TOWARD ASSURANCE AND TRUST FOR THE INTERNET OF THINGS

By
Jeffrey S. Chavis

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Engineering

Baltimore, Maryland
April 2021

Abstract

Kevin Ashton first used the term Internet of Things (IoT) in 1999 to describe a system in which objects in the physical world could be connected to the Internet by sensors. Since the inception of the term, the total number of Internet-connected devices has skyrocketed, resulting in their integration into every sector of society. Along with the convenience and functionality IoT devices introduce, there is serious concern regarding security, and the IoT security market has been slow to address fundamental security gaps. This dissertation explores some of these challenges in detail and proposes solutions that could make the IoT more secure. Because the challenges in IoT are broad, this work takes a broad view of securing the IoT.

Each chapter in this dissertation explores particular aspects of security and privacy of the IoT, and introduces approaches to address them. We outline security threats related to IoT. We outline trends in the IoT market and explore opportunities to apply machine learning to protect IoT. We developed an IoT testbed to support IoT machine learning research. We propose a Connected Home Automated Security Monitor (CHASM) system that prevents devices from becoming invisible and uses machine learning to improve the security of the connected home and other connected domains. We extend the machine learning algorithms in CHASM to the network perimeter via a novel IoT edge sensor device. We assess the ways in which cybersecurity analytics will need to evolve and identify the potential role of government in promoting needed changes due to IoT adoptions. We applied supervised learning and deep learning classifiers to an IoT network connection log dataset to effectively identify varied botnet activity. We proposed a methodology, based on trust metrics and Delphic and Analytic Hierarchical Processes, to identify vulnerabilities in a supply chain and better quantify risk. We built a voice assistant for cyber in response to the increased rigor and associated cognitive load needed to maintain and protect IoT networks.

Primary Reader and Advisor: Aviel Rubin

Secondary Reader(s): Lanier Watkins, Anna Buczak, Anton Dahbora, and Matthew Green

Acknowledgments

I would like to thank many people for their support along this journey. I am very grateful to my advisor, Professor Aviel Rubin, for supporting me during my journey into the Internet of Things and for his guidance along the road toward the Doctor of Engineering (DENG) degree. Avi's profound insights and broad vision in research have been incredibly helpful. I am also thankful to him for always being available and for giving me the latitude to blaze a path as a part of the first cohort of students in the new Johns Hopkins University (JHU) DENG program. This allowed me to explore different research ideas and topics, which not only helped me to shape my specific research interests, but also made the research process enjoyable.

I am also extremely grateful to Dr. Lanier Watkins for his invaluable support of my research. Lanier provided a clear roadmap toward a doctorate degree, and has been with me every step along the way. I have learned from Lanier the right way to conduct scientific research, how to present it, and the value of publishing research papers and reports in peer-reviewed venues. I am forever indebted to him for his motivation, guidance, and tutelage.

I would also like to express my gratitude to Dr. Anna Buczak for serving in my thesis proposal and dissertation committees and to Professor Anton Dahbura and Professor Matthew Greene for serving on my Graduate Board Oral committee.

My work on this thesis and participation in the JHU DENG program were made possible by The Johns Hopkins University Applied Physics Laboratory (JHU/APL). I am thankful for the opportunity, support, and resources to complete this degree and have immense gratitude for the many people at JHU/APL who gave their support in this endeavor. This would not, and could not, have happened without you. I would also like to thank the faculty staff and administrators in the Computer Science Office in the JHU Whiting School of Engineering for their guidance and support.

Special thanks go to Chris Yoon from DHS CISA and Gina Marshall Johnson from JHUAPL for their support for this IoT research.

Versions of the chapters in this dissertation have appeared in various publications or have been submitted to various venues. I would like to take this opportunity to thank my co-authors on each publication. They include, Avi Rubin [1] [2] [3] [4], Lanier Watkins [1] [2] [3] [4] [5], Anna Buczak [1] [2] [3] [4], Aaron Kunz [2] [4], Daniel Syed [3] [4] [6], Otis Brooks [6], Mandira Hedge [5], Gilles Kepnang [5], Mashaal Al Mazroei [5], Tracy Herriotts [4], Kofi Nyarku [4], Samantha Fu [4], Elizabeth Aguirre [4], Antonio Davilla [4], Malcom Doster [4], Syeda Zeeshan [4], and Michelle Feng [4]. These publications and submissions may have been reproduced in whole or in part in this dissertation.

I would like to give special thanks to my wife Sharnet, whose unending support made this possible. I would also like to thank my sons, Jeffrey Brenden and Stefen Chavis, my mother Joan Murphy, brother James Perry Chavis, uncle Jolley Thomas Harris, and editors Barbara Johnson and Rosemary Dodds for reading drafts of this thesis. In addition, special thanks goes to my sister, Jena Wilson for her words and acts of encouragement.

Lastly, I would like to thank my father, James Matt Chavis. Although he is not here in body, you were with me every step of the way of this journey, in spirit. His example of academic achievement was the template that I try to mimic. You, Dad, planted the seeds of a doctorate degree in me when as a child I watched you work toward your Doctorate. This degree has been a lifelong goal instilled in me by you. I am truly appreciative for the opportunity to realize it.

Dedication

This thesis is dedicated to my wife Sharnet and my two sons, Brenden and Stefen, for their eternal love, trust, and support for me during this journey. I would also like to dedicate this work to my parents, Joan Harris Murphy and James Matt Chavis, who through tutelage and example gave me a desire for and an understanding of the value of education. I am indebted to you both for instilling this amazing gift in me.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

This dissertation explores security challenges in the Internet of Things (IoT) in detail, and proposes solutions that will make these devices, systems built from them, and data residing on them more secure. Unlike many other technologies, the IoT space must address unique challenges and requirements that place significant constraints on possible solutions to problems. For this reason, we take a multi-prong approach to IoT security, considering aspects of technology, ecosystem, and policy [7].

Throughout this work, we will explore particular problems in depth and introduce novel technologies that provide solutions. Our work strives to create a more secure IoT-enabled smart system environment centered on the integration of the technologies proposed in this work.

1.1 IoT Concepts and Definition

The IoT is a concept in which everyday devices become more connected, making it possible for them to become smarter; processing becomes intelligent, and everyday communication becomes informative [8]. The term Internet of Things was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors [9]. The National Institute of Standards and Technology (NIST) provides a description, rather than a definition: “The Internet of Things (IoT) is a rapidly evolving and expanding collection of diverse technologies that interact with the physical world” [10]. Although this might seem vague, the NIST description importantly captures the non-static nature of technologies and modalities of interaction with the tangible. However, for practitioners in different domains, such a description can be too loose and therefore not provide essential guidance. The questions arise: What technologies? And, how do these technologies interact?

There is still no universally accepted definition for the IoT. For example, the Internet Engineering Task Force (IETF) view of the “Internet” considers the Transmission Control Protocol/Internet Protocol (TCP/IP) suite and non-TCP/IP suite as protocols and classifies “things” as people, machines, or information that

comprise the IoT that connects objects around us to provide seamless communication and contextual services provided by them [11]. On the other hand, the International Telecommunication Union (ITU) defines the IoT as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [12]. The National IoT Strategy Dialogue recommended that "[t]he IoT consists of "things" (devices) connected through a network to the cloud (datacenter) from which data can be shared and analyzed to create value (solve problems or enable new capabilities)" [13]. The Institute of Electrical and Electronics Engineers (IEEE) has solicited help in developing a definition. They surveyed more than 12 international standards organizations and research bodies to curate a domain-neutral yet more granularly characterized description that is tailored to the underlying "things" of interest, scale, and operational needs—resulting in two definitions for small- and large-scale scenarios [14].

For the purposes of this dissertation, the IoT is defined as all uniquely identifiable physical or virtual objects that exchange information with other uniquely identifiable physical or virtual objects via the Internet (or other communications network) and the evolving technology that supports this exchange as appropriate for the scale and operational needs. Figure 1-1 (from [15]) illustrates this IoT concept and its three constituent elements:

1. Things – Objects in the physical or virtual world that can be identified and integrated into a communications network
2. Devices – Digital equipment with the required capability to communicate and the optional capability of sensing, actuation, data storage, and data processing
3. A communications network (e.g., the Internet)

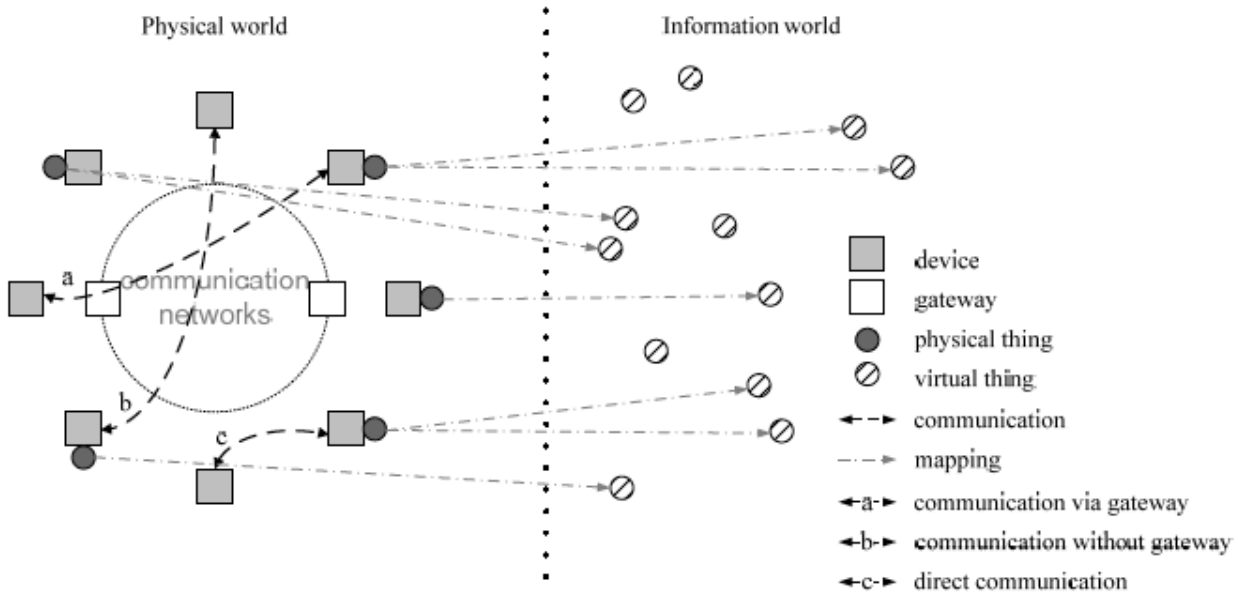

Figure 1-1 Elements of the IoT Concept

The IoT is already very large, consisting of billions of connected devices, and it is expected to grow by an order of magnitude by the end of the next decade [16].

The IoT will be an integral part of every sector of the U.S. economy and will substantially influence daily activities ranging from healthcare to transportation to provision of basic services. Smart grids, with automated sensors and control systems to achieve increased efficiency, will be integrated into transportation and public utility systems. Industrial processes will become increasingly dependent on automation and robotics, affecting the manufacturing, chemical, and defense sectors. Financial and legal institutions housing vast amounts of sensitive data will be increasingly equipped with IoT-enabled sensors and controls including security surveillance systems. Public health and emergency services will be increasingly connected to biometric and geolocation sensors to provide continuous monitoring of patient and first responder status.

1.2 Security Concerns of the Internet of Things

The IoT is more than the connection of an increasing number of “smart” devices. It is one of a number of technology evolutions that will radically alter the way machines communicate. In this sprit, it would be

impossible to assess the cybersecurity requirements of the emerging IoT without also considering the effects of the emerging fifth generation (5G) standard. Although IoT and 5G are independent entities, their development paths are intertwined. Two of three core 5G capabilities [Massive Machine-to-Machine (M2M) Communication and Extremely-Low-Latency and High-Reliability Communications] identified by the ITU relate specifically to the needs of IoT. Similarly, the potential for increased bandwidth, device-to-device communications, and increased adoption of virtualization capabilities presented by the emerging 5G standard will profoundly affect the shape of the IoT. The 5G-powered IoT will dramatically change how networks operate because capability will become increasingly distributed and virtualized and devices will operate more independently of traditional protective infrastructures, dynamically joining and leaving networks. We call attention to 5G's significant impact here in the introduction to offer a complete assessment of IoT challenges; however, 5G is out of scope for this body of work as we focus on issues related more directly to IoT. 5G will be addressed in future work.

We surveyed critical emerging capabilities, technologies, and trends associated with the IoT and identified a number of factors that could create novel and/or heightened cybersecurity risk:

- *Increased scale* – The increased scale of IoT (i.e., more devices, more network traffic to be monitored) will increase the attack surface while stressing defenses.
- *Increased M2M communications* – Increased dependence on M2M and device-to-device (D2D) communications will require adaptation of existing network security techniques to the increased number of devices with limited security operating on networks and increasingly dynamic network architectures.
- *Increasingly virtual networks* – The ability to "stack" virtual networks within a physical network architecture will create new requirements and provide new opportunities for network security, as virtual networks within the same physical architecture may have unique quality-of-service (QoS) and security capabilities.

- *Evolution from cloud computing structures to fog computing structures* – As existing highly centralized cloud computing structures are replaced by more distributed fog computing, centralized security approaches will need to be updated to operate in a more distributed fashion with security applications operating at the edge and within clients on endpoint devices.
- *Increased physical risk* – IoT devices operating in public view, such as security cameras, sensors, and controls, may be compromised by persons with physical access, and such compromises may have significant negative physical effects via IoT actuators. Indeed, the sheer number of these devices will mean broader exposure to supply chain risks.
- *Lower-cost, lower-power devices* – The IoT is enabled by cost-effective devices with low power consumption connected to networks. Many of these devices will not have sufficient capacity to support the processing of intensive cybersecurity analytic capabilities, especially those involving machine learning.

Notionally, cybersecurity risk can be expressed in the following equation, which illustrates why the IoT introduces an increased number of security threats [17]:

$$Cybersecurity\ Risk\ = \frac{Threat\ level\ \times\ Probability\ of\ Attack\ \times\ Points\ of\ Exposure}{Cybersecurity\ Measures\ Implemented}$$

For each device connected to the Internet (i.e., point of exposure), the cybersecurity risk is a function of the threat posed by the attack, the probability that a device will be attacked and the cybersecurity measures implemented (which reduce the risk). As a result, as the number of connected devices increases, the cybersecurity risk increases. In addition, as the number of connected devices and applications (e.g., healthcare, industrial control, smart cities) increases, the potential impacts of a cyber-attack also increases. As basic functions become increasingly dependent on IoT-enabled capabilities, hackers have more targets, and more opportunities, to inflict damage. In addition, as the technology stack becomes increasingly complex to support the often-divergent needs of IoT applications, new threats across the

stack become possible. Critically, each device targeted by cyber-attack becomes a new threat vector, enabling additional attacks on other network-connected devices.

Cybersecurity vulnerabilities in IoT devices can be used to disable or modify IoT-enabled capabilities. An adversary might attempt to disable key industrial processes to inflict economic damage or disable critical safety mechanisms to cause physical harm. Conceivably, an adversary might attempt to exploit gaps in IoT cybersecurity to sabotage specific products or weapon systems or to disable vital public services including water, electricity, or communications. However, the point of attack may not be the ultimate target of a cybersecurity breach. Adversaries may attempt to exploit weaknesses in IoT cybersecurity in one organization or sector to gain access to important financial, legal, health, or other private data in another organization or sector.

IoT solution architectures require multi-layered security approaches to provide complete end-to-end security from device to cloud and everything in between throughout the lifecycle of the solution. Foundational changes are emerging with the IoT that will expose critical networks to potential vulnerabilities. We feel a vision is needed to assess the likely impacts of the IoT and identify potential capabilities to respond to the likely emerging challenges of an increasingly distributed, virtual, and dynamic IoT cyber environment.

1.3 Vision and Approach

The set of issues identified and described above cover multiple areas; thus, solutions to these concerns must go beyond the technical realm, requiring a multifaceted approach that examines the issues from multiple perspectives. Our vision is to explore these issues from the perspective of technology, the supporting ecosystem, and related polices that govern how IoT is implemented. This dissertation explores each issue and introduces approaches, processes, and techniques that can be deployed in IoT settings that work toward addressing each of these unique challenges.

1.3.1 Connected Home Automated Security Monitor (CHASM): Protecting IoT Through Application of Machine Learning [1]

The IoT will dramatically transform the home experience, but it presents significant security risks. We propose a system that helps reduce the cognitive load on a user in keeping their smart home network protected. The system helps prevent IoT devices from becoming invisible or forgotten by the user and provides semi-autonomous capability to address key security concerns in the connected home. In this paper, we describe the problem, explain specifications for the system, present our work in IoT discovery and IoT device classification portions of the system, and show initial results related to our efforts exploring novel application of machine learning to build this capability.

1.3.2 A Capability for Autonomous IoT System Security: Pushing IoT Assurance to the Edge [2]

Complex systems of IoT devices (SIoTD) are systems that have a single purpose but are composed of multiple IoT devices. These systems are becoming ubiquitous; they have complex security requirements and face a diverse and ever-changing array of cyber threats. Issues of privacy and bandwidth will preclude sending all the data from these systems to a central repository and so these systems cannot totally rely on a centralized cloud-based service for their security. The security of these systems must be provided locally and in an autonomous fashion. In this paper, we describe a capability to address this problem, explain specifications for the system, present our work on SIoTD assurance, and show initial results of a novel edge-based application of machine learning to build this capability.

1.3.3 Envisioning Cybersecurity Analytics for the Internet of Things [6]

The IoT represents a rethinking of how we currently envision networks. It involves increases in scale of an order of magnitude over the next decade, increasingly distributed and virtualized architectures, and an ever-changing perimeter. Cybersecurity analytic capability will need to adapt to meet the demands of this evolving environment. In this paper, the authors assess the ways in which cybersecurity analytics will need to evolve and identify the potential role of government in promoting needed changes. This paper provides a brief summary of that effort.

1.3.4 Identification of Botnet Activity in IoT Network Traffic Using Machine Learning [5]

Today our world benefits from IoT technology; however, new security problems arise when these IoT devices are introduced into our homes. Because many of these IoT devices have access to the Internet and they have little to no security, they make our smart homes highly vulnerable to compromise. Some of the threats include IoT botnets and generic confidentiality, integrity, and availability (CIA) attacks.

Our research explores botnet detection by experimenting with supervised machine learning and deep learning classifiers. Furthermore, our approach assesses classifier performance on unbalanced datasets that contain benign data, combined with small amounts of malicious data. We demonstrate that the classifiers can separate malicious activity from benign activity within a small IoT network dataset. The classifiers can also separate malicious activity from benign activity in increasingly larger datasets. Our experiments have demonstrated incremental improvement in results for (1) accuracy, (2) probability of detection, and (3) probability of false alarm. The best performance results include 99.9% accuracy, 99.8% probability of detection, and 0% probability of false alarm. This paper also demonstrates how the performance of these classifiers increases as IoT training datasets become increasingly larger.

1.3.5 A Proposed Trust Model for Assessing Cybersecurity Risk in a Supply Chain Considering IoT's Impact [3]

As the IoT becomes increasingly integrated into industrial processes, cyber-attacks to systems using their supply chains as points of entry become more common. A decade ago, the Stuxnet virus demonstrated the ability to inflict cyber-physical damage on even a closed system via its supply chain. The emergence of increased levels of Internet- and IoT-based capabilities—ranging from collaborative design capabilities to manufacturing processes that enable fabrication directly from digital models to increased use of automation and robotics on factory floor—expands exposure to the Internet and to cyber-attacks. Even as awareness of the need to protect against malware implanted in the supply chain grows, there is still no effective method of assessing the cyber risk in a supply chain. This paper describes how increased integration of IoT-enabled capabilities has expanded the threat surface and proposes a methodology,

based on trust metrics and Delphic and analytical hierarchical processes, to identify vulnerabilities in a supply chain and better quantify risk.

1.3.6 Toward an Ambient Computing Paradigm for IoT Cybersecurity: Lowering the Cognitive Load for Users [4]

The IoT is becoming more pervasive in the home, office, hospital, and many other user-facing environments (UFEs) as more devices are networked to improve functionality. However, this explosion of networked devices in UFEs necessitates that security systems become easier to help users remain aware of the security of the devices on their network. Users may not have the skills or the time needed to continuously monitor networks of increasing complexity using common open-source tools. Specifically, they are not likely to fully comprehend the data that those tools present, nor are they likely to have a working knowledge of the tools needed to monitor and protect their IoT-enabled network environments. This paper expands earlier work on a CHASM to build a system that uses ambient computing to facilitate network security monitoring and administration for the connected home and other UFE-connected environments such as hospitals, smart buildings, and smart cities, through novel integration of voice assistant technology. Our system is designed to combine machine-learning–enriched device awareness and dynamic visualization of IoT networks with a natural language query interface enabled by voice assistants to greatly simplify the process of providing awareness of the security state of the network. The voice assistant integrates knowledge of devices on the network to communicate status and concerns in a manner that is easily understood. These capabilities will help to improve the security of UFEs while lowering the associated cognitive load on the users. This paper outlines continued work in progress toward building this capability as well as initial results on the efficacy of the system.

1.4 Outline of this Work

Each chapter in this dissertation explores an aspect or problem related to security or privacy in IoT.

- Chapter 2 examines IoT security issues in the home and introduces the CHASM concept.
- Chapter 3 extends CHASM to the IoT edge.

- Chapter 4 examines cybersecurity analytics capability in response to the threats from the perspective of a large organizational entity.
- Chapter 5 examines the use of machine learning to identify botnet activity in IoT devices and networks.
- Chapter 6 introduces and examines a proposed framework for assessing IoT motivated cybersecurity risks and its impacts on supply chains.
- Chapter 7 examines ambient computing and its utility in lowering the cognitive load on users when protecting IoT enable networks.
- Chapter 8 is a summary of the preceding topics explored in this dissertation.
- Chapter 9 lists the references cited in the research and performance of work described herein.

2 CONNECTED HOME AUTOMATED SECURITY MONITOR (CHASM): PROTECTING IOT THROUGH APPLICATION OF MACHINE LEARNING

2.1 Introduction

Kevin Ashton first used the term Internet of Things (IoT) in 1999 to describe a system in which objects in the physical world could be connected to the Internet by sensors [18]. Since the inception of the term, estimates predict that by 2020 the total number of Internet-connected devices being used will be 50 billion [19], and the combined markets of IoT will grow to 520 billion by 2021 [20] into every sector of society including the home.

Along with the convenience, functionality, and entertainment options IoT in the home devices introduce, there is a host of related threats. IoT-related compromises are raising the profile of security concerns associated with IoT devices [21] [22] [23], but the IoT security market has been unhurried to address fundamental security gaps. Security is often applied to hardware, software, and system development as an afterthought instead of being included in the initial development process; often, the development of IoT devices, platforms, and protocols suffers from a lack of security [24]. The security concerns with IoT become more apparent when considering the ubiquitous way in which IoT will be integrated into our everyday lives [18] [25].

Specifically, one risk of owning home IoT devices is their ability to become invisible to the users over time [26]. Once devices become connected to the network, owners tend to forget the device is connected and, therefore, do not actively and regularly check them, thereby overlooking the need to perform important updates, which exposes the user to security risks. Furthermore, technical diversity in products combined with immature standards complicates the task of tracking, securing, and maintaining IoT products in the home.

To complicate things further, rapid change in the device vendor space means manufacturers may update a product only for a few years. This directly conflicts with the fact that many smart home appliances (e.g., refrigerators) have lifetimes of 10 to 15 years [26]. This creates a case where connected

devices may go many years without a software update. This is similar to having an unpatched Windows XP service pack 1 device on a current home network today, thereby exposing the network to all of its vulnerabilities.

Compared to the relatively few general-purpose computers (e.g., PCs, smart phones, or tablets) most users have to administer in their homes, there could be several dozen IoT devices in a connected home. A study of such a home conducted in 2017 showed the average consumer has 13 connected devices [27]. With the adoption rate of IoT steadily increasing, that number will continue to rise [28]. With all of these IoT devices, the task of keeping a growing IoT network secure places an unrealistic burden on the average consumer [29]. In response, the task of protecting IoT-connected domains and applications will need to be automated. Therefore, full autonomy likely is not achievable for securing the totality of IoT. We believe novel methods will increase the level of autonomy in applications to provide analytically informed insights and recommendations, and will support homeowners in their efforts to be more efficient and effective in providing assurance for their IoT devices.

To address these security issues that, to our knowledge, are not currently being fully dealt with by a single solution, we propose a framework and solution that combines a novel application of standards-based detection, behavioral-based detection, verification operation of IoT devices, and application of active security measures that address fundamental shortcomings in securing IoT devices in the connected home. This approach and capability will help users understand the IoT devices on their networks and to maintain the security of those devices. Although this system is intended for the home, the system would also be useful to organizations with large numbers of IoT devices.

Our contributions are as follows: (1) we identify a gap in the IoT home security space and propose a system to fill this gap, (2) we experimentally detail the ensemble machine learning model aspect of our system, and (3) we introduce early results from the IoT discovery aspect of our system.

The remainder of this chapter is organized as follows. Section 2.2 introduces and motivates the CHASM system. Section 2.3I introduces IoT discovery and classification efforts for CHASM and discusses related works. Section 2.4 presents our experimentation approaches by describing our data collection, feature engineering, and model development. Section 2.5 describes our results. Lastly, Section 2.6 summarizes the paper and discusses future work.

2.2 CHASM

CHASM is a proposed IoT security system that aims to reduce the cognitive load on owners in managing and securing their connected homes' environment containing multiple devices, CHASM will maintain situational awareness of all of the IoT devices on a home network for the user, monitoring for vulnerabilities and detecting threats. Specifically, it will detect devices; monitor rogue, vulnerable, or compromised devices; and assist the user in managing the overall complexity of maintaining a safe and secure smart home environment addressing the key issues related to securing a connected environment [30].

The cornerstone capability of this system would be to perform automated device discovery, identification, and classification [31] [32] [33] [34]. Through being able to specifically identify and characterize all devices on a network, the user would automatically have a list of every IoT device on their network, and thus gain insights on what should be addressed on their network from a security point of view.

However, situational awareness of connected devices is not adequate to protect home users [35]. To protect a smart home, one needs to have insights into many other aspects of connected devices. Examples include how the devices are operating, whether a device is compromised and operating outside its normal behavior, and whether the device is properly configured and protected via patches and other updates. CHASM also will seek to inform smart home users of the current level of support provided for their device from the manufacturer.

2.2.1 CHASM Capabilities and Requirements

To meet the needs previously outlined, CHASM will provide the following functional capabilities for connected home users to better protect themselves:

- Perform automated IoT device discovery.
- Keep a tally of devices connected to a network (based on devices discovered).
  - Warn when new devices are connected.
  - Perform IoT device profiling and verification.
- Verify devices are operating as they should.
  - Detect for attacks against devices.
  - Determine whether a device is operating outside its intended range.
- Help users keep their devices properly patched and secured.
  - Provide device and device software patch support.
  - Notify users when the last update was made available.
  - Warn users of possible unsupported devices based on changes in the manufacturer or the company, or for issues specific to a device (e.g., no longer supported with patches).
  - Warn users if there are known vulnerabilities in a device.
  - Recommend and remind users to change default passwords and show the last time they were changed.

Considering CHASM's target capabilities and long-term goal to run on the edge in a user's environment, requirements for machine-learning algorithms deployed to the system are as follows:

- They can be deployed on a lightweight, affordable processing platform.
- They do not require total session knowledge for classifications.
- They strive to use the most computationally efficient method to meet a goal.
- They create portable models, selecting features that work across multiple different networks.

- They can provide insights on devices for which they were not trained. If CHASM cannot classify a specific device, it will classify the type of device through its characteristics and relationships to other similar devices.

2.3 IoT Device Discovery and Classification

This chapter describes initial work done to develop the device discovery and classification capability of CHASM. Future work will explore anomaly detection and patch support capabilities. We focus initial attention to the IoT discovery processes because we believe features important for IoT device discovery will be reusable for device profiling and threat analysis model development tasks. Furthermore, we believe IoT discovery is foundational because a user needs to have a good accounting and understanding of the devices on a network to be able to protect and secure them.

Although there are many ways to perform IoT discovery, each has its strengths and limitations. For example, with Media Access Control (MAC) addresses, one can determine the device manufacturer. This approach is relatively straightforward, but it also can be spoofed or be representative of another device in the network chain (e.g., a router or a switch) [36]. Another method to perform direct IoT device discovery is to investigate full TCP sessions; however, this requires waiting until the end of a session to extract the necessary features. In addition, for some IoT devices, the TCP sessions can last for days [31].

