ANALYSIS OF DATA FLOW IN IOT DEVICES AND EVALUATING SECURITY OF MUD IMPLEMENTATION ON SMART HOME NETWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MILAD KAZEMI DARAZAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CYBER SECURITY

DECEMBER 2020

Approval of the thesis:

ANALYSIS OF DATA FLOW IN IOT DEVICES AND EVALUATING SECURITY OF MUD IMPLEMENTATION ON SMART HOME NETWORK

submitted by **MILAD KAZEMI DARAZAM** in partial fulfillment of the requirements for the degree of **Master of Science in Cyber Security Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, Graduate School of **Informatics** _____

Assist. Prof. Dr. Cihangir Tezcan
Head of Department, **Cyber Security** _____

Assoc. Prof. Dr. Cengiz Acartürk
Supervisor, **Cognitive Science, METU** _____

Assist. Prof. Dr. Pelin Angın
Co-supervisor, **Computer Engineering, METU** _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Milad Kazemi Darazam

Signature :

ABSTRACT

ANALYSIS OF DATA FLOW IN IOT DEVICES AND EVALUATING SECURITY OF MUD IMPLEMENTATION ON SMART HOME NETWORK

Darazam, Milad Kazemi
M.S., Department of Cyber Security
Supervisor: Assoc. Prof. Dr. Cengiz Acartürk
Co-Supervisor: Assist. Prof. Dr. Pelin Angın

December 2020, 74 pages

IoT usage has shown significant growth in the past decades. A major target for the IoT market is the living spaces which have become smarter by the integration of IoT devices. However, the network infrastructure have not been developed from the perspective of Cyber security. IoT devices are subject to Cyber security threats in multiple fronts. For example, infected IoT devices may contribute to DDoS attacks that target global Internet services, such as the DNS. For mitigation of the attacks, various solutions have been proposed. In this thesis, we review available solutions with a particular focus on the application of a standardized whitelisting method, namely Manufacturer User Description (MUD). For an evaluation of MUD usage in IoT networks, we analyzed traffic of two devices with the aim of detecting recognizable and distinctive traffic patterns. We established specific MUD files based on the detected traffic patterns and evaluated the MUD files for the validation of their proof of work.

Keywords: IoT traffic pattern, smart home security, DDoS prevention, MUD

ÖZ

IOT CİHAZLARDA VERİ AKIŞININ ANALİZİ VE AKILLI EV AĞINDAKİ MUD UYGULAMALARININ GÜVENLİĞİNİN DEĞERLENDİRİLMESİ

Darazam, Milad Kazemi
Yüksek Lisans, Siber Guvenlik Bölümü
Tez Yöneticisi: Doç. Dr. Cengiz Acartürk
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Pelin Angın

Aralık 2020 , 74 sayfa

IoT kullanımı geçtiğimiz on yılda kayda değer büyüme gösterdi. IoT pazarındaki başlıca hedef yaşam alanıdır. Ancak akıllı ev ağ altyapısı, siber güvenlik açısından, IoT ile uyumlu çalışacak şekilde geliştirilememiştir. IoT cihazları, birçok açıdan siber güvenlik tehditlerine maruz kalmaktadır. Örnek olarak, ele geçirilmiş IoT cihazları, DNS gibi servisleri hedef alan DDoS ataklarına katkıda bulunabilirler. Bu tez çalışmasında, özellikle Manufacturer User Description (MUD) olarak adlandırılan standardize edilmiş whitelisting metodunun uygulanmasına odaklanarak kullanılabilir çözümleri gözden geçirdik. IoT ağlarında MUD kullanımını değerlendirebilmek için iki cihazın ağ trafiklerini, tanınabilir ve karakteristik trafik desenlerini çıkarmak amacıyla, analiz ettik. Tespit edilen trafik desenlerine dayalı olarak özel MUD dosyaları oluşturduk ve MUD dosyalarını çalışmanın doğrulanması amacıyla değerlendirdik.

Anahtar Kelimeler: IoT trafik deseni, akıllı ev güvenliği, DDoS önleme, MUD

To My Family

ACKNOWLEDGMENTS

I want to thank my family, my supervisor and co-supervisor for supporting me during the research, experiment and publishing period. Without their support, I couldn't stay motivated in finding satisfactory solution that fits the criteria that I researched for.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DDoS: Distributed Denial of Service

IoT: Internet of Things

SDN: Software Defined Networking

ML: Machine Learning

MUD: Manufacturer Usage Description

OpenWRT: Open Wireless Router

CHAPTER 1

INTRODUCTION

Internet of Things (IoT) is a term coined in 1999 by Kevin Ashton [1]. More recently the term is mostly used for embedded devices in various computation-enabled environments, where the IoT devices are able to sense, communicate, and provide services to human users or other devices.

IoT market size has been growing fast since the past decade. Cisco Systems Inc. predicts that 48 percent of Machine-to-Machine (M2M) traffic would be generated by smart home networks by the end of 2023 [2]. According to IoT Analytics, which is a provider of market insights for IoT, the number of connected devices increased from 3.8 billion in 2015 to 8.3 billion devices in 2019, accompanied by a projection reaching nearly 16 billion devices by the end of 2023 [3]. Due to the broadening use of IoT technologies in daily life settings, the users of IoT devices, in particular, smart-home technologies have been subject to increased security threats. The stakeholders of the global supply chain, from IoT hardware manufacturers to end-user consumers, have been facing IoT security threats at various fronts. The threats cover a wide spectrum of issues, from malware embedded in hardware to operating system-level and application-level vulnerabilities. The end-users of the IoT supply chain also play a major role in enriching the coverage of security issues due to their low awareness of cyber security, in particular, the lack of knowledge about the configuration of the devices for security is a major problem [4]. More specifically, the growth of the smart home market has been accompanied by growing security threats since the IoT devices are not usually configured by experts with an appropriate technical background. In smart home networks, device misconfiguration has been one of the top vulnerability categories, as reported by OWASP [5]. This makes IoT networks an attractive domain

for adversaries.

The vulnerabilities and security issues related to IoT devices threaten the security of the local networks, as well as providing a basis for DDoS attacks. As the adversaries gain unauthorized access to IoT devices, they infect the devices with specific malware codes, mostly to use them as botnets for Distributed Denial of Service (DDoS) attacks. This is a typical scenario that can be observed in today's IoT networks. DDoS attacks may eventually target large service providers, which may in turn result in the loss of reputation, as well as financial loss [6]. Therefore, the mitigations that provide solutions for the security of IoT devices and IoT networks comprise an important part of today's cyber security solutions.

On the other hand, the improvement of security mitigations is complicated by multiple facets of IoT security, which address specialization in different aspects of IoT ecosystems in smart homes. In this thesis, we focus on the network dimension in the domain of IoT security, the underpinnings of communication of IoT devices with each other and with the rest of the world, since it comprises the backbone of any IoT infrastructure.

There exist multiple approaches to the improvement of the security of IoT networks, especially smart home networks. Recently, software-defined networking (SDN) aims at improving IoT network security by improving network architecture. Machine Learning (ML) models provide tools that are based on learning from available data, such as detected anomalies. Besides SDN and ML approaches, Blockchain applications have been introduced for improving IoT network security, which benefit from its secure nature by design, to solve some of the existing problems in the IoT networks, such as privacy-related issues. The aforementioned approaches (SDN, ML, Blockchain) have their own pros and cons, which are discussed in the following chapters. More recently, MUD (Manufacturer Usage Description) provides an appropriate solution for the improvement of the security of IoT networks, by offering a cost effective solution that is easy to integrate.

The MUD approach is compatible with relevant approaches that offer outsourcing the management and security of smart home networks because of its complexity [7]. The approach is also compatible with the idea that DDoS attack detection is effective

when it is close to the source of the attack, since MUD provides a DDoS prevention mechanism that is close to the source.

As IoT devices are built to be used for specific tasks, it is possible to have predictable network traffic patterns in their communication. By identifying the patterns in the communication of IoT devices in a network, detecting anomalies from normal behaviors can be an effective approach, which can also improve the applicability of machine learning-based solutions. The major research question of this thesis is whether it is possible to extract distinctive and recognizable traffic patterns from IoT devices, which can be used as a major feature for developing identification, anomaly detection, and authorization systems. By verifying this hypothesis machine learning methods become a reasonable approach for generating anomaly detection models and also it helps to conclude that whitelisting methods can be implemented in order to restrict the network communication of IoT devices. The distinctiveness features of traffic patterns can be effective in constructing a comprehensive normal behavior profile. Hence, the model can perform with a lower false positive rate in comparison with previous models that have been used to detect anomalies in the general computer network environments. The high false positives are one of the major reasons behind the low success rates of previous intrusion detection systems [8]. In this thesis, we chose to follow the whitelisting method instead of machine learning methods for IoT network security. This choice was made based on different criteria that we explain in the following chapters.

