deep convolutional neural networks

This paper demonstrates exceptional performance of approximately 10.0 dB learning-based gain using the Deep Convolutional Neural Network (DCNN) for demodulation of a Rayleigh-faded wireless data signal with a simulation of FSK demodulation over an AWGN Rayleigh fading channel with average signal to noise ratios (SNR) from 10 dB to 20 dB [12].

1.5 Structure of the Thesis

The remainder of this thesis is structured as follows:

- Chapter 2 describes the related work for this thesis. This includes thread models, frameworks and methodologies from the areas of Smart Home IoT, FSA based attack model, Risk and Vulneranility analysis.

- Chapter 3 describes the background knowledge for this thesis. This includes security and desired security goals of IoT based Smart Home, attack surface and attack entities.

- Chapter 4 discusses the overview of the FSA-based SHIoT attack model. It explains finite state attack automata state tuples along with a simplified example of an FSAA related to the vulnerabilities of a SHIoT system.

- Chapter 5 explains the more practical contributions of this thesis by showing and discussing the implementation of the attack modeling using finite state automata, which describes the different types of cyber security aspects and their attack models.

- In Chapter 6, an analysis of the Fortification process and Vulnerability. The analysis is done in two parts: Determining arc vulnerability and Fortification analysis. Additionally, we presented a set of representative attack graphs in order to perform the above-mentioned analyses.

- The conclusion and future work in Chapter 7 elaborate on the open issues and future work.

CHAPTER 2

BACKGROUND

2.1 Security Goals of SHIoT

Smart home appliances are intended to be interconnected to home networks. The nature of the interconnected smart home internet resources can be attacked anytime from any location in the world, and this makes smart home security a main issue. Regardless of the threats and vulnerabilities, the more secured smart home IoT devices sometimes get compromised by the poor users expertise. As per [10], and [11], the IoT based smart home revolves around five most significant security goals:

- Authentication enables to keep the network secure by allowing only authenticated users to access its protected resources.
- Authorization ensures that every user access right is defined for the purpose of resource utilization.
- Confidentiality deals with keeping data secure, so that only authorized users can access the private data.
- Integration refers to protecting data from being modified by unauthorized access.
- Availability denotes that only proper authorized users can access data, communications infrastructure, services and computing resources, and making sure that those authorized users are not prohibited from such access.

Based on the above-mentioned security goals, the IoT based smart home threats and vulnerabilities have been analyzed, assessed, and mitigated in the following sections.

2.2 Desired Security Goals of IoT based Smart Home

Attacks in SHIoT can be launched remotely either by direct access to networked control interface or downloading malware to devices. Moreover, even the more secured SHIoT devices sometimes get compromised because of poor user expertise or judgement [6]. Technology on its own is not a sufficient safeguard against this; the human component is one of the most vulnerable elements within SHIoT security. It can be influenced or manipulated to divulge sensitive information that allows unauthorized individuals to gain access to protected systems. Nonetheless, the most common causes of cyber-related smart home attacks are inadequate authentication procedures, limited software updating/patching, poor product design, nonsecure communications protocols, improper implementation or device/application use [13]. For this work, based on our study of the literature [6, 9, 14], we limit IoT based smart home security to the following three significant security goals: authentication of devices/users, authorization of the same, confidentiality of data exchanged between IoT applications and IoT services and access control of the IoT devices and services. In Figure 1, we map the three desired security goals to potential attacks at three different layers of the SHIoT stack.

2.3 Attack Surface

The main purpose of attack surface is to understand, explore and validate security threats in the cyber world [11] and, it is required to understand the motive of the attacker that the attack source (local or public network), how they deploy attacks and what information could be targeted. Figure 1 describes the attack surface infrastructure along with cybersecurity challenges in the different IoT layers. Most of the smart home IoT devices face access control-based cybersecurity problem in the local home network area under the perception layer. Authentication related risks arise between smart home gateway and public network/internet in the network layer. The third class of cybersecurity challenge concerns the confidentiality between IoT services and applications in the application layer. Confidentiality problems occur when the attacker eavesdrops on the private data in the smart home IoT system.

Smart home technologies have large attack surfaces that have several vulnerabilities. Whether its a local or pubic network attack, attackers will use the smart home resources such as methods, channels, devices and data to initiate attacks [5]. Thus, our proposed approach analyzes these two sides of the IoT network attacks, namely local (insiders) and public networks (outsiders). Local networks contain IP, non-IP networks, end devices, gateways and controllers. Public networks contain user controllers, applications, internet/cloud and IoT services [1]. Table 1 shows the IoT smart home environment attack surface including possibilities of the attacks in both sides as well.

2.4 Attack Entities

Since the smart home based cyber-attacks have unpredictable behavior in nature, we cannot directly make a formal description of attack behavior, and therefore we only give a description of it from the perspective of the attack process. The process of the attack is formed by triggering from the corresponding attack entities, which can cause the change of system state, each of which is caused by a class of corresponding attack entities, which contains a variety of attack behaviors. The below mentioned attack entities can be combined as an attack process.

- Reconnaissance entities: also called Intelligence gathering, the first step of network attack, by which to realize the basic outline of the target system.
- Scanning entities: Mainly used for collecting more detailed information for the target environment and systems, such as the port number of the target host, IP address and other information.
- Access and escalation entities: Through cracking passwords and modifying permissions, to obtain the target system and further improve the access permissions.
- Exfiltration entities: By means of encryption or using the communication protocol of the target system to steal data in the target system for interest.
- Assault entities: To attack on the target system, and destroy its confidentiality, integrity and availability.

- Sustainment entities: To ensure that the next attack can smoothly access the target system, sustainment entities added access permission and a back door.

- Obfuscation entities: To attack process with the hidden, confused, traps and others, so that the target system administrator could not track, identify attack source and attack purpose. It may exist in the each stage of the process, or before the attack, or after the attack, so it supports the whole attack process.

Comprehensive analysis on the finite automaton theory and the relationship between the attack entities and system states: the target system can be divided into different states, which are limited. System states can transfer under the action of the attack entities, and the transferred entities are limited, with the system showing the state behaviors. It can bring cyberspace attack behavior into state transfer behavior of finite automaton.

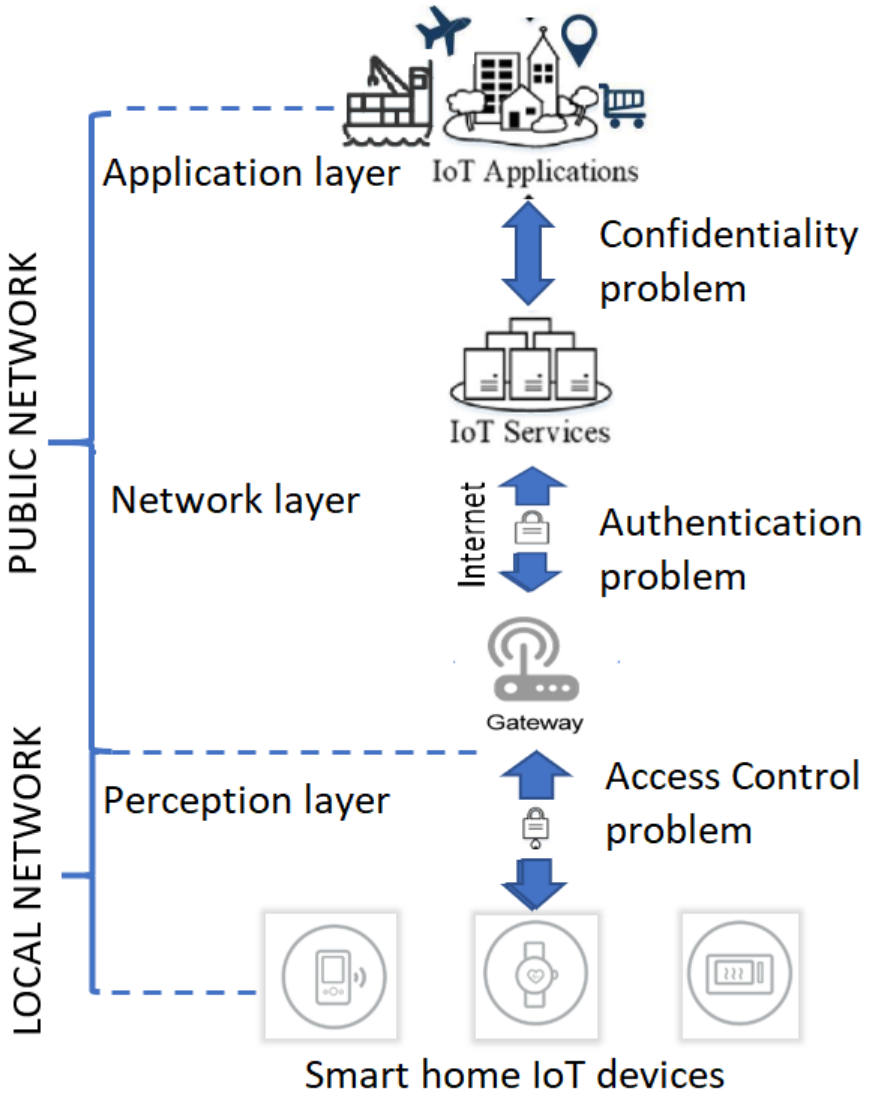

Figure 1: Smart home attack surface problem space with cybersecurity challenges in the different IoT layers

CHAPTER 3

RELATED WORK

In this section, we briefly survey the works that address risks and vulnerabilities associated with smart home-based IoT and attempt to use finite state automata for formally modeling the attacks and analysis the attack vulnerabilities with the objective of evaluating various security metrics.

3.1 Securing IoT in the smart home context

Lee et al. [13] pointed out the increasing popularity of smart home application of the emerging IoT and the need to provide adequate level of protection for potential cyber-attacks against the resource-constrained smart devices. They review the smart home technologies (applications, devices, operating systems, and communication protocols) and discuss the main security challenges and threats against them [6]. A smart environment that integrates IoT technology is considered to be a complex system because it consists of different products from different companies based on different technologies that do not share a universal language. However, the impacts of IoT security vulnerabilities are very dangerous in critical smart environments used in fields such as medicine and industry. In IoT-based smart environments without robust security systems, applications and services will be at risk. Confidentiality, integrity, and availability are three important security concepts of applications and services in IoT-based smart environments; thus, to address these

concerns, information security in IoT systems requires greater research focus [15]. Denning et al. [16] analyzed potential security attacks against home-based IoT and provided a structure for reasoning about the different security needs. They proposed an informative framework to evaluate the risk posed by in-home IoT along on three dimensions: the feasibility of an attack on the system, the attractiveness of the system as a compromised platform, and the damage caused by executing a successful attack.Although their proposed framework evaluates the smart home based risks, It is not clear that how efficiently it will analysis the attacks.