For this reason, the focus of this work is to explore discovery, profiling, and verification of IoT devices solely based on their network behavior or other information contained in individual or constrained groups of packets. The goal is to combine aspects from multiple IoT discovery approaches in an ensemble analytic that fuses insights from each approach.

Outlined next are relevant related efforts from the literature in IoT discovery and classifications that were used for motivations in our work.

2.3.1 Related Work in Behavioral-based IoT Device Discovery

Zhang et al. [32] used passive traffic measures for their research centered on the Samsung SmartThings platform. The authors evaluated 181 SmartThings devices by leveraging side-channel inference capabilities to design and develop a system to monitor SmartApps from encrypted wireless traffic.

Luo et al. [33] developed an automated system that can characterize an IoT device and generate an automated profile. They used a set of observable traits that comprise an IoT device, including protocols being used, data transfer rate, heartbeat frequency, upload and download rate, and the number of global or public traffic packets.

Shahid et al. [31] present a machine-learning–based approach to recognize the type of IoT devices connected to the network by analyzing streams of packets sent and received. Their results are promising, with an overall accuracy as high as 99.9% on their test set achieved by a random forest classifier.

Meidan et al. [34] used random forest applied to features extracted from network traffic data with the goal of accurately identifying IoT device types from the white list. IoT devices were correctly detected as unknown in 96% of test cases (on average), and white-listed device types were correctly classified by their actual types in 99% of cases.

Leung et al. [37] used protocol-independent network flows characteristics to compare a number of IoT discovery research efforts, summarized the results, and created an ensemble algorithm that leveraged the features invented by previous efforts from Shen et al. [38] and Pego and Nunes [36]. They then combined them using a random forest classification algorithm.

The results of these related works and others indicate there is potential for a lightweight security application that runs in the background and is capable of performing automated device identification. Using a similar approach to the one proposed by Meidan et al. and Leung et al., we extend their work on

behavioral-based device discovery to include behavioral-based verification of devices to ensure they are operating normally [39] and behavioral-based detection of attacks against the device [40].

2.4 Experimentation

2.4.1 Data Collection

Our data set consisted of 2 months of network data collected from 16 IoT devices on an Internet-connected network. The devices were part of a larger collection of 120 connected devices including tablets, smart phones, and other IoT devices (e.g., thermostats, smart speakers, and smart outlets). The devices were configured and operated normally by users during the 2-month period. The network data were captured using WireShark on a PC on the IoT network. The resulting captured data were segmented into 1-hour blocks of packet capture (PCAP) data and saved to a file. Each 1-hour block contained heterogeneous captures of the various IoT devices performing their normal operations. The collection of 1-hour PCAP files was then stored in the internal network to support machine-learning exploration. Figure 2-1 shows the architecture for our testbed setup. Figure 2-2 shows the devices and their data activity patterns. Table 2-1 lists the specific devices under test.

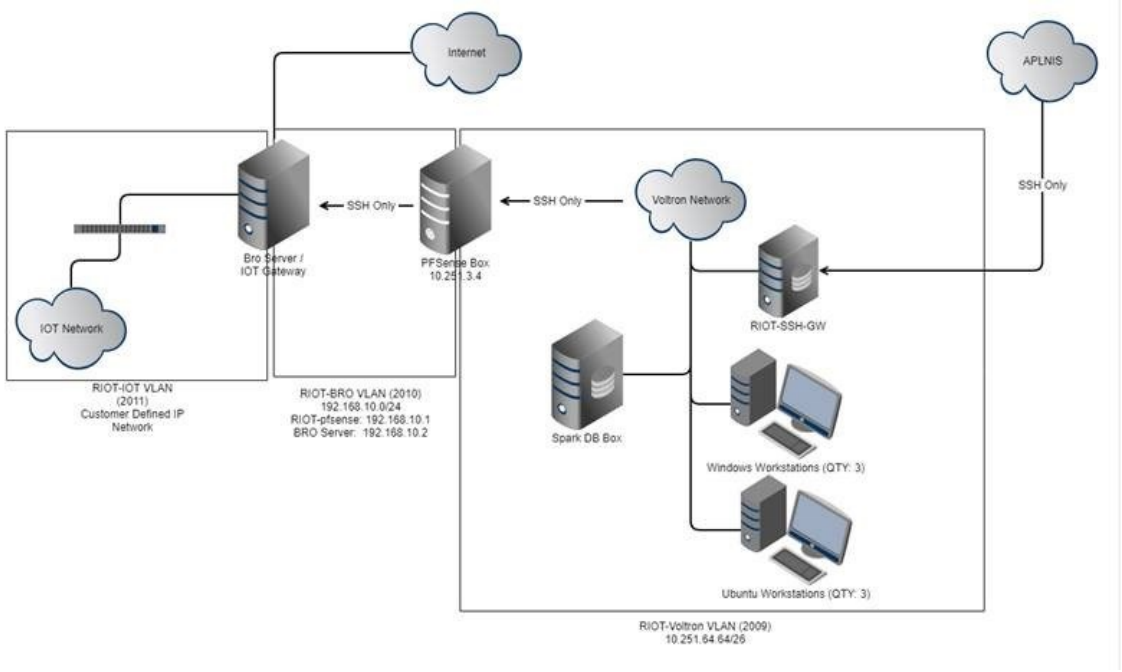

Figure 2-1 IoT Testbed High-Level Architecture

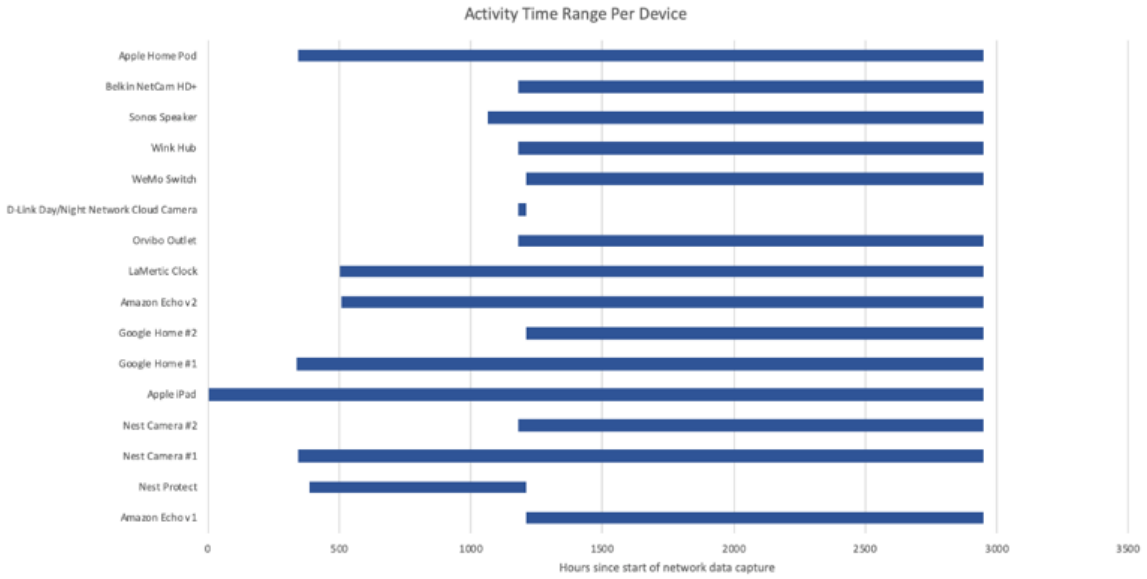

Figure 2-2 Select Device Activity Time Range per Device

Table 2-1 IoT Devices Under Test

Device ID	Device
0c:47:c9:10:d4:6f	Amazon Echo v1
18:b4:30:1f:17:19	Nest Protect
18:b4:30:e4:83:a2	Nest Camera #1
18:b4:30:e4:c6:2d	Nest Camera #2
2c:61:f6:75:0a:72	Apple iPad
48:d6:d5:79:37:85	Google Home #1
48:d6:d5:98:71:cb	Google Home #2
4c:ef:c0:07:94:8e	Amazon Echo v2
54:c9:df:8e:88:3f	LaMertic Clock
ac:cf:23:65:e5:e4	Orvibo Outlet
b0:c5:54:0f:4a:44	D-Link Day/Night Cloud Camera
b4:75:0e:0c:a9:61	WeMo Switch
b4:79:a7:24:38:1a	Wink Hub
b8:e9:37:57:30:be	Sonos Speaker
c0:56:27:54:28:91	Belkin NetCam HD+
d4:90:9c:d5:e1:48	Apple Home Pod
0c:47:c9:10:d4:6f	Amazon Echo v1

2.4.2 Feature Selection and Engineering

Through examining the related works, we identified a combined list of potential single-packet, nonweighted features and multipacket features for machine-learning algorithm development for IoT. We considered a number of factors in identifying potential features to extract from each packet in the PCAP data. Table 2-2, which references Bezawada et al. [41] and Miettinen et al. [42], outlines the single-packet feature set.

Meidan et al. [34] followed each TCP session from each device from successful SYN→FIN, and then extracted more than 300 features from each session. The following are the most important and useful of these features:

- Time-to-live for TCP packets sent from device to server: minimum, maximum, average, first quartile, and third quartile
- Total number of packets in the session that contain the Reset flag

- Ratio between number of bytes sent from device and received by device

Shahid et al. [31] grouped TCP flows by unique combinations of source and destination IP and port. For each conversation, they extracted the following:

- The size of the first N packets sent
- The size of the first N packets received
- The N – 1 packet inter-arrival times between the first N packets sent
- The N – 1 packet inter-arrival times between the first N packets received

Using combinations of these techniques, along with analytic insights motivated from Buczak et al., we enhance and extend the multipacket features by applying the following statistical transforms: average, variance, skewness, and kurtosis [43].

2.4.3 Model Selection and Model Development

Leveraging the Anaconda Data science distribution [44], we used Python pandas in a Jupyter notebook to form test data sets that comprised a sub-collection of the IoT devices. Ground truth was already established for the data set in that we knew a priori the mapping between a specific IoT device and its associated MAC address and associated IP addresses. We created models using Python Scikit-learn libraries and applied our data to these models.

2.4.3.1 Multiclass Decision Forest Details

The first machine learning model explored was a multiclass decision forest [45] classifier. This machine learning method uses decision trees and bagging to predict a target that has multiple values. The random forest [46] is composed of multiple decision trees, and the final answer is decided by majority voting based on the predictions. This produces models that are robust despite noise and have error rates that are similar to neural networks. We chose this model based on its proven performance exhibited in the related works. We trained the multiclass decision forest classifier to distinguish the type of IoT device presented to the classifier.

2.4.3.2 Multiclass Neural Network Details

The second machine learning model explored was a multiclass neural network [47]. This is a variation of a Deep Neural Network that can predict a single target from a collection of multiple candidates. A neural network is a set of interconnected layers, where the inputs are the first layer and are connected to an output layer by an acyclic graph composed of weighted edges and nodes. Between the input and output layers are hidden layers. The number of hidden layers can be configured to optimize performance based on the type of classification being performed. In our experiments, we created a single hidden layer where the output layer was fully connected to the input layers, and the number of nodes in the hidden layer was set to 100. We selected this classifier to attempt to leverage the time series nature of the IoT data.

2.5 Results

The initial machine-learning model results were varied for our initial data set.

2.5.1 Multiclass Decision Forest

Using the single-packet features, the multiclass decision forest posted an overall accuracy of over 98% and an average accuracy of 99%. Comparing these results, this model matched or bested previous efforts cited in related works, and it verifies that the decision forest is a good choice and performs well for the purpose of IoT discovery. The total model results are shown in Table 2-3. The top six features (those with information gain values above 0.01) for the multiclass decision forest are tcp_payload_entropy, tcp_src_port, tcp_window_size, ip_ttl, and TLSv1.2, and tcp_len (Figure 2-3).

Table 2-2 Select Features Used for Device Discovery and Classification – Machine Learning Model Development

Name	Example and Description	Reference
Network Layer Protocol	IP, ICMP, ICMPv6, EAPoL	[41]
Transport Layer Protocol	TCP, UDP, etc.	[41]
Application Layer Protocol	HTTP, HTTPS, DHCP, BOOTP, SSDP, DNS, MDNS, NTP, etc.	[41]
Internet Protocol (IP) options	Padding, RouterAlert	[41]
Packet Length	Integer number	[48]

Name	Example and Description	Reference
TCP: source port class	An integer from 0 to 3 no port ⇒ f = 0 well-known port [0, 1023] ⇒ f = 1 registered port [1024, 49151] ⇒ f = 2 dynamic port [49152, 65535] ⇒ f = 3	[42
TCP: destination port class	Same as source port class	[42]
TCP: payload length	Integer number	[41]
TCP: Shannon Entropy of payload	Used to determine whether a file is encrypted	[41]
TCP: window size	Length of the TCP receive window	[36]
UDP: source port class	Similar to TPC source port class for User Datagram Protocol (UDP)	[34]
UDP: payload length	Length of the UDP payload	[34]]
UDP: Shannon Entropy of payload	Used to determine whether a file is encrypted	[32]
Timestamp	Unix epoch time (in milliseconds)	[37]
Aggregated into sequences of five packets	For each five-packet sequence, all single-packet features were concatenated and scored.	[48]]
Aggregated packets into flow summaries	For each flow summary, extracted: sleep time, active volume, average packet size, mean rate, peak-to-mean ratio, active time, number of servers, number of protocols, unique DNS requests, DNS interval, NTP interval, and most frequent port number.	[38]

Table 2-3 Multiclass Decision Forest Results

Overall accuracy	0.982324
Average accuracy	0.991162
Micro-averaged precision	0.982324
Macro-averaged precision	0.979172
Micro-averaged recall	0.982324
Macro-averaged recall	0.979421

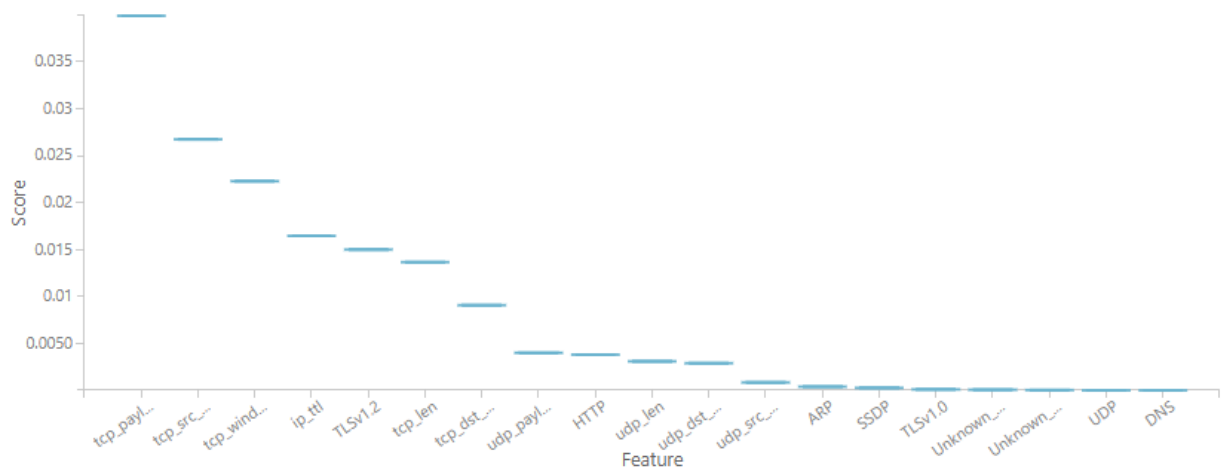

Figure 2-3 Information Gain for Non-Zero Features Multiclass Decision Forest

2.5.2 Multiclass Neural Network

The multiclass neural network had an overall accuracy of 83% and an average accuracy of 91%. Analyzing these results, one sees there was a great deal of variability in this model's performance. Specifically, the model precision (i.e., percentage of results that are relevant) and the model recall (i.e., percentage of total relevant results correctly classified by the algorithm) dropped to 83%, as shown in Table 2-4. The top seven features (those with information gain values above 0.01) for the multiclass neural network are TLSv1.0, Unknown TCP, tcp_payload_entropy, tcp_src_port, ip_ttl, UDP, and TLSv1.2 (Figure 2-4)

Table 2-4 Multiclass Neural Network Results

Overall accuracy	0.832727
Average accuracy	0.916363
Micro-averaged precision	0.832727
Macro-averaged precision	0.862141
Micro-averaged recall	0.832727
Macro-averaged recall	0.78037

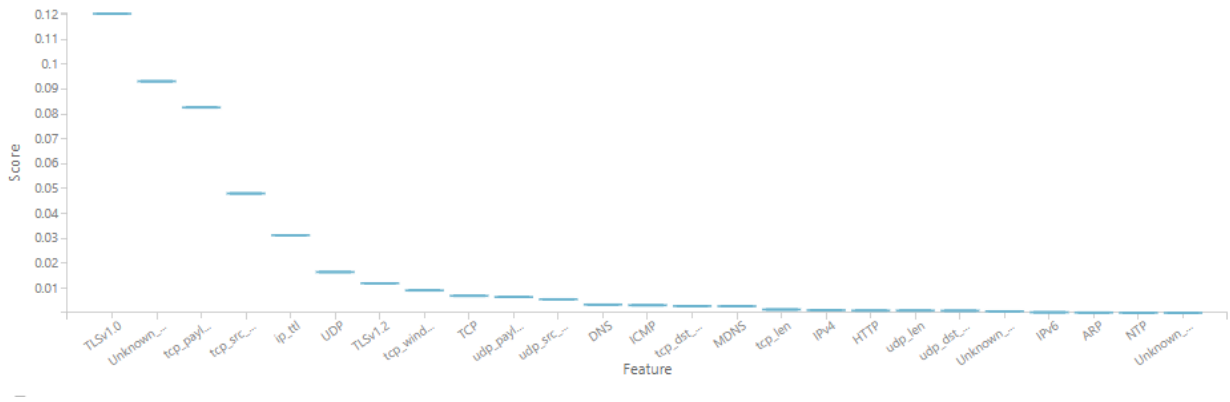

Figure 2-4 Information Gain for Non-Zero Features Multiclass Neural Network

Another source of variability in the results was due to the wide variety in how IoT traffic presents itself based on operating modes. IoT traffic has significantly different characteristics in an active state versus a quiescent state. The takeaway is that as we train machine-learning models on these data, one will need to take into account the variable nature of IoT traffic and train over a broader temporal collection of data that includes all of the operating modes of the device (Figure 2-2, Figure 2-5). We will explore this in future work.

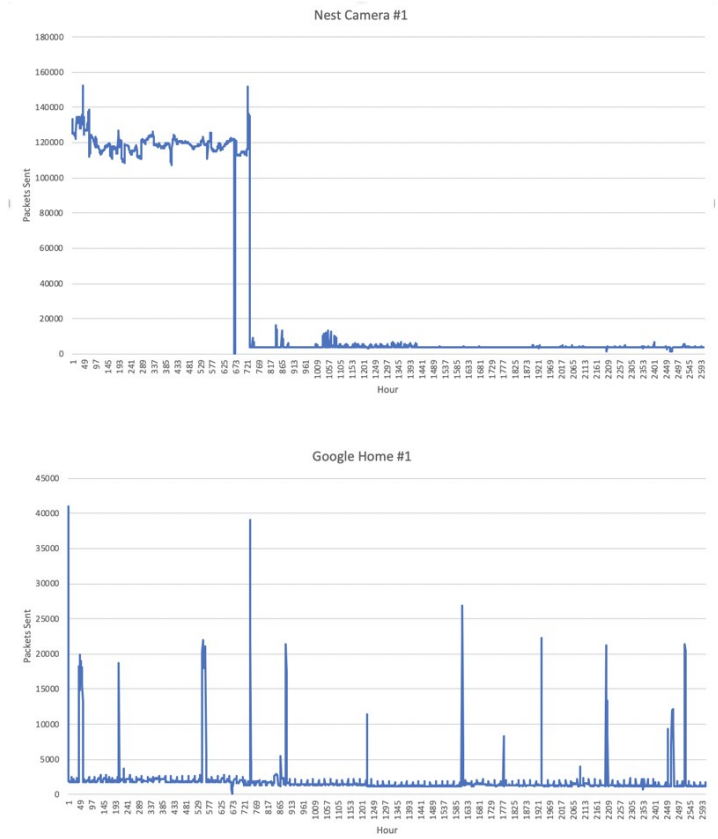

Figure 2-5 IoT Data Characterization Variance (Active vs. Quiescent State)

2.5.3 Ensemble Analytics

Using a collection of different discovery approaches, we created an ensemble analytic that combines multiple discovery analytics into a single score. The ensemble analytic (Figure 2-6) includes:

- Translating device MAC address to manufacturer name
- Quick category classifier (determine IoT device category after 20 packets)
- Slower category classifier (determine category after 10K packets)
- Device classifier (identify specific types devices)

Each approach for discovery provided a view or perspective into the device that can be combined to improve the overall confidence of individual model results. Furthermore, any impedance mismatch or

changes between modes can be used as a general anomy detector for devices. We will develop and explore this concept for improved device detection, classification, and anomaly detection in future work.

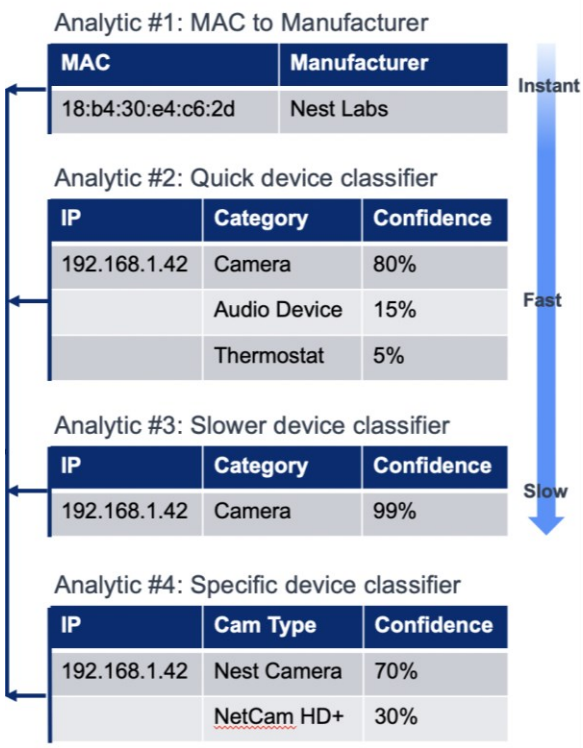

Figure 2-6 Example of IoT Discovery Ensemble Analytic

2.6 Future Work

Future work will further define the CHASM concept and build a reference implementation. We will develop machine-learning models to characterize normal behaviors and detect abnormal behavior for identified IoT devices. We will design an alerting capability that mines open-source data to assist users in keeping their devices properly patched and secured and warn them of issues specific to a device, or which devices are now possibly unsupported based on changes in the manufacturer or the company. Additionally we will explore leveraging a larger set of the features and explore ensembles of individual machine-learning models to try to improve the performance on the IoT discovery models.

In conducting this research, we have captured and characterized over 250 GB of IoT data containing dozens of different devices. We believe this dataset would be of value to the broader academic

community and plan its release to support further research by the academic community at the completion of our research.

Beyond its value for the connected home, we believe CHASM could have direct application to large constantly evolving IoT environments such as in clinical and hospital settings where devices are added and removed from the network frequently, providing entire classes of devices that could be added to the network and forgotten. CHASM could provide monitoring and security capabilities for hospital information technology (IT) administrators to better protect their networks, patients, and data. We also believe CHASM could have utility in industrial control settings, where administrators need to maintain strict control over what devices are connected to their networks. We are actively researching how CHASM can be tailored and tuned for these environments.

3 A CAPABILITY FOR AUTONOMOUS IOT SYSTEM SECURITY: PUSHING IOT ASSURANCE TO THE EDGE

3.1 Introduction

Complex SIoTD are systems that have a single purpose, but are made up of multiple IoT devices. These systems are generally comprised of many individual IoT devices working together to form a larger, cohesive system function. SIoTDs are becoming commonplace as IoT devices proliferate into our society (e.g., our homes, smart cars, smart offices, smart buildings, smart hospitals, and smart cities). These systems often have complex security requirements and face a diverse and ever-changing array of cyber threats. Devices may come and go from these systems at any time, and some systems may only be intermittently connected to central processing hubs (clouds). An impending increase in data transmission rates (5G) and constantly changing network footprints will further complicate the network defender's job of maintaining situational awareness and providing security for these systems.

In addition to these challenges, many SIoTD networks are segmented, meaning that the data from all devices might not be sent back to a single place. Thus, these systems cannot rely on a centralized cloud-based service for their security. Rather, protection for these systems must occur locally and in an autonomous or semi-autonomous fashion.

To obtain a complete picture of an SIoTD, and to meet the security and related analytic requirements, processing will need to occur close to the point of data generation. This region is known as the edge. (Figure 3-1, derived from [49]).

There are three data-focused use cases related to SIoTD that motivate pushing IoT situational awareness and system security to the edge:

IoT applications will generate too much localized data, making it impractical to ship all data back to a central location.

Data privacy concerns from the system owner will restrict sensitive data from being stored in a public cloud. System owners will want to maintain control of their personal, valuable data.

Many IoT use cases have stringent timing requirements that cannot tolerate extensive latencies based on round trip timing from the point of data generation to the cloud and back. These cases require data to be processed and intelligence generated locally.

This chapter describes ongoing work exploring novel techniques to build a capability providing real-time, edge-centric, automated situational awareness and assurance to IoT networks.

Our contributions are as follows: (1) we identify a gap in the security of SIoTD and propose a solution to fill this gap, (2) we experimentally detail the machine learning (ML) pipeline and corresponding model development, and (3) we introduce results of porting the assurance algorithms from a server to a low size, weight, and power (SWaP) device running autonomously at the edge collocated at the point of data generation.

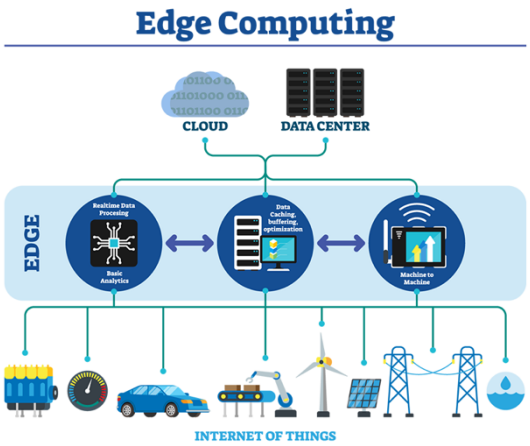

Figure 3-1 IoT Edge Computing

The remainder of the chapter is organized as follows. Section 3.1.1 provides an overview of the automation types guiding this capability. Section 3.1.2 discusses related works. Section 3.1.3 outlines the IoT assurance architecture. Section 3.1.4 describes the feature selection, engineering, and generation activities. Section 3.1.5 presents the IoT data pipeline and the associated design. Section 3.1.6 presents an overview of model selection and training. Section 3.1.7 describes the results. Lastly, Section 3.1.8 presents the conclusions and discusses future work.

3.1.1 Automation Types

Artificial intelligence (AI) initiatives can be broadly categorized according to their goals [50]. Two of the most common are:

- Process automation
  - Automation of digital and physical tasks using "robotic (physical or virtual via computer code)" process automation technologies.
- Cognitive insight
  - Using algorithms to detect patterns in vast volumes of data and interpret their meaning

To meet the needs of this system, we will employ elements of both process automation and cognitive insights. This capability will operate in a semi-autonomous fashion, acting as an agent or virtual robot and is made up of a collection of related hardware, frameworks, algorithms and edge devices. This capability will essentially be an autonomous realization of CHASM for SIoTD [1].

3.1.2 Related Works

A literature survey was conducted to gather a list of relevant features and techniques that might be useful for fingerprinting/profiling IoT devices.

Meidan et al. [51] extracted features from TCP sessions and trained a multiclass random forest classifier to identify IoT devices from a white list and distinguish between known and unknown devices.

Shahid et al. [52] presented an ML-based approach to identify IoT devices by analyzing streams of packets sent and received with an overall accuracy as high as 99.9% on their tests et using random forest classifiers.

Truong et al. [53] used a recurrent neural network to identify and classify IoT devices in network traffic.

Buczak et al. [43] used random forest augmented with the statistical moment of average, variance, skewness, and kurtosis to develop an ensemble algorithm to identify the presence of covert DNS tunnels in network traffic.

Sivanathan et al. [54] used random forest in a smart-campus environment instrumented with a diversity of IoT devices over a 3-week period to distinguish IoT devices from non-IoT devices and to classify IoT device with 95% accuracy.