Another contribution of this research is to develop a dataset of captured traffic of IoT devices and related analyses. We had two IoT devices available in our lab to capture their traffic flows while they generated normal traffic in the network. In order to reach the maximum coverage of variation in their generated traffic, we had multiple capture sets. Based on our conclusion after analyzing data, we decided to work on one of the specific sets of capture, which covered all the features we needed to have for the next step. By creating this dataset, we aimed to contribute to a library of datasets, including the network traffics of two popular IoT devices. This helps security researchers access the normal behaviors of each device. Also by generating useful reports on raw data, it is possible to have an abstract view of the dataset. By utilizing the generated reports, we have analyzed the variation of particular features on the dataset in order to find

a pattern to prove the distinctiveness of the traffic flow in the IoT network. This conclusion led us to follow normal profiling and whitelisting methods as an applicable and promising approach to the IoT network security problems. Based on our survey of proposed solutions and regarding the security of smart home at the network level, we suggest a general architecture of a smart home network by utilizing open-source approaches.

This architecture, which is based on a new standard is called Manufacturer Usage Description or MUD. MUD is theoretically explained by the Internet Engineering Task Force (IETF) [9] and technically reviewed, documented, and implemented by NIST [10]. This suggestion is based on the theoretical evaluation of the researches that have reviewed MUD and based on our implementation of MUD in our laboratory to evaluate it from aspects of implementation easiness and promised security protections. For this, we generated MUD profiles for the IoT devices we used in our labs by analyzing their traffic patterns and following the Yang model presented by IETF. Also, we built a MUD enabled network to test our generated MUD files.

In order to find a reliable approach for smart home network security, we surveyed theoretical and technical approaches from different categories and reviewed some of them from the perspective of completeness and methodologies. This survey is presented in the next chapter.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we present a survey of proposed solutions on the security of smart home networks that majorly focus on IoT devices. The motivation of detecting and preventing DDoS traffic in smart homes is mainly the reason why researchers have been involved in this topic and created a lot of novel approaches for this question.

In this chapter, we first introduce a set of definitions and report relevant statistics. Next, we review each solution in its own category and critically analyze the solution from different aspects.

2.1 IoT Definition

The IoT term was coined by Kevin Ashton, who established the MIT Auto-ID Center[1]. In the same year, IBM developed the first machine-to-machine protocol for connected devices and named it MQTT [11]. Since then, other organizations have proposed a definition for this term, too. For example, International Telecommunication Union used the term in the report of the year 2005, and then in 2012 defined it as a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on the existing and evolving inter-operable information and communication technologies or a network which is available anywhere, anytime, by anything and anyone [12]. Similarly, IEEE defined it as a network of items each enabled with sensors which are connected to the Internet [13]. ETSI, officially recognized by the EU as a European Standard Organization, uses the machine-to-machine (M2M) term instead of IoT, and describes it as communication between two or more entities that do not necessarily need any human interven-

tion. M2M services intend to automate the decision and communication process [14]. NIST also uses Cyber-Physical System for describing IoT and defines it as a new way for connecting daily used objects in order to enhance the efficiency and sustainability and improve the quality of life.

2.1.1 The Growth of the IoT Market

Cisco predicted that by the end of 2023, 48 percent of M2M traffic would be generated by smart home networks [2]. According to IoT Analytics, the number of connected devices increased from 3.8 billion in 2015 to 8.3 billion devices in 2019 and it is also predicted that it is going to reach nearly 16 billion by the end of 2023 (Figure 2.1) [3].

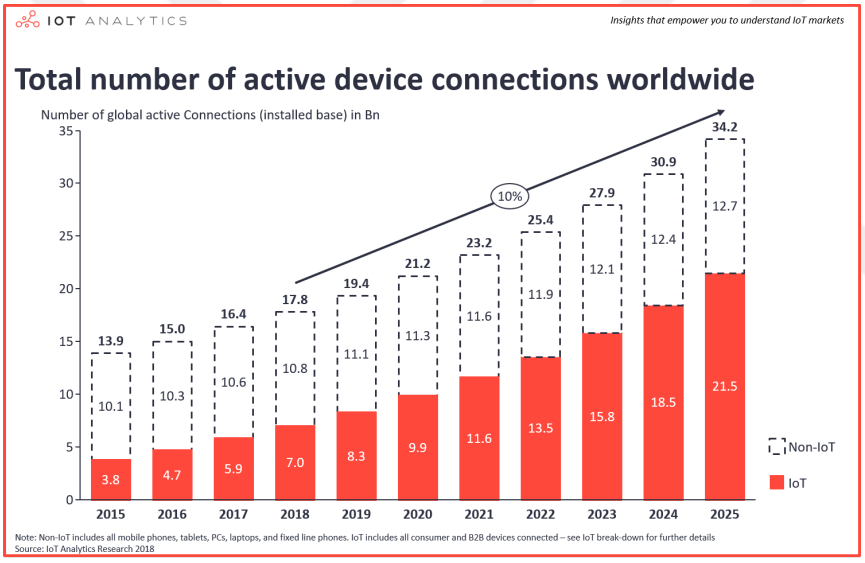

Figure 2.1: Number of active IoT devices worldwide

Accordingly, IoT devices will have a significant role that shapes the way we live, the way we communicate, as well as the way we work and learn in the near future. Given its major role in daily life, the security of IoT devices and their connections will also be of utmost importance.

The importance of IoT has already been addressed by numerous studies ([15], [16],[17] and [18]). Nevertheless, IoT threats and research on them are expanding rapidly. The goal of this research is to bring pieces together and present a systematic review of

IoT security in relation to DDoS attacks and possible solutions for mitigation within the context of use. The ultimate goal of the thesis is to offer an effective solution (MUD – Manufacturer User Description) as a mitigation to the problem. Below, we describe IoT use cases in terms of their most common context of use, namely smart environments.

2.1.2 The Concept of Smart Environment

The word "smart" has become a prefix to many objects that people are using daily, in their homes, offices, around the city, etc. It is related to the ability of connection and interaction of an object with humans and other objects. Our daily useful objects are embedded with processors, sensors, and software to be able to connect with humans and other objects and transfer the processed data to help in decision-making processes. Our living environments, such as homes and cities, have also changed significantly in the past several decades. A revolutionary change in environments is the way they are becoming smart.

2.1.3 Smart Home

One of the main environments that have changed rapidly with the growing IoT market is our homes. It is also one of the early stage usage of IoT devices as in the examples of using temperature sensors or locking doors remotely. For example, refrigerators had been one of the pioneers of smart home devices that opened their way into the homes. Despite that, even coffee machines have become smart, and this points out that smart devices need a modern design of home networks, which will be powered by SDN in the next few years. Also, it is necessary to mention that because the users of the home networks are not always IT experts, misconfiguration and neglecting security hardening happens a lot and this opens a door for adversaries to gain access from these devices.

2.1.4 Smart City

Besides the home network, the cities are getting smart. The term smart city had been defined in various forms from the perspective of different academicians, but despite the differences in architecture, researchers have some descriptions in common. The usage of machines or embedded systems or in other terms IoT, as the main component in planning the infrastructure of a computer network, is the root in descriptions. The usage of IoT devices in city-related services have been growing fast because of the investments in this area. One of the earliest usages of IoT in the context of the smart city is in the transportation systems [19] and traffic management [20]. Technological dependencies of these systems are usually based on the municipalities and in the case of attacks, many of even traditional services can be interrupted.

2.1.5 E-Care

IoT also made significant development in the healthcare market and nowadays it plays a doctor's assistant role in developed countries and does tasks like continuous glucose monitoring [21]in patients.

2.1.6 Architecture of IoT

Like different definitions, different functioning models were also designed for IoT architecture. The model proposed by IEEE can be a good reference, as it provides a simple approach for describing the layers of the functional components of IoT devices (Figure 2) [13].

In this design, the bottom layer is the perception or sensing layer which can be also referred to as the hardware layer and is responsible for the functionality of components, sensors, etc. One upper layer is the network layer and transmission of data through a variety of protocols like 5G, 4G, WiFi, ZigBee, infrared, etc. are popular protocols of this layer. The upper layer is called the application layer due to the main responsibility of it in establishing connections between services, and also business logic relies on this layer.