3.2 Risk assessment in cyber systems

There are several existing frameworks that are useful for risk assessment in cyber systems such at the MITRE ATT&CK framework [17], TARA [18], NIST SP 800-30 Guide for Conducting Risk Assessment [19], OCTAVE [20], and the various graph-based frameworks [21]. Most of these frameworks, except a few on the graph-based ones, generate a textual narrative (list) of vulnerabilities in the system and are not suitable for automated analysis; in fact, even manual what-if analysis is also challenging in many cases. The major shortcoming of the graph-based frameworks is that they cannot be easily updated and/or re-used when systems evolve. However, a majority of these models fail to consider the attackers capabilities and the likelihood of a particular attack being executed. To alleviate such drawbacks, Dantu et al. [22] propose a probabilistic model to assess network risks. They model network vulnerabilities using attack graphs and apply Bayesian

logic to perform risk analysis. Liu and Man [23] use Bayesian networks to model potential attack paths in a system, and develop algorithms to compute an optimal subset of attack paths based on background knowledge of attackers and attack mechanisms. In both Dantu et al. and Liu and Mans works, nodes in the attack graph are assigned a probability value that describes the likelihood of attack on a node. They compute the likelihood of system compromise by chaining Bayesian belief rules on top of the assigned probabilities.

3.3 Threat Model

In order to understand the IoT security landscape, a general IoT threat model is needed. A threat model defines threat scenarios with associated risk distributions. It helps in analyzing a security problem, design mitigation strategies, and evaluate mitigation solutions. When created in the design phase, a threat model helps to identify changes that need to be made to the design to mitigate potential threats [24]. When a threat model is created for a deployed system, it can be used to prioritize the mitigation actions [25]. Several studies have focused on modeling attacks and intrusions with the objective of evaluating various security metrics. Michael and Ghosh [26] employed a finite state machine (FSM) model constructed using system call traces. By training the model using normal traces, the FSM could identify abnormal program behaviors and thus detect intrusions. In [27], a finite state machine based technique to automatically construct attack graphs was described. The approach can be applied in a networked environment consisting of several users, various services, and a number of hosts. However, its applicability in the SHIoT environment is unclear.

3.4 Finite State Machine based Attack Model

Chen et al. [28] combined an analysis of data on security vulnerabilities and a focused source-code examination to develop a finite state machine (FSM) model to describe and reason about security vulnerabilities. An in-depth analysis of the vulnerability reports and the corresponding source code of the applications led to three observations: (i) exploits must pass through multiple elementary activities, (ii) multiple vulnerable operations on several objects are involved in exploiting a vulnerability, and (iii) the vulnerability data and corresponding code inspections allow us to derive a predicate for each elementary activity. These three observations motivated them to develop the FSM model to describe and reason about security vulnerabilities. Zhang et al. [29] presented an attack modeling method based on system states aggregation. In this model, the basic principles of finite state automaton were investigated and attack entities of cyberspace were classified by attack process. This work combines finite automaton with the changes of system state caused by attack entity, building the attack model of finite automaton, making an analysis of the model algorithm, and making a quantitative evaluation on attack cost, the success rate, exposure rate and evaluating severity of attack on cyberspace. Mouton et al. [30] described that human operators were one of the weakest links in the security chain as they are highly susceptible to manipulation. A social engineering attack targets this weakness by using various manipulation techniques to elicit individuals to perform sensitive requests. This paper proposed the underlying abstract finite state machine of the Social Engineering Attack Detection Model (SEADM) to formally address social engineering. This model is, however, only applicable for social engineering attacks and it's not clear

that how efficiently it will detect the smart network attacks. Therefore, this paper improves on the SEADM by providing the underlying finite state machine, which allows researchers to better understand and utilize the SEADM.

3.5 Risk and Vulnerability Analysis

There are several research endeavours on risk and vulnerabilitu analysis of IoT and its home application to gauge the impact of security attacks against them. Andreas et al. [5] review the state of the art in the context of smart home IoT security and privacy, and apply a risk analysis to evaluate smart home automation system vulnerabilities and threats and their potential impact. They used Information Security Risk Analysis (ISRA) method. In the ISRA method, the systems risk exposure is systematically reviewed based on the three basic requirements: confidentiality, integrity, and availability. Since smart home automation systems often have heterogeneous architectures, the proposed risk analysis would not be applicable to different technology designs. Furthermore, their approach is more beneficial in the design and development phases rather than attack identification and prevention during the operational phase. Costa et al. [31] presented a practical method supported by open source tools that can identify high risk vulnerabilities present in smart home IoT devices. Wang et al. [32] focused on vulnerability assessment of industrial internet of things and proposed a vulnerability graph model based on attack graph and a vulnerability algorithm based on maximum loss stream. Chen et al. [28] combined an analysis of data on security vulnerabilities and a focused source-code examination to

develop a finite state machine (FSM) model to describe and reason about security vulnerabilities. Davis et al. [33] mentioned that the vulnerability studies of IoT devices to date are not all inclusive and, in some cases, target well-known vendors or devices.

3.6 Main obervations

In summary, the above studies on smart homes focused mainly on possible security issues that may occur in a IoT based smart home environment. There is no work that covers the entire IoT architecture layer for smart homes shown in Figure 1 from the cyber security aspects. We address this missing part by proposing a finite state attack model as a more generalized form of cyber attacks. The analysis of the attack model is geared to meet the needs of cyberspace attack modeling with an aim to be effective on cyber attack detection and safety warning. Moreover, none of the above studies discussed how to build an automatic tool for the vulnerability analysis. Our comprehensive understanding of these public and local network of IoT smart home-based FSA models will enable us to examine how to analyze attack graphs for worst-case survivability and how to fortify for IoT security management.

CHAPTER 4

OVERVIEW OF FSA-BASED SMART HOME IOT ATTACK MODEL

In this work, we present a framework for modeling attacks in SHIoT that is based on the paradigm of finite state automata. FSA are a computational formalism that can be represented as a directed graph. However, in the graph-based risk modeling domain, there is no consensus as to what a node or an arc means. Nodes have been variously used to represent vulnerabilities in assets, actions, events, states and even a combination of these, and accordingly, arcs have been used to represent pre/post conditions of vulnerability exploitation, sequence of actions or events and state transitions. Thus, these graph-based models serve very well as a visualization tool for the defender in situational awareness campaigns; however, since most lack precision and formalism, they cannot be easily used for automated analysis. FSA by definition are used to capture state transitions, which reduces ambiguity in our modeling efforts. FSA processing can be easily automated and is not computationally intensive.

4.1 Review of Finite state Attack Automata

A finite state attack automaton is a non-deterministic or deterministic finite state machine that models attack of any complexity against the system. It describes the attack model through regular languages [34]. A deterministic machine has exactly one path for

every input-state pair. In a non-deterministic machine, there may be multiple valid transitions for every input-state pair, and the chosen transition is not defined; any transition can be followed. Using non-deterministic machine, we can make multiple valid attack paths for SHIoT state transitions. A deterministic finite state machine is a state machine that is guaranteed to complete for all inputs in a finite amount of time, while a non-deterministic finite state machine may execute indefinitely or fail to progress toward completion for certain input sets. A finite state machine is provably deterministic if and only if it is both free of cycles (that is, no state is ever revisited after being processed once) and defines a transition to a new state for each potential input in every state (that is, any valid input into a state results in a transition to a new state) [30]. As explained below, for modeling purposes we can safely assume that cycles are non-existent in an attack and hence we resort to a deterministic finite state automata for our work. We formally define the finite state automata for SHIoT attack (FSAA) as a tuple that includes the following elements:

$$FSAA = (S, \Sigma, \delta, S0, F)$$

where S is a non-empty finite set of states representing various states of interest in modeling the attack. In particular, S0 represents initial state representing a system steady state when no attack had been launched. We use σ to denote the finite set of input symbols representing the transition alphabets. A transition is an action that causes the system to change from one state to another, denoted by δ. Finally, F is the set of terminal states which can be one of the potential attack success states or attack failure states. Thus, $F \subseteq S$. Figure 2 is a simplified example of an FSAA related to vulnerabilities in a system like SHIoT, where each state of S represents an instance of the SHIoT environment attack

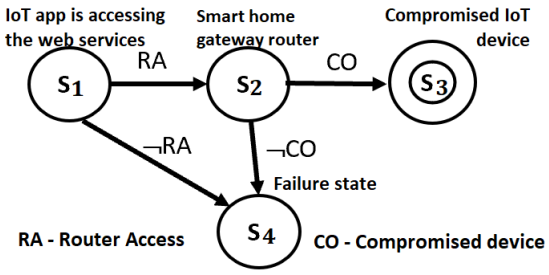

Figure 2: A finite state machine illustration

scenario. Here, we have $S = S0, S1, S2, S3$.

4.2 A Smart Home Internet of Things based finite state machine illustration

Based on Figure 2, when a users phone that is compromised is trying to access the smart home gateway router from the public network through IoT app web services, a transition $\delta(S0, RA) = S1$ occurs, where RA (Router Access) is an input alphabet symbol. When the attacker is not able to access the home router, a transition $\delta(S1, CO) = S3$ occurs where S1 is read into the input alphabet symbol ¬CO, in which ¬ is used for negating symbol CO (Compromised) to indicate "not compromised", and the new state becomes S3. When the attacker successfully compromises an IoT device, the successful final accepting state (S2) is denoted by the transition symbol CO (Compromised), and the failure state is S3 with the transition input symbols ¬RA and ¬CO. Table 1 and Table 2 shows the State Transition Table and State Transition Table for all Input Alphabets for the SHIoT based illustrtaion. We posit that the FSAA attack graph for our problem is a cyclic. When an attack is conducted, there may be one or more attack steps that take the system

Input / State	S_1	S_2	S_3	S_4
RA	S_2	-	-	-
¬RA	S_4	-	-	-
CO	-	S_3	-	-
¬CO	-	S_4	-	-

Table 1: State Transition Table for the SHIoT based illustration

	Input		Output	
No	Σ_1	Σ_2	Σ_3	Σ_4
1	RA	CO	✓	-
2	¬RA	$\emptyset$	-	✓
3	RA	¬CO	-	✓

Table 2: State Transition Table for all Input Alphabets

from an initial compromised state to other states (which may be the initial compromised state too - the case of a self loop) and then back to the initial compromised state. This is a cycle and hence to properly model such a scenario in the form of a finite state attack automata, it must allow for cycles. However, in terms of value gained, a cycle does not increase the likelihood of an attack or change the outcomes of the attack. If we consider that each automaton state corresponds to a set of transitions that takes the attacker closer to its desired goal, any cycle includes at least one attack that cannot further increase the advantage towards its goals [7]. Thus, it is safe to assume that a sequence of transitions cannot visit the same state twice or a previously visited state.