Bai et al. [55] employed a long short-term memory (LSTM) convolutional neural network (CNN) cascade model to automatically identify the semantic type of a device to achieve automatic device classification in network traffic streams of IoT devices.

Bezawada et al. [56] extracted a wide range of features from packet headers to build device fingerprint vectors, and used Gradient Boosting classify IoT devices.

The features selected for implementation are listed in Table 3-1.

3.1.3 IoT Assurance Architecture

Our goal in this effort is to train models that can autonomously identify IoT device types. We want those models to perform well in both a controlled laboratory setting *and* the real world. Therefore, it is essential that we train the models on datasets that include large numbers of IoT devices with large amounts of traffic from each. It is assumed that in the case of device classification (i.e., guessing what category a device belongs to), it would also be best to have many different devices in each category so that the model does not overfit to the behavior of a few devices.

An IoT testbed was assembled containing 56 IoT devices of various types. The IoT devices were installed on a network demilitarized zone (DMZ), also referred to as an isolated perimeter network, that had access to the Internet. This is because the devices were untrusted but needed Internet access to function properly. We used a span port on a switch to send all network traffic to a Jetson Nano, which ran

tcpdump and archived the data in PCAP format. That data was then transferred to a local, on-premises "cloud" server outside the DMZ to be used as training data for the models.

Table 3-1 Features Used for Fingerprinting and Profiling IoT Devices

Feature Name	Description	Reference
Packet Length	Integer, range 0 to 65535, but typically 0 to 1500	
Ethernet Protocol	IPv4/IPv6/ARP/LLDP/EAPoL/Unknown	[56]
IP Protocol	ICMP/TCP/UDP/ICMPv6/Unknown	[56]
IP Time-to-Live (TTL)	Integer, range 0 to 255	
TCP Application Protocol	HTTP/SSHv2/SSLv3.0/TLSv1.0/TLSv1.1/TLSv1.2/Unknown	[56]
UDP Application Protocol	SSDP/DNS/MDNS/NTP/DHCP/ISAKMP/Unknown	[56]
TCP/UDP Source Port Class	Each value has its own column, with a 0 or 1 value SYSTEM= port 1-1023 USER= port 1024-49151 DYNAMIC= port 49152-65535 UNKNOWN= port > 65535	[57]
TCP/UDP Destination Port Class	Each value has its own column, with a 0 or 1 value SYSTEM= port 1-1023 USER= port 1024-49151 DYNAMIC= port 49152-65535 UNKNOWN= port > 65535	[57]
TCP: payload length	Integer, range 0 to 65535	[56]
TCP: Shannon Entropy of payload	Float, range 0.0 to 8.0	[56]
TCP: window size	Integer, range 0 to 65535	[56]
UDP: payload length	Integer, range 0 to 65535	
UDP: Shannon Entropy of payload	Float, range 0.0 to 8.0	
From Internet	Integer, 0 or 1 Set to 1 if the packet is coming from a public IP address	
To Internet	Integer, 0 or 1 Set to 1 if the packet is being sent to a public IP address	

3.1.4 Feature Selection, Engineering and Generation

When determining what features to extract from network packet data, it is important to keep in mind that many IoT devices are already using encryption methods such as Transport Layer Security (TLS) to secure their traffic against eavesdropping and that more devices will be using this type of security in the future. Therefore, it would be naïve to assume that features extracted from unencrypted packet payload data will remain useful in the future. Rather, the selected features should be those that can be easily

extracted from both encrypted and non-encrypted traffic. For example, features extracted from packet headers such as packet length, IP time-to-live, TCP window size, and payload entropy can be extracted from packets regardless of whether the payload is encrypted or not.

Furthermore, any features used for IoT discovery might be useful for IoT profiling, threat detection, and anomaly detection as well (future work).

3.1.5 Data Pipeline Design

A successful edge deployment involves more than simply pushing a trained model out to an edge device to be executed. The feature extraction code and other data processing logic that are part of the associated analytic pipeline must also be ported to run at the edge. This is because those processing components transform the raw sensor data into a form that is ready for ML processing. Thus, any feature extraction or data processing code that is part of the analytics pipeline must be designed in such a way that it can either be massively parallelized in a central data center for training purposes or pushed to the edge to run on a less capable computing platform. It is an added productivity bonus if the code can run in the same form on both types of platforms without modification, thus streamlining the deployment process.

With this in mind, the feature extraction and processing pipeline for this system was designed to be portable from the ground up (Figure 3-2), making the deployment of the complete analytics processing pipeline to the edge straightforward. We implemented the feature extraction in C++ on top of the open-source *libtins* library [58], which provides efficient methods for collecting and parsing network packet data. Higher layers of the software that aggregate the packet data by device and transform the data into vectors that are ready for ML were written in Python using a combination of *scikit-learn*, *pandas*, and custom Python code. Keras and TensorFlow were used for model training and inference.

These software modules were combined into a reusable Python package named *iot-ai*, which allows the user to create modular data pipelines for subsequent ML development. Since model training is a batch

operation and model execution on a real-world network is ideally a streaming operation, the *iot-ai* library was designed to be a streaming capability. Batch operations are handled as streaming ones by streaming individual lines of a static file through the pipeline. This limits the system complexity by unifying two very similar processing pathways and code bases into one. With this design, no code conversion is needed when porting a feature extraction pipeline from batch mode in the cloud (for training) to streaming mode at the edge (for live inference).

The nodes in the pipeline have flexible interfaces and can be combined in different ways to facilitate the many operations typically involved in training an ML model on network data. Furthermore, individual pipeline nodes can be parallelized to take advantage of multi-core CPUs when available.

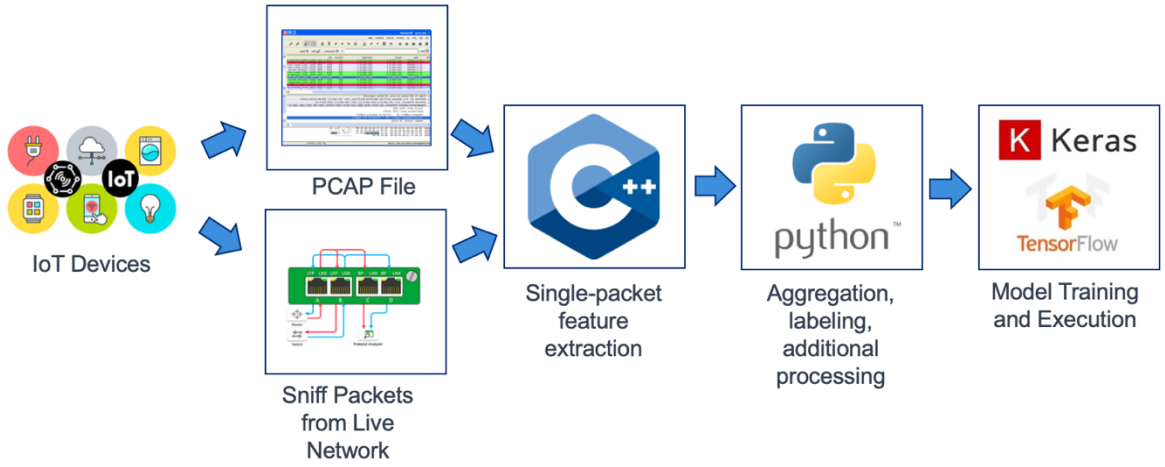

Figure 3-2 IoT Feature Extraction and Processing Pipeline

Additionally, the *libtins* library provides the option of reading packets from a PCAP file or reading the packets directly from a network interface, which means our pipeline can be easily transitioned from processing static data to running on live data.

The result is a modular pipeline design that can be easily reused for data collection at the edge, training in the cloud, execution of models in the cloud, and execution of models at the edge.

3.1.6 Model Selection and Training

The chosen model architecture was a recurrent neural network consisting of a single LSTM single layer of 128 units, followed by softmax output of width 10 (one for each category of device). The LSTM layer was fed with fixed-length input sequences of 20 feature vectors, one vector per network packet.

The model was trained on PCAP data that was collected from the IoT testbed over the course of 3 months (Figure 3-3). This included data from 56 different IoT devices and amounted to 94 GB of PCAP (about 262M packets). This dataset was then split into 80% for training, 19% for validation during training, and a 1% holdout for final evaluation of model performance.

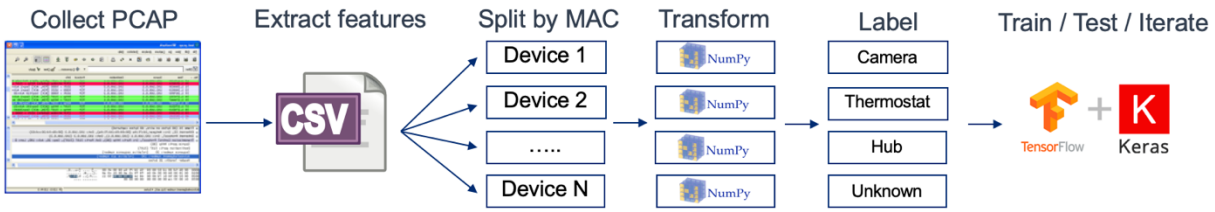

Figure 3-3 Collection, Extraction, Training, and Testing Pipeline

Traffic from each IoT device was grouped by MAC address and arranged in time order. The data from each device was then transformed into sequences of 20 packets each. The values in these sequences were then normalized, one-hot-encoded and labeled according to device category (e.g., this 20-packet sequence is from a camera, this other 20-packet sequence is from a thermostat, etc.).

3.1.7 Results

The initial ML model results were varied for our initial data set. However, the observed variation was in execution speed, not accuracy.

3.1.7.1 Model Execution and Results (On the Server)

The model was trained successfully on a single Nvidia Tesla V100 GPU in under 1 hour, and achieved 93% accuracy on each of the training, validation, and holdout datasets. The model's detailed performance results on the holdout set for each category are presented in Table 3-2.

Table 3-2 Model Performance Results on Holdout Set per Category

Category	Precision	Recall	F1-Score	Support
Unknown	0.98	0.95	0.96	10893
assistant	0.98	0.85	0.91	19765
audio device	0.95	0.93	0.94	1307
clocks	1.00	0.81	0.89	3437
hub	0.76	1.00	0.86	22254
mobile device	0.78	0.66	0.71	397
router	1.00	0.89	0.94	17382
television	0.98	0.96	0.97	44895
thermostat	1.00	1.00	1.00	5241
triggers and switches	1.00	0.97	0.98	5463

At the time of this writing, the amount of traffic being generated by the IoT devices in the testbed is relatively small (20 MB every 10 minutes). With a parallelization level of 3×, the end-to-end data pipeline including the trained model can process the aforementioned 10 minutes of network data in 47 seconds on a large virtual machine (VM) server with 30 Intel(R) Xeon(R) CPUs at 2.50 GHz and 512 GB of random access memory (RAM). In this configuration, the processing pipeline consumed only 6 CPUs and 3 GB of RAM.

3.1.7.2 Port to Low-SWaP Devices

To execute the model at the edge, a Raspberry Pi 3B was obtained, having a four-core, 1.2-GHz Broadcom, 64-bit ARMv7 CPU and 1 GB of RAM. Tensorflow2 was installed, followed by the iot-ai library and its dependencies (*libtins*, *libpcap*, etc.). This device was then connected to the same network as the IoT devices, and a span port was set up on the network switch to forward all the network traffic from each device to the Raspberry Pi (Figure 3-4).

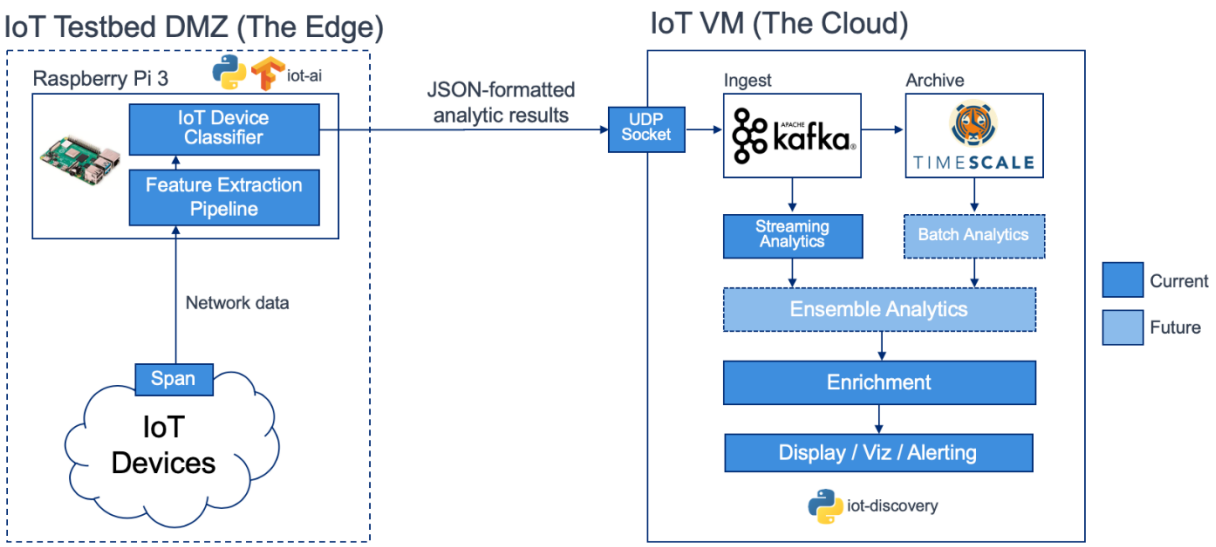

Figure 3-4 Cloud-to-Edge Analytic Extraction Configuration

3.1.7.3 Results of Model at Edge

On the edge platform, the data pipeline was able to process 10 minutes of testbed traffic in 6 minutes. While this is encouraging and indicates that the edge device is currently capable of keeping up with the live stream of testbed data, this data volume is still somewhat small.

The speed of pipeline execution at the edge could likely be improved by running the model on an ML-focused edge device such as an Nvidia Jetson TX2 or by converting the model to TensorFlow Lite, which Google claims will soon offer support for LSTM networks with fixed sequence lengths [59].

3.1.8 Conclusions and Future Work

These results indicate that it is possible to run IoT device discovery, classification, and verification models at the edge on low-SWaP devices; however, we feel there are areas of improvement that can be explored to increase the performance of the system.

3.1.8.1 Model Tuning

We did not perform extensive hyper-parameter tuning, actively exercise the individual IoT devices to collect traffic during all phases of device operation, nor normalize the number of training examples from each device category. This could be explored in future work.

3.1.8.2 Complex Network Topologies

The network topology used in this effort was a relatively straightforward one; all devices were connected to the same router, and the network traffic was captured at that router. In future work, we intend to apply the concepts and systems developed in this effort to more complex network topologies.

3.1.8.3 Model Training at the Edge

We believe that, over time, advances in low-SWaP devices will allow for model training at the edge, running alongside model execution. We will explore this in future work.

3.1.8.4 Analytic Ensembles

The IoT landscape is complex and evolving; new devices are appearing all the time. Therefore, training a single model for IoT discovery would be problematic, as there would be a constant need to retrain that model as new types of devices are produced (often).

Two additional challenges for the single-model approach are the inherent trade-offs between generating results quickly and observing devices long enough to make good predictions, and the fact that individual models may be biased, brittle, or prone to error in certain situations.

Combining multiple models into an ensemble may help in this regard. If some models are noisy or inaccurate due to a rapidly changing device landscape, they can be balanced out by others. For example, if four models say a given device is a camera and one says something else, it is probably a camera.

Using an ensemble approach would also allow some models to focus on quick but less accurate predictions and others on much slower but more accurate predictions. This way, the user would receive a notification almost immediately, and the result would be refined over time. Allowing the user to drill down and see the results from individual models would provide an additional degree of explanation in these situations.

Other findings of this work include the following:

- The same analytics that were designed for IoT Device Discovery also could be used for IoT Device Verification and Threat Detection.
  - Example 1: All ensemble analytics agree that device X is a camera, but all of a sudden, half of them are now claiming it is a thermostat.
    - Perhaps the device has begun behaving differently (may indicate a potential compromise)
    - OR this indicates an issue with the models
  - Example 2: None of the ensemble analytics agree on device X.
    - Perhaps this is a new type of device not seen before, and further investigation will be needed

We believe these results indicate that a distributed, edge-based, semi-autonomous capability could be developed and deployed in complex, dynamic SIoTD environments, and the associated model may be able to maintain good performance, even if the model is presented with data on which is has not been trained. We will explore this in future work.

4 ENVISIONING CYBERSECURITY ANALYTICS FOR THE INTERNET OF THINGS

4.1 Introduction

The IoT is large; it is ubiquitous; and it is growing. Although it is likely not possible to place a definitive number on the size of the IoT, estimates of its current size range in the tens of billions, and it is expected to increase in size by an order of magnitude by the end of the next decade [16]. The IoT will be an integral part of every sector of the U.S. economy and will substantially influence daily activities ranging from healthcare to transportation to provision of basic services. “Smart” grids, with automated sensors and control systems to achieve improved efficiency, will be integrated into transportation and public utility systems. Industrial processes will become dependent on automation and robotics, affecting the manufacturing, chemical, and defense sectors. Financial and legal institutions housing vast amounts of sensitive data will be equipped with IoT-enabled sensors and controls, including security surveillance systems. Public health and emergency services will be connected to biometric and geolocation sensors to provide continuous monitoring of patient and first responder status.

However, the IoT is more than the connection of an increasing number of smart devices. It is a consequence and a driver of an evolution in computer and communications technology that will radically alter the way machines and humans communicate and interact. The IoT is enabled by technological innovation, and it is having a profound effect on the development of new capabilities. As envisioned, the emerging IoT would be impossible without increased bandwidth; new decentralized network topologies; compact, low-power “intelligent” devices that connect to networks; and the virtualization of networks and network functions. In turn, the demands of the IoT for increased scale and for customized applications with higher reliability and lower latency to support increasing levels of computation, automation, derived intelligence, and robotics is a driver of new technologies.

With the capability, scale, and complexity of the emerging IoT comes vulnerability to cyber-attack. As network traffic increases in both scale and complexity, cybersecurity analysts will require new tools to enable rapid interpretation and response to a new, expanding set of cybersecurity threats.

This chapter summarizes the results of an 8-month analysis of large-scale networks in support of the National Critical Function (NCF) set [60]. An analysis was performed to evaluate the impact of the IoT and associated technologies on existing cybersecurity analytics tools and capabilities. The goal of this analysis was to define a vision to guide the evolution of cybersecurity analytics, and to define a governmental role in promoting that vision. It was envisioned that government's primary focus would be on protecting public sector networks, but, as a result of the interconnectedness of the emerging IoT, there was also desire to define a role in promoting a more secure IoT for the private sector.

The contributions of this chapter are as follows: (1) we define an IoT taxonomy supporting the NCF; (2) we identify several trends that could create novel and/or heightened cybersecurity risk; (3) we define the role of government in developing IoT scale cybersecurity analytics; and (4) we propose recommendations based on key IoT cybersecurity vision elements that will guide organizations to realize the necessary level of capability to secure the IoT.

This remainder of this chapter is divided as follows. Section 4.2 describes the IoT, the enabling technologies that enable it, and the cybersecurity approaches affected by it. Section 4.3 lays out a vision for an emerging IoT cybersecurity analytics capability. Section 4.4 attempts to define a potential role for government in making this vision a reality. Lastly, Section 4.5 presents conclusions.

4.2 IoT Taxonomy

There is no consensus definition for what constitutes the IoT. Organizations including NIST, Institute of Electrical and Electronics Engineers (IEEE), and International Telecommunications Union have put forward their own definitions or descriptions that are both roughly the same and slightly different [10] [14] [15]. Nonetheless, there is agreement that the IoT consists of an increasingly large number of

machines exchanging data across interconnected networks. Connected machines provide varying levels of processing capacity (i.e., intelligence) and perform a wide range of tasks. Devices connected to the IoT can act as sensors and actuators supporting relatively simple automated tasks, or they can be complex intelligent agents functioning as part of autonomous or robotic systems. They can be connected to networks using IPs (i.e., devices connected directly to the Internet or via later-generation mobile networks), or they can connect via gateway technologies or other non-IP communications protocols.

IoT-enabling technologies can be divided into four categories [61]:

- Devices
- Networks and computing
- Communications
- Software

The past decade has seen a steady rise in the number of personal devices (e.g., phones, tablets, and laptops) connected to the Internet. Today, a majority of citizens in the United States owns a smart phone, providing them access to a camera, a capable processor, and the Internet via one readily portable, lightweight device. [62]. In addition, advances in processor manufacturing have enabled integration of small processors in a wide range of household devices. The ability to provide significant processing in small devices with limited energy consumption is leading to increasing automation of everything from everyday household appliances to advanced industrial control systems. Networks are populated by personal communication and computing devices as well as a broad range of smaller sensors, actuators, and control systems providing remote control over a variety of processes.

The rise in IoT has been abetted by and influenced an evolution in network architecture. Increased automation, autonomy, and robotics have led to more network traffic and a need for reduced latency and higher reliability. To reduce network traffic and latency, while increasing reliability, there has been a move from centralized “cloud” computing structures to more decentralized “fog” and “mist” computing

structures. Fog and mist computing enable critical computing capabilities, including analytics, to move toward the “edge” of the architecture to various locations capable of supporting analytics and levels of decision-making, as well as into the IoT devices themselves. Additionally, the demands of the varied IoT applications will in the future be met by enhanced use of virtualization. Virtualization, in the form of software-defined networks and virtualization of network functions, is already making networks more responsive and flexible. The advent of network virtualization will make it possible to operate networks within networks (i.e., network slicing). Virtual networks, all operating within a single physical space, will be capable of meeting the customized needs of specific applications or network segments.

A third area of technological change is communications. Just as the IoT is a product of increased communications connectivity and bandwidth, the envisioned future of the IoT will also depend on greater bandwidth and the ability of connected devices to operate with low power consumption. Implementation of the 5G telecommunications standard will enable wireless devices to operate at higher frequencies, where bandwidth is less constrained. In addition, the 5G standard will support direct device-to-device (D2D) communications connectivity, enabling devices to connect directly to other devices without the need to access intervening service provider infrastructure. Implementation of D2D will enable ad hoc networking for devices outside of a coverage envelope.

The IoT is being supported by availability of open-source software products designed specifically for IoT applications and the development of IoT-related standards. Industry working groups such as the Open Web Applications Security Project, IEEE, and the IETF have developed standards and open-source software capabilities to support independent development of IoT-related software [63] [64] [65]. The ZigBee Alliance, a consortium of industry organizations involved in the development of IoT capabilities, has recently established the Connected Home Over IP project to promote standardization [66].

Furthermore, there is a trend toward increasing levels of intelligence in networked devices and applications. Systems hosted across networks can display varying levels of automation, autonomy, or even

intelligence. In particular, systems employing machine learning—a technique in which machines are trained through controlled repetition to perform tasks in a process that mimics the way humans learn to do tasks through practice—are becoming relatively common. Machine learning can range from applications that employ statistical methods and feedback to more sophisticated applications (e.g., deep neural networks) that actually attempt to model human-learning processes. Although the use of sophisticated algorithms that learn can significantly improve processes and controls, the processing required to enable even relatively simple learning can be intensive. Thus, the drive for automation and autonomy within the IoT creates a need for more processing than can be supported within most devices, and subsequently encourages the use of more fog and mist computing structures.

Based on our survey of IoT-related technology, we identified several trends that could create novel and/or heightened cybersecurity risk:

- *Increased scale.* The increased scale of the IoT will expand the attack surface while stressing defenses. Cybersecurity analysts will have a larger quantity and variety of network traffic to monitor, because the IoT supports higher levels of human-to-human exchange of multimedia material via enhanced mobile broadband (eMBB) and increased levels of machine-to-machine (M2M) communications. Analytic tools will need to be both scalable and adaptable.
- *Evolution from cloud-computing to fog-computing structures.* Increased automation, autonomy, and robotics have contributed to an evolution from highly centralized cloud computing to more responsive, more distributed fog and mist computing. Analytic capabilities are hosted at points within the communications architecture, including within devices themselves when possible, closer to the network edge. By distributing analytics and other processing, latency and the volume of network traffic are reduced, whereas reliability increases. However, the evolution from centralized to decentralized architectures will demand increasingly portable analytic tools.

- *Move toward virtual networks.* Increased use of "virtualization"—software-defined-networks, virtualization of network functions, and even virtualization of whole networks—is critical to meeting the varied needs of IoT applications. The ability to "stack" virtual networks, each with its own discrete quality of service and security, within a physical network architecture enables customized service to applications with unique requirements. Increased virtualization, however, creates new attack surfaces as network components become potential targets, and it creates additional challenges for analytics that must operate on a holistic level as well as provide cybersecurity for customized slices. Figure 4-1 illustrates the concept of network visualization in a notional urban environment. Both radio access and core networks are divided into virtual networks operating across the same physical space. Different services, each with various quality of service needs supporting critical functions within a shared physical space, have access to a virtual network customized to their discrete needs.

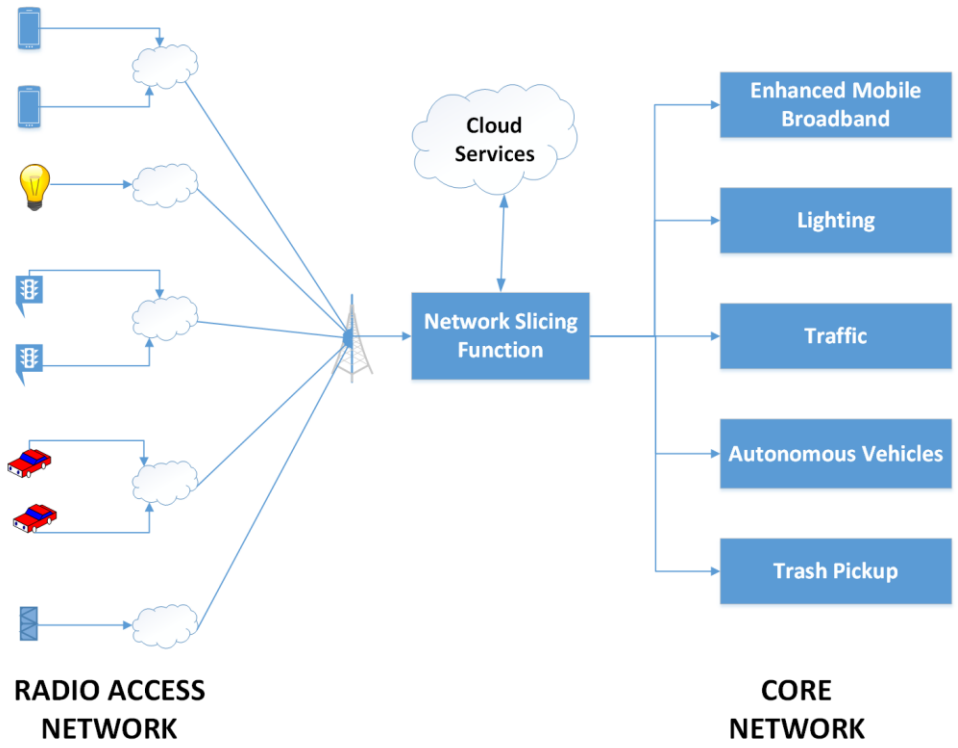

Figure 4-1 Network Virtualization

- *M2M communications.* Increased dependence on M2M and D2D communications will require adaptation of existing network security techniques to the large number of devices with limited security operating on networks and more dynamic network architectures. The growing number of devices with limited processing and the dynamic nature of D2D connectivity will require reevaluation of the concept of "endpoint" in terms of cybersecurity. Specifically, robust endpoint security will become more challenging, and networks will require additional analytic capabilities to evaluate the content of a network at any moment and what is happening at its endpoints.
- *Greater physical risk.* The expanding number of connected devices with at least some processing power at the heart of the IoT also presents new risks. Connected devices such as security cameras, sensors, and controls will frequently operate outside of security perimeters and may be compromised by persons with physical access. The number of these devices will also mean broader exposure to supply chain risks.
- *Lower-cost, lower-power devices.* The IoT is enabled by cost-effective devices with low power consumption connected to networks. Consequently, many connected devices will not have sufficient processing capacity to support comprehensive security. The presence of relatively unsecured devices on the Internet will place a burden on analytic tools to detect anomalous and undesired behaviors.

Best practices include adoption of a layered model of cybersecurity, as illustrated in Figure 4-2. It is important to put measures in place at each of these levels to secure hardware, software, networks, and data. As outlined in the list of trends, the emerging IoT will place new stresses at each level of cybersecurity as well as additional dependence on existing cybersecurity practices, including the adoption of Zero Trust Architectures and Endpoint Detection and Response. As the definition of the network perimeter shifts because more endpoint devices lack sufficient capacity to perform robust security, and

as other parts of the perimeter become more dynamic, robust cybersecurity will require a different approach to securing networks [67].