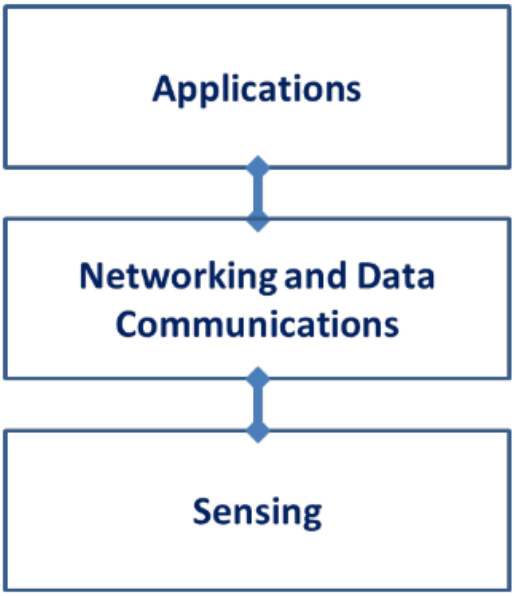

Figure 2.2: Architecture of IoT

2.2 IoT Security Challenges

Many of the security challenges in IoT devices are due to the fast-growing market and rapid development which usually results in not developing them with security in mind [22]. Other parts of challenges are born due to the ease of configuration which in most devices can be done by non-technical people and as a result, we see many devices left with default credentials and unpatched and outdated software [22].

There is a research [17] that have been categorized the security challenges of IoT into four groups. The first group of challenges is related to authentication-related problems which can be directly linked to the perception layer with physical threats due to the accessibility of IoT devices. Confidentiality problems as the name implies, relate to the security of communication protocols and media of data transmission in the network. Integrity as the other main principle of security is a matter of concern when data can be modified in the transmission process or even interrupted due to the denial of service attacks. Therefore, integrity and availability have been considered in the same group of security challenges in this study. Privacy challenges are the last group that has gained significant focus lately. As IoT devices would process the critical and confidential data, the policy of processing these data and keeping them

secure without knowing much about them is another challenge. Security solutions may not be able to do deep packet inspection for detecting anomalies due to the privacy policies and limited resources.

Besides all these major categories, there are other security challenges that are specific to IoT devices and can be categorized in the mentioned groups. A limited amount of resources for processing is one of these key challenges, especially in the area of cryptography and intrusion detection. Despite all these, many researchers believed that the lack of a universal platform and security architecture standards for IoT is a big issue that should be solved immediately.

Being aware of IoT security challenges, issues, and problems is a key point for manufacturers and developers in this area. Fortunately, OWASP had started an Internet of Things project with the aim of increasing awareness about the mentioned security challenges [5]. This project helps manufacturers, developers, and consumers to understand security issues and enable them to make better decisions in the process of building, deploying, and assessing IoT technologies.

The OWASP project consists of various resources earned through years of research and includes attack surface areas, testing guides, and top vulnerability listings. Also, it has specific testing guides for firmware analysis and medical device testing.

Observations show that the major use case of infected IoT devices is for the botnets. These infected devices can be used to take down or interrupt real-time services, as presented in the following section.

2.3 Distributed Denial of Service (DDoS)

Distributed Denial of Service (DDoS) is a kind of attack in which the attackers try to interrupt the normal behavior of the system (usually server) in a way that it is unable to function and serve the real clients. Nowadays this attack happens with the help of distributed attacker machines which are unknowingly taking part in it [23]. DDoS is also an important threat against smart environments. For this reason, we present the concept of DDoS in detail in this section.

There are many challenges with DDoS attacks. For example, finding the source of the attack is one of the major problems due to the nature of distributed contribution of nodes in this attack. Besides that, command and control (CC) servers are also accessed through proxy of connections and it makes it very hard to track down the real identities behind the attacks. The other problem with it is that many times the kind of traffic that are sent to the server through compromised nodes (zombies) are like the legitimate traffic, so distinguishing between these as benign and malicious by looking at the traffic patterns is an extremely hard and costly process.

Authors in [23], categorized DDoS attacks into two major groups, in terms of resource and bandwidth depletion. The bandwidth depletion DDoS attacks divides into two subgroups, namely the protocol exploit attacks, such as flooding UDP traffic with random ports or ICMP messages, and the amplification attacks, which exploit DNS and NTP protocols. The resource depletion DDoS attacks have two subgroups, too: The protocol exploit attacks, in which an attacker can do TCP SYN, PUSH ACK, SIP flood, slow HTTP attacks, and malformed packet attacks, such as ping of death or Teardrop.

DDoS attacks may have a high potential to induce hazardous effects in our daily communication via the Internet and its services. It may cause voluminous data loss, in addition to its time cost. As an example, consider a DDoS attack on an electronic voting system. The service failure problem does not only have economic, reputation, and social impacts but also can have an impact on international relations.

The motivation behind DDoS attacks may vary depending on the context. For instance, a DDoS attack on a game server may aim at disconnecting game players thus reducing the reputation of the game service. It may be conducted by hacktivists to disseminate a political message as a follow-up to a DDoS attack on a governmental service. It may also be due to the market competition, or the simple motivation behind a DDoS attack may even be the motivation of a script kiddie to slow down public services for fun.

Recently, the major targets of DDoS attacks are technology companies due to the market competition and also the governmental organizations are regularly under attack by APT (Advanced Persistent Threat) groups. In terms of its technical impact,

the most influential attacks have been made on DNS servers, especially the root DNS servers. The DDoS attack on October 21, 2003 exemplifies a DNS server attack that dates back to more than a decade ago [24]. More recently, the DDoS attack on October 21, 2016 aimed at attacking Dyn DNS servers. This attack is peculiar in that the slave computing devices of the attacking architecture consisted of IoT devices that were infected by the Mirai malware [6].

Mirai is an IoT malware with published source [25] that basically uses a short dictionary of default username and passwords to brute force the authentication mechanism of IoT devices that have default credentials. Mirai had been used to attack Krebs on Security blog, OVH data centers, and also Dyn name servers [26]. The latter one caused interruption on accessing popular websites like Twitter in North America during the attack time period.

Although Mirai was not the only DDoS capable malware and not even the best of them, those attacks were a key point in the history of IoT security, when all began to propose their solutions to mitigate such attacks. It actually revealed the potential problems of the insecure IoT devices and experts started to take the security of IoT as a serious problem.

Authors in [27] have made an analysis of DDoS capable malware based on the features that any malware can have and listed all known DDoS capable malware with their features. Later in [28], they published their work with a focus on Mirai investigation and also proposed a white worm named AntibIoTic as a solution to protect IoT devices against these malware [29].

This type of malware first targeted Linux operating systems. Hydra, which was an open-source application for FTP server password hijacking, targeted MIPS CPU architectures and used IRC-based communication with the master. It was capable of SYN and UDP flood attacks. After four years malware authors became interested in other CPU architectures like ARM and they released malware named Aidra, Zendran, Spike, MrBlack, etc., some of which were still using the IRC-based communication mechanism, although the later ones used an agent-handler. Also, the layer 7 flood was an interesting type of DDoS attack that had been added to the next generation of these malware(s). Among all these known malware(s), Mirai was probably a compre-

hensive one due to its capability of doing different kinds of attacks and compatibility of targeting different CPU architectures [27], [28], [29].

2.4 DDoS Mitigation Techniques

Mitigation of DDoS attack relies on early-stage attack detection, which becomes efficient when it happens near the source [30]. Mostly, we have two popular detection methods in network intrusion detection systems which are the main mechanisms to detect DDoS attacks in local networks. The first method is misuse detection which is very successful in terms of detection rate for known attack types and has been used for a long time in intrusion detection and prevention systems. But the problem with this method is it fails in the detection of new attack types or better said, zero-day attacks. The second method is anomaly detection, which may detect the zero-days but sometimes it can lead to high false-positive alarms due to the optimization issues in different kinds of networks with a variety of protocol usages. DWARD [31] and MULTOPS [32] are examples of DDoS mitigation algorithms that detect anomaly traffic potentially for DDoS attacks.

We surveyed different approaches to detect DDoS attacks in smart home networks. There are completely different kinds of approaches with different architectures and disciplines. However, some of these approaches have suggested hybrid solutions and got benefits of different methods to propose a comprehensive and efficient all-in-one solution.

In this chapter, first, we look at the management view of smart home networks because nowadays experts believe that smart home security should be outsourced. Then we look into different categories of approaches with some background and definitions as an introduction. We start with intrusion detection systems as the main and traditional approach to detect DDoS in outgoing traffic. After that, we survey learning approaches that have recently become popular in intrusion detection systems. Next, we explain software-defined networks that are believed to change the structure of home networks and will help the procedure of out-sourcing security. Then we have an analysis of Blockchain-based approaches and finally, we mention other types of

approaches that cannot be categorized in the previously mentioned groups.

2.4.1 Outsourcing Security as a Business-Level Solution Against IoT Attacks

In this section, we address some of the problems with the current home network structure that is based on traditional networking design and then mention some challenges that should be reviewed for a modern design of the home network.