4.3 Why Finite State Attack Automata?

An attack in SHIoT environment is very complex, and this is very important to use mathematical method description in order to study various complex attack behavior. The literature survey on the FSA based attack model focused mainly on the threat and vulnerability identification and investigation of the attack entities. What is lacking is a comprehensive model that would allow the security analysts to capture and analyze the nature of the interactions between the different devices, applications, and human users, as well as the vulnerabilities and misconfigurations, in order to understand the weak spots and vulnerabilities in the SHIoT system and prepare for potential security attacks. The finite state automata based attack model is intended to help facilitate the attack execution flow by grouping attack vulnerabilities. It takes the attack behavior as the associated process, classifying the attack entity, then studying the state transfer under the attack behavior, and finally being able to identify attack vulnerabilities. Thus, the development of a Finite State Attack model methodology capable of expressing the process of exploitation by composing the vulnerable operations and possible exploits in the SHIoT system. A major advantage of using FSA to model SHIoT risk is that an FSA by definition is used to capture state transitions, which reduces ambiguity in our modeling efforts and FSA processing can be easily automated and is not computationally intensive. Moreover, it can be converted to a regular grammar, which in turn, can be used to generate regular expressions, thus providing opportunities to harness the power of regular expression tools and techniques.

CHAPTER 5

ATTACK MODELING USING FINITE STATE AUTOMATA: A FORMAL TREATMENT FOR SHIOT

The main purpose of attack model is to understand, explore and validate security threats in the cyber world. An attack model can be used to understand the motive of the attacker, that is, why the attack happened and what information could be targeted [7]. In the SHIoT environment, attack model can identify the attack plan, a sequence of actions that allow attackers to achieve their goals, such as access to specific sensitive information [35]. Through this attack model, the smart home system administrator can easily analyze different attack paths and then decide which vulnerabilities to prioritize for patching. During such analysis, the FSAA attack model captures the following valuable aspects related to the attack: (i) Attack source: who are the attackers, e.g., internal vs external, and their capabilities. (ii) Attack goal: what they want to achieve. (iii) Attack method: how attackers deploy attacks. (iv) Attack consequence: the damage will be resulted from attacks. Different types of cyber security aspects and their attack models are described in the following section. These correspond to the reference attack space shown in Figure 1.

5.1 Modeling confidentiality cyber security: Public network

Different properties of the smart home network stimulate different ways for an attacker to compromise a SHIoT system. We first define vulnerable states that allows us

to categorize the public/local network attack model properties for further analysis.

Definition 1: Vulnerable States in SHIoT environment:

A vulnerable state is a common attack model property that includes the following: (a) system vulnerabilities and network vulnerabilities (as reported in vulnerability database) (b) insecure system properties such as unsafe security policy, no mech anism for updating software, corrupted file access permission (read/write access) (c) insecure public network properties such as public Wi-Fi and hotspot connection. (d) insecure smart home network properties such as unsafe network condition, unsafe hard-coded passwords, unsafe IoT device/peripheral access permission. Each vulnerable state property helps us to categorize the vulnerabilities of the public/local network that may be useful to find out attackers intention as where he is going to hit first or which rout the attacker will take in order to attack the smart home, For example, "joining the insecure public Wi-Fi networks access" can be considered as an instance of the network vulnerabilities. Similarly, "unsafe IoT device/peripheral access permission" is an instance of the SHIoT network vulnerable property. Such vulnerable states and properties let us specify the different types of the smart home based public and local network attacks.

Definition 2: Transitions in SHIoT environment:

Each transition is a property of the public/local network elements that controls traverseability of actions over the smart home network. Let $\mathcal{S}$ be the set of states and $\mathcal{T}$ be the set of transitions. Here, the transition is represented as $\mathcal{T}: \mathcal{S}_{pre} \rightarrow \mathcal{S}_{post}$ where $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq \mathcal{S}$. Transitions are further associated with a truth value, True ($T = 1$) or False ($T = 0$) representing either successful or failure exploitation. For example, the

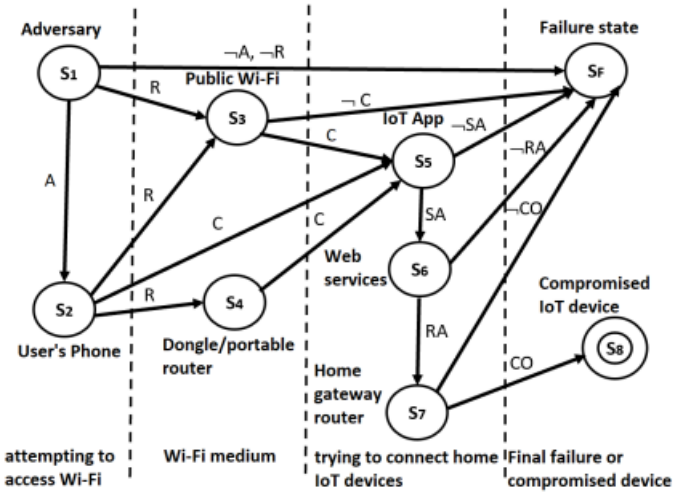

Figure 3: Confidentiality based attack model

state S: "joining the insecure public Wi-Fi networks access" is associated with a truth value signifying whether an attacker has compromised the users mobile phone. We shall also use the term "compromised" to indicate the true (or T = 1) state of an attribute. The success or failure of an attacker reaching its goal depends mostly on the states transition in a public or private network. Thus, We formally define a finite state attack model to capture the consequence relationships between such vulnerable transition states along with a most vulnerable attack path.

Definition 3: FSA based attack model components:

The FSA based attack model consists of transitions and states. The transition and state count will be varying from attack to attack and network to network. Consider S to be the set of states. δ is the transition function that takes (state, input symbol(Σ)) and maps to a resulting state: $\delta: \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$, where S_{pre} and S_{post} denote the set of starting

Input Alphabet	Description
R	Request
C	Connect
A	Access
SA	Service access
RA	Router access
CO	Compromised

Table 3: Input alphabet symbols

states and the set of ending states, respectively. The successful or compromised transition is noted by the true value 1, while a failure is noted by 0.

A FSA based attack model consists of a set of successful transition states and a set of failure transition states. Therefore, the set of successful transitions lead to a successful final accepting state and a failure transition leads to a reject state. For example, the successful transition path from the state S2: "User controller (mobile phone) is trying to connect to the smart home IoT device through the public network" to the state S8: "The compromised IoT device" and the failure transition path from the state S3: "Public Wi-Fi" to the state S5: "The IoT application connection". The transition function $\delta^*: \mathcal{S}_{pre} \times \Sigma^* \rightarrow \mathcal{S}_{post}$ denotes the set of successful state transitions (extended transitions or a walk of transitions).

Definition 4: Confidentiality based cyber-attack (public network): Let $\mathcal{S}$ be the set of states. We define a compromised state between a pair of transition states as the mapping $C: \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$. Then, the function $a: S \rightarrow S$ is called a confidentiality based cyber-attack if for $S_{pre}, S_{post} \in \mathcal{S}$:

1. $S_{pre} \neq S_{post}$,

2. with S_{pre}, S_{post} a compromised state transition $C(S_{pre}, S_{post}) > 0$, and

3. $\exists S_1, \cdots, S_n \in \mathcal{S}$ such that $C(S_{pre}, S_1) > 0$, $C(S_1, S_2) > 0$, . . ., and $C(S_n, S_{post}) > 0$.

A confidentiality based cyber-attack allows an attacker to compromise the state S_{post} from S_{pre} with a true value of success (T=1). Although, given a compromised state, another state can be compromised with a successful true value using a chain of other states. Thus, in the third condition, each step in such a chain is a confidentiality based cyber-attack. Informally, an attack is associated with a vulnerability exploitation, denoted by e_i, which takes the attacker from one network state (S_{pre}) to another (S_{post}) where i denotes the i-th vulnerability exploitation from among all exploitations. Consequently, S_{pre} and S_{post} are respectively called a precondition and postcondition of the attack a, denoted by $a(S_{pre})$ and $a(S_{post})$, respectively. An attack relates the two different states to embed a cause-consequence relationship between the two. For example, for the states S_{pre} = "public Wi-Fi access" and S_{post} = "IoT application connection", the attack $S_{pre} \rightarrow S_{post}$ is associated with the e_i = "IoT application" exploit. Using this exploit, an attacker can monitor legitimate user′s online traffic and manipulate the private messages as well.

A description of the finite state attack automata machine in mathematical notation follows. The finite state machine is a 5-tuple consisting of the finite set of input alphabet symbols Σ representing the transition alphabet (For example, consider a transition $\delta(S_1, R) = S_3$ where R is an input alphabet symbol), the finite set of states $\mathcal{S}$, the start

States	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_F
A	S_2	-	-	-	-	-	-	-	-
R	S_3	-	-	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-	-	-
¬R	S_F	-	-	-	-	-	-	-	-
R	-	S_3	-	-	-	-	-	-	-
R	-	S_4	-	-	-	-	-	-	-
C	-	S_5	-	-	-	-	-	-	-
C	-	-	S_5	-	-	-	-	-	-
¬C	-	-	S_F	-	-	-	-	-	-
C	-	-	-	S_5	-	-	-	-	-
SA	-	-	-	-	S_6	-	-	-	-
¬SA	-	-	-	-	S_F	-	-	-	-
RA	-	-	-	-	-	S_6	-	-	-
¬RA	-	-	-	-	-	S_F	-	-	-
CO	-	-	-	-	-	-	S_8	-	-
¬CO	-	-	-	-	-	-	S_F	-	-

Table 4: State Transition Table for Confidentiality based Attack Model

state S_0, the set of accepting states $\mathcal{F}$, and the set of state transitions δ that contains 3-tuples representing state transitions, consisting of a current state, a current input, and the next state.