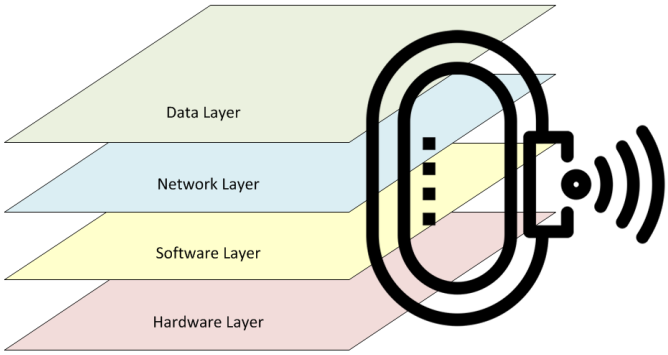

Figure 4-2 Key Layers of IoT Security

4.3 Cybersecurity Analytics Vision

A number of vignettes, scenarios, and use cases describing potential IoT-enabled attack vectors were developed to serve as the bases for more detailed future analyses. In developing these use cases, a number of new vulnerabilities arising from IoT-enabled applications were envisioned, and scenarios and use cases were constructed based on those vulnerabilities. Automation of industrial control systems and similar applications (e.g., smart home applications) was seen as not only introducing a large array of devices with limited processing to support cybersecurity, but also as opening a potential conduit for attacks on infrastructure and public services. Use of biometric measurement devices connected to personal area networks was seen as creating new concerns regarding privacy because cybersecurity vulnerabilities could be exploited to access private data. D2D communications-supporting applications ranging from emergency rescue to autonomous vehicles present highly dynamic, hard-to-characterize perimeters that were expected to present unique challenges to secure. Highly virtualized networks, which may be required to support applications with high levels of autonomy, will require that network analysts have the ability to rapidly and correctly interpret behaviors across a wide range of highly specialized topologies.

Additionally, a range of potential cyber-physical effects with impact well beyond the range of target surfaces were identified. Control systems that operate public utilities (e.g., water and electricity), industrial processes, and even home and building physical plants are becoming more automated and connected, and therefore vulnerable to cyber sabotage. Medical processes collect greater amounts of data using IoT devices and store them in databases, potentially enabling substantial amounts of sensitive private data to be compromised in a breach. Financial and banking agencies move large amounts of information and money electronically, increasing the potential for loss of private financial information and possibly enabling theft or manipulation of transactions by external actors. Transportation systems and autonomous vehicles become subject to attacks capable of causing large-scale disruptions to daily life or even injuries. To make things more complicated, the ultimate target of an attack may be at some distance from the attack surface. For example, one can readily envision an adversary attempting to access financial services software via an attack on a bank's smart building network or attempting to access legal or medical records through an attack on an ad hoc emergency services network (see Figure 4-3).

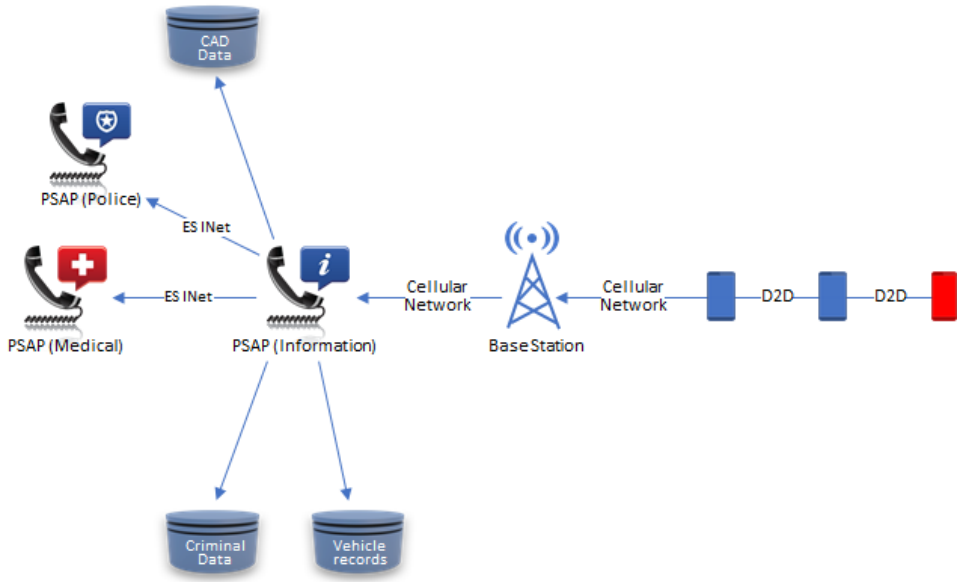

Figure 4-3 Hypothetical Attack on Legal Records via an Ad Hoc Network

The diversity, complexity, and scale of the IoT mean that analytics will become increasingly important in providing cybersecurity. Specifically, cybersecurity can no longer depend on defending the perimeter

and keeping dangerous and malicious entities off the network—it will need to be supplemented by continuous monitoring and assessment of the networks to detect and identify malicious actors that have circumvented cybersecurity measures and entered the network. This requires a robust analytic capability—consisting of cybersecurity analytic software as well as trained analysts—to continuously monitor network performance and identify anomalies. We identified eight vision elements characterizing the needed evolution in cybersecurity analytics to meet the needs of the IoT.

- Analytics will need to be increasingly portable and scalable. As processing with IoT-enabling architectures becomes increasingly distributed, analytics will need to be hosted in a variety of platforms capable of accommodating various levels of processing. In addition, they will need to monitor and evaluate networks with increasingly varied topologies and sizes, creating a need for increased scalability.
- Analytics will need to be decomposed to enable them to support distributed execution. Advanced cybersecurity analytics will require decomposition to enable operation across sectors, closer to the edge and endpoint, and on virtualized networks with customized quality of service and security capabilities. It is easy to conceive of analytics consisting of thin clients in smaller devices with larger applications hosted in fog- or cloud-computing facilities.
- Analysts will need the ability to compose advanced analytics from existing analytics. Even as analytics require decomposition for hosting in devices and in processing facilities near the network edge, cybersecurity analysts will need to be able to compose simple analytics operating at the edge into advanced cross-sector analytics while maintaining acceptable levels of analytic rigor.
- Decentralized and virtualized architectures will have a profound effect on machine-learning–based cybersecurity analytics. Similar to a human operator becoming more proficient at a task by performing the task, machine-learning–based analytics become more effective by repeatedly

performing their assigned tasks. As cybersecurity analytics become more physically distributed or customized to meet the needs of specific network "slices," it is likely the analytics deployed toward the edge or to monitor customized slices will be exposed to a less representative cross-section of network traffic. This constrained training could affect the quality of training, and mechanisms to present distributed analytics with more robust training sets may be required. It may be necessary to generate training sets to support supervised machine learning or to implement forms of federated learning to augment the data presented to the analytics during normal operation.

- With the growth of the IoT, analysts will be required to rapidly interpret larger, more complex cybersecurity environments. Visualization capabilities of analytics packages will need to evolve to enable users to maintain situational awareness of larger-scale and potentially more complex network traffic. In particular, they may need to be adapted to enable analysts to make sense of more dynamic network perimeters in which devices enter and join a network on an ad hoc basis (e.g., a network supporting autonomous vehicles) and in which significant portions of the network traffic are virtualized.
- The need for responsiveness to a cyber-attack drives the need for dynamic analytic tasking near the edge. As the cyber environment shifts, analysts will need to dynamically task analytic software with responding to the changes. Analytics deployed in platforms with limited processing may not be able to execute a full suite of cybersecurity analyses concurrently; it may be necessary to modify analytic performance dynamically based on the current environment. Analysts may need to modify search characteristics or insert intelligence into analytic packages to enable them to perform more effectively in a prevailing environment.
- As the cybersecurity environment increases in scale and complexity with the growth of the IoT, automated analytics will need to help human analysts manage the ever-increasing workload.

Cybersecurity analytics need to be designed to maximize human-machine coordination. Increasingly, analytics will require the capability to interpret the cybersecurity environment and generate alerts for network operators.

- Lastly, cybersecurity analytics developers need to be cognizant of the ability of threats to extend across critical sectors.

4.4 Defining the Role of Government

At a minimum, government will need to secure their own networks from intrusion and compromise of important data. As the IoT continues to grow and become increasingly ubiquitous, protecting government and partner computers' networks from cyber threats will become increasingly challenging. Government agencies will be housed in smart buildings connected to city-wide smart grids. They will have fleets of vehicles sharing the transportation network with autonomous vehicles and connected to the transportation grid. Government employees will work using phones, tablets, and computers connected by the same service provider networks that serve the IoT. Government will require an evolving cybersecurity analytics capability aligned with the larger-scale, increasingly distributed, and fluid IT architectures described in the previous paragraphs.

Beyond that, government has an interest in protecting interconnected networks that extend beyond the government domain, although it may have limited ability to do so. The IoT threatens to put industrial processes, public utilities, public health, transportation, and financial services increasingly at risk of cyber-attack. It heightens the risk of economic damage (e.g., a shutdown of critical industries or services or theft of financial assets), compromise of sensitive data (e.g., medical, financial, or legal), or cyber-physical terrorism (e.g., hijacking of drones or aircraft, release of toxic materials). Yet, while each of these results may have clear national security implications, the fact that many of these attacks will be targeted at private computing networks may limit government's ability to act.

The public sector will need a vision for the likely impacts to its programs and initiatives, and potential cyber-analytic capabilities to respond to the likely emerging challenges of an increasingly distributed, virtual, and dynamic IoT cyber environment. That vision needs to include comprehensive security for government and partner networks, and should, where possible, include increased cognizance of and participation in securing the entirety of the IoT. As this paper is being written, much of the planet is in lockdown, possibly the result of trafficking animals in a relatively small corner of the planet [68]. In much the same way, computer systems and international networks everywhere can be placed at risk from failures to police networks anywhere. The ubiquity of the IoT makes it harder, if not impossible, to "quarantine" parts of it, and calls for a broader government role in securing the IoT.

The public sector vision includes the following:

- Government must first secure its own networks. This means having the ability to evaluate cybersecurity analytic capability, as well as maintaining access to the most up-to-date and effective tools and deploying them on government networks.
- Government requires increased situational awareness, not only over its own networks, but over the entirety of the IoT. As the IoT grows, the ability of malicious actors to propagate will only increase, thus early awareness and understanding of threats will be increasingly vital to securing computing and network resources. Government will need to partner with its constituents to expand the data types, sources, and corresponding analytics to gather the appropriate situational awareness in IoT devices and IoT-enabled systems to meet its role as the nation's risk advisor.
- Government should be proactive in partnering with the private sector to provide guidance and assessment related to cybersecurity analytics. While its ability to affect private networks may be limited, government has a clear stake in promoting a secure cyber environment beyond the public sector. Government should, wherever it can, leverage its role as an advisor and a public advocate to communicate and promote adoption of best practices throughout the cyber environment.

- Government may need to take an active role in guiding the development of cybersecurity analytic capabilities to meet the demands of the IoT. Where existing tools demonstrate gaps or vulnerabilities, government may need to take the lead in sponsoring technical development to address observable shortcomings.

4.5 Conclusions

The IoT represents a significant change in the way humans and machines will interact in the future. Not only does it represent a massive change in scale as more and more devices exchange increasing quantities of network traffic, it also will incorporate radical new technologies that redefine what a network is. New physical connections, including D2D connectivity, will enable ad hoc networks that can rapidly be deployed in emergencies and support autonomous vehicular traffic. Virtual networks will enable the creation of networks within networks, each with its own unique quality of service and security.

To respond, cybersecurity analytic capabilities will need to undergo dramatic reevaluation. Increasingly, the analytics designed for cybersecurity applications will need to become more portable, to enable them to operate in smaller devices with less computing capacity, and more scalable, to monitor larger and more diverse networks. Functionality will require decomposition, to support interpretation of local networks and network slices, and aggregation, to enable analysts to discern network trends based on behaviors observed on subsets of the larger network. Machine-learning–based algorithms will require new ways to train algorithms that are exposed only to customized traffic. Analysts will need capabilities that allow them to visualize the larger, more dynamic IoT, to handle the additional workload associated with maintaining cyber awareness, and to dynamically respond to it.

To ensure success, government has a role in promoting a more secure IoT. It must first protect government networks, providing the ability to access, evaluate, and deploy analytic capabilities that meet the challenges of the emerging IoT. Government will need to be increasingly cognizant of what is happening on the IoT, and will have a stake in leveraging its role as an advisor to ensure that evolving best practices are implemented throughout both the public sector and the private sector. Lastly, government may need to promote and invest in cybersecurity analytic tools to address critical gaps.

5 IDENTIFICATION OF BOTNET ACTIVITY IN IOT NETWORK TRAFFIC USING MACHINE LEARNING

5.1 Introduction

In recent years, botnets have emerged as a serious threat to cybersecurity because they provide a distributed platform for performing a multitude of malicious activity such as distributed denial-of-service (DDoS) attacks [69], exploitation of command and control (CnC) vulnerabilities [70], phishing, malware dissemination, and click fraud [71]. Although computers were the original target for botnet attacks, IoT devices have become an increasingly common target. In 2016, the Mirai botnet succeeded in remotely controlling nearly half a million IoT devices by scanning the Internet for open telnet ports and logging into those devices, using a set of default username and password combinations [72].

Several Mirai variants now exist to exploit diverse IoT security; thus, protecting IoT devices from botnet attacks remains a priority. IoT devices use multiple means of communication such as Wi-Fi, Bluetooth, or Zigbee. This causes a new issue—volume and variety in traffic flow. Typically, machine learning classifiers create models from monitored traffic data and then detect malicious traffic flows. However, with increasing IoT traffic flows, there is a need to improve detection of malicious activity in IoT traffic.

The main contributions of this chapter are as follows: (1) we demonstrate that machine learning, supervised learning, and deep learning methods can be developed to effectively identify varied botnet activity and (2) we also show how the performance of these classifiers scale when a large dataset of IoT network connection logs is used with additional types of botnets.

The remainder of the chapter is organized as follows. Section 5.2 discusses related works on which our research builds. Section 5.3 describes the IoT network connection log dataset used in this research along with the additional data we captured to augment this dataset. Section 5.4 describes the supervised learning and deep learning classifiers used, the performance metrics used, and our methodology.

Section 5.5 describes our experimental evaluation. Section 5.6 provides our results and discussion. Section 5.7 summarizes our research and describes avenues for future work.

5.2 Related Works

5.2.1 Anomaly Detection Models for Smart Home Security

Our research focused on detection of botnet traffic in smart-home environment networks. Prior work by Ramapatruni et al. [73] focused on anomaly detection within similar home environments through analysis of device behavior using Hidden Markov Models (HMMs). HMMs were used to learn common behaviors, such as actions and signals, from devices in the smart-home environment. Using HMM with tuned hyperparameters, a detection accuracy of 97% was achieved.

Although this prior work focused on the analysis of device actions and signals to detect anomalous behavior, our research focused strictly on network traffic to detect botnet activity.

5.2.2 Detection of Mirai Attacks

Our research focused on the detection of Mirai botnet and similar botnet activity. Previous work by Kambourakis et al. [72] proposed a network-based algorithm for detecting IoT bots in a Mirai simulation. Their research considered preconditions (e.g., telnet port open and default credentials) and performed signature-based detection against incoming Mirai packets, only sampling a fraction of the devices at a time. For the purpose of optimization, the algorithm minimized the cost associated with average detection delay in detecting a compromised device.

Our work assessed the effectiveness of machine learning classifiers to detect botnet activity in home IoT environments, seeking to improve on current top performance results of the botnet detection capability in an IoT environment. Our work uses accuracy, detection, and false alarm rate as performance metrics. This differs from previous work that considered sampling rate and detection delay cost as performance metrics.

5.2.3 IoT Security Using Deep Learning and Big Data

Our research was geared toward the use of big data and machine learning to enhance security for large volumes of packet traffic. Prior work by Amanullah et al. [74] used an LSTM model combined with a CNN on popular security datasets, such as NSL-KDD, KDD99, and UNSW-NB15, and achieved results of accuracy of 97%, detection of 98.03%, and false positive rate of 4.08%.

Our research attempted to improve upon these results using a different dataset including benign data we captured from a testbed environment as well as a different set of machine learning classifiers.

5.2.4 Imbalanced Datasets for Traffic in Industrial IoT Environments

Our research considered a scenario in which network traffic is a "haystack" of benign data, with a few "needles" of malicious traffic data within. Thus, the dataset used in our research to train our classifiers was unbalanced. Prior research by Zolanvari et al. [75] also applied machine learning to detect attacks within unbalanced datasets. Their research focused on network traffic collected within Industrial Internet of Things (IIoT) environments; their collected traffic had low amounts of attacks, thereby resulting in an unbalanced dataset for training their classifiers.

5.3 Datasets

5.3.1 Stratosphere Lab IoT-23 Data

The Stratosphere Lab IoT-23 dataset (hereafter simply IoT-23) contains network traffic from the Amazon Echo home intelligent personal assistant, Phillips Hue smart light-emitting diode (LED) lamp, and Somfy smart door lock IoT devices; this network traffic is labeled as either benign or infected with the name of the executed malware sample. The downloadable dataset, as well as details on the IoT devices used and the network setup, are available on the Stratosphere Lab website [76].

5.3.2 Additional Benign Data Capture

Additional benign network traffic PCAPs were captured from an Amazon Echo home intelligent personal assistant and a Phillips Hue smart LED lamp. Additional captures used the testbed setup depicted

in Figure 5-1. These devices provide similar real-time activity (e.g., requests, responses, service calls), and the network setup is a similar unrestrained Internet connection without induced malicious activity.

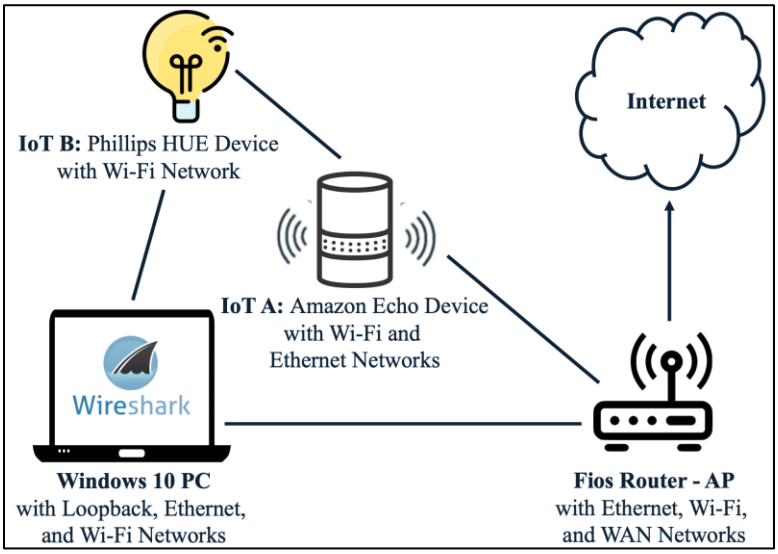

Figure 5-1 IoT Testbed Architecture

Wireshark was used to capture the network traffic in this testbed and generate new PCAPs from it. The Zeek tool [77] then converted the PCAPs into separate connection log files. These log files provided traffic flows similar to IoT-23, with attributes matching those of IoT-23.

5.3.3 Small Dataset and Large Dataset

A small dataset and a large dataset were created for testing the performance of machine learning classifiers to detect malicious botnet behavior.

The small dataset was created using a subset of the IoT-23 dataset. First, the Zeek network connection logs from the IoT-23 3-1, 8-1, 20-1, 34-1, and 42-1 captures, as described in [76], were combined. Then, from this combined dataset, 32,382 benign data points and 1,676 malicious data points were randomly extracted, creating a benign to malicious ratio of 95% to 5%. Of note, the malicious data included in the small dataset was infected with Muhstik, Hakai, Torii, Mirai, and Trojan botnet malware. The total size of the small dataset was 11.1 MB.

The large dataset was created by combining all the Zeek network connection logs from all the IoT-23 data captures shown with the Zeek network connection logs created from the benign data captured from the testbed described in the previous section. There is a wider variety of botnet behavior in the large dataset compared to the small dataset—the malicious data contained in the large dataset was infected with Mirai, Torii, Trojan, Gagfyt, Kenjiro, Okiru, Hakai, IRCBot, Hajime, Muhstik, and Hide and Seek botnet malware. The large dataset contained a benign to malicious ratio of 95% to 5%, the same ratio contained in the small dataset. This ratio gives a needle-in-the-haystack representation of malicious behavior representative of a real-life scenario. Altogether, the large dataset contains 30,854,774 benign data points and 1,623,938 malicious data points. The total size of the large dataset is 3.7 GB.

5.4 Methodology

We used several different machine learning algorithms and performance metrics to demonstrate the robustness and efficacy of our approach.

5.4.1 Machine Learning Algorithms

The following supervised deep learning and machine learning algorithms were used to develop classifiers for botnet activity detection: decision tree, random forest, multiclass decision forest, two-class neural network, and multiclass neural network.

The *decision tree* classification algorithm creates a tree structure representing a sequential decision process in which an input data point's attribute values are consecutively tested to determine the data point's classification. A decision tree is composed of a root node, internal nodes, and leaf nodes that contain the possible class labels. Within each internal node, the value of the attribute assigned to the node is tested to determine the next node to progress to along the tree's path. Once a leaf node is reached, its label will represent the input data point's identified classification [78].

The *random forest* classification algorithm is an ensemble-learning method in which multiple decision trees are constructed, and the mode value of the classes predicted by the individual trees is used to

classify the input data point. Each decision tree in the forest considers a random subset of features and only has access to a random subset of the training data, thereby increasing diversity and potentially resulting in more robust classification predictions compared to individual decision tree classifiers [79].

The *multiclass decision forest* algorithm uses decision trees and bagging to predict a target that has multiple values. The underlying model upon which this is based is the random forest, which is composed of multiple decision trees, with the final classification decided by majority voting based on the predictions [80].

The *two-class neural network* classification algorithm is a set of interconnected layers of nodes that perform binary classification. In concept, a neural network consists of an input layer, any number of hidden layers, and an output layer. This design has activation nodes at each layer, and this is done over iterations. Input layers use input vectors, hidden layers calculate weighted values, and the output layer generates a weighted sum from each input data point. The identified classification is determined after all iterations complete. One of the main benefits with this neural network is that it decreases the number of false positives in larger datasets, making it a strong selection for needle-in-the-haystack datasets [81].

The *multiclass neural network* classification algorithm is similar to a two-class neural network, but is capable of identifying multiple labels. Additional considerations are that the inputs are the first layer and connected to an output layer by an acyclic graph composed of weighted edges and nodes. Between the input and output layers are hidden layers. The number of hidden layers can be configured to optimize performance based on the type of classification being performed. The nature of this multiclass neural network is that it performs predictions over multiple candidates and then uses the weighted values to predict a single output target value [82].

5.4.2 Performance Metrics

The following metrics were used to measure the performance of the machine learning classifiers on the small and large datasets:

- Accuracy = (True Positive + True Negative) / (True Positive + False Negative + False Positive + True Negative)
- Probability of Detection = True Positive / (True Positive + False Negative)
- Probability of False Alarm = False Positive / (False Positive + True Negative)

The accuracy is equivalent to the proportion of correctly classified data points, the probability of detection is equivalent to the proportion of correctly predicted positives, and the probability of false alarm is equivalent to the proportion of negatives incorrectly predicted as positives.

K-fold cross validation was used to train, test, and evaluate the classifiers using the three metrics described. Using K-fold cross validation where K=10, the dataset was divided into 10 sections or folds. During the first iteration, the first fold was used to test the model and the remaining folds were used to train the model. During the second iteration, the second fold was used to test the model and the remaining folds are used to train the model. This process is repeated until all folds were used as the test set [83].

Additionally, information gained for each feature in the dataset was calculated to rank the relative importance of features used by each classifier. Information gain is a feature evaluation method that measures the amount of information a feature provides in terms of the change in entropy.

Entropy is defined as:

$$H = -\sum_{i=1}^{K} p_k \log_2 p_k \tag{1}$$

where p_k denotes the proportion of instances belonging to class k (K = 1, …, k).

Following from this, the change in entropy, or information gain, is defined as:

$$\triangle H = H - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R \tag{2}$$

where m is the total number of instances, with m_k instances belonging to class k, where K = 1, …, k [84].

5.4.3 Methodology

Our methodology was as follows: (1) We leveraged the IoT-23 dataset, using a small fraction of it to form a small dataset containing 95% benign data and 5% malicious data from five botnets; we combined the IoT-23 dataset without our captured benign data to form a large dataset, containing 95% benign data and 5% malicious data from 11 botnets. (2) We considered several machine learning models and chose the models with the most promising initial performance on the small dataset to further optimize. (3) We ran our optimized machine learning models on the small dataset and increasingly bigger subsets of the large dataset, using accuracy, probability of detection, and probability of false alarm as performance metrics. (4) We described the best practices from our research that has produced competitive results.

5.5 Experimental Evaluation

In our experimental evaluation, we considered several machine learning algorithms and both small and large datasets. We used this approach to determine whether the problem of effectively identifying anomalous IoT network traffic is dependent on the amount of data and/or the type of machine learning method used.

5.5.1 Experimental Setup

For small datasets, the Waikato Environment for Knowledge Analysis (Weka) was used to calculate information gain on the features of the datasets. Classification accuracy is the standard on which Weka calculates information gain for each feature and rank each feature of the set. Using Weka and the Python Scikit-learn library, the machine learning classifiers were applied to the small dataset within a local environment, as shown in Figure 5-2.

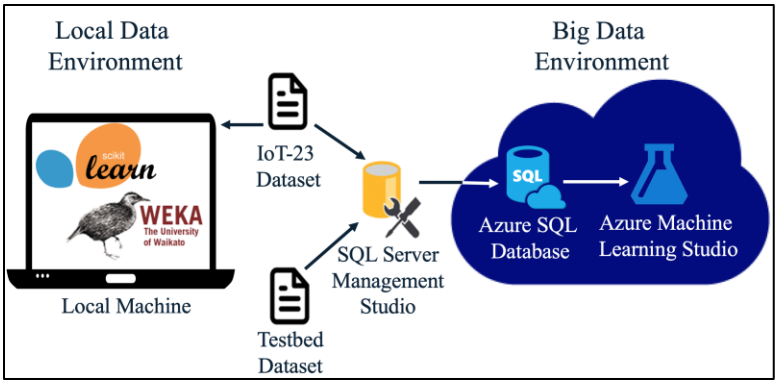

Figure 5-2 Local and Big Data Environment

For large datasets, the Microsoft Azure environment was used to store and combine IoT-23 data with the captured benign data from the testbed to create the large dataset. The Azure environment includes the Azure Machine Learning studio that was used to calculate information gain on the features of the datasets, specifically a Permutation Feature importance module that calculates information gain on a set of features [85]. Classification accuracy and recall are the standards on which the Azure Machine Learning studio calculates information gain for large datasets. The Azure environment further includes a Structured Query Language (SQL) server hosting an SQL database that stores the large dataset and natively connects to Azure Machine Learning services where machine learning pipelines can be created. Machine learning classifiers were applied to the large dataset within the Azure environment as shown in Figure 5-2.

5.5.2 Small Dataset Experiment

The following models were generated on the small dataset: decision tree, random forest, multiclass decision forest, two-class neural network, and multiclass neural network. This was done to measure initial performance feasibility of the models and to choose the models with the most promising initial performance to optimize. From this initial comparison, the best performing models in terms of the performance metrics described in the previous section were the multiclass decision forest and multiclass neural network models.

Fields from the Zeek network connection log files were used as features for each of the machine learning classifiers tested. Preprocessing was performed to one-hot encode the non-numerical features in the dataset. Additionally, a packet spacing feature was derived by calculating the timestamp-delta of packets sent from the same IP address.

For the multiclass decision forest, the resampling method, number of decisions trees, maximum depth of the decision trees, number of random splits, and minimum number of samples per leaf node hyperparameters were optimized. Specifically, the bagging resampling method was used. With bagging, each tree is grown on a new sample, created by randomly sampling the original dataset with replacement until the dataset is the size of the original. The number of decision trees used was 16, the maximum depth of the decision trees used was 32, and the number of random splits used was 128. The minimum number of samples per leaf node used, representing the minimum number of cases required to create any leaf node in a tree, was 20.