First, according to [25], the operational cost of implementing traditional networks is like half of the whole cost. The most important issue with these networks is that they are configured by non-expert people. The result can be seen as outdated firmware, outdated operating systems, default authentication credentials and etc. Therefore, people tend to be offered some all-in-one and easy-to-configure solution as their gateways and home network management systems. We think in order to reach such a solution, the management and security of home networks should be outsourced. This source can be ISPs or any other security company involved with these services.

The author of [7] mentioned the challenges that a modern smart home network solution may have. Although outsourcing would be beneficial in order to design an ideal architecture, it is good to consider the challenges. With a contribution of home networks activity report, the collected data sets can be useful in the security approaches. One of the challenges with this feature is how to handle this size of data and how to analyze them. The second challenge is with privacy because any leakage in the ISP would target customers directly. The third challenge is that the components of the home networks should be remotely manageable and finally, the controller of the gateways should be considered as secure devices.

One of the problems in the proposed solutions in the literature for detecting DDoS and other anomalies in the home network is that they need a high number of processor units and it is not cost-effective to have them in home networks. These solutions can be gathered as modules and used in network structures like ISPs. So by outsourcing security and management of home networks to experts, a modular solution for intrusion detection can be applied which can always be learned from new data that is going to be gathered from distributed, updated, and live nodes. It seems that

with growing rate of incoming traffic of IoT devices from homes, people will tend to outsource security, because a variety of software and hardware will make it hard to configure, manage and secure all of them independently.

2.4.2 Misuse Based Detection and Prevention Systems as Signature-Based Solutions Against IoT Attacks

The main principle of intrusion detection systems is based on that the destructive activities like attacks have different characteristics than the normal activity and behavior [34]. There are many approaches to intrusion detection systems that can be categorized into three main models [35]. The most traditional one is misuse detection that is based on patterns extracted from data gathered from sensors, and generate a signature which is going to be compared to system input. Whenever the input matches with signatures, the system detects an intrusion and raises an alarm if it is only a detection system or forces the policy if it is also a prevention system.

The other main approach for intrusion detection systems is anomaly detection, which is used to monitor the system or infrastructure at first with this assumption in mind that there is no intrusion in it. After collecting data, a model based on this normal behavior is going to be constructed which will consider any activity that doesn't fit in the model as an intrusion.

Specification-based detection as the name of it implies is a collection of a defined specification that specifies intrusion and normal activity, so whenever an activity matches intrusion specifications, it is considered as a malicious activity.

After surveying intrusion detection systems for IoT environment, authors of [17] suggested that a desirable IDS for IoT environment should be able to operate in real-time due to the role that some IoT devices have in the network. Also, as there is a major trend in the market of IoT, the IDS should interoperably communicate, which means it should be compatible with different protocols. The other major feature can be dedicated to host-based IDSs, and it's the lightness of the algorithm. Because of the low resource of computation and power in many IoT devices, the host-based intrusion detection algorithms should be lightweight enough to be cost-effective. Placement

of the IDS in IoT network is another matter of concern, some suggest a distributed form of IDS while the others point to the benefits of central decision-making systems. However, it completely depends on the nature of the network and the heterogeneity of IoT devices that works on the network. Of course, many of these IDSs have been built with the idea of protecting the local network or the outside network from a particular kind of attack, and some of them are focused on DDoS attacks.

For example, authors in [36] adopt Suricata [37] which is a popular intrusion detection and prevention system, to a 6LoWPAN network to detect DDoS detection and reduce the false alarms. Also [38] proposed a model to detect DDoS attacks based on the energy consumption of the devices in the 6LoWPAN network. As the algorithm relies only on one parameter, it is considerably lightweight.

In [39] the authors experimented with a solution for DDoS detection with an inline approach by using Snort [40] that is the most popular intrusion detection system, to monitor, detect, and prevent the incoming traffic to the IoT network to protect the smart devices. Their results show promising detection rates. However, they haven't mentioned the false positive rate and latency in the network.

Another experiment utilizing Snort can be found at [41]. They did an experiment with the usage of Snort in home networks and tried to get common alerts that smart home networks may involve like login failure from outside the network, UDP flood, etc.

Another usage of IDS in smart homes with a different approach in placement and performance can be found in [42]. Authors have proposed a two-layer architecture for intrusion detection systems in which one of the agents resides in the home network and is designed to collect data and send them to the others which reside in the ISP to perform data analysis. After analysis, based on the data that might come from the other home networks, the expert system generates rules and policies for the home network and obligates these policies by the home network gateway agent.

A combination of learning methods with signature-based IDSs are also available as we discuss in the next section. [43] can be an example of this hybrid system. Their research aims to achieve fast pattern matching to scale up the performance.

2.4.3 Machine Learning Methods for Mitigating IoT Attacks

By using learning methods, there is the capability to construct a model of the normal behavior in systems. These models then can be used to detect any deviation from the borders of normal behavior and depending on the violation type and the amount of violation, alarms can be raised in the intrusion detection system. It might seem that anomaly detection approaches are one of the effective approaches to detect zero-days but on the other hand, as we cannot make a comprehensive profile of so-called normal behavior, the rate of false-positives usually tends to be high in these systems.

Another major problem that a researcher might face in developing these kinds of systems is the lack of updated and classified data sets that include normal behavior of today's IoT devices. The other challenge is the implementation of such models. Because usually the learning models need sufficient resources to work on and when we talk about IoT network there is a lack of resources, researchers tried to give lightweight algorithms as a solution like [38] which depends on only one parameter. In another scenario, the major process of training and evaluation part can be done in non-IoT devices that can reside as a gateway or even ISPs when the security is outsourced. Meanwhile, lightweight algorithms seem more practical and interesting to use in current home network infrastructures, even with a low amount of resources.

For example, in [44] authors have proposed an approach to classify IoT devices in smart environments based on their traffic behavior. As the IoT devices may have certain characteristics in their network traffic, they might be classified into major groups and by using this type of classification, anomalies can be detected, especially anomalies like DDoS. They claim that their proposed method is designed to need low resources of computation with the idea in mind that many gateways in home-like environments are suffering from low amount of computation power.

Another approach that can fit into the home networks very well is described in [45]. This modular solution is based on a machine learning model for the classification of network traffic as benign and malicious. It can be considered not only for DDoS malware but also for any kind of attack that threatens IoT devices. The experiments show that the best accuracy among the three algorithms of Gaussian Naive Bayes,

KNN, and Random Forest is achieved by KNN. The goal of this research is to detect malware in the early stages when the attacker starts to scan for vulnerabilities and because of this goal, they have classified the malicious traffic into three major malware targets as HTTP GET, HTTP POST, and TELNET classes.

Besides these approaches, simulations show us that combining machine learning algorithms with a simple structure of SDN can dramatically reduce the effects of attacks. Authors in [46] implemented a test bed with Mininet to examine their SVM-based solution for the detection of DDoS attacks in the SDN network. They have used the Ryu controller as a target and generated sample DDoS traffic and successfully reduced 36 percent of the attack traffic.

Combining machine learning approaches with SDN architecture have great benefits. For example, the study of [47] shows how machine learning and SDN can come into one platform. This solution combines the manufacturer's usage description (MUD) standard with SDN networks and utilizes a Snort IDS to prevent a common type of attack in the IoT network. In this solution, the MUD profiles are generated from Pcap files that are captured from IoT devices. Then in real time, these profiles can be captured by an inspection engine and converted to flows and fed to the controller. By this solution, the majority of attacks from the source can be detected and prevented with the method of whitelisting. An open question in this research is that it doesn't show significant attack prevention in the case of the camera-type IoT devices and the reason is that the majority of these cameras provide service to any external IP address for streaming. Identifying IoT devices is an approach like we mentioned in [44], it requires statistics and logic to rely on for detecting anomalies in the network. Another challenge is that when people want to install and implement new features on their IoT devices, that needs to access completely different locations. It can result in a pattern that goes beyond the whitelisted flows of the network and that particular device.

Another combination of SDN and machine learning to detect and prevent DDoS attacks can be seen at [48] where authors have proposed a framework for DDoS prevention with the benefit of using a self-organizing map algorithm. This framework consists of two major components, the central controller which is called MECshield, and the other one which is called MECshield-agents. The agents have a duty of col-

lecting data and extracting the features and sending them to the controller. Then based on the behavioral analysis, the policies are generated and executed in the gateway by the smart SOM filter.

When it comes to machine learning and choosing algorithms over another for anomaly detection, we can mention the studies of [49] . They have implemented an anomaly detection solution using a variety of algorithms of machine learning to detect the DDoS traffic that can be generated from IoT devices. They have implemented a private laboratory of home usage IoT devices and captured normal traffic and also captured DDoS traffic and spoofed the IP address of attack packets with the IP address of their IoT device. Their proposed solution was successful with very high accuracy. Between the tested algorithms neural network has been chosen as the best one and besides that decision tree and K-nearest neighbor have provided accurate results. Their research shows that classification methods with low-dimensional features can effectively find anomaly traffic and even using only stateless features would help much in detecting DDoS traffic.