The successful confidentiality based cyber-attack notations are: $\Sigma = \{$R, C, A, ¬A, ¬R, ¬C, SA, ¬SA, RA, ¬RA, CO, ¬CO$\}$

$\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_F\}$

S_0 = S_1

δ = {((S_1, R), S_3), ((S_1, R), S_4), ((S_1, C), S_5), ((S_2, A), S_1), ((S_2, ¬A), S_F), ((S_2, R), S_3), ((S_2, ¬R), S_F), ((S_3, C), S_5), ((S_3,¬C), S_F), ((S_4, C), S_5), ((S_5, SA), S_6), ((S_5, ¬SA), S_F), ((S_6, RA), S_7), ((S_6, ¬RA), S_F), ((S_7, CO), S_8), ((S_7, ¬CO), S_F) }

	Input Alphabet							Output	
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	S_8	S_F
1	A	R	C	$\emptyset$	SA	RA	CO	✓	-
2	A	R	$\emptyset$	C	SA	RA	CO	✓	-
3	A	C	$\emptyset$	$\emptyset$	SA	RA	CO	✓	-
4	R	$\emptyset$	C	$\emptyset$	SA	RA	CO	✓	-
5	¬A	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	✓
6	¬R	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	✓
7	A	R	¬C	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	✓
8	R	$\emptyset$	¬C	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	✓
9	R	$\emptyset$	¬C	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	-	✓
10	A	C	$\emptyset$	$\emptyset$	¬C	$\emptyset$	$\emptyset$	-	✓
11	A	R	$\emptyset$	C	¬SA	$\emptyset$	$\emptyset$	-	✓
12	A	R	C	$\emptyset$	¬SA	$\emptyset$	$\emptyset$	-	✓
13	A	R	C	$\emptyset$	SA	¬RA	$\emptyset$	-	✓
14	A	R	C	$\emptyset$	SA	RA	¬CO	-	✓
15	A	C	$\emptyset$	$\emptyset$	SA	¬RA	$\emptyset$	-	✓
16	A	C	$\emptyset$	$\emptyset$	SA	RA	¬CO	-	✓

Table 5: State Transition Table for all Input Alphabets

δ^* = {((S_1, R), S_3), ((S_3, C), S_5), ((S_5, SA), S_6), ((S_6, RA), S_7), ((S_7, CO), S_8)}

Table 6 describes the public network transition state vulnerabilities. Using both Figure 3 and the provided mathematical notations, it is easy to imply a state transition table. Table 4 depicts all the possible state transitions given a specific input for each state. For all input states, the output is either a failure state or a state with a next high level state index. To further show that the FSA attack model provides a valid outcome of either success or failure for all given alphabet sequences, a transition table with all possible input alphabet sequences (paths) and their corresponding results are shown in Table 5.

Each row in the table represents a path. Σ_i shows the i-th input character of the path. The symbol $\emptyset$ indicates no transition occurred in the i-th position of the path.

Figure 3 explains the public network confidentiality based attack model. State S_1 is between the user controller device and the actual public Wi-Fi network, so the attacker can see the legitimate users online traffic with the transition alphabet A (Access). While the attacker is trying to initiate the man in the middle (MITM) attack, any disruption occurs due to out of range signal or the user changed the current public wi-fi service, the current transition goes to the failure state S_F. Subsequently, the attacker can directly access the user's mobile phone by launching malware and phishing attack with input symbol A. If the attacker fail to succeed or compromise the user's phone, the transition goes to the failure state S_F with the input symbol $\neg$A.

State S_2 denotes the user controller and it deals with the Wi-Fi request connection. Initially, the user tries to connect to the public Wi-Fi with the connection request R (Request). Similarly, the user can use the portable Wi-Fi router or dongle (S_4) to get the Wi-Fi access with the connection request R or the user can directly connect to the IoT application using the mobile data with the connection transition input symbol (C).

Once the user controller got connected into the public Wi-Fi, the user next connects to the IoT application and use the web server as well. In that case, the MITM attack directs to monitor all the legitimate users transactions one by one, Thus the attacker can travel virtually with the user from the transition states S_5 to S_6, S_7, S_8 with the input symbols SA, RA, CO.

State S_5 deals with the IoT application connection. The user can access the IoT

application through the public Wi-Fi internet/dongle/LTE. The successful transition alphabet will be marked by C (Connection). If there is any problem occurs due to poor signal, the transition goes to S_F with the transition input symbol ¬C.

State S_6 deals with the web server connection along with the transition state symbol SA. If the attacker is not able exploit the web server by injecting commands and scripts, the failure state transition (S_F) will occur with the input symbol ¬SA.

State S_7 deals with the home router gateway connection. If the gateway allows the IoT application request, the user can easily control the IoT device with the transition alphabet RA or else it will go to the failure state S_F with the input symbol ¬RA.

State S_8 deals with compromising the IoT device. Through the MITM attack, the attacker can travel with the user controller. Once he got the home router gateway access, it is easy for him to compromise the home IoT devices. The state S_8 is the final successful state where the attacker can easily read, insert, and modify messages and data after successfully compromise the device that can be denoted by the transition CO (Compromised).

5.2 Modeling authentication cyber security: Public network

In this cyber security aspect, brute-force attack is a major threat to most of the smart home environment as it is hard to discover that the smart network system does not seem to be operating abnormally. When an attacker executes brute force attack via the public network, he initially tries to hack the login credentials by making a number of login attempts. Since the attack happens in the public network, the attacker can try to hack

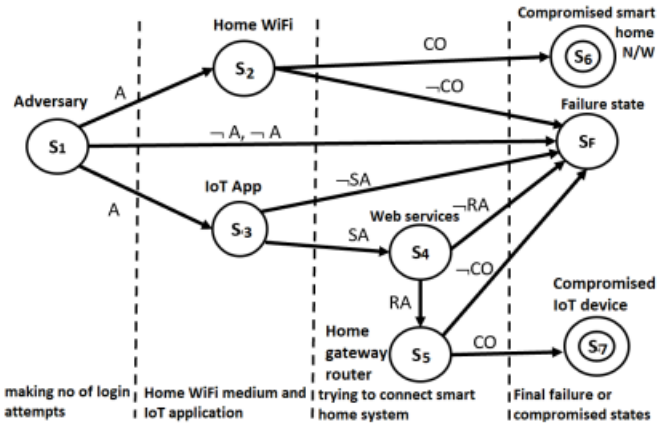

Figure 4: Authentication based attack model

the home Wi-Fi credentials as well as IoT application authentication credentials. Due to the diverse exposure of SHIoT, IoT applications are prime candidates for authentication brute-force attempts.

The successful authentication based cyber-attack notations are:

$\Sigma = \{$A, ¬A, SA, ¬SA, RA, ¬RA, CO, ¬CO$\}$

$\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_F\}$

S_0 = S_1

δ = {((S_1, A), S_2), ((S_1, A), S_3), ((S_2, CO), S_6), ((S_2, ¬CO), S_F), ((S_1, ¬A), S_F), ((S_1, ¬A), S_F), ((S_3, SA), S_4), ((S_3, ¬SA), S_F), ((S_4, RA), S_5), ((S_5, ¬RA), S_F), ((S_5, CO), S_7), ((S_5, ¬CO), S_F)}

δ^* = {((S_1, A), S_3), ((S_3, SA), S_4),((S_4, RA), S_5), ((S_5, CO), S_7)}

Definition 5: Authentication based cyber-attack (public network):

Given a directed graph G, let $\mathcal{S}$ be the set of states and $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq S$. We define C, a compromised state between a pair of transition states after the credentials have been breached. Thus,$C(S_{pre}, S_{post}) = 1$ is called an *authentication control based cyber-attack* where a is attack with A is its input symbol to denote brute-force attempts.

1. Initially, $S_{pre} \neq S_{post}$
2. If a: $\mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$ is an attack, then $C(S_{pre}, S_{post}) = 1$.

An authentication based cyber-attack allows an attacker to compromise the home Wi-Fi/IoT application credentials with the successful transition δ: $\mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$. For example, S_{pre} = "The attacker is making the authentication credentials attempts" and S_{post} = "Home Wi-Fi router/IoT application" with the associated transition state symbol A. Thus, the attack $a(S_{pre}, A) = S_{post}$.

Table 7 illustrates all the possible state transitions and Table 8 shows the transition table with all possible input alphabet sequences (paths) and their corresponding results. Figure 4 explains the public network authentication based attack model. State S_1 deals with the attacker login attempts. The attacker can hack home Wi-Fi and IoT app credentials by making no of login attempts with the transition input symbol A (Attempt). If the attempts did not work for a certain amount of time, the transition goes to the failure state (S_F) with the input symbol ¬A (Not a successful attempt).

State S_2 deals with the home Wi-Fi medium.If the attacker breaks the home Wi-Fi credentials, he can adversely control the smart home network system with the input symbol CO. once the attackers have access to the network, they are much harder to catch.

If the attacker is not able to break the credentials after several attempts, the transition goes to the failure state (S_F) with the input symbol ¬CO.

State S_3 deals with the IoT application brute force attempts.If the attacker is able to hack the IoT application authentication credentials, the transition goes to the next level with the transition input symbol SA. If he fails to hack the credentials after a several attempts, the transition goes to failure state with the input symbol ¬SA.

State S_4 deals with the web server connection along with the transition state symbol RA. If the attacker is not able exploit the web server by injecting commands and scripts, the failure state transition (S_F) will occur with the input symbol ¬RA.

State S_5 deals with the home router gateway connection. If the gateway allows the IoT application request, the attacker can easily control the IoT device with the transition alphabet CO or else it will go to the failure state S_F with the input symbol ¬CO.

State S_6 deals with compromising the smart home network system and it is the final successful compromised state. Once the attacker found the correct Home Wi-Fi credentials, it is easy for him to compromise the smart home network system.

State S_7 deals with compromising the IoT device. Once the attacker hacked the IoT application authentication credentials, he can compromise an IOT device through smart home gateway router. Thus, State S_7 is the final successful compromised state.

5.3 Modeling access control cyber security: Local network

A Denial-of-Service (DoS) is an attack meant to shut down a machine or network, making it inaccessible to its intended users and it plays a major role for access control

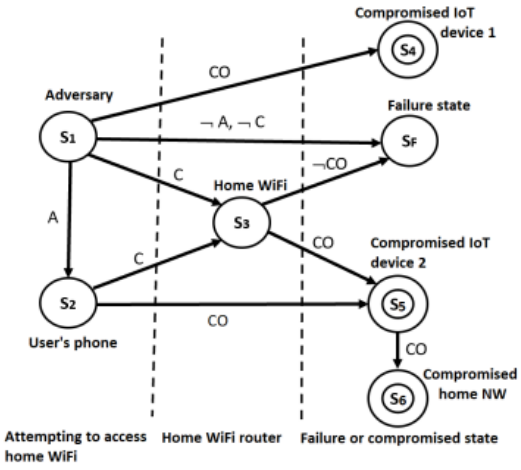

Figure 5: Access control based attack model

based cyber security aspect. The attacker accomplish this attack by flooding the target with traffic or sending it information that triggers a crash. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and effort. Table 9 describes the local network transition state vulnerabilities.