For the multiclass neural network, the hidden layer specification, number of hidden nodes, number of learning iterations, learning rate, initial learning weight, and normalizer hyperparameters were optimized. Specifically, one hidden layer with 100 nodes was used, with the number of nodes in the input layer equal to the number of features in the training data. One thousand learning iterations were used. A learning rate, representing the speed at which the model changes according to estimated errors, of 0.1 was used. An initial learning weight of 0.95 was used. A binning normalizer was used, which creates bins of equal sizes, then normalizes every value in each bin by dividing by the total number of bins.

5.5.3 Large Dataset Experiment

Increasingly large subsets of the large dataset were used to test the performance of the multiclass decision forest and multiclass neural network classifiers until high performance was achieved.

The following sizes of large dataset subsets were tested:

- 80,000 data points
- 160,000 data points
- 320,000 data points
- 480,000 data points
- 560,000 data points

Importantly, each subset contained 95% benign and 5% malicious data as well as the same ratio of the 11 botnets as found in the large dataset.

The Microsoft Azure machine learning environment was used for its big-data storage and supplementary compute power. Experimental pipelines were created within the Azure environment. The large dataset experimental methodology allowed for determining how well the classifiers performed on increased data, with a variety of malicious botnet behavior.

The large dataset comprised multiple PCAP files, exported into connection logs format using Zeek, imported and aggregated at the Azure SQL database using Microsoft SQL Server Management Studio (SSMS). This centralized all the traffic data so that it could be used for experiments in the Microsoft Azure machine learning studio with optimized performance.

For the multiclass decision forest, the bagging resampling method was used. With bagging, each tree is grown on a new sample, created by randomly sampling the original dataset with replacement until the dataset is the size of the original. The number of decision trees used was 32, the maximum depth of the decision trees used was 64, and the number of random splits used was 1024. The minimum number of samples per leaf node used, representing the minimum number of cases required to create any leaf node in a tree, was 1.

For the multiclass neural network, the hyperparameters optimized were the number of learning iterations, an initial learning rate, an initial learning weight, a momentum parameter, a shuffle parameter,

and normalizer; 160 learning iterations were used. A learning rate of 0.04 was used. A momentum of 0 was also used, with a shuffle parameter set to True. The initial learning weight of 0.1 was used. A min-max normalizer was used that linearly rescales every feature to a closed-interval from 0 to 1.

5.6 Results and Discussion

Table 5-1 shows the features of the dataset that were used by the multiclass decision forest and multiclass neural network classifiers.

Table 5-1 Classifier Features

Feature	Feature Description
Derived packet spacing	The timestamp-delta of packets sent from the same IP address.
id.orig_p	Source port.
id.resp_p	Destination port.
proto – tcp	Transport layer protocol of the connection is tcp.
proto – udp	Transport layer protocol of the connection is udp.
proto - icmp	Transport layer protocol of the connection is icmp.
resp_bytes	Number of payload bytes the responder sent.
conn_state – OTH	A partial connection that was not later closed.
conn_state – SF	Normal connection establishment and termination.
conn_state – REJ	Connection attempt rejected.
conn_state – S0	Connection attempt seen, no reply.
conn_state-RSTO	Connection established and aborted by originator using RST.
history – C	History of a tcp/udp connection packet with a bad checksum.
history – Sr	History of a tcp connection with a SYN request followed by a RST (reset) flag set to 1 from the responder.
history – ShAdDaFf	History of a tcp connection with a successful three-way handshake, packets with payload exchange from both ends and terminated gracefully.
history – I	History of an inconsistent packet (e.g., both FIN+RST bits set).
history – S	History of a tcp connection with a SYN without ACK bit set.

Feature	Feature Description
history – R	History of a connection packet with RST bit set.
history – ShR	History of a tcp connection with SYN, ACK and RST bits set.
history – ShAFr	History of a tcp connection that includes SYN+ACK requests as well as inconsistent packets of FIN+RST bits set.
resp_pkts	Number of packets that the responder sent.
resp_ip_bytes	Number of IP-level bytes the responder sent.

5.6.1 Small Dataset Classification Results

For the multiclass decision forest classifier applied to the small dataset, the most impactful features were the derived packet spacing with an information gain score of 0.076, id.resp_p with an information gain score of 0.061, and history – S with an information gain score of 0.014. The least impactful features were history – Sr, conn_state – REJ, and conn_state – S0, each with an information gain score of 0.

For the multiclass neural network classifier applied to the small dataset, the most impactful features were the derived packet spacing feature with an information gain score of 0.087, conn_state – SF with an information gain score of 0.071, and id.resp_p with an information gain score of 0.067. The least impactful features were conn_state – REJ, conn_state-RSTO, and history – I, each with an information gain score of 0.

Table 5-2 shows the performance results for the multiclass decision forest and multiclass neural network run on the small dataset.

Table 5-2 Results of Classifiers Run on Small Datatset

Performance Metric	Multiclass Decision Forest	Multiclass Neural Network
Dataset size	34,058 data points	34,058 data points
Percent True Positive	63.9%	71.0%
Percentage True Negative	99.6%	99.4%
Percentage False Positive	0.4%	0.6%
Percentage False Negative	36.1%	29.0%
Accuracy	81.75%	85.2%

Performance Metric	Multiclass Decision Forest	Multiclass Neural Network
Probability of Detection	63.9%	71.0%
Probability of False Alarm	0.4%	0.6%

5.6.2 Large Dataset Classification Results

For the multiclass decision forest classifier applied to the large dataset, the most impactful features were the id_resp_p with an information gain score of 0.075, the derived packet spacing with an information gain score of 0.059, and resp_bytes with an information gain score of 0.033. The least impactful features were history – R, conn_state – S0, and history – S, all with an information gain score of 0.

For the multiclass neural network applied to the large dataset, the most impactful features were the id_resp_p with an information gain score of 0.026, id_orig_p with an information gain score of 0.023, and derived packet with an information gain score of 0.022. The least impactful features were proto – udp, conn_state – RSTO, and conn_state – SF, all with an information gain score of 0.

For the large dataset, the multiclass decision forest exhibited an upward trend in accuracy, growing from 99.93% to 99.98% as dataset size increased. Interestingly, accuracy of the multiclass neural network fluctuated as dataset size increased; the accuracy ranged from ranging from 99.49% to 99.70%, with no steady trend as data size increased.

Table 5-3 shows the performance results for the multiclass decision forest and multiclass neural network run on the large dataset.

Table 5-3 Results of Classifiers Run on Large Datasets

Performance Metric	Multiclass Decision Forest	Multiclass Neural Network
Dataset Size	560,000 data points	480,000 data points
Percent True Positive	99.8%	99.7%
Percentage True Negative	100%	100%
Percentage False Positive	0%	0%
Percentage False Negative	0.2%	0.3%
Accuracy	99.9%	99.8%

Performance Metric	Multiclass Decision Forest	Multiclass Neural Network
Probability of Detection	99.8%	99.7%
Probability of False Alarm	0%	0%

5.6.3 Results Discussion

Considering the unbalanced nature of our dataset (95% benign to 5% malicious ratio), the classifiers performed well as the dataset size increased. The following were the most consistently important features in terms of information gain: packet spacing, id_resp_p, resp_bytes, history – S, conn_state – SF, and resp_pkts. The results showed how increased dataset size and botnet diversity (i.e. amount of network traffic flow considered) improved classifier performance. Larger dataset sizes had more traffic flow; with additional traffic, the overall information gain of the features increased. The classifiers benefited from features with higher information gain, as they contributed to improving the probabilities of accuracy and detection, while minimizing the probability of false alarm.

Results also represent real-world scenarios of IoT environments and attack vulnerabilities. Botnets infect IoT devices by exploiting vulnerable nodes in the network that have open ports of protocols such as telnet/23 and control these devices using a CnC server. The botnet binary is then transferred and downloaded on the target. The Mirai botnet and its variants (e.g., Okiru, Gafgyt, and Hajime) use infected devices to recruit new bots to join the botnet network using different techniques such as performing a horizontal port scan or a SYN flood attack where the bot floods the network with non-legitimate SYN requests to scan for random IP addresses. Having a successful connection indicates a new target was found, and the CnC server is notified about the connection.

These parameters of botnet functionality can be aligned with the features our research found to have the highest information gain score using multiclass decision forest and multiclass neural network classifiers. For example, the responding port feature (id.resp_p) indicates that the victim responded to a request targeting one of its vulnerable open ports such as telnet to initiate a connection and load the

botnet binary. The history feature (history – S) indicates there was an attempt to initiate a TCP three-way handshake by sending multiple SYN requests without waiting for acknowledgment from the responder. This is evidence of a SYN flood attack to recruit new bots. Furthermore, a history of complete connections (history – ShAdDaFf, conn_state – SF) as well as the number of responding packets (resp_pkts) indicate a successful connection between the infected device and scanner and that the device is compromised. On the other hand, there were other features that had zero information gain scores, such as rejected connection attempts (conn_state – REJ), where the IoT device has potentially rejected a connection request from a botnet or a CnC. Similarly, (conn_state – RSTO) and (history – I) are features of connections that were reset and aborted by the source before reaching the destination. The presence of these features is an indication that the data point is negative because the compromise attempt has failed.

The performance results of our classifiers are competitive with recently published results. Our research uncovered best practices that can be applied when attempting to detect malicious botnet activity in relatively small, unbalanced datasets. First, it is advantageous to understand the behavior of botnets and the aspects of network activity their presence may impact. Through this understanding, researchers may identify and prioritize features to use within detection classifiers that would provide the most information gain. Second, additional features may be derived that may make network activity patterns more apparent to machine learning classifiers. For example, in our research, we calculated and included a packet-spacing feature that made more obvious to the classifiers the time between packets sent from the same IP address. This derived feature was discovered to provide high information gain and greatly improved the performance of our classifiers. Although our research used only features found in and derived from the Zeek connection log, additional features may be found and derived from information provided in the original PCAP files. Lastly, our research showed how increasing the amount and variety of botnet behavior in our datasets greatly improved performance of our classifiers. Even by using just a small

subset of our large dataset, we were greatly able to improve performance in terms of accuracy, probability of detection, and probability of false alarm compared to the small dataset.

5.7 Summary and Future Work

In our research, we created small and large datasets leveraging the IoT-23 dataset and our own captured benign data. These data had an unbalanced needle-in-the-haystack makeup of small amounts of malicious behavior compared to benign behavior. We considered several machine learning models, ultimately choosing to optimize the multi-forest decision tree and multiclass neural network classifiers to detect malicious botnet activity. We ran these optimized models on the small dataset and increasingly bigger subsets of the large dataset, using accuracy, probability of detection, and probability of false alarm as performance metrics. We found that performance of our classifiers increased as the size of the dataset, amount of malicious activity, and diversity of malicious activity increased. We highlighted how our results are competitive with recently published results attempting to solve similar problems and described the best practices from our research that produced the competitive results.

Future work could be extended to consider a wider range of IoT devices, larger datasets, and the use of unsupervised learning classifiers. Our research only considered network activity from three devices, although many more smart-home devices exist and are continuing to be manufactured. There is motivation to perform anomaly detection on specific brands of the same IoT devices; this would provide insight into network and application layers of the specific IoT devices. Lastly, our research only considered training from a labeled dataset; additional work could consider the implementation of unsupervised learning techniques, such as K-means clustering, on unlabeled data, to detect malicious activity.

6 A PROPOSED TRUST MODEL FOR ASSESSING CYBERSECURITY RISK IN A SUPPLY CHAIN CONSIDERING IOT'S IMPACT

6.1 Introduction

The IoT is the product of a number of intersecting technologies that have enabled ever-increasing levels of processing and communications to be integrated into ever smaller and less expensive devices. It is already ubiquitous and having a profound effect on many aspects of human activity, including the design and development of most products. However, even as integration of IoT-enabled capability into a supply chain can enhance efficiency and productivity, it also increases exposure to Internet-based threats. One area of particular concern is the potential for introduction of malware into a system via its supply chain.

In this chapter, we propose a methodology for objective evaluation and decision-making related to cyber vulnerabilities in a supply chain. We describe potential threats to systems within their supply chains, identify potential trade-offs between IoT-related capabilities and vulnerabilities, and discuss a methodology for describing cybersecurity risks as trust relationships. Although it is not deemed possible to create a fully objectively quantifiable method to measure cybersecurity vulnerability, it should be possible to use the proposed methodology to support decision-making and identify weak links in the supply chain.

Our contributions include (1) a brief description of the evolving cybersecurity threat, (2) an analysis of areas in which IoT-enabled capabilities enhance productivity at a cost of introducing additional cyber vulnerability, (3) a proposed trust model, and (4) a proposed methodology for evaluating cybersecurity vulnerabilities in a supply chain using the Delphic Hierarchy Process (DHP) and Analytic Hierarchy Process (AHP). Our analysis differs from other supply chain analyses in that it focuses on threats to a system via the supply chain, rather than on threats targeted at disrupting the supply chain.

6.2 Background

It has long been projected that the IoT would be a target for future cyber-attacks. The combination of large numbers of connected devices and the weak protections for many of those devices make IoT devices an ideal point of entry. This possibility became a reality a little over a decade ago. Although neither the

perpetrators nor their goals are known, the Stuxnet attack on the Natanz uranium enrichment plant in Iran demonstrated the ability of a cyber-attack to inflict physical damage across international boundaries. Running undetected for months, it damaged more than 1000 centrifuges, thereby degrading facility capability by more than 20% [86]. Stuxnet presents a model of an IoT-directed attack in that it has the following characteristics:

- It targets industrial control systems. In this case, the targets were Siemens programmable logic controllers.
- It depends on lateral movement to move from an opportunistic attack surface to its ultimate goal. In this case, the virus had to be introduced into a number of domains outside the target domain, and it eventually found its target even though that target was inside an air-gapped domain.
- It is stealthy. The proliferation of devices on the IoT, even in a local slice of the IoT, makes it increasingly difficult to detect persistent agents.

Another potential risk is the introduction of counterfeit components, especially electronics. Despite efforts to protect the integrity of supply chains, it is estimated that 15% of all spare and replacement parts for military systems are counterfeit [87]. There is growing concern that a nation state might attempt to exploit this increased dependence on automation to introduce counterfeit parts as part of an intentional program of sabotage. In June 2014, a U.S. contractor admitted to conspiring to ship counterfeit semiconductors from Hong Kong to a U.S. Navy base in Connecticut for use in nuclear submarines [88].

The problem is exacerbated by IoT technology. Not only are more devices autonomous and connected, components and subsystems within systems are connected. The average automobile has more than 30 connected computers, whereas a luxury automobile may have as many as 100 [89]. A modern Airbus A320 has approximately 2000 computers [90] that control everything from entertainment and environment, to maintenance and diagnostics, to key functions like fuel injection and braking in

automobiles and most aspects of flying an airplane. Increasingly, systems are themselves self-contained IoT ecosystems.

Future technology evolutions may further exacerbate the threat:

- Reverse engineering and cloning technologies are becoming more sophisticated. (In part, this is driven by foreign governments that want to reduce their dependence on U.S.-produced components, for fear that they contain malware.) [91]
- The number of system components containing processors is increasing.
- Trusted Foundry programs are no longer keeping pace with the state-of-the-art in integrated circuit (IC) design. Where military and aerospace applications were once a large driver of the IC market, they now account for only about 1% to 2% of the market [92].
- Lastly, introduction of Micro-Electrical-Mechanical Systems (MEMS) and nanotechnology make it possible to develop smaller and lower-power devices, like smart dust. This may eventually enable miniature electronic components to be stealthily introduced into larger non-electronic components.

Critically, there is no way of knowing how many vulnerabilities exist in deployed systems or even how many attacks have already been launched, with persistent agents awaiting activation. In August 2020, the Department of Homeland Security Cybersecurity and Infrastructure Security Agency's U.S. Computer Emergency Readiness Team issued a warning regarding a collection of vulnerabilities, referred to as Ripple20. Ripple20 contains approximately 20 vulnerabilities, including four rated as critical using NIST guidelines that an adversary could use to launch zero-day cyber-attacks against systems with the Ripple20 vulnerabilities. These vulnerabilities exist within the TCP/IP stack of a popular line of commercial products. It is estimated these vulnerabilities are present in systems across a wide range of critical infrastructure [93].

AI and ML are frequently applied to protect systems from cyber threats. Both of these technologies can enable security systems to assess and adapt to a changing threat environment more rapidly than humans can react. To date, neither AI nor ML driven malware have been observed "in the wild." Although there are threats capable of autonomous behavior, none exhibit the adaptability of an AI/ML-capable agent. Agents can be programmed to respond to cues within their environment, but to date malware capable of altering its behavior spontaneously based on its environment has not been observed.

AI/ML-enhanced malware could take two forms. In the first, the malware agent itself would be intelligent; however, this may be difficult to achieve. Both AI and ML are processing intensive. The processing required by an intelligent agent might limit its ability to operate in smaller, less well-protected devices. It might also render it easier to detect because the need for processing resources would make it more readily suspicious to any antivirus or security analytics. In the second form, an agent might be intelligently designed, controlled, and reprogrammed. Once launched, an agent could provide "cyber reconnaissance" information back to the control module, which in turn could modify instructions or agent programming intelligently. Either implementation could provide a capability for an agent that learns based on what works and what does not, responding to successful detections by adaptation.

Integration of AI into malware has the potential to increase the scale and speed of future attacks and make them ever harder to detect.

6.3 Developing a Framework for Evaluating the Cyber Supply Chain

The first step in developing a framework for evaluating risks within the cyber supply chain was to identify how the IoT was being integrated into existing industrial systems throughout the engineering life cycle. To do this, we performed a literature review, looking at both currently existing practices and likely future practices.

Since the advent of ICs in the 1980s and the Internet in the 1990s, there has been a move toward increasing connectivity and mobility. As the ability to implement complex processing in ever smaller

devices has increased, devices housing critical functions have become smaller and more mobile, and processing has been integrated into ever more devices. It has also changed the way work forces operate.

- Devices, ranging from household appliances to complex machine tools, are often designed with processing capability and Internet connectivity. It is not unusual that the first task a new device performs is a search for local Wi-Fi to access the Internet.
- Design processes are becoming distributed and collaborative. It is common for complex systems to be designed by teams of engineers and stakeholders, located in different parts of country (or even the planet), to leverage cloud-based collaborative services and models to develop system designs (a practice that has become even more common because of the Covid-19 pandemic).
- Software development has been profoundly affected. New Internet-friendly programming languages ranging from Java to Python are becoming more common. Common software development tools, including those for managing requirements, tracking system defects, and configuration management, are often cloud-based and leased as software-as-a-service (SaaS). Lastly, as software grows in size and complexity, thereby making comprehensive testing more difficult, software development has become dependent on the ability to "push" upgrades and patches to systems via the Internet. The Windows 10 operating system has an estimated 50 million lines of code (Google may have as many as 2 billion) [94]. An entire software industry has emerged that depends on selling imperfectly tested software and identifying and fixing problems faster than its customers can become aware. IoT devices do not contain operating systems with tens of millions of lines of code, but they are designed by developers whose practices are built around the ability to implement system upgrades via Internet [95].
- In the near future, manufacturing processes will be changed by the adoption of greater levels of automation and robotics. It is already possible for an engineer to develop a digital model of a component on their laptop using customized software, and have that design implemented on a

machine without a human intermediary. There may come a time when the entire role of "machinist" disappears, replaced by automation.

6.3.1 IoT in the Supply Chain

Figure 6-1 provides a notional view of the Internet-enabled functions that may be performed by the notional factory of the future, with functionality expressed at levels varying from cloud services and edge services down to processing performed at a factory or even at a component (tool) level. Potential functionality of this factory of the future include the following:

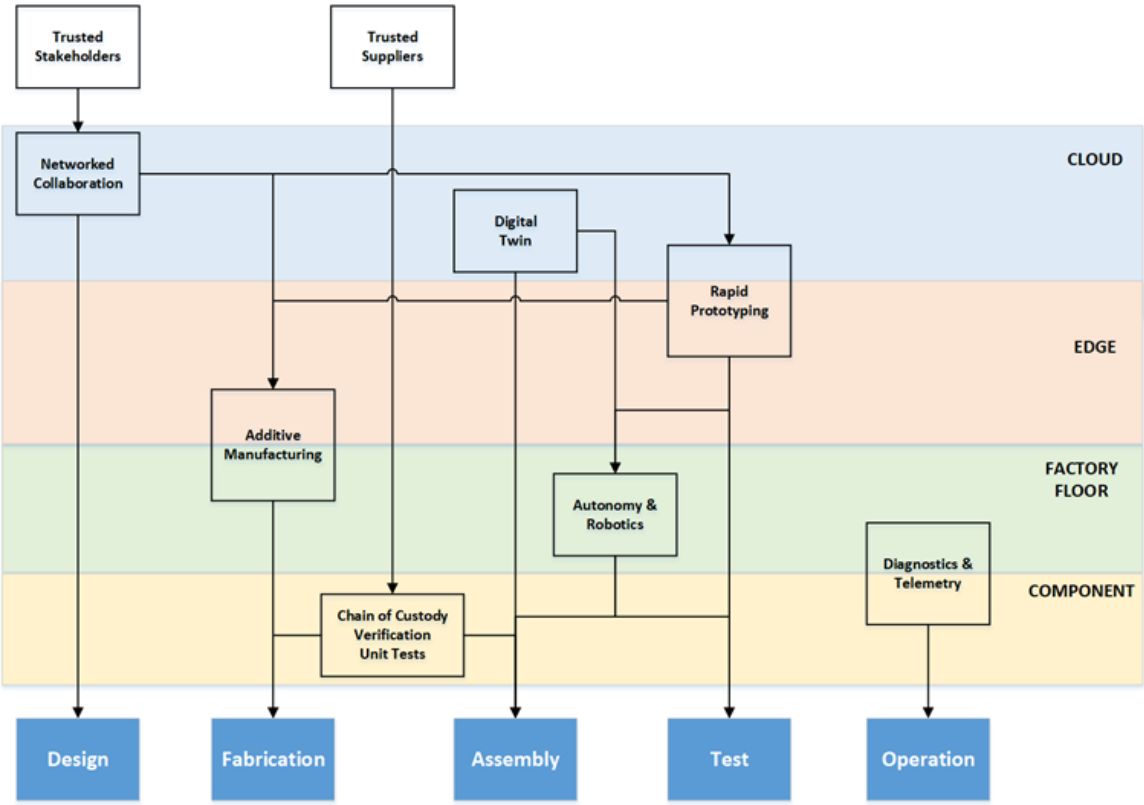

Figure 6-1 Functional View of IoT-enabled Manufacturing

Network-enhanced collaboration. Industries and government are moving to commercial cloud computing services. Commercial platforms such as Amazon Web Services and Microsoft Azure offer infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and SaaS. These structures are readily scalable and tailored to collaborative use of software tools. Future designs will be executed in the cloud, with geographically dispersed design teams having access to designs, models, and software tools,

including requirements, configuration, and risk management tools. This concept was arguably implemented on a large scale for the first time by Boeing for use in the design and development of their 777-commercial airliner. For the Boeing 777, approximately 2000 workstations were linked to eight mainframes in the Seattle area to provide development teams with the ability to electronically assemble and analyze the airplane. Engineers were required to develop designs at workstations, using the CATIA computer-aided design (CAD) system. Those designs were shared and incorporated into an evolving model and design. Initial models were low fidelity, but over time evolved into a manufacturing quality model [96].

Use of virtual and augmented reality to improve process and design. The Internet of the future will support use of virtual and augmented reality in design processes. Users will have the ability to visualize design and performance and even interact with design models. One application of virtual processing that more closely models the physical world is the use of digital twins. The concept of a digital twin is two decades old. It was a mirrored system, a precursor of the digital twin concept, that enabled NASA engineers to determine how to rescue the Apollo 13 astronauts. In the future, digital twin technology is expected to integrate real-time sensor data to support product, process, and service enhancements. The eMBB capability built into 5G-enabled IoT networks will move real-time telemetry data collected in edge servers to digital twins hosted in cloud platforms [97].

Rapid prototyping. This involves several technologies that enable rapid design, fabrication, and testing of prototype components for systems. In the context of this paper, it is arguably a combination of networked collaboration and direct digital fabrication (as discussed later). It will soon be possible for a team of design engineers to collaboratively develop a design for a component using cloud-based modeling, convert that design into an executable digital model, and deliver that software to a 3D printer for fabrication [98].

Automated, autonomous, and robotic assembly and fabrication. Automation and robotics are already widely used in manufacturing. However, the IoT-enabled factory floor, in which IoT-enabled sensors, controls, and actuators support collection of greater amounts of data and increasing levels of intelligence, will be capable of greater levels of autonomy. Processing that is currently performed by programmable logic controllers operating in closed systems with their own unique operating systems is expected to migrate into edge processors.

Direct digital manufacturing. A key component of future rapid prototyping is the ability to go directly from a digital design to a manufactured component or model. Engineers using CAD programs at their workstations can develop designs, have them converted into digital models, and transmit them directly to an additive manufacturing or computer numerically controlled machining center. This process enables more efficient rapid prototyping and production, which in turn facilitates the exploration and testing of novel concepts [99]. In addition, additive manufacturing processes, such as 3D printing, can save costs through increased efficiency. By constructing components layer by layer, rather than by cutting, these processes can substantially reduce waste, and, when working with expensive materials, costs. Although additive manufacturing is primarily used in the development of models, it has been applied in the aerospace and defense industries, particularly in the fabrication of titanium components [100].

Diagnostics and telemetry. IoT-enabled technology will support enhanced diagnostics and telemetry process for product improvement. Systems of the future will be equipped with sensors to monitor environmental conditions and effects. It will be possible to detect and identify conditions that could lead to system degradation and set maintenance schedules to be proactive. Telemetry data can be input into models to predict component failures before they happen.

Chain-of-custody verification and unit testing. Lastly, the envisioned factory will require enhanced capabilities to verify the integrity of components and raw materials entering the manufacturing process. IoT-enabled technologies, including radio frequency identification (RFID) and micro-tagging, will enable

monitoring of the supply chain and environmental conditions throughout the distribution process. In addition, all components with processing capability entering the factory will need to be scanned for introduced malware.

6.3.2 Areas of Potential Exposure

As design, development, and production become increasingly tied to Internet-enabled capabilities, supply chain exposure to the Internet, and by extension to cybersecurity threats, will increase. Figure 6-2 illustrates the life cycle of the system from the mining of required raw materials through the fabrication of components to the assembly, integration, and testing of components to form a system, and lastly through deployment and upgrade of a system during its life cycle. Throughout that process, there are interactions with external organizations and entities, and each of these interactions presents at least some level of risk.