Authors in [50] proposed a system that aims to protect IoT networks from DDoS attacks, although it is a different goal, it can be useful to get inspired. The architecture of the protection system is distributed and the main algorithm is based on the Naive Bayesian classification. Through the network, there are agents which can be classified with their tasks as a collector, monitoring, actuator, and communication agent. Results show that as it uses a learning method, with progress in learning, the detection rate and overload in the network become low, so during the next phase of the attack, it is going to be much easier to detect and prevent attacks on IoT networks.

It is important to mention that authors in [51] proposed an early solution in this area, inspired by their previous work on the prevention of DDoS attacks in wireless mesh networks. Their solution is based on learning automata with the design of service-oriented architecture. Their approach consisted of three phases of detection, identification, and defense. The network model for preventing DDoS attacks result in cross-layer model of detection and prevention, which is a real concern due to the diversity of the attacks in different layers. They have simulated their solution with learning automata and without it to compare the results. The results show a very

large amount of efficient results are achieved using the SOA model, especially their proposed DALERT method for notifying the other nodes of the network is played the main role in the performance.

2.4.4 SDN Based Solutions for Mitigating IoT Attacks

Software defined networking gives a promising paradigm shift in computer networking. The main idea behind the SDN is to separate the forwarding functions that rely on hardware from the management functions that rely on the software. Also one of the great benefits of SDN is its programmability with great vast of platforms. As mentioned before, home networks became complex and SDN can be a good solution for its architecture. Authors in [52] have surveyed many research on utilizing the SDN in smart homes and as expected, the majority of the works are focused on the QoS and management of internet and wireless in the home networks. Only 4 of 42 surveyed articles have used SDN for security purposes and 5 of them included IoT device management in the home networks. All of the related works have studied separately and included in this review in different sections.

At [53], authors had described the challenges and vision of software defined home networks. Authors propose a software-defined architecture that benefits of virtualization, central management, and intelligent decision system for smart homes and offers APIs for external services. One of the concerns with the emerging of these architectures is the central unit of decision which contains user activity information. This amount of centralization attracts the attention of adversaries and any breach cost for privacy and safety of the whole home network and the users at a critical level.

Authors in [54] suggest that for efficient and high-performance anomaly detection in the network there is a need for SDN infrastructure. They proposed an inter-domain approach and claim that analysis based on flow/packet is required due to the high level of today's traffic and limitation in the computational resources.

In [55], authors have proposed security as a service solution using SDN structure. There is an orchestrator that is placed in the cloud and gives Restful API to both the customer network and the ISP. Security mechanisms implemented by service

providers can be enabled or disabled through web interfaces by customer and ISP.

By utilizing SDN and NFV, a flexible protection architecture proposed with the aim of reducing the risk of attacks in the home networks [56]. The powerful feature of the solution is the functions that are available over the network and they acknowledge the central system with evidence of happening events. Then the central system orchestrates the functions with policies to mitigate the attack and put the damage to the lowest level.

Authors in [30] have designed a smart home level SDN network in which the controller is working with the intrusion detection system closely in order to detect the DDoS attacks near the source. The soft switch of the network mirror all the traffic to the intrusion detection system and if there is any malicious behavior in the traffic, the intrusion detection system gives an alert to the controller to generate a flow for blocking the further traffic from that device, As this solution seems to guarantee reducing delay in the network and assure the QoS, but the problem can be raised with a large number of signals that may come from intrusion detection system to the controller, which may target the controller to be a point of failure in a centralized system.

The work in [57] proposes IoT-IDM which is a network-based intrusion detection and mitigation framework to protect the IoT devices in the smart home networks. They consider SDN architecture because of its features like programmability, flexibility and etc. The framework is consist of three key modules: Device manager which is a collection of IoT device profiles including their characteristics and vulnerabilities, the sensor element is there to handle the inline monitoring phase, feature extractor module to get the best of the collected network traffic, the detection unit which can be signature-based or anomaly-based and finally the mitigation module which benefits of Openflow to generate rules for blocking or executing any other behavior on malicious flows.

In another source-near detection approach, [58] authors have implemented a test bed for their proposed source-near DDoS detection algorithm by using SDN networks that are implemented on Raspberry pi devices as a soft switch and controller. They have used an entropy formula to detect the attack with the assumption of decreasing the entropy in network packets when the network becomes a part of the attack.

Authors in [59] claim that they enhanced the FS-OpenSecurity architecture model by utilizing KNOT in the controller and the orchestrators in an SDN based smart home network architecture. KNOT consists of three components: packet parser. Flow graph builder and attack mitigation. They have studied three cases of protecting privacy, protecting voice assistants, and burst on communication channels, to evaluate their solution.

There is another research [60] with an aim to give a solution to secure smart homes with minimum load in traffic. For this reason, they suggest using SDN network architecture with an analyzing engine that serves on the cloud. The aim of the research is to show that cost of flow-based traffic analysis is much lower than the packet analysis, therefore they have implemented their testbed and experimented with the proposal with different network rules and IoT devices.

Authors in [61] proposed a general framework for IoT structure in software defined networks and also an algorithm for detecting the DDoS attacks which are originating from IoT devices. Their experiment also shows that with the use of this framework, controllers can detect the devices that are sending DDoS traffic. Their DDoS detection algorithm is based on the cosine similarity of packets and there is a threshold for this similarity. When the similarity is above the threshold, it is detected as a DDoS attack.

2.4.5 Blockchain Based Solutions for Mitigating IoT Attacks

Blockchain is a distributed database that is agreed on the peer to peer network. It is consists of sequences of block that is holding transaction data and each block is linked to the previous and the next one in the network. The verification of each block after the generation of it depends on the agreement in the community of the network nodes [62].

Due to the secure nature of Blockchain, there is a lot of proposed solution of it in smart homes. Authors in [63] have surveyed these solutions in which the majority of them are focused on the integrity of the data, which is the key point of Blockchain technology. Besides that, we have found two contributions of using Blockchain for detecting malicious activity in smart homes, and surprisingly, the authors promise

high performances with these solutions, which means they have taken care of the QoS in designing the solutions.

There is research [64] that focuses on detecting smart home bots by considering that bots communicate with each other and with their master. The solution utilizes the Byzantine Fault Tolerant Blockchain to find the mutual contacts in the communication and mark them to the blacklist. Two main components of this solution are home gateways which are considered to be secure and they have the duty of collecting network flows, pre-processing them, and send it to the Blockchain pools provided by second components which are device regularities and manufacturers.

Also, researchers in [65] proposed a Blockchain based framework that can control and enable trust-free operation in a network of IoT devices, with the feature of DDoS attack detection and prevention. In this proposed solution, Ethereum miners are assumed as safe against DDoS attacks and also gateways are immune from being compromised. The security of the framework relies on Ethereum for generating addresses for nodes that are IoT devices and custom coded smart contracts. It registers a device and dedicates a specific amount of gas to it. Gas represents the amount of resources that device allowed to use like bandwidth, and the dedicated gas depend on the specification of that device and its role in the network. So adversary faces a couple of challenges in order to interrupt the operation of the network. The first challenge is that the architecture of the network is decentralized one and failure in one system does not affect the whole network. The other challenge is the impossibility of entering into the network as a new device, though all devices are registered. Also in the case of compromisation of all IoT devices in the network, as their gas amount is limited, they cannot cause a denial of service and interruption for the server.

2.4.6 Other Methods Against IoT Attacks

Out of the box of an ongoing trend for security, especially smart home security, authors have proposed other methods that cannot be classified in the previous sections. But of course, there are similar type of approaches with them, which are not considered to be reviewed in this thesis because it does not fit in the aim of the thesis. It is good to mention that in some of these approaches, there are parts of them that can

be classified in the previous section. But as the main idea of the solution is not about utilizing the previously mentioned technologies, it is decided to mention them in this section.

Authors in [29] proposed a solution for securing IoT devices automatically, using a white worm they created, namely AntibIoTic. This malware exploits the spreading capability of other malware like Mirai in order to compete with them in infecting the IoT devices. Instead of making use of them for DDoS attacks, AntibIoTic tries to patch the security holes and alarm the administrators for further security operations. This white worm is compared in the related article to the other instances that already exist, they claim that the advantages of AntibIoTic over them are the good documentation and the openness of source code which shows their ethical approach. Besides that, AntibIoTic can do permanent security operations in the case that the administrator or owner of the device doesn't care about the alarms. For example in the case of using a weak password for SSH, AntibIoTic can change the user and password permanently to more secure ones.