The successful access control based cyber-attack components are: Σ = {A, ¬A, C, ¬C,CO, ¬CO }

$\mathcal{S}$= {S_1, S_2, S_3, S_4, S_5, S_6, S_F}

$\mathcal{F}$ = {S_4, S_5, S_5, S_F}

S_0 = S_1

δ= {((S_1, C), S_3), ((S_1, CO), S_4), ((S_1, A), S_2), ((S_2, C), S_3), ((S_1, ¬A), S_F), ((S_1, ¬C), S_F), ((S_2, A), S_5), ((S_3, CO), S_5), ((S_3, ¬CO), S_F), ((S_5, CO), S_6)}

δ^*={((S_1, C), S_3), ((S_1, A), S_2), ((S_2, A), S_5), ((S_5, CO), S_6)}

Definition 6: Access control based cyber-attack (local network):

Given a directed graph G, Let $\mathcal{S}$ be the set of states. We define C, a compromised state between a pair of transition states, as a mapping $C: \mathcal{S} \times \Sigma \rightarrow S' = [0, 1]$, where Σ is an input alphabet. Then, given $\mathcal{S}_{pre}, \mathcal{S}_{post} \subseteq S$, $a: \mathcal{S}_{pre} \times \Sigma \rightarrow \mathcal{S}_{post}$ is called access control based cyber-attack.

1. Initially, $S_{pre} \neq S_{post}$,
2. Given S_{pre}, S_{post} a compromised state transition $C(S_{pre}, S_{post}) > 0$.

An access control based cyber-attack allows an attacker to compromise the state S_{post} from S_{pre} with a true value of success $(T = 1)$. Although, given a compromised state can be compromised a whole smart home network using direct access, an attack is associated with a vulnerability exploitation (e_i), which takes the attacker from one network state (S_{pre}) to another S_{post}. Therefore, we say that $C(S_{pre}, S_{post}) > 0$. For example, for the states S_{pre} = "attacker launch or misuse the smart home insecure network properties" and S_{post} = "IoT device", the attack $S_{pre} \rightarrow S_{post}$ is associated with the e_i = "Compromised IoT device" exploit. Using this exploit, an attacker can control the entire smart home network.

Table 10 illustrates all the possible state transitions and Table 11 shows the transition table with all possible input alphabet sequences (paths) and their corresponding results. In Figure 5, State S_1 deals with the attacker who is trying to access the local IoT device and user controller (User's phone). If the attacker is compromised the user

controller, he can access any of the IoT devices through the user controller. At the same time, the attacker can directly compromise an IoT device without entering the home Wi-Fi. The successful transition of the current state S_1 would be CO. If the attacker is not able to connect/access the IoT devices/user controller, the transition goes to the failure transition state S_F with the transitions ¬A/¬C.

State S_2 deals with the user controller. If the user controller is already compromised, the attacker can easily monitor user's online traffic while the user is trying connect/access the home Wi-Fi router/the IoT device with the transition C/A.

State S_3 deals with the home Wi-Fi. Once the attacker gets the home Wi-Fi connection, he can compromise an IoT device S_5 with the transition CO. Eventually, he can make the smart home network inaccessible. Through the compromised states S_4 or S_5, the attacker can control the whole smart home network and it is denoted by the state S_6 with the transition CO.

States Description	Vulnerability	Impact	CVE#
S_1 (adversary) - trying to access user's phone	Malware, Phishing	Take control of device	CVE-2021-27612
S_2 (User's phone) - trying to connect to a public wi-fi medium	Malware, Synchronization, Buffer Overflows, Phishing	Monitor user's online activities, take control of device	CVE-2021-23977
S_3 (Public Wi-Fi) - Accessing the IoT application	possibility of joining a fake or rogue Wi-Fi hotspot	allows cyber attackers to monitor userâĂŹs online traffic	CVE-2018-11477
S_4 (Dongle/Portal router) - Accessing IoT application	It becomes âĂIJdiscoverableâĂİ to malicious attacker seeking to exploit connection	allows attackers to sniff on network traffic and inject malicious scripts	CVE-2019-13053
S_5 (IoT application) - Accessing the web services	Infect associated smart application with malware	User credentials and private data could be stolen	CVE-2019-1698
S_6 (Web services) - Accessing the home Gateway Router	SQL Injection, Cross Site Scripting	user data can be modified (Insert/Update/ Delete)	CVE-2021-3340
S_7 (Home Gateway Router) - trying to compromise the IoT device	Uses UPnP to modify firewall settings, to reconfigure routers, and opens ports to IoT devices	Botnet creation as part of larger attacks such as DDoS	CVE-2009-2257
S_8 (Compromised IoT device)	Add fake/Sybil nodes to network and spread malware	Affect the whole network system, Increases the power consumption of sensor nodes	CVE-2019-1957

Table 6: Public Network Transition State Vulnerabilities

Input / State	S_1	S_2	S_3	S_4	S_5	S_6	S_F
A	S_2	-	-	-	-	-	-
A	S_3	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-
¬A	S_F	-	-	-	-	-	-
CO	-	S_6	-	-	-	-	-
¬CO	-	S_F	-	-	-	-	-
SA	-	-	S_4	-	-	-	-
¬SA	-	-	S_F	-	-	-	-
RA	-	-	-	S_5	-	-	-
¬RA	-	-	-	S_F	-	-	-
CO	-	-	-	-	S_7	-	-
¬CO	-	-	-	-	S_F	-	-

Table 7: State Transition Table for Authentication based Attack Model

	Input					Output		
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	S_F
1	A	CO	-	-	-	✓	-	-
2	A	¬CO	-	-	-	-	-	✓
3	¬A	∅	∅	∅	∅	-	-	✓
3	¬A	∅	∅	∅	∅	-	-	✓
5	A	∅	SA	RA	CO	-	✓	-
6	A	∅	¬SA	-	-	-	-	✓
7	A	∅	SA	¬RA	-	-	-	✓
8	A	∅	SA	RA	¬CO	-	-	✓

Table 8: State Transition Table for all Input Alphabets

States Description	Vulnerability	Impact	CVE#
S_1(adversary)- accessing home router	executing dictionary attack, Synchronization, Buffer Overflows	take control of the device	CVE-2021-23977
S_2(User's phone)- accessing IoT device	Insecure hard coded default password, UPnP system	allowing hackers and malware to hijack firmware, software, and IoT devices.	CVE-2018-20100
S_3(Home Wi-Fi router)- accessing an IoT device	It can add fake nodes to the network and spread malware to the network	affect the whole system, Increases the power consumption of sensor nodes	CVE-2019-1957
S_4-Compromised IoT device	executing code/scripts remotely and gain superuser rights in the system	Overall network performance will become unusually slow, IoT devices start operating on its own, compromised connected devices are pulled into a botnet	CVE-2020-2035
S_5-Compromised home network	executing code/scripts remotely and gain superuser rights in the system	Overall network performance will become unusually slow, compromised connected devices are pulled into a botnet	CVE-2020-2035

Table 9: Local Network Transition State Vulnerabilities

Input / State	S_1	S_2	S_3	S_4	S_5	S_F
C	S_3	-	-	-	-	-
CO	S_4	-	-	-	-	-
¬CO	S_F	-	-	-	-	-
A	S_2	-	-	-	-	-
¬A	S_F	-	-	-	-	-
C	-	S_3	-	-	-	-
A	-	S_5	-	-	-	-
CO	-	-	S_5	-	-	-
¬CO	-	-	S_F	-	-	-
CO	-	-	-	S_6	-	-

Table 10: State Transition Table for Access Control based Attack Model

	Input			Output		
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	S_F
1	C	$\emptyset$	CO	-	✓	-
2	A	C	CO	-	✓	-
3	A	CO	$\emptyset$	-	✓	-
4	CO	$\emptyset$	$\emptyset$	✓	-	-
5	¬C	$\emptyset$	$\emptyset$	-	-	✓
6	¬A	$\emptyset$	$\emptyset$	-	-	✓
7	A	C	¬CO	-	-	✓
8	C	$\emptyset$	¬CO	-	-	✓

Table 11: State Transition Table for all Input Alphabets

CHAPTER 6

SMART HOME IOT ATTACK MODEL WORST VULNERABILITY ANALYSIS: A GRAPH-BASED APPROACH

We cast an IoT environment as a complex network where the intent is to support the mission rather than just as a set of connected entities. We envision the steps to be as follows:

1. Formulate a graph model to capture all possible mission dependencies that are relevant as well as their relationships to the broader mission objectives, which will allow mission survivability related analysis.

2. Identify graph metrics and graph analysis techniques to answer queries about the resiliency of individual components to security threats, as well as answer questions about the overall mission survivability.

We consider the graph model to help us perform two types of analyses: (i) graph-theoretic and (ii) quantitative. The graph-theoretic analysis is geared more towards what-if queries on survivability. It helps us to identify critical components of the mission, their relationships with each other and to the overall mission objectives. It might also help us identify potential but yet undiscovered attacks on the mission continuity. It also drives the quantitative analysis that may help answer questions about the overall robustness of the mission. Thus, we focus on developing a graph-theoretic analysis framework for the mission based on the above philosophy. We (i) propose graph metrics to evaluate roles played

by various arcs in component graphs and identify mission critical arcs, (ii) develop graph metrics that can be used to answer what-if queries on the mission continuity, and (iii) develop efficient techniques to re-evaluate core mission graph and propose enhancement.

6.1 Graph-based attack modeling analysis

We outline three different types of analysis on the mission network:

1. Structural analysis – This will determine which individual components are most critical to the continuity of the mission and if attacked can lead to serious (potentially cascading) failures in the mission. The result of these analyses (there can be different types depending on the chosen metric) will provide different ranked sets of critical arcs representing different aspects of mission continuity (what are those aspects of mission continuity), and will serve as the basis for making quantitative analysis.

2. Vulnerability analysis – This analysis will use the ranked set of mission critical arcs and determine which cyber vulnerabilities on these critical arcs can be exploited resulting in the compromise of the mission. The analysis is similar to a traditional attack graph based analysis but will also perform newer analysis that will allow us to fortify the environment to provide enhanced survivability to attacks. For this, we need to perform quantitative analysis on the resulting graph.