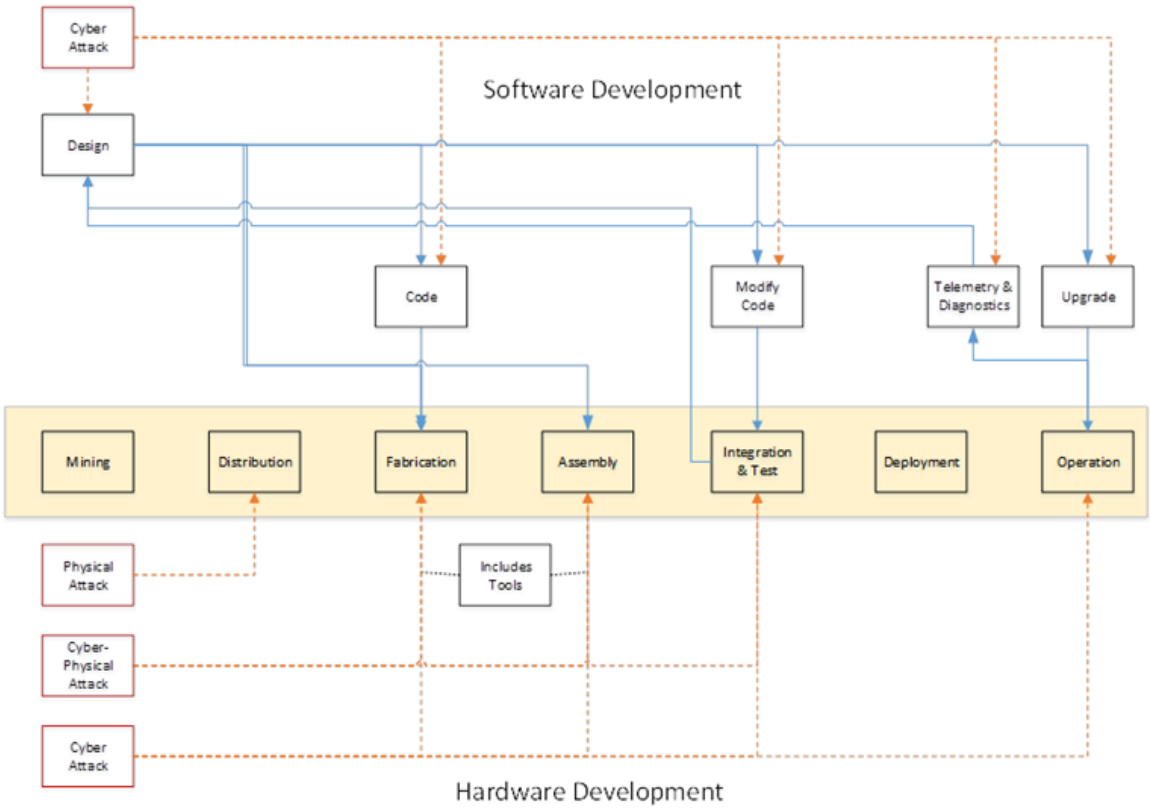

Figure 6-2 Internet Exposure within a Product Life Cycle

Interactions identified in Figure 6-2 include the following [101]:

- *Collaborative interactions with stakeholders and developers.* Design of complex systems is likely to migrate toward the adoption of collaborative virtual environments. Many industries have adopted their own versions of this process, including digital design models, cloud services, and widespread access for engineers operating remotely from workstations. This efficiency comes at a cost of increased Internet access and thus increased risk.
- *Digital design to fabrication.* The ability to go directly from a digital design to a fabricated component provides its greatest benefits when design engineers can develop components designs, possibly using cloud services such as those previously described, convert model output into an executable digital model, and input that directly into an additive manufacturing device such as a 3D printer. However, this highly efficient rapid prototyping capability will lead to broader access and Internet exposure.
- *Access to software development sites and tools.* There are numerous tools available to support all aspects of software development, including coding, configuration management, requirement management, and defect tracking. Development sites may even be hosted in cloud services, and tools are frequently hosted on external servers.
- *Software upgrades.* A typical system will require numerous software builds prior to deployment. It is unusual for software builds to be performed manually for large systems.
- *Software upgrades to manufacturing tools.* In addition to the product that will be dependent on external software upgrades, any automated tool used in the assembly, fabrication, or production process will also require periodic software upgrades.
- *Distribution processes.* One of the existing vulnerabilities in the supply chain is the distribution of components and raw materials. Processes are required to ensure the integrity of system components and raw materials throughout the distribution chain.

- *Post-deployment telemetry and diagnostics.* A common feature of modern systems is the use of diagnostic and telemetry data to support maintenance and to identify design flaws.
- *Post-deployment upgrades.* A key feature of the existing software production life cycle is the timely deployment of software upgrades to operational systems via the Internet.

6.4 Applying Trust

Evaluating cybersecurity vulnerabilities in an architecture or a supply chain is currently a subjective process, often requiring specific subject matter expert input. Although NIST provides guidelines to help organizations protect themselves against cybersecurity vulnerabilities within their supply chains, they do not provide guidance on how to measure those vulnerabilities. There is a need for a transparent method for measuring cyber risk within a supply chain.

Trust measurement deals with a problem analogous to the problem of cybersecurity measurement in a supply chain. A number of processes, including supply chain management, rely on trust relationships—relationships in which one party is dependent on a second party to meet specific obligations. A substantial portion of the field of game theory is devoted to finding ways to model trust ("The Prisoners' Dilemma," the ubiquitous example used to introduce students to the concept of game theory, is a model to understand and measure the effects of trust). The cyber supply chain presents as a trust problem, and a critical hypothesis of our effort is that principles associated with the measuring and modeling of trust relationships can be applied to this problem. Specifically, every entity within the supply chain has a trust relationship with the entities below it in the supply chain; each must trust that antecedent entities have controlled exposure to cyber risks and applied best practices. Trust metrics are present in a number of common applications, including search engines [102].

There are three steps to developing a methodology for applying trust:

- Identifying the system or network architecture to which a trust model can be applied.

- Identifying a trust model. For this particular study, this requires identification of the attributes that contribute to cybersecurity trust between two entities in the supply chain. It also requires development of techniques for evaluating and weighting attributes.
- Conducting aggregation. Results from the previous two steps must be aggregated to provide an estimate of the cybersecurity trust of a proposed supply chain.

The goal of this study is not to provide an absolute measure of the cybersecurity risk inherent in a particular supply chain, but to develop a methodology that can be used to compare supply chains and identify likely vulnerabilities. Development of a more precise measure would require data not currently existing.

6.4.1 Identify an Architecture

Identifying the architecture is straightforward—the supply chain itself represents an architecture. Each of the suppliers (of components, subsystems, and services) is an entity in the architecture. Layers can be identified, as can flows between entities and layers.

6.4.2 Identify a Trust Model

Five attributes of a cybersecurity trust relationship are proposed. The first is HISTORY, which is a measure of the reputation of an entity within the supply chain. Critical aspects of a HISTORY measurement focus on an entity's past performance. In a supply chain in which Entity A has contracted Entity B to deliver a product or a service, HISTORY centers on Entity B's reputation (i.e., does Entity B have a reputation for strong cybersecurity) and on Entity A's past experiences with Entity B. Because cybersecurity incidents frequently are not publicized, a strong historical relationship may be more important than an industry reputation.

The second attribute identified for this analysis is EXPOSURE, which is intended as a measure of exposure to Internet threats. A primary tenet of this analysis is that all contact with the Internet represents a risk, although the risk is not uniform for all processes that use the Internet. It is proposed that each

entity in the supply chain would report any activities requiring access to the Internet. Not all of these activities would have equal risk, and subject matter experts would have to be enlisted to evaluate the risk of each activity. However, it is intended that EXPOSURE would provide an estimate to the risk involved in each supplier's development processes.

The third attribute, PRACTICES, provides a measure of the things an entity performs to reduce cyber risk. There are a number of steps an entity can take to lower its cybersecurity risk, even if the Internet is integrated into its processes. First, it can enforce industry best practices and compliance with security standards. It can employ tools to detect, track, and remediate cybersecurity attacks. It can also implement processes for vetting employees to reduce the risk of insider threats. Again, subject matter experts would need to assess each entity's security practices. PRACTICES provide a measure of what an entity does to mitigate cybersecurity risk.

The fourth attribute, DEPENDENCE, reflects inherited cyber risk. In particular, cyber threats entering lower levels of the supply chain may, if not detected and remediated, propagate up the supply chain. DEPENDENCE is intended as an aggregate measure of cyber vulnerability inherited from lower levels of the supply chain. Measuring DEPENDENCE requires a knowledge of all the entities in the supply chain, including those that supply materials, products, and services. As discussed previously, there are a number of services in a supply chain that present a low risk of disruption to the supply chain, but could be large sources of risk to its cybersecurity. These services can include the following:

- *IT contractors.* Outside companies contracted to supply IT services (e.g., set up and operate a company's Internet) could represent an additional risk.
- *Security services.* Services contracted to support Internet security, ideally, serve to reduce the risk; however, whom a company uses to secure its IT services will affect its cybersecurity.
- *Cloud services.* Any PaaS and IaaS used by lower-level suppliers can affect a supplier's cybersecurity.

- *Automated tools.* Automated tools connected to the Internet and any tools implemented as SaaS can present vulnerabilities.
- *SaaS, IaaS, and PaaS.* Capabilities as a service that rely on the Internet as a delivery and access mechanism could represent risk.
- *Distribution services.* Distribution services refers to the transport and warehousing of components. Delivery mechanisms for software also need to be evaluated.

The final attribute, TRANSPARENCY, indicates a measure of a supplier's willingness to provide complete and accurate information. Trust in cybersecurity depends on transparency between organizations. If Company A contracts Company B to provide a product or service, Company A has to be confident that Company B is being transparent about processes that require Internet access and about its practices to protect its networks and information. Figure 6-3 illustrates how these attributes can be applied.

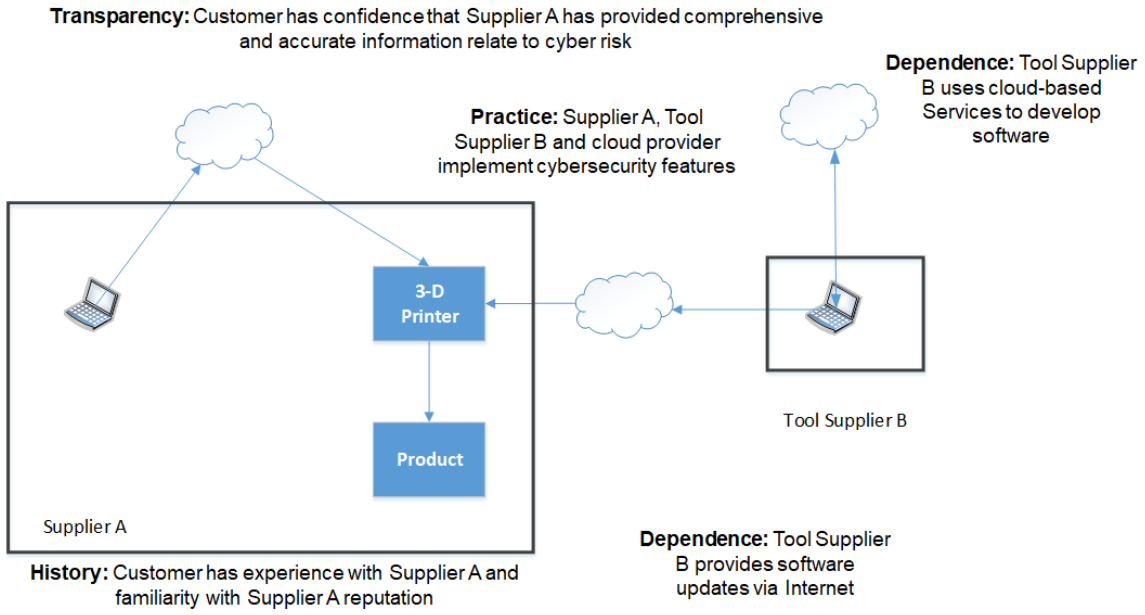

Figure 6-3 Applying the TRUST Framework

Perhaps the greatest challenge in using trust to evaluate cyber risk is identifying methods for assigning values and weights to the attributes. Because of the large quantity of unknowns associated with

cybersecurity and cyber-related threats, these evaluations are largely subjective and dependent on subject matter expertise. Thus, there is a need for a process to elicit subject matter expertise and present it in an objectively quantifiable manner.

A number of techniques are available to support objective assessments based on subjective criteria. Two popular techniques are DHP and AHP. The two can be used in tandem [103].

DHP is a systematic procedure for eliciting expert opinion, relying on anonymity, controlled feedback, and statistical group response. Critical features of DHP are that it is structured, it is executed in a step-wise fashion, and all responses are, to some extent, reflected in the final result. A goal of DHP is to move toward a consensus result [103].

In AHP, experts are presented a comprehensive series of pairwise comparisons, pairing options to selection criteria. Experts are asked to provide input as to which is better and to rate the comparisons based on importance. They may also be asked to provide a rough estimate of the magnitude of the difference between options. Posing more concrete comparisons, rather than abstract assessments, enables experts to provide assessments that are more accurate. AHP is commonly used in trade studies and other situations in which a group decision is required to select between options. DHP and AHP are candidate methods for identifying weights and evaluating attributes.

6.4.3 Aggregation

A number of techniques have been proposed for the aggregation of attribute values in trust relationships:

- A *weighted sum* approach is the simplest approach. It enables evaluators to input a value and weight to each attribute. Weight and value are multiplied for each attribute and then summed for all the attributes pertaining to a particular entity in the supply chain.
- *Fuzzy-logic–based modeling* is a form of mathematics specifically intended for Boolean mathematical-like problems in which multiple values are possible. Specifically, it arose out of the

need for methods to indicate partial truths. Trust calculations, in particular, would seem well suited to fuzzy-logic–based approaches. Trust and its attributes are seldom likely to be absolute (no network is absolutely safe from a cyber-attack, and no system is absolutely vulnerable). Fuzzy models have the advantage in that they are designed to provide useful responses to ambiguous input. Fuzzy logic has a number of applications, including computer-aided medical diagnoses.

- *Belief theory* (also referred to as the Dempster-Shafer theory) provides a method for combining the "beliefs" of multiple experts, taking into consideration their confidence in the result as part of the calculation. In assessing a supply chain, experts provide estimates of their confidence in the levels of protection and exposure of each entity.
- Lastly, *Bayesian methods* provide a methodology for updating assessments as additional information becomes known.

All of these methods are valid candidates for assessing the amount of trust a company can have it its supply. Because of the limited data available, it is proposed that initially aggregation be performed using a weighted sum as presented in the following equation:

$$\text{TRUST} = \alpha\text{HISTORY} + \beta\text{DEPENDENCE} + \chi\text{EXPOSURE} + \delta\text{PRACTICES} + \varepsilon\text{TRANSPARENCY}$$

Weighted sum aggregation is simpler than the other methods; however, given the unknowns still associated with the use of trust to measure trust in cybersecurity, it is deemed the best approach for initial efforts.

6.5 Conclusions

The IoT is a ubiquitous, intersecting set of technologies that is changing much of the modern world. Because of the IoT, systems and systems of connected systems of unprecedented scale and autonomy are possible. These capabilities are available to both the U.S. and its adversaries; in both cases, they increase what is possible as well as introduce new vulnerabilities for an adversary to exploit. This paper identifies some instances of the former, but its focus is not the latter.

Military and industrial systems historically have been closed systems; they have had limited to no outside exposure. Although it is feasible military systems will continue to be isolated from the Internet, industrial systems are becoming increasingly integrated with the IoT. A consequence will be that while military deployed systems will continue to have limited exposure to cybersecurity threats, the industrial systems used to design, develop, and produce them will become increasingly dependent on the Internet and thus increasingly exposed to cyber threats. Although it may remain possible to operate an infrastructure insulated from the Internet, the cost of doing so will likely be prohibitive. A deployed system may have limited exposure to the Internet, but its supply chain will be intertwined with the Internet.

At the same time that it is becoming harder to insulate a product supply chain from the IoT, the threats present within the IoT are becoming more sophisticated. The huge number of devices on the IoT, the limited processing power (and limited anti-malware capability) of many of those devices, and the complexity and heterogeneity of communications architectures that connect them make the IoT a target gateway for adversaries looking to breach networks. Developing capabilities to protect networks of such large scale and different topologies represents a possibly insurmountable challenge. At the same time, the avenues of attack have become more sophisticated and capable. The last decade has seen the advent of persistent agents and agents capable of achieving cyber-physical effects. The next decade may see attempts to implant counterfeit components with malware and intelligent malware.

Although the risk of malware introduced through a product supply chain has become an accepted threat, there is still no accepted methodology for assessing that threat. This paper includes a nascent methodology, using trust measurement, for assessing cybersecurity risk in a supply chain. It includes a system engineering analysis of a supply chain, identifies potential use cases, and identifies architectures, attributes, and methods of aggregation for applying principles of trust. However, the proposed methodology requires validation primarily because the actual product supply chain is decades away from

definition. Further analyses are recommended, including efforts to refine and validate the proposed methodology by applying it to a number of supply chains.

6.6 Related Efforts

During the course of our research, we identified a several related concurrent efforts focused on the same problem:

NIST has an ongoing effort to evaluate and protect against supply chain cybersecurity risks [104]. In August 2020, it released a tool to evaluate risk [105]. However, unlike our project, NIST efforts focus on preventing disruptions to the supply chain.

The National Telecommunications and Information Administration (NTIA) has launched the Software Bill of Materials to protect supply chains from introduced malware [106]. The NTIA effort focuses on identifying software with known vulnerabilities.

The Department of Defense (DoD) administers the Defense Federal Acquisition Regulation Supplement (DFARS) [107] and the Cybersecurity Maturity Model Certification (CMMC) [108] to identify requirements and policies on federal contractors to ensure compliance with cybersecurity best practices, including best practices for securing the supply chain.

6.7 Voice Assistant Cyber Scenarios and Use Cases

Users interact with a large amount of data and associated computing and processing through a variety of devices. Incorporating ambient computing through a voice interface for cyber capabilities opens the aperture on its value to a wider range of situations. Provided next are examples where this voice assistant interface could improve current user experiences.

6.7.1 Use Case #1: Low Cyber Skilled (Average Smart Home, IoT Devices Owner)

A homeowner with limited understanding of computing and the Internet installs a personal cyber assistant to help protect their devices and personal information. The personal assistant connects with the main router in the house; the homeowner provides the serial number. Once installed, the personal

assistant detects and inventories all the devices connected to the associated IP address. The personal assistant provides that inventory to the homeowner, who then provides confirmation for each device. Whenever a new device is detected for the first time, the personal assistant sends an alert and requests confirmation that the device belongs on the network.

In addition, the personal assistant continues to monitor the home network for suspicious activity and alerts the homeowner. Because the personal assistant is envisioned to consist of a primary application connected to a router and a client, which could be loaded into a phone, the personal assistant could provide remote alerts to the homeowner.

6.7.2 Use Case #2: Traveler's Assistant (High-end IoT Application Owner with Minimal Networking)

A tech-savvy businessperson spends a large amount of time traveling and is highly dependent on Internet and cellular. Their business requires the secure transmission of large amounts of proprietary and otherwise sensitive data. As a result, they are not comfortable transmitting that data from or receiving it at a hotel room, local coffee shop, or the airport.

The personal assistant can provide real-time assessments of the security of local networks and access information about potentially more secure networks nearby. In addition, the personal assistant can help the businessperson set up a virtual private network (VPN) or other security structures to ensure secure transmission of information.

6.7.3 Use Case #3: The Factory of the Future (Highly Skilled Cyber Defender)

The IT chief of a modern factory complex has to manage a sophisticated 5G-enabled network. The factory contains several robotic systems connected to a low-latency, highly reliable machine-to-machine communications network. It also employs several digital twins. These virtual systems do not have the low-latency requirements of the robotics systems; however, they depend on large volumes of data transported across broadband. Lastly, the factory deploys a state-of-the-art health-monitoring system to

ensure the health and safety of its workforce. Because this health-monitoring system transmits large amounts of personal data, it requires additional security to protect against compromise.

Each of these networks is implemented on a single physical infrastructure. The factory leverages 5G features like network slicing to divide this physical infrastructure into separate virtual networks, each with its own discrete QoS and security requirements. The IT chief can ask the personal assistant questions, and the assistant can provide the IT chief guidance in constructing network slices and providing the customized security required for each virtual network. In addition, the personal assistant can send alerts when anomalous traffic is detected and interface with the IT chief to assist in assessing whether an anomaly is indicative of a malicious act or actor or is just an anomaly.

6.7.4 Use Case #4: Ad Hoc Emergency Response Network (Field Operator)

In the aftermath of a hurricane, a large urban area is flooded and loses power and communications. Rescue teams operating in the affected area will need to bring their own communications equipment. However, although their radios support limited voice communications, which will help to coordinate activities, many of their devices require connectivity to the Internet, and their radio systems will not support the anticipated level of IP traffic.

To achieve robust IP communications, the first responders depend on an ad hoc mobile networking capability. Key features of this capability will be self-organizing networks and 5G-enabled device-to-device communications. The personal assistant will be able to assist in establishing and maintaining these ad hoc communications. The personal assistant will have the ability to interact with the onsite officer in charge of communications to identify objectives and set priorities (e.g., identifying which communications to drop first if networks become overburdened). It can also support stationing of unmanned aerial vehicles as relays.

7 TOWARD AN AMBIENT COMPUTING PARADIGM FOR IOT CYBERSECURITY: LOWERING THE COGNITIVE LOAD FOR USERS

7.1 Introduction

The IoT is becoming pervasive in the home, business, and mission-critical environments as more consumer, business, and industrial control devices are networked and IoT enabled, thus improving functionality in UFEs. UFEs are smart environments that have deep technological foundations, but strive to present only a portion of the technology to the user. This explosion of networked devices exposes users to many security vulnerabilities thus necessitating that those smart-system owners become more aware of the activity on and security of the devices on their network.

System owners likely do not have the skills necessary nor the time needed to continuously monitor their network using common open-source tools. Furthermore, network defenders are often inundated by the sheer number and diversity of devices and associated traffic and alerts. No one individual is likely to be able to fully use and comprehend the data that those tools present. We propose a system that uses ambient computing to facilitate network security monitoring and administration for smart and connected environments. Ambient computing refers to technologies that allow people to use a computer without realizing they are doing it [109]. This work is an extension of earlier work on CHASM [1]. In this chapter, we combine dynamic visualization of IoT networks with a natural language query interface enabled by voice assistants to simplify the process of providing information about the security state of the network to the casual user as well as the more seasoned network defender.

Using ML, the voice assistant integrates knowledge of the name, type, and function of devices on the network to communicate potential security concerns in a manner that is easily comprehensible and to recommend the appropriate actions needed. These capabilities will help improve the security of connected domains by providing the system owner a unique view into the IoT network and devices. When combined with other information, such as a topographical map, the enriched view could give the network owner the ability to monitor and protect their network with minimal cognitive load.

This chapter (1) demonstrates a voice assistant capability for IoT security applications that lowers the cognitive load on the end user, independent of their security and network skill level, (2) integrates trigger-based PCAP capability for deep packet inspections and IoT ML development, (3) integrates an ML-based discovery capability to characterize devices on a network, and (4) introduces a graph analytics capability for visualizing network connectivity and situational awareness.

The remainder of this chapter is organized as follows. Section 6.7 discusses voice assistant cyber scenarios and use cases. Section 7.2 describes voice assistant applications and related works on which our research builds. Section 7.3 presents an overview of the system architecture. Section 7.4 overviews voice assistant interface design and associated commands. Section **Error! Reference source not found.** d escribes the embedded underlying capability. Section 7.6 details the experimental evaluation we conducted. Section 7.7 explores voice assistant security concerns. Lastly, Section 7.8 presents conclusions and describes avenues for future work.

7.2 Voice Assistant Applications and Related Works

The voice assistant capabilities can be catalogued into two areas: (1) network monitoring smart assistance and (2) general voice user interface (VUI) applications.

7.2.1 Network Monitoring Smart Assistants

The growth of common IoT devices in UFEs and their respective networks opens up considerable risk possibilities. To combat this issue, network monitoring is useful for alerting and guiding network users through the security statuses of their devices. Narayanan et al. [110] developed systems that use smart assistance technology to regularly scan networks and users in search of needed backups, notifying a previously designated client in the event of positive detection. Without regular backups, systems become vulnerable to permanent data loss from a data breach or attack. With the combination of technology and human-driven backup systems, users can eliminate this vulnerability through double-layered authentication and the removal of sole human or computer error.

Additionally, with the constantly changing landscape of cybersecurity threats, early detection of such attacks has become more challenging. Even with advanced monitoring protocols, hackers can be present on a system for more than 100 days before being detected. Bassett et al. [111] developed a cybersecurity system that intakes data from a number of sources to have multiple collaborative smart agents complete a collection of network security tasks. This results in more data-informed decisions for security administrators, while simultaneously lowering their cognitive load and the potential for human error.

7.2.2 General VUI Applications

The diverse applications of VUI-enabled smart home devices can be further divided into two categories: convenience and practical. Convenience applications simplify lifestyles, without working to resolve any specific issue. Practical applications replace human activity to achieve necessary actions using the connected IoT devices.

The majority of convenience in both home and professional UFEs involves the replacement of traditional appliances with smart devices for easier use. Smart home assistants can control an interconnected network of Wi-Fi-connected light bulbs, alarms, thermostats, and other instruments without any practical need for automation [112].For instance, the Amazon Alexa smart home interacts with users through a VUI, and is able to control any device connected to Wi-Fi through either their own application or Wi-Fi-connected smart plugs. Although mainly implemented through VUI, Amazon Alexa's nature allows communication through e-text on mobile devices and on Amazon-provided screen displays. Other examples of the simplified lifestyle effects these VUI assistants provide include starting workouts, ordering online purchases, controlling smart homes, scheduling routines, and performing calculations.

The variety of practical applications includes uses with real demand and necessity in home and office use. An IoT-based fall-detection system proposed by members of the Department of Electrical and Computer Engineering at the University of Kentucky uses cameras and motion sensors to identify potentially life-threatening falls in homes with vulnerable occupants [113]. Once activated, the connected

smart assistant launches a VUI-based dialogue, prompting endangered users to notify the police or caregivers. Such application of the smart assistant relies heavily on the vocal component of the device because the use of a touch-based system may not be possible in a risk situation. This allows for more independent living, thereby saving money and time.

7.3 System Architecture

The ambient computing capabilities presented in this paper leverage both internally facing and cloud-based capabilities. This section provides an overview of the computing and network infrastructure and describes an end-to-end flow of a user query and its associated response through the system.

The voice interface provides access to a set of underlying cyber-focused capabilities exposed as edge-based RESTful services running on Raspberry Pi devices. We used two Raspberry Pi 3B devices, each having a four-core, 1.2-GHz Broadcom, 64-bit ARMv7 CPU, and 1 GB of RAM. The REST services return results via JavaScript Object Notation (JSON) to an ESXi server running Linux VMs. The Pi edge devices monitored collection of more than 70 IoT network devices connected to the same network, and a span port on the network switch forwarded all the network traffic from each device to the Raspberry Pi (Figure 7-1).

Figure 7-1 Raspberry Pi

We used Amazon Alexa for business (A4B) to create the user interface. Any Alexa user registered on our network can connect via Wi-Fi and access A4B services in the cloud. Upon joining, the user is authenticated and their voice commands go to the A4B cloud account and they can access related backend

services in the IoT testbed. (See Section **Error! Reference source not found.** for an overview of the b ackend embedded cyber services.) Figure 7-2 illustrates the data and information flowing though the network.

1. An ESXi IoT VM retrieves IoT device classification from an edge-deployed Raspberry Pi (IoT Pi) running ML-based device classifiers. The IoT Pi monitoring the collection of Pi sends JSON messages via a standard syslog port (UDP port 514).
2. The ESXi IoT VM stores the Feature File on the IoT Pi in the DMZ. All data collected here are processed and stored on a DMZ that can be reached by the Internet.
3. The user's voice commands are streamed by Alexa, using TLS 1.2, to an A4B account for processing by Amazon Voice Services (AVS). Amazon's Automatic Speech Recognition (ASR) processes the stream into text strings that are then forwarded to Amazon's Natural Language Understanding (NLU) system. NLU interprets the result and produces an intent. The service then routes the intent to one of our custom skills.
4. The Skill retrieves the IoT network data from the IoT host located in our DMZ.
5. The Skill formulates the raw data into Simple Speech Markup Language (SSML) text . The response system then takes the SSML and uses text-to-speech to generate an audio speech file. The resulting audio is then streamed back to Alexa.
6. The user sees a visualization of their IoT network from the IoT DMZ host.

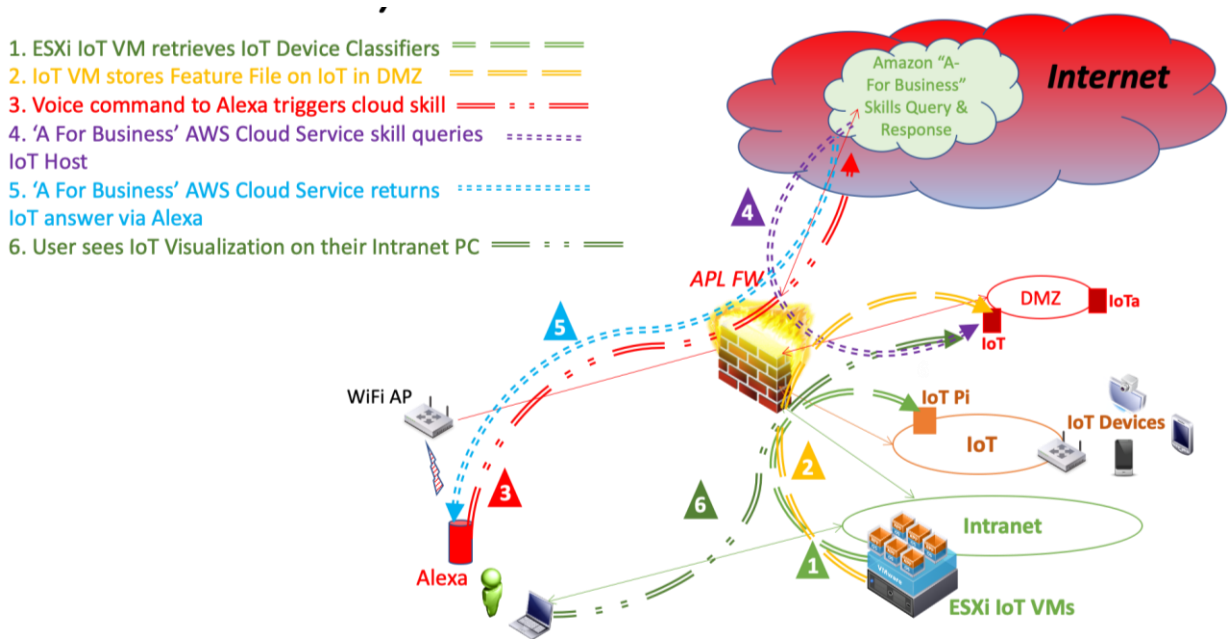

Figure 7-2 Alexa IoT Device and Network Status

7.4 Voice Assistant Interface Design

Although there are many existing frameworks for the creation of voice assistant capabilities, our cyber assistant was developed specifically as an Amazon Alexa skill. Amazon allows for the creation of custom cyber-assistant capabilities through the creation of Alexa skills, which can be published and then added to a specific Alexa device. This framework was chosen specifically because Amazon provides a robust Software Development Kit (SDK) for Alexa skill development and handles any necessary speech-to-text and natural language processing, allowing us to focus on the development of the VUI and underlying logic. Internally, an Alexa skill splits an individual piece of functionality into intents, each having frontend and backend pieces. A specific intent's frontend consists of the VUI, also known as "utterances" or invocation phrases that trigger this specific piece of functionality. The backend consists of an IntentHandler, and deals with the actual logic of the triggered intent [114]. The development of the Alexa skill thus began with the design of the voice interface and the intended functionality, with a focus on user experience.