There is a study on a modular framework that is focused on the integrity of stored and transmitted data on the smart home but it also provides availability and confidentiality in the form of authentication. The architecture of the framework is pretty modular and it is consists of two major levels, the kernel level, and the module level. At the module level, it has a smart appliance module and integrity module plus authentication module and in these modules are interacting with the kernel by system calls [66].

Researchers in [67] implemented a testing tool for checking if a particular IoT device is secure against DDoS attack or not. For this purpose, they have used an odroid and Mirai attack for testing IoT devices. The results are monitored in a dashboard created by Elasticsearch and Kibana. This solution is portable and patented by the makers, who are thinking of release it as OSS.

NCCoE proposed a network architecture design with the help of web services, for the prevention of DDoS attacks using Manufacturer Usage Description and service. In this scenario, any new IoT device connecting to the network asks the MUD file from the DHCP server, and then the request is passed to the MUD controller. Based on the device type the MUD controller downloads the MUD file and converts the rules in the

form of an understandable form for the router [68].

Authors in [69] have proposed an application layer DDoS attack detection algorithm by using Chaos theory. They have selected 8 features: request and packet numbers, data rate, average packet size, the combination of between request and response time as three features, and finally parallel requests. By using the NS3 simulation tool as s test bed they have implemented different scenarios to evaluate their detection algorithm. One of the key points of their experiments is that they have considered real-world conditions that effecting the results like the relation between the number of devices in smart home and the slowness of the connection time and etc.

The study of [70] offers collaborative network security in the name of Community Guard which has two major components, the guardian node which servers between the home router and modem and the second one is community outpost that runs in the cloud server and is a central managing unit. The nodes are pulling part of rules and blacklisted IP(s) from the intrusion detection system which runs on an outpost and push the suspicious traffic to the outpost. Also they can generate anti-DDoS rules if necessary. By the collaboration of nodes in homes, the outpost will get updated from new attacks and generate new rules to detect them actively.

Academicians with concerns in the consequence of DDoS attacks in E-Health care presented a study which in that [71] authors have suggested an algorithm that is based on the buffer utilization of the server and TTL of incoming packets. They assume that incoming attack packets are generally from geologically far places. They have put a threshold of 70 or 75 percent for the buffer utilization of the server and it is the trigger point for starting the algorithm to execute the prevention mechanism.

Authors in [72] suggest fog computing with distributed security prevention processes by utilizing virtual network functions. The architecture is collected in three layers from field control and local control to central control which is run on the cloud. By using and analyzing the Modbus/RTU protocols they try to detect attacks in the local control layer by signatures they have implemented only if the attack has bypassed the rule-based detection in the firewall. When suspicious traffic is detected, the central get alarmed and does further analysis by gathering data from distributed controllers in other networks.

ShadowNet [73] is another solution propose for the mitigation of DDoS attacks at the source level, using Edge computing. The node device is designed to run application-specific edge functions that are going to handle the requests correlated with web service. With this part of the solution blocking certain blacklisted IP(s) are possible without any effort on the gateways. Prototyped services show that this solution can detect DDoS attacks against web services, 10 times faster, and also it prevents 82 percent of the traffic from entering the internet.

2.5 Summary and Evaluation of the Literature

The security of IoT devices can be described in multiple layers. However, an important feature of IoT that threatens the network is that these devices use to communicate with each other permanently and it is what the market develops forward with it. The goal is to produce devices that can be inter-operable and communicable with each other. Therefore a leak in any of the IoT devices can expand the threat to the others and the whole network.

The limitation of proposing a solution for IoT security, especially in the network layer, is that there is a heterogeneous type of devices and manufacturers. So a proposed solution should become compatible with all of them, at least in specific environments and conditions.

The other limitations that can be mentioned about the IoT, is the limited resources in the matter of power, memory, and process units. These limitation needs to be addressed with lightweight solutions for detection mechanism, encryption and etc. Also usually the gateways and hubs of the smart home suffer from the limited resources and it is one of the most important reasons that researchers believe that the security of smart homes should be outsourced to the more powerful processing units like ISPs.

The literature review shows, that DDoS protection performs better when it's done near the source. Smart home security tends to rely on SDN structure as the architecture and intrusion detection strategies need to benefit both from misuse based and anomaly based methodologies as a hybrid approach to reach the maximum outcome. Besides that, there is a need for reliable standardization like manufacturer usage de-

scription. The recommended standard need to be implementable both for open source and proprietary network devices in order to be acceptable on a worldwide scale.

CHAPTER 3

METHODOLOGY

As a response to the growing number of vulnerabilities in IoT devices, novel mitigation methods have to be developed for network security. We ask if IoT devices have distinctive and recognizable traffic patterns that can help network administrators, and in the automated environment, network management services to identify them and white-list their pattern of network traffic or block them if exceeding the thresholds. Revealing regularities in IoT traffic patterns has the potential to bring a set of advantages in the design of mitigation methodologies. For example, lightweight machine learning algorithms can be designed for protecting IoT networks from scalable attacks. Besides developing ML algorithms, pre-identified rule sets may be used for developing hybrid systems with ML-based mitigation models.

In this chapter, we explain how we captured data for analysis, the data analysis methodology, as well as the description of the implementation of a test-bed infrastructure in two case studies, with the purpose of demonstrating the real-world applicability of the proposed approach. The chapter also introduces the details of the application and evaluation of the MUD concepts within the framework of the development of mitigations to protect MUD-enabled IoT networks.

MUD is now an IETF standard that can enforce white-listing access control policies in IoT networks. The main object of this standard is the MUD file, which is a specific file for every type of IoT device. These files include the needed communication rules for each IoT device. By following the MUD file, it is possible to restrict the communication in the network, in order to avoid anomalies.

In particular, in search of identifying distinctive, recognizable network traffic patterns,

we collected our dataset and analyzed the data to find the features for recognizing patterns in the traffic flow. For the purpose of evaluation, we generated MUD files to observe the proof of work for the whitelisting approach. Moreover, we further investigate MUD as an acceptable solution to the problem. In particular, we employ MUD for investigating the easiness of implementation and the security mechanism that it promises [74]. In the next section, we describe the equipment and network that have been used for this purpose.

3.1 Equipment and Network Topology

A review of publicly available datasets of IoT network traffic reveals the potential for the usability of MUD concepts on a frequently used IoT device, namely video cameras. A major issue with the video cameras is that they are not fully compatible, in most cases, with MUD concepts, by their design [75]. That is due to the nature of monitoring systems, where system administrators may want to monitor the scenes from different places and from different IP addresses. A similar situation applies to general-purpose smart home voice assistants since they may also have a wide range of diversity in their network traffic patterns. Accordingly, the diversity of use in these IoT devices may result in in-distinctive traffic patterns. This thesis aims at answering this research question: whether IoT devices exhibit distinctive, regular traffic patterns in spite of their diversity in use at different fronts. Below we present how we approached the problem in two specific devices. We selected the Motorola baby care camera [76], due to its wider variety of functionalities, compared to commercially available, common video cameras for end-users. Also, we selected a Google voice kit [77] as the simple-to-implement and ready-to-develop home assistant. Figures 3 and 4 show the Motorola baby care camera and the Google voice kit used in the present study. IoT networks usually require a gateway to connect to the internet,

Figure 3.1: Motorola baby care camera

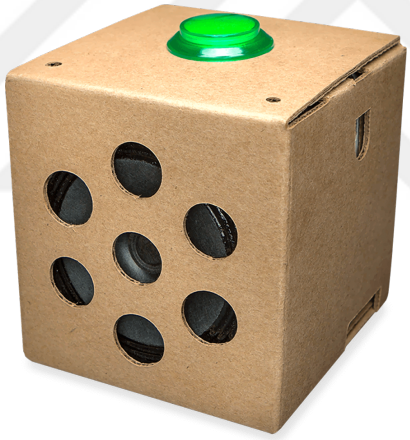

Figure 3.2: Google voice kit

either directly or by means of being a subnet of a larger network. In the present study, we utilized a TP-Link Archer c7 access point as the gateway [78]. Another reason to select this particular gateway is that it is one of the affordable access points with the compatibly of OpenWRT as a firmware [79]. Figure 5 shows the network architecture of the established IoT network in the lab. Below are further characteristics of the configuration and assembling procedure of the aforementioned components:

- The Google Voice kit needs to be physically assembled with a Raspberry pi [80] as a processor unit. Next, there needs an installation of an operating system and configuring the software and scripts recommended by Google.

- The Motorola baby care needs to be configured through a synchronization process with a smart mobile phone, which has the Motorola Hubble application installed.

- The TP-Link access point is flashed with the latest version of OpenWRT (v19.3). The requirement packages for storing, configuring, and capturing traffic.

These selected devices were used for the implementation of the METU IoT laboratory for capturing benign traffic. In the following section, we describe the data collection procedure.