3. Fortification analysis – The purpose of this analysis is to study the impact of different component's compromise on the overall mission and how to strategically

address vulnerabilities of different arcs in an attack graph. This helps one to determine strategies for mission survivability towards fortification of the system.

The above-mentioned analysis are used to employ an iterative technique to propose fortification.

6.2 Vulnerability Analysis through an Algorithmic Approach

Our vulnerability analysis for an attack graph is based on determining the worst vulnerable path in an attack graph; a public network confidentiality based cyber-attack illustration has shown in chapter 5 on how an attack graph is generated for an SHIoT system. Since the vulnerability of an arc in the graph is represented as a probabilistic value, the usual shortest path based approach cannot be used. Thus, we present an approach to tackle the problem.

Consider an attack graph $\mathcal{G}$ of N nodes in which v_{ij} represents the vulnerability probability of an arc associated with attack $S_i \rightarrow S_j$, where $0 < v_{ij} \leq 1$, in this attack graph. If two states S_i and S_j are not connected, then v_{ij} is assumed to be zero.

Given v_{ij}, the vulnerability of path p from state S_i to state S_j is given by

$$v_p^{(i,j)} = 1 - \prod_{(i',j') \in \mathcal{P}_{ij}} (1 - v_{i'j'}) \tag{6.1}$$

where $\mathcal{P}_{ij}$ is the path consisting of the set of arcs (i', j') for path p from state S_i to state S_j. Thus, the problem of finding the most vulnerable path in a graph between two states S_i and S_j among the set of paths Ω may be written as

$$\max_{p \in \Omega} v_p^{(i,j)} \tag{6.2}$$

Observe that v_p has non-additive properties in terms of arc vulnerability probabilities. Thus we cannot directly apply a shortest path algorithm based on the vulnerability probabilities.

Instead, we bank our approach on another observation. The complement of vulnerability for an arc is reliability where we denote the reliability of $S_i \rightarrow S_j$ for an arc to be $r_{ij} = 1 - v_{ij}$. While there has been work on determining the most reliable path in a graph ([36]), the problem of finding the most vulnerable path has not been explored.

Now, given r_{ij}, the vulnerability of path p in (6.1) can be written as

$$v_p^{(i,j)} = 1 - \prod_{(i',j') \in \mathcal{P}_{ij}} r_{i'j'} \tag{6.3}$$

Now we introduce the term w_p to be $1 - v_p$, i.e., we can rewrite (6.3) as

$$w_p^{(i,j)} = \prod_{(i',j') \in \mathcal{P}_{ij}} r_{i'j'} \tag{6.4}$$

Note that w_p is not the reliability of path p.

Based on w_p, we can write (6.2) as the following equivalent problem

$$\min_{p \in \Omega} w_p^{(i,j)} \tag{6.5}$$

Since (6.3) has product terms, for the minimization problem (6.5) we cannot directly apply a shortest path algorithm. On the other hand, taking logarithm of both sides in (6.4), we can write

$$\log w_p^{(i,j)} = \log \left(\prod_{(i',j') \in \mathcal{P}_{ij}} r_{i'j'} \right) = \sum_{(i',j') \in \mathcal{P}_{ij}} (\log r_{i'j'}) \tag{6.6}$$

Thus, we can now solve the minimization problem (6.5) by using the arc weight to be $\log r_{i'j'}$ since we now have additive properties of path in terms of arc cost $\log r_{i'j'}$. We

do still have an additional issue to address. Since $0 \leq r_{i'j'} \leq 1$, the term $\log r_{i'j'} < 0$, i.e., in the log-space, the arc weights are always negative. Recall that the attack graph we described has acyclic property, which means that we can apply Bellman-Ford algorithm for arcs with negative weights $\log r_{i'j'}$ [37].

To summarize, our overall approach to determine the most vulnerable path in an attack graph is as follows from the initial state S_1 to the compromised state S_{comp}:

- Instead of arc vulnerability v_{ij}, use the transformed term $\log(1 - v_{ij}) = \log r_{ij}$ for each arc as the abstracted arc weight.
- Instead of solving (6.2), solve (6.5) by considering the arc weight as $\log r_{ij}$ using the Bellman-Ford shortest path algorithm on the acyclic attack graph.

This is summarized in Algorithm 1.

Algorithm 1 Vulnerability Analysis: Determining Worst Vulnerability in an N-node Attack Graph from state S_1 to the compromised state S_{comp}

Algorithm 1: Vulnerability Analysis: Determining Worst Vulnerability in an N-node Attack Graph from state S_1 to the compromised state S_{comp}

1. **Require:** Input: Attack graph $\mathcal{G}$ of N nodes with arcs associated with attack $S_i \rightarrow S_j$ and their vulnerabilities v_{ij}
2. $D_{1,1} \leftarrow 0$
3. for ($k = 2$ to $k = N - 1$) do
4. $\quad D_{1,k} \leftarrow \infty$
5. for ($h = 0$ to $N - 1$) do
6. $\quad D_{1,comp} \leftarrow \min_{\forall k \neq comp} \left\{ D_{1,k} + \log(1 - v_{k,comp}) \right\}$
7. $\quad$ Update p
8. endfor
9. return $F \leftarrow D_{1,comp}$ (cost), p (the most vulnerable path)

CHAPTER 7

SHIOT FORTIFICATION PROCESS AND VULNERABILITY ANALYSIS

The Fortification process builds on the vulnerability analysis discussed above. Our approach on fortification is based on first identifying the weakest arc on the most vulnerable path of the attack graph. We assume that once we know this, we can take measures to reduce its weakness through efforts such as any software updates to reduce its vulnerability.

We assume, in this study, that we can do this improvement in a certain boosting value on the weakest arc on the most vulnerable path. We then re-run the vulnerability analysis on the attack graph with this change in the arc weight due to improvement. We continue this process of improvement iteratively until a desirable threshold on fortification is attained. Since in our scenario, no arc can have v_{ij} below 0.0, we added a condition in this iterative process to check for this possibility. This process is captured in Algorithm 2. In practice, it may not be possible to reduce vulnerability on every arc of the attack graph. This variation can be easily captured in our fortification process by marking such arcs as not candidates for boosting.

Our fortification process is assessed in representative attack graphs to quantify the number of iterations needed to reach a particular fortification threshold. Note that our process is quite generic and can be used for a wide range of attack graphs for vulnerability assessment beyond the realm of SHIoT.

Algorithm 2 Fortification Process

Algorithm 2: Fortification Process

1. Input: an attack graph $\mathcal{G}$ with N nodes with arc vulnerabilities v_{ij}
2. Input: Fortification threshold: $F_{threshold}$
3. Input: Boost parameter: B
4. Initialize: $F_{now} = 1.0$
5. $F_{now}, p_{now} \leftarrow$ Algorithm 1
6. While ($F_{now} \geq F_{threshold}$) do
7. $\quad (i', j') \leftarrow \text{argmax}_{(i,j) \in p_{now}} v_{ij}$
8. $\quad v_{i'j'} = \max\{v_{i'j'} - B, 0\}$
9. $\quad F_{now}, p_{now} \leftarrow$ Algorithm 1
10. Endwhile
11. return # of iterations to reach $F_{threshold}$

Metric	Metric Value	Numerical Value
Attack Vector	local	0.7
	remote	1.0
Attack Complexity	high	0.8
	low	1.0
Privileges Required	required	0.6
	not-required	1.0
User Interaction	none	0.8
	required	0.6
Confidentiality	partial	0.7
	complete	1.0
Integrity	partial	0.7
	complete	1.0
Availability	partial	0.7
	complete	1.0

Table 12: Base metric elements and values of the base metric group based on the CVSS [1]

7.1 Determining arc vulnerability

Our approach for attack model vulnerability is based on the probabilistic estimation of arc's vulnerability on the attack graph. To compute the probability of an attack arc, the probability of success needs to be estimated while an attacker exploits a vulnerability exploitation. We use the metrics defined in VCE database Common Vulnerability Scoring System in this work to evaluate the attack probability. CVSS is the most commonly used vulnerability scoring system and it is supported by the US national vulnerability Library (NVD) [32]. It comprises three distinct groups of metrics such as base, temporal, and environmental. The base metrics measure the intrinsic characteristics of a vulnerability with two subscores: (1) the exploitability score, composed of the access complexity

Increment \Threshold	P(0.4)	P(0.5)	P(0.6)	P(0.7)
10%	152	179	197	216
15%	104	111	126	150
20%	73	89	93	102
25%	60	66	73	80

Table 13: Iterations required for attack graph with 9 nodes shown in Figure 3

and authentication (AU) occurrences and (2) the impact score, expressing the potential damage on confidentiality(C), integrity, and availability(AC). The temporal metrics measure dynamic aspects of a vulnerability in the environment around the smart home. The environmental metrics measure two aspects of impact that are dependent on the environment surrounding the smart home. More information on CVSS metrics and their scoring computation can be found in the CVSS documentation [1].

In this work, we considered only the base metrics score such as authentication, confidentiality, and access control in the analysis. Since this paper focuses on the vulnerability probability assessment of the smart home network system, in order to simplify the problem, we do not consider the temporal and environment metrics group.

The Base Score formula depends on sub-formulas for Impact Sub-Score (ISS) and Exploitability, which are defined below:

$$\text{ISS} = 1 - [(1 - \text{Confidentiality}) \times (1 - \text{Integrity}) \times (1 - \text{Availability})]$$

Given the vulnerability exposure information (CVSS attributes), the probability of vulnerability v of an arc (i, j) is computed from CVSSs Exploitability subscore as the

following:

$$v = \text{ISS} \times \text{AttackVector} \times \text{AttackComplexity} \times \text{PrivilegesRequired} \times \text{UserInteraction}$$

All metrics are determined under the assumption that the attacker has already located and identified the vulnerability. Thus, the analyst need not consider how the vulnerability was identified. Additionally, many different sectors' individuals will be scoring vulnerabilities, such as software vendors, vulnerability bulletin analysts, and security product vendors. However, vulnerability scoring is expected to be skeptical of the individual and their organization. For example, the privilege required metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability, and this metric value can be categorized as none (1.0), and required (0.6) [1].

7.2 Fortification Analysis

For our what-if analysis, we consider the 9-node attack graph for SHIoT shown in Figure 3 in the Appendix. In addition, we used a 15-node attack graph (Figure 6) from [38], and generated 26-node, 50-node and 60-node attack graphs.

We applied our fortification process on these graphs. For the boost parameter, we started with 10% improvement and conducted the study till 25% for an arc. For the fortification threshold, we used values 0.4 to 0.7. The results for 9-node, 15-node, 26-node and 50-node attack graphs are presented in Tables 13, 14, 15, and 16, respectively.