The Alexa skill's goal is to provide an easy-to-use, easy-to-understand, natural cyber voice assistant; therefore, we gave particular care to the multiple ways in which a user could interact with the skill, as well

as the types of functionality provided by the skill; for instance, the voice interface needed to be able to handle cases where users may phrase the same request differently. Alexa specifically handled this by allowing multiple utterances to be tied to each intent. The specific intents and commands supported by the cyber assistant were also chosen by considering the kinds of information both technical and nontechnical users may want to know about their network.

The different commands supported by the Alexa skill can be divided into three categories. The first includes administrative commands that control the overall skill and the underlying ML capability. For instance, a user can start and stop the Alexa skill, change certain settings including the setting of certain triggers and alerts, and ask for help. On the backend side, a user will also be able to start, stop, and reboot the ML pipeline as well as check its status.

The second category of commands forms the bulk of the assistant's core functionality. These are the commands that query the ML pipeline for specific information about the user's network, including the number of devices, categories of the devices, newest device, etc. The goal of these commands is to give a user increased access to information about their network in a way that is user friendly and nontechnical so that any network owner may use this capability.

The final category of commands includes those aimed toward users with more advanced knowledge of cybersecurity and networking to support a range of users. These commands will give a network owner access to more technical information regarding their network and allow for the configuration of combinations of event triggers. This category also includes the commands that integrate other cyber capabilities that the skill supports, including the directed PCAP and network visualization.

This effort was focused on implementing this functionality as an Amazon Web Services (AWS) Lambda-hosted Alexa skill, with corresponding utterances and intents. Table 7-1 briefly summarizes the commands that the skill currently supports, including sample invocation phrases and intended behavior.

Table 7-1 Core Functionality

Intent Request Name	Purpose	Sample Utterances	Slots (optional)	Returns
GetNetworkSummaryIntent	Provide an overview of connected devices and status of network.	"Tell me about my network." "Summarize my network." "Tell me a summary."		Number of total connected devices, number of devices active in the last 24 hours, and last device added
GetDevicesSummaryIntent	Provide a summary about the connected devices and their categories.	"How many devices do I have?" "What devices are on my network." "Tell me about the devices on my network."		Number of connected devices and number of connected devices in each category
GetNewestDeviceIntent	Provide an overview of the newest device on the network.	"What's the newest device?" "Get the newest device." "Tell me about the newest device."		The newest device added to the network, the time it was added, and the type of device it is
GetLatestDevicesIntent	Provide a list of devices added after a certain specified time. If no time is specified, default to last time the skill was used.	"Any new devices since yesterday?" "Any new devices?" "How many new devices do I have since last week?"	Amazon.DATE Amazon.TIME	A list of devices added to the network after the specified time and the category breakdowns
GetNetworkMapIntent	Provide to the user a graphical representation of the network and devices connected to it.	"Show me my network map."		A user-viewable network map with labeled nodes and edges representative of the network and activity on the network
GetPacketDataIntent	Provide the user PCAP data collected from the network.	"Capture data from the network for me."		PCAP data from *tcpdump*
HelpIntent	Suggest commands to ask Alexa about your network.	"Help." "What can I ask?" "What can you tell me about my network?"		A list of commands available to ask the Alexa skill

7.5 Embedded Underlying Capability

The voice interface provides an interface to a set of underlying cyber-focused capabilities. This section describes the collection of applications exposed to the user.

7.5.1 IoT Device Discovery and Classification

IoT discovery is foundational to providing good cybersecurity because a user needs to have a good accounting and understanding of the devices on a network to be able to protect and secure them. Furthermore, because IoT networks are dynamic by nature, the landscape will change often and thus require constant monitoring. Therefore, automated methods are needed to maintain situational awareness of networks. Our IoT device discovery and classification capability can autonomously identify IoT device types. Although there are many ways to perform IoT discovery, each has its strengths and limitations. For example, with MAC addresses, one can determine the device manufacturer. This approach is relatively straightforward, but it also can be spoofed or be representative of another device in the network chain.

Therefore, the focus of this work is to explore discovery, profiling, and verification of IoT devices solely based on their network behavior or other information contained in individual or constrained groups of packets [1].

To support this goal, an ML model was trained to analyze packet sequences and predict what type of IoT device each packet sequence came from [2]. To construct the training data set, traffic from more than 60 IoT devices was collected, grouped by MAC address, and arranged in time order. The data from each device was then transformed into sequences of 20 packets each. The values in these sequences were then normalized, one-hot encoded, and labeled according to device category (see Table 7-2), and an ML model was trained on that data. The result was a model that can take a 20-packet sequence of network data from a device and provide a prediction as to that device's category.

Table 7-2 Supporting Metadata

Metadata Tag	Description
“predicted_category”: “television” # This line and the next two are a summary of the category_scores	Category with the highest evaluated score
“confidence_level”: “Low”,	Confidence level of the prediction
“confidence_percent”: 49,	Score of the prediction
“device_id”: “b8:27:eb:3d:c2:a2”, #	Unique identifier for tracking devices inside the ML pipeline
“first_seen_ts”: #	Unix timestamp in ms (This is the time that the device was first detected on the network.)
“first_seen_utc”: “2020-06-05 14:33:52.013460+00:00”, #	Human-readable version of first_seen_ts
“last_seen_ts”: 1591887289012.231, #	Unit timestamp in ms (This is the most recent time that the device was seen on the network.)
“last_seen_utc”: “2020-06-11 14:54:49.012231+00:00”, #	Human-readable version of last_seen_ts
“ground_truth”: {	Ground truth for the MAC address (if known)
“iot_testbed_alias”: “”,	Testbed alias for a device (if available)
“iot_testbed_category”: “”	Category with the highest evaluated score
“ip_addresses_used”:	IP addresses that the device was seen using
“mac_address”: “b8:27:eb:3d:c2:a2”, # MAC address of the device	MAC address of the device
“mac_manufacturer”: “Raspberry Pi Foundation”, #	Possible manufacturer of the device, based on first three octets of MAC address

IoT device categories are as follows:

- Television
- Assistant
- Unknown
- Hub
- Audio device
- Clocks
- Router
- Mobile device
- Triggers and switches
- Cellular device

- Thermostat
- Camera
- VPN router
- Human interface device

The model was deployed to run in streaming mode on a Raspberry Pi, resulting in a stream of output predictions. A dashboard was created to aggregate and display these predictions (see Figure 7-3).

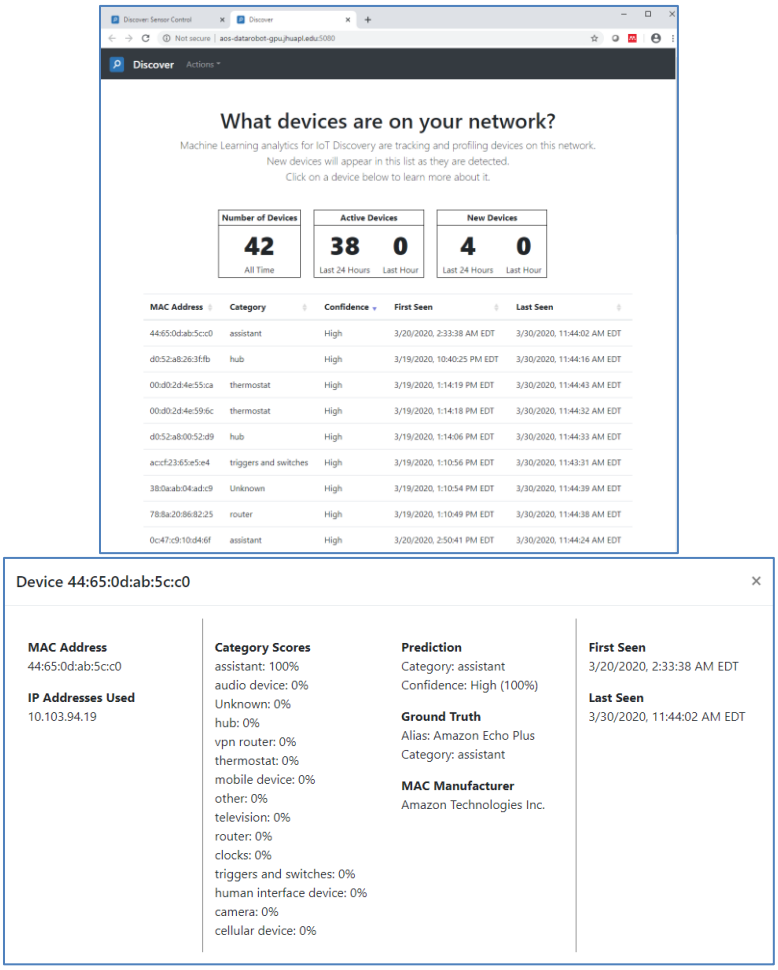

Figure 7-3 Dashboard to Aggregate and Display Predictions

7.5.2 Integrated PCAP

The key element to providing cyber protections to a system is to have timely relevant data with labels related to the time, event, and situation that generated the event of interest. Monahan et al. states that

organizations using PCAP as part of their normal toolsets were more confident in the information they received about their environments and therefore were better prepared to protect them. Specifically, they had:

- Shorter breach detection and response time
- More confidence in their workflows and processes

Therefore, the use of PCAP is key to providing useful cybersecurity in a network [115] [116]. However, gathering these data is hard because the storage capacity for full PCAP is expensive. In addition, being able to capture data coincident with a specific incident can be difficult because of the unpredictability of cyber events. To meet this challenge, we integrated a trigger-based PCAP capability that captures data based on passed-in configuration parameters. This gives the users the ability to perform offline deep packet inspection on data that are coincident with an event, and have relevant, insightful metadata stored with the data itself. To provide PCAP, we integrated *tcpdump* [117], a software program that allows the user to capture network packets being transmitted or received over a network to which a device is attached.

The PCAP capability is composed of two parts: an Edge Sensor Service that runs on each network sensor and a Sensor Management Service that runs in the cloud.

The network sensors are currently deployed as Raspberry Pis, each running the ML pipeline as well as a copy of the Edge Sensor Service. The Edge Sensor Service sends periodic heartbeat messages to the Sensor Management Service, indicating the sensor's name, location, status (e.g., CPU utilization, CPU temperature, remaining disk space), and universally unique identifier. The Edge Sensor Service also listens for commands from the Sensor Management Service and manages the execution of those commands.

The Sensor Management Service listens for heartbeat messages from the network sensors and uses that data to determine which sensors are available for tasking. The Sensor Management Service exposes

a REST Application Programming Interface (API) that allows a user (or other software program) to perform the following actions:

- Obtain a list of all available sensors and their status.
- Reboot a specific sensor.
- Start the ML pipeline on a specific sensor.
- Instruct a specific sensor to capture packets via *tcpdump*. The full range of *tcpdump*'s filtering capability is available, including the ability to filter by IP addresses, IP address ranges, MAC addresses, port numbers, and protocols.
- Obtain the status of ongoing and completed PCAPs on a specific sensor.
- Retrieve completed PCAPs (in PCAP file format) from a specific sensor.

Because the Sensor Management Service exposes the above functionality through the REST API, it could easily be connected to other software in support of several novel use cases:

- When a new device is detected on the network, automatically capture some of its network traffic for later analysis.
- Allow the user to ask Alexa for a PCAP from a device by name. For example, "Get me some network data from that new thermostat that showed up on the network today."
- To support a cybersecurity analyst with some knowledge of ML, build additional APIs for retraining and redeploying the ML pipeline on the sensor, which would then allow an Alexa conversation such as:

  "Alexa, capture packets from that unknown device that showed up on my network today."

  "Alexa, that device is a thermostat."

  "Alexa, retrain the machine learning model with the new data from the thermostat."

  "Alexa, is my new model ready?"

  "Alexa, deploy my new model to sensor X."

- For the home user, a similar set of steps could be performed automatically because the average home user would not know about the intricacies of ML model training. An example conversation might be:

  Alexa: "There is one new device on your network this morning. I can't figure out what it is."

  User: "Oh, that's my new smart refrigerator."

  Alexa: "Got it. I'll update my list." (In the background, Alexa retrains the existing ML model to be able to recognize smart refrigerators, and automatically deploys the new model.)

- In addition to training models to categorize IoT devices, this capability could be used to learn "normal" patterns of life of different IoT devices on the network, and alert the user if any devices deviate from those patterns. An example conversation might be:

  User: "Alexa, how's my network looking today?"

  Alexa: "Your back-porch security camera started acting like a router at 3 a.m. today. You might want to get that checked out."

7.5.3 Network Mapping

As previously discussed, the IoT device testbed implements a pipeline model in which data are ingested from the IoT devices via a sequential process of PCAP, feature extraction, model training, and model inference. To gain a sense of network situational awareness, an additional processing layer is needed that is responsible for network visualization, analytics, and general user interaction. This layer, termed IoTA for IoT Analytics, primarily taps into the pipeline after feature extraction is complete; however, it was designed to fuse information across several points in the pipeline (e.g., in the model training and execution phases). The rationale behind this design is to enable the user to aggregate and view IoT data from a low-level (device connections) point of view while still leveraging higher abstractions of data processing to provide a more complete understanding of the device network while minimizing user cognitive loads. IoTA leverages the IoT ML processing pipeline to enrich the visual rendering of low-

level network data with context-aware inferences, such as device type labeling and potential threat quantification.

As with most computer networks, the IoT device testbed generates large amounts of data. When implementing a visualization interface, one critical decision is where to position data ingest within the processing pipeline and where to implement the frontend to maximize deployment flexibility while minimizing local compute resources. For this reason, the IoTA architecture consists of two processes: the backend processor and the frontend renderer. The backend processor is designed to run in a convenient location behind the enterprise firewall, where access to network and computing resources is most optimal. This processor gathers and fuses information from the various levels within the processing pipeline and then presents a streamlined collection of data to the frontend for rendering. This process is intended to run on a server with sufficient computational resources to enable real-time transformation of data. In determining how best to aggregate data for visual representation, the system identifies repeated communications within a given timeframe specified by the frontend and generates a data structure that contains a summary of these communications. The level of detail of the summary is dependent on information from the frontend renderer about the context of the visualization (e.g., topology visualization, anomaly detection, network querying).

The frontend renderer runs in a standard web browser for maximum flexibility. This enables practically any network-enabled computing device to interface with the processing pipeline across the enterprise firewall. Because of the limited computational resources available at the frontend, the backend processor essentially compresses the available data through context-based slicing and aggregation operations. To minimize operator cognitive load, the interface of the renderer is intentionally minimalistic because research has shown that cluttered interfaces greatly increase operator distraction and negatively influence their focus on other tasks [118]. The renderer is divided into four main sections in which content changes are context sensitive. In this way, only the information pertinent to the task is displayed.

Figure 7-4 shows the primary interface for IoTA, where the display is segmented into (1) data transformations and rendering controls, (2) network device listing with filtering, (3) micro network rendering, and (4) macro network rendering.

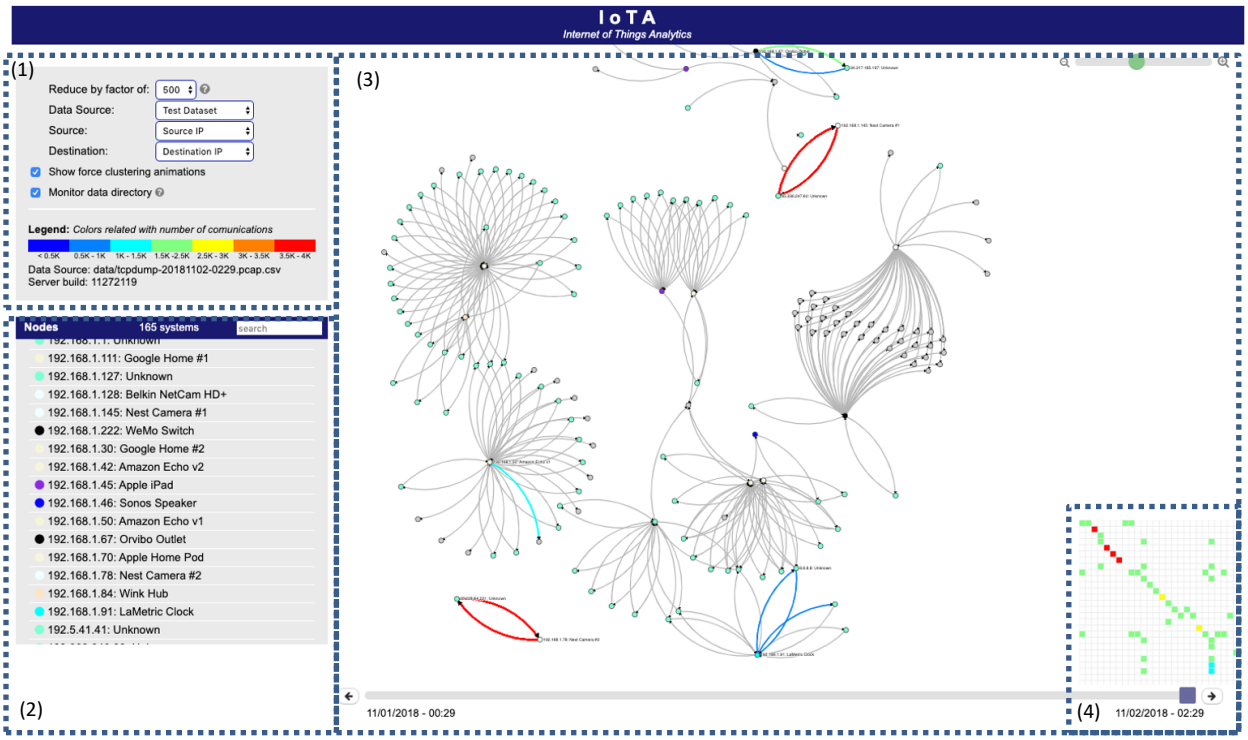

Figure 7-4 IoTA Frontend Renderer with Fixed Context Areas: (1) data transformation and rendering, (2) network device listing and filtering, (3) micro network rendering, and (4) macro network rendering

The primary visualization method used by IoTA is force-directed clustering with semi-radial layouts that minimize link crossings. After careful consideration and experimentation, this method proved to be the most robust in presenting device network information to highlight relationships between devices while minimizing clutter and occlusion. Figure 7-5 presents a screenshot of this visual layout. The display supports zoom operations and an infinite canvas to scale well with large network collections, while using minimal resources on the client device. All node positions can be altered by simply clicking and dragging nodes across the canvas. During this operation, the layout manager continuously updates the force-directed layout through smooth animations. This aspect is important for preserving operating context as the network analyst interacts with the visualization [119].Furthermore, node positions can be correlated

to physical spatial coordinates and overlaid on spatial maps that can represent building schematics, geographic maps, or any other spatially relevant maps, as illustrated in Figure 7-5.

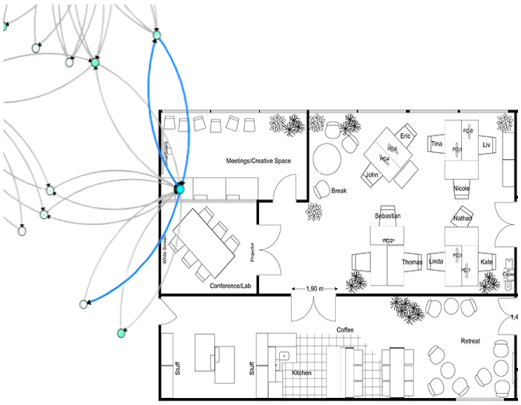

Figure 7-5 Node Position Correlation with Spatial Mapping

The visualization employs a streamlined list of visual cues, such as node/link position and size, node/link color, and dynamic labels, to always present an uncluttered interface that minimizes operator cognitive load. For example, node labels are selectively displayed based on criteria such as communication frequency, external network device connections, and alerts derived from the model inference output of the pipeline. Additional information about devices can be obtained by simply hovering the mouse over the device node, with click events activating various charts that capture communication patterns to and from the node in question. The visual renderings automatically update with new data while still presenting a simple horizontal scroll bar to scrub backward in time. As new network nodes are loaded, smooth animation sequences help preserve context as devices pop in and out of existence based on the specified window size and aggregation level.

The Data Transformation and Rendering section houses controls to adjust which data sets are loaded for rendering. For example, the user is able to adjust the aggregation factor for communications between the same source and destination units, the attributes to be considered source and destination for the node-link model, and when and how often to update the display. Depending on the task the network analyst wishes to the perform, they can select from preconfigured source and destination combinations

that are designed to highlight important aspects of the network, such as identifying unusual system-to-system communications or uncharacteristic communication bursts for the device involved or determining the type of devices with which systems communicate.

The Network Device Listing and Filtering section of the renderer displays lists of information about the network (e.g., device IP address, device name, device type with associated color encodings for the renderer). Lists can be sorted and filtered through inexact queries, which enables rapid information retrieval. In addition, the visual representation in the Macro Network Rendering section is always mirrored by the filtered lists in this section. This approach preserves application context as the network analysist explores the network through query operations.

The Macro Network Rendering section employs various visual renderings to provide a macro (or birds-eye) view of the network under study. The purpose of this section is to provide the analyst with a quick method of assessing the global behavior of devices on the network. In the view shown in Figure 7-4, a matrix heat map is used to show the connections between all devices as well as the amount of communication between devices through cell color. Hovering the mouse over each cell provides more details about the communication between the associated devices, with click events leading to charts that capture historical communication patterns between devices.

The IoTA client and server models enable additional functionality to improve human-machine interfacing. The server processor supports extensions to external devices to control the transformation and aggregation stages, which in turn affects what is rendered on the client. An extension was built to support interfacing with the AVS API so network analysts could perform natural language queries against the device network and have the filtered network displayed as instructed on the client renderer. The use of natural language queries combined with uncluttered visual renderings has the potential to significantly improve the effectiveness of network defenders and casual users alike.

7.6 Experimental Evaluation

Because the goal of this work was to reduce the cognitive load on the user and make the system owner/user more effective, we conducted a limited study that compared aspects of preforming commands associated with each utterance to evaluate the cognitive load and relative difference in difficulty performing a command manually versus using the voice assistant. Table 7-3 compares the cognitive load needed for a skill and a comparative measure of reduction of the cognitive load achieved by leveraging the voice assistant.

Table 7-3 Preliminary Evaluation and Comparison of Intents

Intent Request Name	Returns	Tools Needed To Perform Intent	Level of Difficulty To Perform the Intent – Manual (1–5)	Level of Difficulty To Perform the Intent – Voice (1–5)	Time (voice assistant)
GetNetworkSummaryIntent	Number of total connected devices, number of devices active in the last 24 hours, and last device added	Wireshark	4	1	15 seconds
GetDevicesSummaryIntent	Number of connected devices and number of connected devices in each category	Wireshark	4	1	22 seconds
GetNewestDeviceIntent	The newest device added to the network, the time it was added, and the type of device it is	Wireshark	3	2	20 seconds
GetLatestDevicesIntent	List of devices added to the network after the specified time and the category breakdowns	Wireshark, router admin console	3	2	25 seconds
GetNetworkMapIntent	User-viewable network map with labeled nodes and edges representative of the network and activity on the network	*tcpdump*, python, tableau	5	3	40 seconds

Intent Request Name	Returns	Tools Needed To Perform Intent	Level of Difficulty To Perform the Intent – Manual (1–5)	Level of Difficulty To Perform the Intent – Voice (1–5)	Time (voice assistant)
GetPacketDataIntent	PCAP data from *tcpdump*	*tcpdump*	3	4	Variable based on *tcpdump* query

To formulate Table 7-3, we asked a cyber subject matter expert and a typical home IoT device owner to quantitatively evaluate the level of difficulty to perform each of the skills manually and with the voice assistant using a five-level Likert scale. We also measured the time needed to perform the skill using the voice assistant. This timeframe was measured from the moment the user began speaking to invoke the Alexa skill to the moment when Alexa stopped speaking her response to the user. The measurement for each intent also included the time it took for the intent to reach out to the REST API to receive a network summary, although the skill caches that data within sessions for increased efficiency when multiple intents are invoked per session.

From these preliminary results we see that the voice can lower the relative level of difficulty to perform a command. Specifically, intents that provide a summary of the networks were most impacted. As we continue to add skill and intents, we will expand the testing to more completely evaluate the effectiveness of the voice assistant

7.7 Security Concerns

Smart Personal Assistants (SPAs) will change the way users interact with UFEs; however, they also present significant security and privacy issues. Keeping these applications safe without sacrificing the benefits of efficiency is key. Some security concerns of SPAs include synthesized speech, voice squatting, weak authentication, and profiling [120].

7.7.1 Synthesized Speech

Smart assistant technology lacks the ability to recognize whether the user is the legitimate owner of the device or an illegitimate user making a request to the device. This weakness makes SPAs unable to detect whether inaudible sounds or signals are requests to the device. As a result, radio signals or other sound waves from other technology can interfere with SPAs and cause the smart assistant technology to approve illegitimate requests. For example, a Burger King TV ad requests Google Home to open up Wikipedia and read information about the "Whopper hamburger."

7.7.2 Voice Squatting

Voice squatting is a method wherein a threat actor takes advantage of or abuses the way a skill or action is invoked. This threat can be activated when a user prompts a request, but receives a response that can be a potential threat to the owner. For example, a user can prompt Alexa by saying "Alexa, open Capital One" to run the Capital One skill. Then, an unknown threat actor can potentially create a malicious app with a similarly pronounced name, such as Capital Won.

7.7.3 Weak Authentication

To ensure the safety of users and their networks, proper implementation of authentication controls is necessary for the usability of SPAs. Vulnerabilities regarding weak authentication include inadequate lockout implementations and the inability to determine one authenticated user from another. An example of weak authentication is when an SPA confuses an unauthenticated user as the owner of the device. For example, the daughter of an owner prompts an SPA to order a dollhouse; as a result, the SPA takes her order as a legitimate request to the device and orders the dollhouse without authenticating the identity of the user.

7.7.4 Profiling

Beyond the issue of authorization, SPA users face the threat of profiling. Profiling is when data are collected about the user's personal information such as their interests, behaviors, and preferences. There

are three main types of profiling: en route profiling, profiling by third-party developers, and profiling by SPA providers. Attackers leverage *en route profiling* to determine a user's presence during traffic analysis. The attacker can then use these techniques to conduct threats that are more serious. *Profiling by the third-party developers* involves the sharing of valuable personal and network information, resulting in malicious apps that combine various data the user has shared, thereby creating a complete profile of the user that may compromise the privacy of the SPA owner. In the *profiling by SPA providers* threat, the SPA makes compromises to uphold the user's privacy by collecting sensitive data such as the user's conversations, online search habits, and other information stored on the SPA. The SPA then may have access to personal data of the SPA user, which poses a security and privacy risk to the user [120].

Addressing security concerns is beyond this specific effort and therefore an area of focus for future work.

7.8 Conclusions and Future Work

7.8.1 Conclusions

This chapter summarizes ongoing work toward integration of voice assistant technology to lower the cognitive load placed on a user to monitor and maintain their smart environments by providing access to complex capabilities in a natural way. We developed a proof-of-concept cyber voice assistant as an Alexa skill, which implements a set of intents that allow users to access information on their networks and the associated devices. Together, these intents make up the most basic user commands, which provide a network owner awareness of their connected devices. We evaluated the effectiveness of the voice assistant by capturing qualitative perceptions on the ease of use of an intent and quantitatively capturing the time to perform an intent from a limited user base.