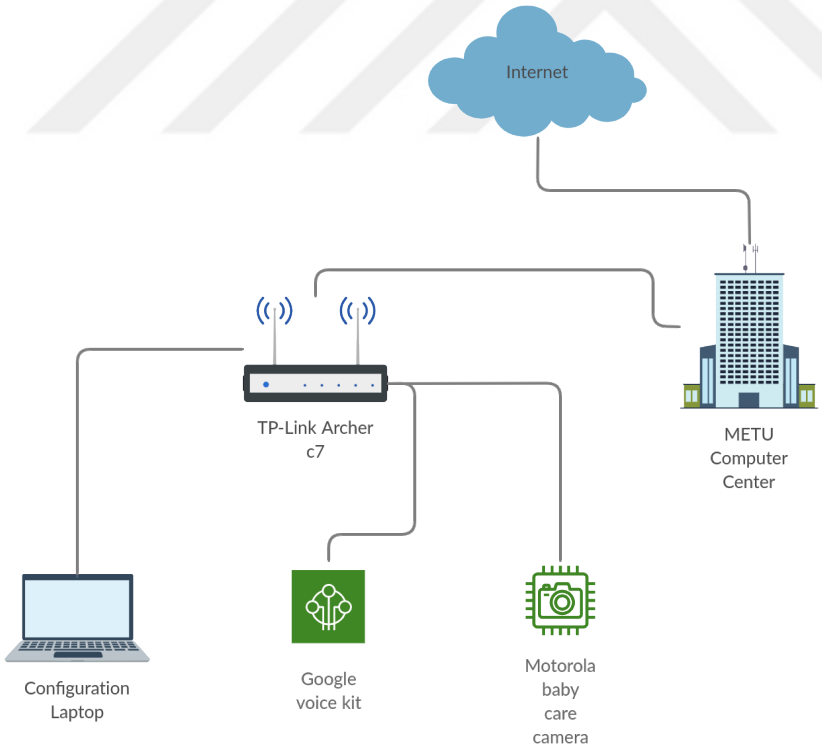

Figure 3.3: Architecture of METU IoT Lab

3.2 Data Collection

In order to capture network traffic, a tcpdump 4.9 was installed on OpenWRT, and block-mount, e2fsprogs, kmod-fs-ext4, kmod-usb-storage, kmod-usb2, and kmod-usb3 packages were deployed for creating a permanent storage directory, which was simply a USB storage attached to the access point for storing the pcap files. In terms of the physical infrastructure, we had an access point connected via WAN port to the internet, and IoT devices were connected to the WLAN. No other devices were connected to the access point.

Next for preparing the required packages for the access point, we started to collect data from Motorola baby care and Google voice kit. For the procedure of capturing traffic from Motorola baby care, we started monitoring and executing all of the features of the application from capturing images, playing music, talking to the baby and etc. We captured two files, one of them is a set of capture for 24 hours, and the other is about 1.5 hours. Since the two pcap files were similar to each other, we selected the smaller pcap file for the analysis.

After assembling and running the Google voice kit, we started to ask voice commands to it in two sessions, and each of them took about 1.5 hours to capture. The list of the asked commands is derived from the relevant support website and we tried to cover virtually all categories of voice commands reported in the manuals. We paid particular attention to foreground the voice commands that are mentioned in discussion forums.

During data capture, IoT devices were isolated from the rest of the network, so the traffic completely belonged to the communications of the IoT devices. Also, these devices were not in communication with other IoT devices, and the only destinations were the ones on the WAN side. However, in some cases like Google voice kit, it is possible to have communication with other devices in the local network if permission was acquired. But the aim of the research was to analyze the communication pattern with the WAN side. Although the communication in the LAN side can be interesting for the researchers working on the security of IoT devices, solely.

3.3 Datasets

3.3.1 METU IoT Dataset

In this part, we describe our dataset and its features. Also, we mention other publicly available datasets and two research groups on IoT security and their efforts on publishing these datasets for comparison.

After collecting the benign data, reports and log files are generated with tools suggested by the Stratosphere lab [81]. Accordingly, in our study Capinfos [82], PassiveDNS [83], TCPdstat [84], DNStop [85], and Zeek [86] are the tools used for the analysis. Also, flows are bidirectionally mapped using Argus [87], which made the dataset files easier to use with the ML techniques. Below is a description of the tools and the outputs of their processing. Capinfos: Capinfos is a program published by the Wireshark team. It can generate statistical reports from the input, which is a traffic capture. The reports can be in two formats: long and table. Long reports are suitable for human readability, and table format is suggested to do analysis with other tools.

- DNStop: DNStop is a tool built on libpcap. It provides statistics related to DNS queries, and it is used to provide a map of destinations and their popularity from the local network.
- Passivedns: This tool also filters DNS queries and give useful information about the directions of the traffic flow. It can continuously be used on interfaces or for digital forensic operations.
- TCPdStat: It is a network forensic analysis tool that has been used for many years. Today, it is not well supported, but it provides great results to focus on.
- Zeek: Previously known as Bro, Zeek is a network monitoring tool that benefits from hybrid solutions in detecting intrusions. By processing pcap files, it generates different related reports based on the protocols and services found on the pcap.

Besides statistical analysis, pcap files are used as an input file for generating MUD files. There is a tool called Mudgee, written by the UNSW research group [88],

which can create MUD profile that are not provided by IoT companies. Also, there is an alternative for that, named mudmaker.org [89] which is also helping manufacturers and network admins to produce their MUD files. Besides that, following the syntax of the YANG model [90], suggested by IETF, there is a more convenient way to generate MUD files.

3.3.2 The UNSW Dataset

In this section, we present the UNSW dataset for the purpose of comparison with the METU IoT Dataset. The research group at the University of New South Wales conducted an experiment in which they collected traffics of 28 distinctive IoT devices [91]. They also have provided attack traffics and as a result, they have proposed solutions, and some of them are based on manufacturer usage description, anomaly detection with machine learning methods, combining SDN for leveraging the security of IoT networks, as well as identifying IoT devices based on their traffic patterns. The collected dataset has been provided in PCAP and CSV file formats. Also, using the Mudgee software that they have published publicly, they had generated MUD profiles for 20 of their IoT devices. These profiles can be accessed on their webpage. MUD profiles provided in JSON files to help controllers to generate appropriate access controls for the network.

3.3.3 The CTU Dataset

The Stratosphere research lab published a dataset in January 2020 [92], which is a necessary and useful dataset for security researchers. They captured the traffics of 20 IoT malware in the lab and analyzed them both manually and also with Zeek anomaly detector. Besides this, they captured benign traffic of three IoT devices with normal behaviors.

This research team also works on profiling device behaviors and to improve the detection of anomalies easier with whitelisting even for general-purpose network environments. Their previous research led to the release of Stratosphere IPS [93] that aims to help NGOs and other nonprofit organizations to secure their networks from

attacks. Their collected traffics published in pcap files originally, plus there are other useful files generated from different tools, that can help you analyze the traffics.

3.3.4 Comparison of Datasets

There are similarities and differences in these three datasets. For example, all three datasets captured benign network traffics in order to present the normal behavior of the target IoT devices. Also, the authors selected different categories of IoT devices, but the majority of those devices are designed for a smart home network.

The differences in benign captures are that in CTU and METU IoT datasets, the captures targeted IoT devices solely, but in the UNSW dataset, the captures have traces of different devices in pcap files. There are no attack traces of the METU IoT dataset, but the attack traces for CTU belongs to the traffics of IoT malware, and the UNSW captured attacks that are more general and not generated from malware but from known attack tools.

3.4 Data Analysis

In this section, the collected traffic from Google voice kit and Motorola baby care is analyzed in order to find the variability of the protocols and the destinations. The question is to find out if the traffic analysis can help to find a pattern of communication for each device. Also, if it does, can whitelisting methods control the flow of communication in IoT networks.

3.4.1 Google Voice Kit

For the purpose of getting an overview of the protocols, flows and the data volume transmitted between the Google voice kit, and the rest of the network including the internet, our collected pcap visualized using the CapAnalysis tool [94]. Here is an overview picture of the Google voice kit pcap file. Figure 3.4 shows that only three distinctive known protocols plus at least one unknown protocol are involved during

the capturing of the Google voice kit traffic.