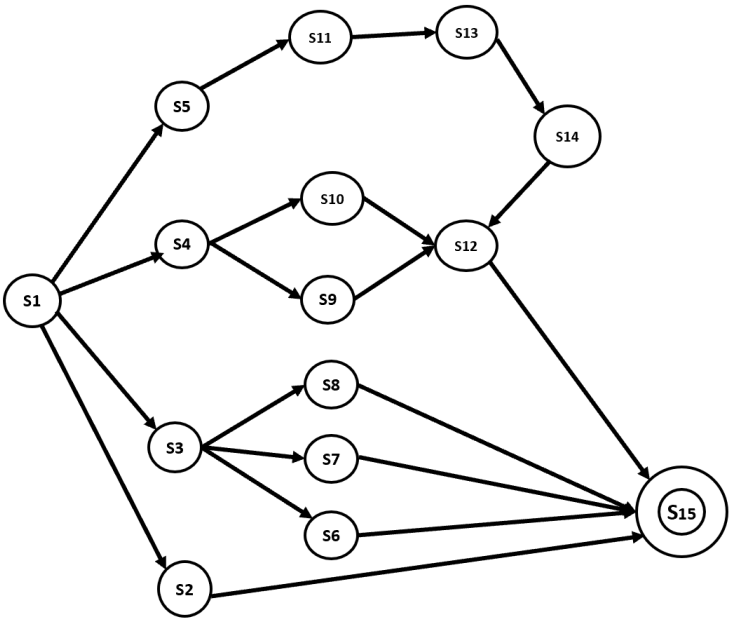

Figure 6: An attack graph with 15 nodes

As can be seen from the results, a higher fortification threshold and larger attack graphs require more iterations to reach the goal, while small boost steps also require a higher number of iterations. A higher number of iterations reflects that more efforts are needed for fortification.

Finally we explained our fortification process with the 60 node attack graph as shown in the (Figure 9). Our previous attack graphs were examined with boost parameter till 25% for an arc. We added an additional boost parameter 30% on this 60 node attack graph and we used values 0.4 to 0.8 for the fortification threshold. The result of the 60 node attack graph is presented in Table 17.

As shown in the table results, a higher fortification threshold value 0.8 requires

Increment \Threshold	P(0.4)	P(0.5)	P(0.6)	P(0.7)
10%	246	274	312	356
15%	186	205	278	303
20%	154	178	211	284
25%	112	136	187	223

Table 14: Iterations required for an attack graph with 15 nodes shown in Figure 6

more iterations 749 to reach the threshold. The iteration represents the effort needed to fortify the SHIoT system. As per the first 10% improvement booster value and the fortification threshold 0.8, the number of iterations to reach the threshold is 749. Since the threshold value is high, the vulnerability of that arc is also high. However, the fortification process relies on the number of iterations until it reaches the threshold. A resource should be strategically allocated in each iteration to ensure the fortification of an SHIoT system in order to protect and secure an arc vulnerability from an attack impact. More iterations of the fortification process improve security functionalities by allocating different resources and diverse control systems that cover the technical details of system architecture and functionality. Thus, the fortification is ideally implemented at every point of system alteration or expansion via the number of iterations.

This type of what-if analysis is helpful in systems management for system administrators as they can strategically allocate resources towards fortification of an SHIoT system. Secondly, a real resource can be associated with each iteration to determine the overall cost of such fortification.

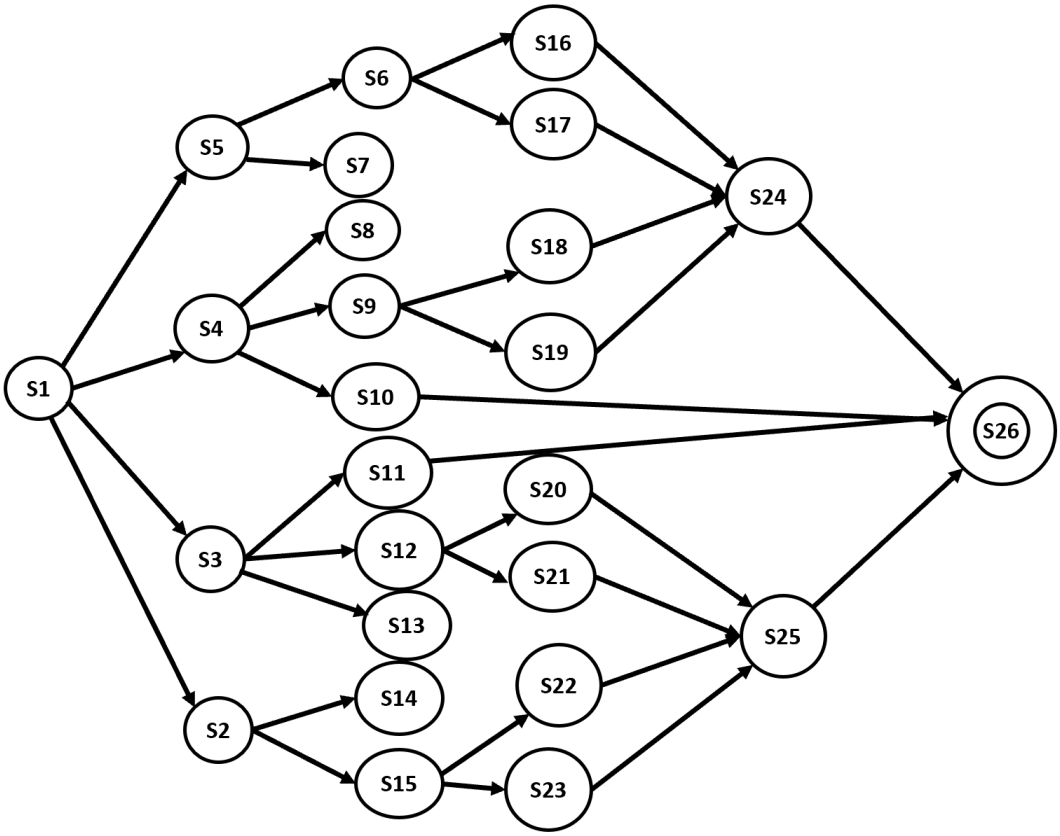

Figure 7: An attack graph with 26 nodes

Weight Increment	P(0.4)	P(0.5)	P(0.6)	P(0.7)
10%	468	494	531	588
15%	411	432	467	498
20%	392	421	458	487
25%	368	394	421	464

Table 15: Iterations required for an attack graph with 26 nodes shown in Figure 7

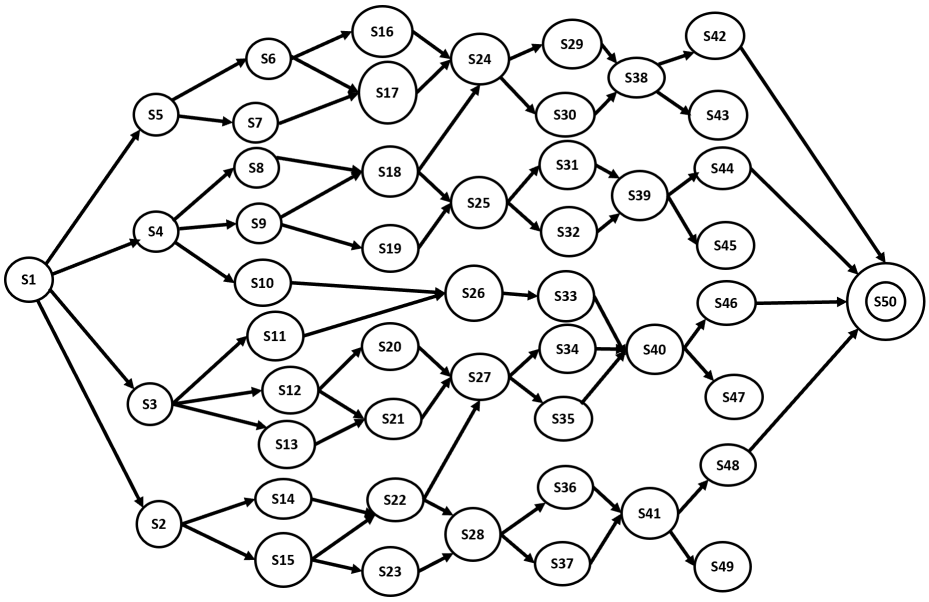

Figure 8: An attack graph with 50 nodes

Weight Increment	P(0.4)	P(0.5)	P(0.6)	P(0.7)
10%	578	597	628	661
15%	523	564	592	628
20%	481	512	568	594
25%	452	489	529	576

Table 16: Iterations required for an attack graph with 50 nodes shown in Figure 8

Increment \Threshold	P(0.4)	P(0.5)	P(0.6)	P(0.7)	p(0.8)
10%	636	658	681	702	749
15%	594	621	674	697	718
20%	536	578	613	658	686
25%	489	541	587	622	657
30%	425	461	497	528	576

Table 17: Iterations required for an attack graph with 60 nodes shown in Figure 9

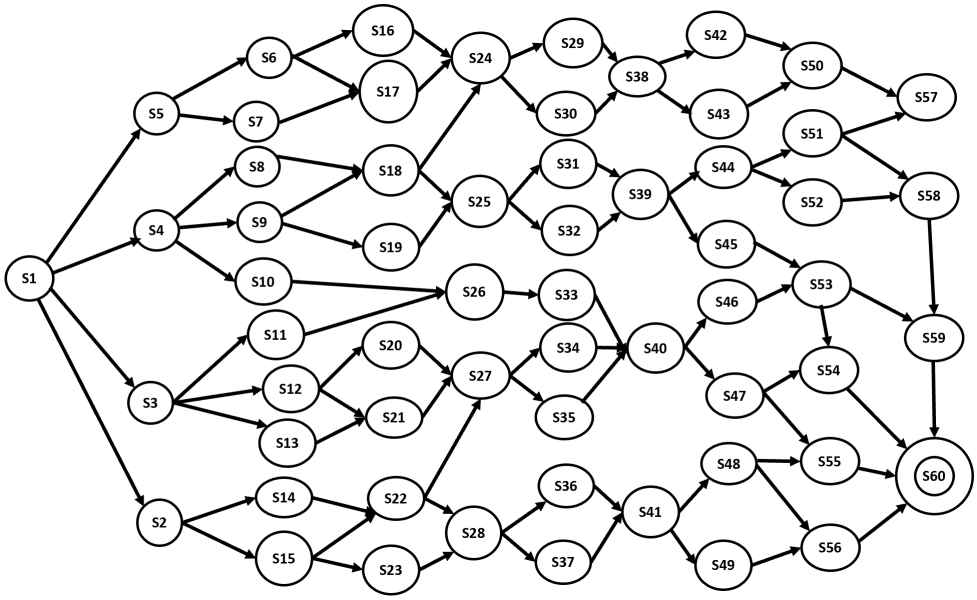

Figure 9: An attack graph with 60 nodes

CHAPTER 8

CONCLUSION AND FUTURE WORK

Inspired by investigating the vulnerability of the threat and attacker motive in an SHIoT environment, we introduced a finite state automata-based attack model for three different smart home based cyberattacks: a confidentiality attack, an authentication attack and an access control attack. We then presented a formalism for each smart home based cyber security aspect attack model and showed how the model enables a better understanding of the security posture of the system. We presented a graph based framework to represent the SHIoT based attacks. In this framework, an attack graph is first represented through Finite state automata in order to analysis the vulnerability for an arc for a confidentiality based cyberattack, followed by a fortification process to enhance the overall smart home system security. In particular, we showed how vulnerability analysis can be done using a Bellman-Ford algorithm with modified arc weights, and how a fortification process can use this vulnerability analysis through an iterative process. It may be noted that our fortification process can be used for any acyclic attack graphs, not just limited to SHIoT. We then studied our approach on representative attack graphs. Our approach allows to identify the worst vulnerability if the vulnerability of an arc changes.