The advantage of integrating voice capability in a cyber personal assistant is that it can guide the user to ask the right questions the right way despite limited expertise. We understand that the ability to use natural language processing to support dialogue using a specialized language in which the machine may

be more proficient than the user will be a real challenge to achieve. Nonetheless, even with limited voice, the analytics that enable even rudimentary monitoring of home networks integrating voice is a potentially large step in securing the IoT as a whole.

7.8.2 Future Work

In future work, we will continue developing the capabilities of the Alexa skill to add more intents and incorporate more in-depth uses of visuals. Visuals are useful for enriching a user's experience in combination with the voice assistant, and there are many devices that support screen visuals. We intend to enhance the display of connected devices through a dashboard to assist in following along with the information returned from the Alexa skill about the network.

In addition to providing statistics about the network and its devices, we believe there exist other security capabilities worth integrating into the Alexa skill intents. The same analytics used for IoT discovery can be used to identify threats. For instance, if the system classifies device X as a camera but then later classifies it as a thermostat, this may be indicative of a compromised device. Incorporating these findings allows users to remain aware of and monitor the integrity of their devices to remain protected.

Beyond work on the assistant itself, we would also like to expand our understanding of how the voice assistant for cyber can lower the cognitive load for a wider range of users. Our current testing methods rely on feedback from a cybersecurity expert and a typical home IoT owner; however, this is not a complete profile of all possible users. To address this, we intend to perform a more complete study to evaluate the efficacy of the voice assistant in lowering the cognitive load on a user by expanding the test user base to include a broader set of participants that span skill level and experience. Lastly, we plan to address the security vulnerabilities of the virtual assistants, port the edge devices to more capable hardware to support more in-depth integration of anomaly detection capabilities, and provide support for contextual conversations, thus allowing follow-on utterances and responses to be conversationally dependent on a previously asked question.

8 SUMMARY

In this dissertation we have explored many aspects of security and privacy in the IoT space. We examined IoT security issues in the home and introduced the CHASM concept. We extended CHASM to the IoT edge deploying machine learning algorithms coincident to where data is generated. We examined the cybersecurity analytics capability in response to the threats from the perspective of a large organizational entity. We examined the use of machine learning to identify botnet activity in IoT devices and networks. We researched and introduced a proposed framework for assessing IoT-motivated cybersecurity risk and its impact on supply chains. Lastly, we examined ambient computing and its utility in reducing the cognitive load on users when protecting IoT-enabled networks. Lastly, we examined applications of these research topics to projects at JHU/APL.

9 REFERENCES

1. Chavis, J.S., L.A. Watkins, A.L. Buczak, and A. Rubin, Connected Home Automated Security Monitor (CHASM): Protecting IoT Through Application of Machine Learning. In: 10th Annual Computing and Communications Workshop and Conference (IEEE CCWC 2020). Las Vegas, NV: IEEE Explore; 2020:6.

2. Chavis, J.S., A. Kunz, L.A. Watkins, A. Rubin, and A.L. Buczak, A Capability for Autonomous IoT System Security : Pushing IoT Assurance to the Edge. In 2nd Annual Workshop on Assured Autonomous Systems (IEEE WAAS 2020). San Francisco, CA, United states: IEEE Explore; 2020:6.

3. Chavis, J.S., D. Syed, O. Brooks, A. Rubin, L. Watkins and A. Buczak, A Proposed Trust Model for Assessing Cybersecurity Risk in a Supply Chain Considering IoT 's Impact. In: ; 2021.

4. Chavis, J.S., A. Kunz, S. Fu et al., A Voice Assistant for IoT Cybersecurity. In IEEE Integrated STEM Education Conference 2021. Princeton, NJ: IEEE Explore; 2021.

5. Hegde, M., J.S. Chavis, G. Kepnang et al., Identification of Botnet Activity in IoT Network Traffic Using Machine Learning. In: VALENCIA, SPAIN: IEEE Explore; http://intelligenttech.org/IDSTA2020/.

6. Chavis, J.S. (JHU/APL), D.P. Syed (JHU/APL). Envisioning Cybersecurity Analytics for the Internet of Things. In: IEEE, ed. *IEEE 5G World Forum*. Mumbai, India: IEEE Explore; 2020 doi:10.1109/5GWF49715.2020.9221018.

7. "Home - Johns Hopkins Institute for Assured Autonomy." n.d. Available at https://iaa.jhu.edu/ [accessed 27-December-2020].

8. Chavis, J.S. (JHU/APL), "Internet of Things (IoT Cybersecurity Analytics Capabilities Vision Paper, Version 1.0," AOS-20-0035, February 2020.

9. Internet Society, "The Internet of Things: an Overview – Understanding the Issues and Challenges of a More Connected World," 12 October 2015. Available at https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf

10. Boeckl, Katie et al., NISTIR 8228, "Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks," June 2019. Available at https://doi.org/10.6028/NIST.IR.8228

11. https://tools.ietf.org/html/draft-rizzo-6lo-6legacy-03

12. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060

13. HP Security Labs Bromium website. Available at https://www.bromium.com/ [accessed 30-December-2019].

14. IEEE, "Toward a Definition of the Internet of Things, Revision 1," May 2015. Available at https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

15. Recommendation ITU-T Y.2060, "Overview of the Internet of Things," International Telecommunications Union – Telecommunications Standards Sector of ITU, June 2012. Available at https://www.itu.int/rec/T-REC-Y.2060-201206-I [accessed 23-January-2020].

16. Columbus, Louis, "2018 Roundup of Internet of Things Estimates and Market Forecasts," Forbes, 13-December-2018. Available at https://www.forbes.com/sites/louiscolumbus/2018/12/13/2018-roundup-of-internet-of-things-forecasts-and-market-estimates/#5414222e7d83

17. IIRA 1.9, "The Industrial Internet of Things Volume G1: Reference Architecture," Industrial Internet Consortium, 19 June 2019. Available at https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

18. Internet Society, "The internet of things: an overview," *Internet Things A Rev.*, p. 53, October 2015.

19. Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)," 2018. [Online]. Available at https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. [accessed 01-December-2018].

20. Bain and Company, "IoT Market Size Will More Than Double," Press Release, 2018. [Online]. Available at https://www.bain.com/about/media-center/press-releases/2018/bain-predicts-the-iot-market-will-more-than-double-by-2021/. [accessed 14-May-2019].

21. SeungJin Lee, "Hacking, Surveilling, and Deceiving Victims on Smart TV," *Blackhat USA*, 2013.

22. Levy, J., "Sophoslabs 2019 Threat Report," 2018.

23. Zeng, E., S. Mare, F. Roesner, and P. G. Allen, End User Security and Privacy Concerns with Smart Homes End User Security and Privacy Concerns with Smart Homes, 2017. Available at https://www.usenix.org/conference/soups2017/technical-sessions/presentation/zeng.

24. R. Contu, P. Middleton, S. Alaybeyi, and B. Pace, "Forecast : IoT Security, Worldwide," 2018.

25. Granjal, J., E. Monteiro, and J. Sá Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Communications Surveys and Tutorials*, 17(3), pp. 1294–1312, 2015. Available at https://ieeexplore.ieee.org/document/7005393 [accessed 11 January 2021]

26. Palani, K., E. Holt, and S. Smith, "Invisible and Forgotten: Zero-Day Blooms in the IoT," n.d. Available at https://www.cs.dartmouth.edu/~sws/pubs/phs16.pdf [accessed 11-December-2018].

27. Martin, C., "North American Consumers To Have 13 Connected Devices," *Connected Thinking*, 12 June 2017. [Online]. Available at https://www.mediapost.com/publications/article/302663/north-american-consumers-to-have-13-connected-devi.html. [accessed 12-May-2019].

28. Knud Lasse Lueth, "State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating," *IoT Analytics*, 2918. [Online]. Available at https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/. [accessed 12-May-2019].

29. Ion, Florence, “How to better secure your smart home,” *engadget*, 2018. [Online]. Available at https://www.engadget.com/2018/08/14/how-to-secure-your-smart-home/ [accessed 14-May-2019].

30. Mosenia, A., “Addressing Security and Privacy Challenges in Internet of Things,” 2017 Available at https://arxiv.org/abs/1807.06724

31. Shahid, M., G. Blanc, Z. Zhang, H. Debar, and M. R. Shahid, “IoT Devices Recognition Through Network Traffic Analysis,” pp. 5187–5192, 2018.

32. Zhang, W., Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhe, “HoMonit: Monitoring Smart Home Apps from Encrypted Traffic,” *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security*, pp. 1074–1088, 2018.

33. Luo, R.H., W. Chen, and P. Venkateswaran, “Profiling and Discovering Internet of Things Devices,” Baltimore, 2018.

34. Meidan, Y., M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, “Detection of Unauthorized IoT Devices Using Machine Learning Techniques,” 14 September 2017.

35. Smith, S. W., The Internet of Risky Things : Trusting the Devices That Surround Us, O’Reilly Media, September 2017.

36. Pego, P.R.J., and L. Nunes, “Automatic discovery and classifications of IoT devices,” *Iber. Conf. Inf. Syst. Technol. Cist.*, pp. 1–10, 2017. https://doi.org/10.23919/CISTI.2017.7975691.

37. Leung, L.Y., U. Tunku, and A. Rahman, “Universal IoT Devices Discovery System Using Protocol-Independent Network Flows Characteristics,” Kampar Malaysia, 2018.

38. Shen, J., Yi. Li, B. Li, H. Chen, and J. Li, “IoT Eye An Efficient System for Dynamic IoT Devices Auto-discovery on Organization Level,” *Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017*, vol. 7, pp. 294–299, 2017.

39. Pacheco, J., and S. Hariri, “Anomaly behavior analysis for IoT sensors,” *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 4, p. e3188, Apr. 2018.

40. Hodo, E., X. Bellekens, A. Hamilton, P-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, “Threat analysis of IoT networks using artificial neural network intrusion detection system,” *2016 International Symposium on Networks, Computers and Communications (IEEE ISNCC’16)*, pp. 1–6, 2016.

41. Bezawada, B., M. Bachani, J. Peterson, H. Shirazi, and I. Ray, *IoTSense: Behavioral Fingerprinting of IoT Devices*, n.d. Available at https://arxiv.org/pdf/1804.03852.pdf

42. Miettinen, M. et al., “IoT sentinel automated device-type identification for security enforcement in IoT,” in *Proc. 2017 IEEE 37th International Conf. on Distributed Computing Systems (ICDCS)*, pp. 2177–2184. IEEE (2017)

43. Buczak, A. L., P. A Hanke, G. J Cancro, M. K Toma, L. A Watkins, and J. S Chavis. "Detection of Tunnels in PCAP Data by Random Forests," *Proc. of the 11th Annual Cyber and Information Security Research Conference*, pp. 1–4, 2016. https://doi.org/10.1145/2897795.2897804.

44. "Home – Anaconda." n.d. Available at https://www.anaconda.com/ [accessed 16-May-2019].

45. "Multiclass Decision Forest – Azure Machine Learning Studio | Microsoft Docs." n.d. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-forest [accessed 15-May-2019].

46. Breiman, L., "Random Forest," Berkeley, CA, 2001. Available at https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

47. "Multiclass Neural Network – Azure Machine Learning Studio | Microsoft Docs." n.d. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-neural-network [accessed 15-May-2019].

48. Sivanathan, A., D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, A., and V. Sivaraman, "Characterizing and classifying IoT traffic in smart cities and campuses," *2017 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS*, pp. 559–564, 2017. https://doi.org/10.1109/INFCOMW.2017.8116438.

49. *Real-Life Use Cases for Edge Computing - IEEE Innovation at Work*. (n.d.). Retrieved from https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/ [accessed 05-February-2020].

50. Ronanki, R. (2018). Artificial intelligence for the real world. *Harvard Business Review*, January-February, pp. 108–116.

51. Meidan, Y., M. Bohadana, A. Shabtai, M. Ochoa, N.O. Tippenhauer, J. D. Guarnizo, and Y. Elovici. (2017). Detection of Unauthorized IoT Devices Using Machine Learning Techniques. Retrieved from http://arxiv.org/abs/1709.04647.

52. Shahid, M., G. Blanc, Z. Zhang, H. Debar, and M.R. Shahid. (2018). IoT Devices Recognition Through Network Traffic Analysis, 5187–5192. Available at https://doi.org/10.1109/BigData.2018.8622243

53. Truong, J., and T. Davison. (2017). IoT Discovery: Device Profiling. Laurel, Maryland.

54. Sivanathan, A., D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman (2017). IEEE Conf.Computer Communications Workshops. "Characterizing and Classifying IoT Traffic in Smart Cities and Campuses," https://ieeexplore.ieee.org/document/8116438

55. Bai, L., L. Yao, S. Kanhere, X. Wang, and Z. Yang (2018). Automatic Device Classification from Network Traffic Streams of Internet of Things. Retrieved from https://arxiv.org/abs/1812.09882

56. Bezawada, B., M. Bachani, J. Peterson, H. Shirazi, and I. Ray (2018). IoTSense: Behavioral Fingerprinting of IoT Devices. Retrieved from https://arxiv.org/abs/1804.03852

57. Miettinen, M., S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma (2016). IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT. Retrieved from https://arxiv.org/abs/1611.04880

58. Libtins C++ packet processing library (Open Source) http://libtins.github.io/

59. https://www.tensorflow.org/lite/convert/rnn, accessed 03-February-2020.

60. "National Critical Functions Set | CISA." n.d., available at https://www.cisa.gov/national-critical-functions-set [accessed 7-May-2020].

61. NIST SP 1500.201, E. R. Griffor, C. Greer, D. A. Wollman, and M. J. Burns, "Framework for cyber-physical systems: Volume 1, Overview," 2017.

62. Pew Research Center Internet and Technology, "Mobile Fact Sheet," 12 June 2019, available at https://www.pewresearch.org/internet/fact-sheet/mobile/ [accessed 18-Mar-2020].

63. OWASP Internet of Things Project, OWASP website, available at https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=Main [accessed 27-Dec-2019].

64. IEEE, "About the IEEE Internet of Things (IoT) Initiative," available at https://iot.ieee.org/about.html [accessed 27-Dec-2019].

65. Internet Engineering Task Force, "The Internet of Things," available at https://ietf.org/topics/iot/ [accessed 18-Mar-2020].

66. "Project Connected Home over IP," available at https://www.connectedhomeip.com/ [accessed 18-Mar-2020].

67. "IoT Security Guidelines for Endpoint Ecosystems," GMSA, 31 October 2017, available at https://www.gsma.com/iot/wp-content/uploads/2018/12/CLP.13-v2.0.pdf [accessed 27-Dec-2019].

68. "Pangolins are Suspected as a Potential Coronavirus Host," New York Times, 10 February 2020, available at https://www.nytimes.com/2020/02/10/science/pangolin-coronavirus.html [accessed 18-Mar-2020].

69. Watkins, L., K. Silberberg, J. A. Morales, and W. H. Robinson, "Using inherent command and control vulnerabilities to halt DDoS attacks," *Proceedings of the IEEE International Conference on Malicious and Unwanted Software (MALCON)*, October 2015.

70. L. Watkins, C. Kawka, C. Corbett, and W. Robinson, "Fighting banking botnets by exploiting inherent command and control vulnerabilities," *Proceedings of the IEEE International Conference on Malicious and Unwanted Software (MALCON)*, October 2014.

71. Feily, M., A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," Third International Conference on Emerging Security Information, Systems and Technologies, pp. 268–273, June 2009.

72. Kambourakis, G., C. Kolias, and A. Stavrou, "The Mirai botnet and the IoT zombie armies," 2017 IEEE Military Communications Conference (MILCOM), pp. 267–272, October 2017.

73. Ramapatruni, S., S. N. Narayanan, S. Mittal, A. Joshi, and K. Joshi, "Anomaly detection models for smart home security," 2019 IEEE 5th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 19–24, May 2019.

74. Amanullah, M. A., R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. M. Nainar, N. M. Akim, and M. Imran, "Deep learning and big data technologies for IoT security," *Computer Communications*, January 2020.

75. Zolanvari, M., M. A. Teixeira, and R. Jain, "Effect of imbalanced datasets on security of industrial IoT using machine learning," 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 112–117, November 2018.

76. Stratosphere Laboratory. ***A labeled dataset with malicious and benign IoT network traffic***. January 22th. Agustin Parmisano, Sebastian Garcia, Maria Jose Erquiaga. [Online]. Available at https://www.stratosphereips.org/datasets-iot23 [accessed: 6 September 2020].

77. "Zeek overview." [Online]. Available at https://docs.zeek.org/en/current/intro/index.html#overview [accessed: 6 September 2020].

78. Scikit-Learn. (2019). Decision Tree Classifier. [Online]. Available at https://scikit-learn.org/stable/modules/tree#classification [accessed: 6 September 2020].

79. Scikit-Learn. (2019). Random Forest Classifier. [Online]. Available at https://scikit-learn.org/stable/modules/ensemble.html#forest [accessed: 6 September 2020].

80. "Multiclass Decision Forest – Azure Machine Learning Studio | Microsoft Docs." [Online]. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-forest [accessed: 6 September 2020].

81. "Two-Class Neural Network | Microsoft Docs." [Online]. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-neural-network [accessed: 6 September 2020].

82. "Multiclass Neural Network | Microsoft Docs." [Online]. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-neural-network [accessed: 6 September 2020].

83. Scikit-Learn. (2019). K-fold Cross Validation. [Online]. Available at https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-iterators [accessed: 6 September 2020].

84. Zhou, V., 2019. A Simple explanation of information gain and entropy. [Blog] Available at https://victorzhou.com/blog/information-gain [accessed: 6 September 2020].

85. "Permutation Feature Importance – ML Studio (classic) – Azure | Microsoft Docs." [Online]. Available at https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/permutation-feature-importance?redirectedfrom=MSDN [accessed: 6 September 2020].

86. Linsay, Jon R. 1 August 2013. "Stuxnet and the Limits of Cyber Warfare." Security Studies. 22(3):365–404 https://www.tandfonline.com/doi/full/10.1080/09636412.2013.816122 [accessed 5 June 2020].

87. Greenmeier, Larry. 28 April 2017. "The Pentagon's Seek-and-Destroy Mission for Counterfeit Electronics." Scientific American. https://www.scientificamerican.com/article/the-pentagon-rsquo-s-seek-and-destroy-mission-for-counterfeit-electronics/ [accessed 9 June 2020].

88. Sawyer, Don. 6 October 2014. "Counterfeit Threat Taking Malicious Turn?" Embedded Military Systems website. https://militaryembedded.com/comms/rf-and-microwave/counterfeit-taking-malicious-turn [accessed 7 August 2020].

89. Motavelli, Jim. 4 February 2010. "The Dozens of Computers That Make Modern Cars Go (and Stop)." The New York Times. https://www.nytimes.com/2010/02/05/technology/05electronics.html [accessed 6 August 2020].

90. Traufetter, Gerald. 3 August 2009. "Do Computers Make Planes Less or Safer?" *ABC News*. https://abcnews.go.com/Travel/story?id=8236562&page=1 [accessed 6 August 2020].

91. Tehranipoor, Mark M., Ujjwal Guin, and Swarup Bhunia. 24 April 2017. "Invasion of the Hardware Snatchers: Cloned Electronics Pollute the Market." IEEE Spectrum website. https://spectrum.ieee.org/computing/hardware/invasion-of-the-hardware-snatchers-cloned-electronics-pollute-the-market [accessed 10 August 2020].

92. Lapedus, Mark. 22 October 2018. "A Crisis in DoD's Trusted Foundry Program?" Semiconductor Engineering website. https://semiengineering.com/a-crisis-in-dods-trusted-foundry-program/ [accessed 7 August 2020].

93. ICSA-20-168-01. 20 August 2020. "Treck TCP/IP Stack (Update G)." US-CERT CISA website. https://us-cert.cisa.gov/ics/advisories/icsa-20-168-01 [accessed 24 April 2020].

94. Metz, Cade. 16 September 2015. "Google is 2 Billion Lines of Code – And it's All in One Place." *Wired*. https://www.wired.com/2015/09/google-2-billion-lines-codeand-one-place/ [accessed 25 August 2020].

95. Cheng, Jiangfeng, Weihai Chen, Fei Tao, and Chuh-Liang Lin. June 2018. "Industrial IoT in 5G environment toward smart manufacturing." In *Journal of Industrial Information Integration*. 10:10–19. https://www.sciencedirect.com/science/article/pii/S2452414X18300049 (accessed 25 September 2020)

96. Glende, Wolf L. 22–25 September 1997. "The Boeing 777: A Look Back." *AGARD Conference Proceedings 602*. https://apps.dtic.mil/dtic/tr/fulltext/u2/a354350.pdf#page=53 [accessed 9 June 2020].

97. Marr, Bernard. 6 March 2017. "What is Digital Twin Technology – and Why is it so Important?" *Forbes*. https://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twin-technology-and-why-is-it-so-important/#5005c9af2e2a [accessed 4 June 2020].

98. Berman, Barry. March-April 2012. "3-D printing: The new industrial revolution." Business Horizons. 55(2):155–162. https://www.sciencedirect.com/science/article/pii/S0007681311001790 [accessed 25 September 2020].

99. Gibson, Ian, David Rosen, and Brent Stucker. 2015. *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Second Addition. Springer. https://link.springer.com/content/pdf/10.1007/978-1-4939-2113-3.pdf [accessed 2 August 2020].

100. Dutta, B., and Francis H. (Sam) Froes. 2015. "The Additive Manufacturing (AM) of Titanium Alloys." *Titanium Powder Metallurgy: Science, Technology and Applications*. https://www.sciencedirect.com/science/article/pii/B9780128000540000241 [accessed 2 August 2020].

101. "Software Supply Chain Attacks." n.d. DNI website. https://www.dni.gov/files/NCSC/documents/supplychain/20190327-Software-Supply-Chain-Attacks02.pdf [accessed 2 August 2020].

102. Slawski, Bill, "Trust Metrics at Google" gofishdigital.com. 1 May 2019 Available at https://gofishdigital.com/trust-metrics/ [accessed 25 September 2020].

103. Khorragshahgol, Reza, and Vassilis S. Moustakis. January 1988. "Delphic Hierarchy Process (DHP): A Methodology for Priority Setting Derived from the Delphic Method and Analytical Hierarchy Process." In European Journal of Operational Research. 37(3):347–354. https://www.sciencedirect.com/science/article/pii/037722178890197X [accessed 25 September 2020].

104. Boyens, Jon, Celia Paulsen, Nadya Bartol, Kris Winkler, and James Gimbi. February 2020. NISTIR 8276 (Draft). "Key Practices in Cyber Supply Chain Risk Management: Observations from Industry." https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8276-draft.pdf [accessed 14 September 2020].

105. Boyens, Jon, Celia Paulsen, Jeffrey Ng, Kris Winkler, and James Gimbi. August 2020. NISTIR 8272. "Impact Analysis Tool for Interdependent Cyber Supply Chain Risks." https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8272.pdf [accessed 14 September 2020].

106. NTIA website. Software Bill of Materials. https://www.ntia.gov/SBOM [accessed 15 September 2020].

107. DFARS website, https://www.acquisition.gov/dfars [accessed 24 September 2020].

108. Office of the Under Secretary of Defense for Acquisition and Sustainment Cybersecurity Maturity Model Certification website, https://www.acq.osd.mil/cmmc/ [accessed 24 September 2020].

109. "What Is Ambient Computing, and How Will It Change Our Lives?" https://www.howtogeek.com/547655/what-is-ambient-computing-and-how-will-it-change-our-lives/ [accessed 16 October 2020].

110. Narayanan, S. N., A. Ganesan, K. Joshi, T. Oates, A. Joshi, and T. Finin, “Early detection of cybersecurity threats using collaborative cognition,” in *Proceedings - 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018*, Nov. 2018, pp. 354–363, doi: 10.1109/CIC.2018.00054.

111. Bassett, G., “System and Method for Cyber Security Analysis and Human Behavior Prediction,” US 2016/0205122 A1, 2016.

112. Kodali, R. K., V. Jain, S. Bose, and L. Boppana, “IoT based smart security and home automation system,” in *Proceedings - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, Jan. 2017, pp. 1286–1289, doi: 10.1109/CCAA.2016.7813916.

113. Greene, S., H. Thapliyal, and D. Carpenter, “IoT-Based fall detection for smart home environments,” in *Proceedings - 2016 IEEE International Symposium on Nanoelectronic and Information Systems, iNIS 2016*, Jan. 2017, pp. 23–28, doi: 10.1109/iNIS.2016.017.

114. “Technical Documentation | Amazon Developer Portal.” https://developer.amazon.com/documentation [accessed 21 October 2020].

115. “How important is PCAP for cyber defense? - Help Net Security.” https://www.helpnetsecurity.com/2019/09/23/packet-capture/ [accessed 06 September 2020].

116. Monahan, D., “Report Summary : Unlocking High Fidelity Security 2019,” 2019. [Online]. Available: https://www.endace.com/ema-2019-research-report-download.pdf.

117. “TCPDUMP/LIBPCAP public repository.” https://www.tcpdump.org/ [accessed 15 October 2020].

118. Ellis, G., and A. Dix, “A taxonomy of clutter reduction for information visualisation,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1216–1223, Nov. 2007, doi: 10.1109/TVCG.2007.70535.

119. Steinberger, M., M. Waldner, M. Streit, A. Lex, and D. Schmalstieg, “Context-preserving visual links,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2249–2258, 2011, doi: 10.1109/TVCG.2011.183.

120. Edu, J. S., J. M. Such, and G. Suarez-Tangil, “Smart Home Personal Assistants: A Security and Privacy Review,” Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.05593 [accessed 16 October 2020].

Appendix A. Acronyms

5G	Fifth Generation
ACK	Acknowledgment
AHP	Analytic Hierarchy Process
AI	Artificial intelligence
AM	Additive Manufacturing
ASR	Automatic Speech Recognition
AVS	Amazon Voice Services
AWS	Amazon Web Services
CAD	Computer-Aided Design
CHASM	Connected Home Automated Security Monitor
CIA	Confidentiality, Integrity, and Availability
CnC	Command and Control
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DENG	Doctor of Engineering
DHP	Delphic Hierarchy Process
DMZ	Demilitarized Zone
DNS	Domain Name Server
DoD	Department of Defense
HMM	Hidden Markov Models
HPSC	High Performance and Smart Computing
IaaS	Infrastructure-as-a-Service
IC	Integrated Circuit
IDS	Intelligent Data and Security
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoTA	IoT Analytics
IP	Internet Protocol
ISI	Intelligence and Security Informatics
IT	Information Technology
ITU	International Telecommunication Union

JHU	The Johns Hopkins University
JSON	JavaScript Object Notation
LED	Light-Emitting Diode
LSTM	Long Short-Term Memory
MAC	Media Access
MEMS	Micro-Electrical-Mechanical Systems
MILCOM	Military Communications
ML	Machine Learning
NCF	National Critical Function
NIST	National Institute of Standards and Technology
NLU	Natural Language Understanding
NTIA	National Telecommunications and Information Administration
PaaS	Platform-as-a-Service
PCAP	Packet Capture
RFID	Radio Frequency Identification
SaaS	Software-as-a-Service
SDK	Software Development Kit
SIoTD	Systems of Internet of Things Devices
SPA	Smart Personal Assistants
SQL	Structured Query Language
SSML	Simple Speech Markup Language
SSMS	SQL Server Management Studio
SWaP	Size, Weight, and Power
TCP	Transport Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UFE	User-Facing Environment
U.S.	United States
VM	Virtual Machine
VPN	Virtual Private Network
VUI	Voice User Interface

Appendix B. Curriculum Vitae

Jeffrey S. Chavis received his B.S. degree in Electrical Engineering from Howard University in 1989. He received his M.S. degree in Electrical Engineering from the University of Maryland in 1991 as a GEM fellow.

Since 1991, he has worked for Raytheon, Northrop Grumman and the Johns Hopkins University Applied Physics Laboratory (JHU/APL) in various engineering capacities. In addition, since 2005, he has been an adjunct assistant professor in the Johns Hopkins University (JHU) Whiting School of Engineering (WSE), teaching courses in computer science and information science engineering.

Over his career, he has achieved a variety of awards and forms of recognition including induction into the Tau Beta Pi national engineering society, recognition as the Black Engineer of the Year by CCYMAG, and appointments as senior member of the IEEE and Principal Professional Staff member at JHU/APL.

He enrolled in the Doctor of Engineering program in the JHU WSE in 2018. His research interests include the Internet of Things, Artificial Intelligence, Machine Learning and the application of these technologies in smart environments. His hobbies include mentoring, volunteering in the STEM community, music, sports of every kind, and traveling.