Since our work focuses on SHIoT security management through worst attack vulnerability, there are a number of limitations of our work. Specifically, the worst attack vulnerability may not be the most important issue for certain IoT security management.

For instance, instead of probability, the cost to fortify an arc may be important or a combination of both probability and cost with different priorities may be important. Furthermore, while we used CVSS to estimate the probability of an arc in our illustration, in practice, this could be very difficult to determine. Also, estimates of the probability could be erroneous, and thus, a straight-forward application of our worst attack vulnerability approach may lead to error propagation.

Though we implemented our approach in the IoT environment as a complex network in our work, it would be useful to examine and address the SHIoT attack graphs dynamically. It allows us to verify the dynamics and scalability of the SHIoT environment's attack surface vulnerabilities. We can dynamically add the state transition based on the attack scenarios in an attack graph. However, the more ways IoT devices can connect, the more vulnerable the SHIoT security system becomes due to the large attack surface. We will further implement our attack graph dynamically in our future work by constructing a logical attack graph. The advantage of a logical attack graph is that it clearly specifies the underlying relations between system configuration information and an attacker's potential privileges, and it is possible to enumerate all possible attack scenarios by depth-first traversing as well. In addition to its scalability and extensibility, MulVAL (multi-host, multi-stage vulnerability analysis language) [39] can be used to generate an attack graph. MulVAL is a well-known open-source framework for constructing logical attack graphs. It requires four main inputs to construct a logical attack graph: security domain knowledge, such as CVE (Common Vulnerabilities and Exposures); information regarding the environment state, such as the principals and network and host configuration, the security

policy, the IoT devices, applications, services and reasoning rules. MulVALâĂŹs reasoning engine relies on interaction rules, which describe how facts and privileges are used by actions to achieve attack goals. On the other hand, nodes may have vulnerabilities that affect the overall system performance. It is necessary to develop an approach to tackle the node's vulnerability. These issues would be important to consider in future research.

REFERENCE LIST

[1] "Common vulnerability scoring system version 3.1: Specification document." [Online]. Available: https://www.first.org/cvss/

[2] "Common smart home iot attacks that compromise security." [Online]. Available: https://easydmarc.com/blog/7-common-internet-of-things-iot-attacks-that-compromise-security/

[3] X. Wu, J. Wang, P. Li, X. Luo, and Y. Yang, "Internet of things as complex networks," *IEEE Network*, vol. PP, pp. 1–8, 06 2021.

[4] J. Pacheco and S. Hariri, "Iot security framework for smart cyber infrastructures," in *Foundations and Applications of Self* Systems, IEEE International Workshops on*. IEEE, 2016, pp. 242–247.

[5] "Smart home security: Security and vulnerabilities." [Online]. Available: https://www.wevolver.com/article/smart-home-security-security-and-vulnerabilities

[6] W. Ali, G. Dustgeer, M. Awais, and M. Shah, "Iot based smart home: Security challenges, security requirements and solutions," in *2017 23rd International Conference on Automation and Computing (ICAC)*, 2017.

[7] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: A survey," *Journal of Cloud Computing*, 2018.

[8] L. Allodi and S. Etalle, "Towards Realistic Threat Modeling: attack Commodification, Irrelevant Vulnerabilities, and Unrealistic Assumptions," in *Proc. the 2017 Workshop on Automated Decision Making for Active Cyber Defense*, ser. SafeConfig '17, 2017.

[9] F. James, "Iot cybersecurity based smart home intrusion prevention system," in *2019 3rd Cyber Security in Networking Conference (CSNet)*, 2019.

[10] ——, "A risk management framework and a generalized attack automata for iot based smart home environment," in *2019 3rd Cyber Security in Networking Conference (CSNet)*, 2019.

[11] F. James, I. Ray, and D. Medhi, "Situational awareness for smart home iot security via finite state automata based attack modeling," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2021, pp. 61–69.

[12] A. S. Mohammad, N. Reddy, F. James, and C. Beard, "Demodulation of faded wireless signals using deep convolutional neural networks," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 969–975.

[13] "Cybersecurity Consideration for connected smart homes and devices." [Online]. Available: https://industrie-4-0.ul.com/wp-content/uploads/2018/02/UL_Cybersecurity_SmartHome_White_Paper_en.pdf

[14] H. Lin and N. W. Bergmann, “Iot Privacy and Security Challenges for Smart Home Environments,” *Information*, vol. 7, no. 3, 2016.

[15] M. F. Elrawy, A. I. Awad, and H. F. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–20, 2018.

[16] T. Denning, T. Kohno, and H. M. Levy, “Computer Security and the Modern Home,” *Commun. ACM*, vol. 56, 2013.

[17] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, “MITRE ATT&CK®: Design and Philosophy,” MITRE Corporation, McLean, VA, Tech. Rep. MP18036R1, July 2018.

[18] J. Wynn, “Threat Assessment and Remediation Analysis (TARA),” MITRE Corporation, McLean, VA, Tech. Rep. 14-2359, 2014.

[19] Joint Task Force Transformation Initiative Interagency Working Group, “NIST SP 800-30, Revision 1 – Guide for Conducting Risk Assessment,” NIST, NIST Special Publication, 2012.

[20] C. J. Alberts, S. G. Behrens, R. D. Pethia, and W. R. Wilson, “Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Framework, Version 1.0,” Software Engineering Institute, CMU, Pittsburgh, PA, Tech. Report CMU/SEI-99-TR-017, 1999.

[21] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, "Survey of Attack Graph Analysis Methods from the Perspective of Data and Knowledge Processing," *Security and Communications Network*, vol. 2019, 2019.

[22] R. Dantu, K. Loper, and P. Kolan, "Risk management using behavior based attack graphs," *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, vol. 1, pp. 445–449 Vol.1, 2004.

[23] Y. Liu and H. Man, "Network vulnerability assessment using bayesian networks," in *SPIE Defense + Commercial Sensing*, 2005.

[24] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated Security Test Generation with Formal Threat Models," *IEEE Transactions on Dependable and Secure Computing*, 2012.

[25] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "Computing the impact of cyber attacks on complex missions," in *2011 IEEE International Systems Conference*, 2011.

[26] C. C. Michael and A. Ghosh, "Simple, State-Based Approaches to Program-Based Anomaly Detection," *ACM Trans. Inf. Syst. Secur.*, 2002.

[27] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proc. 2002 IEEE Symposium on Security and Privacy*, May 2002.

[28] S. Chen, Z. Kalbarczyk, J. Xu, and R. Iyer, "A Data-Driven Finite State Machine Model for Analyzing Security Vulnerabilities," in *Proc. 2003 International Conference on Dependable Systems and Networks*, 2003.

[29] Z.-W. Zhang and Y. Yun-Tian, "Research of attack model based on finite automaton," in *2012 National Conference on IT and CS*, 2012.

[30] F. Mouton, A. Nottingham, L. Leenen, and H. Venter, "Finite State Machine for the Social Engineering Attack Detection Model: seadm," *SAIEE Africa Research Journal*, vol. 109, 2018.

[31] L. Costa, J. Barros, and M. Tavares, "Vulnerabilities in iot devices for smart home environment," in *Proceedings of 5th ICISSP*, 2019.

[32] H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu, "A vulnerability assessment method in industrial internet of things based on attack graph and maximum flow," *IEEE Access*, 2018.

[33] B. D. Davis, J. C. Mason, and M. Anwar, "Vulnerability studies and security postures of iot devices: A smart home case study," *IEEE Internet of Things Journal*, 2020.

[34] F. Baiardi, F. Martinelli, L. Ricci, C. Telmon, and L. B Pontecorvo, "Constrained automata: A formal tool for risk assessment and mitigation," *Journal of Information Assurance and Security*, 01 2008.

[35] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Generation Computer Systems*, vol. 56, 2016.

[36] M. Roosta, "Routing through a network with maximum reliability," *J. of Math. Analysis & Apps.*, vol. 88, no. 2, pp. 341–347, 1982.

[37] J. Fakcharoenphol and S. Rao, "Planar graphs, negative weight edges, shortest paths, and near linear time," *J. Comp. & Sys. Sci*, vol. 72, pp. 868–889, 2006.

[38] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, "Survey of attack graph analysis methods from the perspective of data and knowledge processing," *Security and Communication Networks*, 2019.

[39] X. Ou, S. Govindavajhala, and A. W. Appel, "Mulval: A logic-based network security analyzer." in *USENIX security symposium*, vol. 8, 2005, pp. 113–128.

VITA

Fathima James was born in India. She was educated in local public schools and earned her Bachelor of Engineering in Computer Science and Engineering from MS University in India, a Master of Engineering in Computer Science and Engineering at Anna University in India, a second Master of Science from the University of Tennessee at Chattanooga. she was a Computer Science Instructor at the University of Missouri at Kansas City, where she was responsible for designing IT courses and taught classes in design and analysis of algorithms and network architecture and discrete structures. She also taught as an Assistant Professor at Chattanooga State Community College and Lecturer for Computer Science at Anna University in India. In addition to her teaching experience, she was a Software Engineer for Centralized Showing Services. She is currently working as an Assistant Professor at the Benjamin Franklin Cummings Institute of Technology.