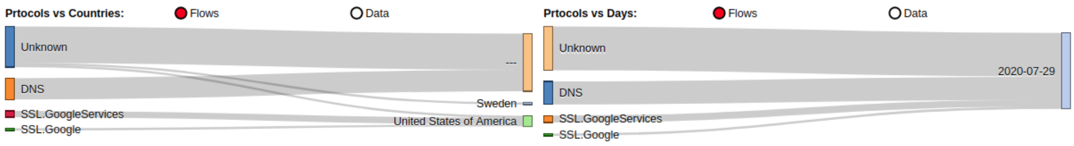

Figure 3.4: Overview of Google voice kit pcap file

Figure 3.4 also shows that the majority of the flows are dedicated to the unknown protocol(s). For finding out what is the unknown protocol and how does it communicate, there is a need to look at the flows. Here again, for the purpose of having a better picture, we used the flow table generated by the CapAnalysis tool, but it is better to analyze the Argus file, that we have mentioned earlier in this chapter. As mentioned before, the Argus files and other customized statistical reports are available for more accurate analysis on pcaps (Figure 3.5).

	Date ↑	Time	Source IP	Destination IP	Destination Name	Source Port	Destination Port	L4	Protocol
	2020-07-29	17:17:09	192.168.1.223	239.255.255.250		43012	1900	UDP	Unknown
	2020-07-29	17:15:09	192.168.1.223	239.255.255.250		58557	1900	UDP	Unknown
	2020-07-29	17:13:09	192.168.1.223	239.255.255.250		41131	1900	UDP	Unknown
	2020-07-29	17:11:09	192.168.1.223	239.255.255.250		52310	1900	UDP	Unknown
	2020-07-29	17:09:09	192.168.1.223	239.255.255.250		41091	1900	UDP	Unknown
	2020-07-29	17:07:09	192.168.1.223	239.255.255.250		48004	1900	UDP	Unknown
	2020-07-29	17:05:09	192.168.1.223	239.255.255.250		41701	1900	UDP	Unknown
	2020-07-29	17:03:09	192.168.1.223	239.255.255.250		35869	1900	UDP	Unknown
	2020-07-29	17:01:51	fdc1:2295:f366::5ef	fdc1:2295:f366::1		49931	53	ESP	DNS
	2020-07-29	17:01:51	192.168.1.223	172.217.169.202	safebrowsing.googleapis.com	59754	443	TCP	SSL.GoogleServices
	2020-07-29	17:01:51	192.168.1.223	192.168.1.1		18059	53	ESP	DNS
	2020-07-29	17:01:09	192.168.1.223	239.255.255.250		49988	1900	UDP	Unknown
	2020-07-29	16:59:09	192.168.1.223	239.255.255.250		52761	1900	UDP	Unknown

Figure 3.5: A part of Google voice kit flow table

Protocol	Country	Bytes Sent	Bytes Received	Bytes %	Lost bytes Sent	Lost bytes Received	Packets Sent	Packets Received	Packets %	Duration
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3
DNS	---	45	61		0	0	1	1		0
SSL.GoogleServices	US	1.2 K	1.9 K		0	0	6	8		240
DNS	---	45	61		0	0	1	1		0
Unknown	---	668	0		0	0	4	0		3
Unknown	---	668	0		0	0	4	0		3

Figure 3.6: A part of Google voice kit flow table

In figure 3.5 and figure 3.6, we have some parts of the flow table for the Google voice

kit. The flows related to the unknown protocol are started to communicate periodically within 2 minutes and by looking at the destination IP address, it shows that these flows belong to the Simple Service Discovery Protocol (SSDP) [95]. Besides the SSDP, the other three protocols that are used for the communication of Google voice kit are DNS, SSL.Google and Google service protocol. For analyzing the specific destinations of the Google service protocol, we need to find out the involved IP addresses (Figure 3.7).

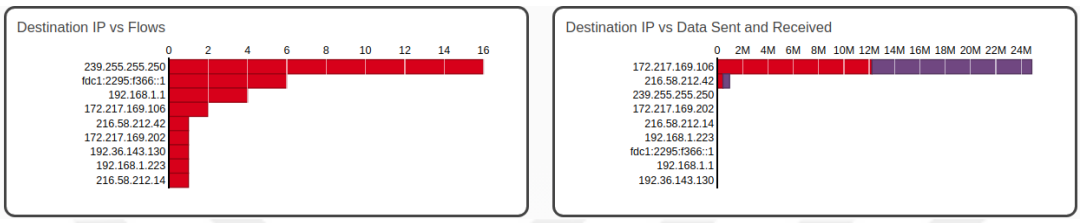

Figure 3.7: Destinations of Google voice kit

For this purpose, we can use Passivedns and DNStop tools to have a better and detailed look at the data, but again for the purpose of better visualization, we used the statistics section of the CAPAnalysis. Figure 9 shows that the majority of sent and received data belong to a specific IP address, so we can calculate a ratio for the communication of the device and even put a threshold for the volume of data sent and received in repeating the time period loops. It is necessary to mention that all the public IP addresses in the above picture belong to Google servers. In the following, figure 10 also describe the ratio of data transmitted with the number of flows.

3.4.2 Motorola Baby Care Camera

All of the procedures of generating reports and information on pcaps for better analysis of Google voice kit are repeated for Motorola baby care. The CapAnalysis tool is used here to show a visualized version of some statistics and information regarding this capture. Figure 3.9 is an overview of flows and protocols for Motorola baby care.

Motorola baby care camera is using SSL, HTTP, DNS, RTMP, and unknown protocols. It also shows that this device is using cloud-based services from Google and Amazon. The HTTP connection is the source of question that it is transmitting sensi-

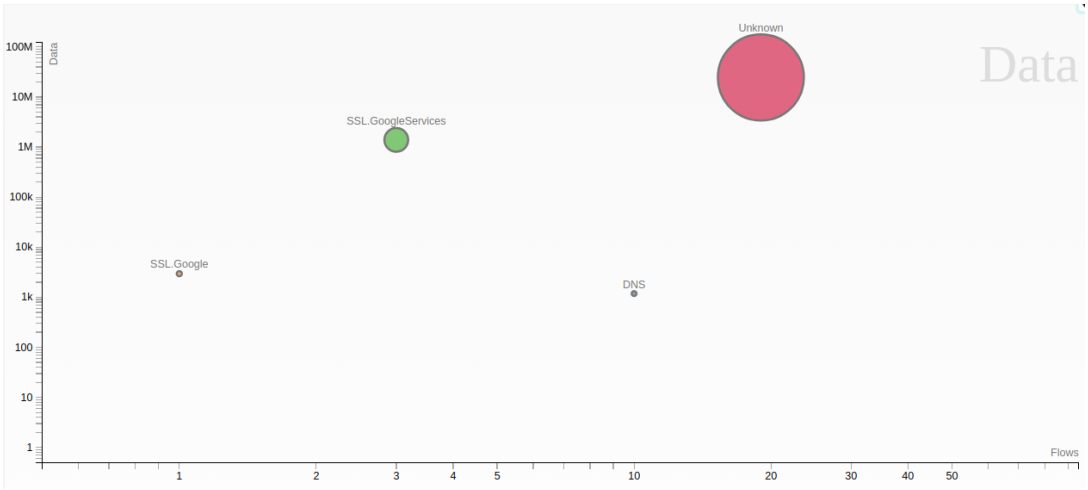

Figure 3.8: Destinations vs. flows in Google voice kit

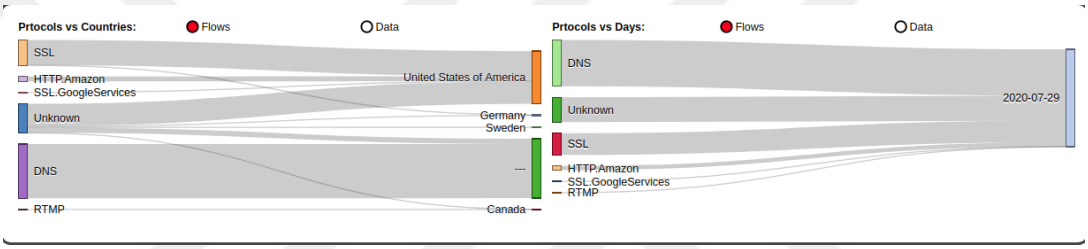

Figure 3.9: Overview of Motorola baby care camera pcap file

tive data or not. But this should be examined in different research, and the topic is out of the scope of this thesis. To find out about the unknown protocols, the flow table needs to be investigated.

	Date	Time	Source IP	Destination IP	Destination Name	Source Port	Destination Port	L4	Protocol ↓	Country	Bytes Sent	Byte
ⓘ	2020-07-29	16:43:08	192.168.1.242	52.86.200.176	upload1.hubble.in	58387	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:45:01	192.168.1.242	52.54.48.98	api.hubble.in	53718	443	TCP	Unknown	US	0	0
ⓘ	2020-07-29	16:42:35	192.168.1.242	3.212.59.137	upload1.hubble.in	56058	80	TCP	Unknown	US	8.2 M	716
ⓘ	2020-07-29	16:42:01	192.168.1.242	3.232.128.34	upload1.hubble.in	41129	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:41:28	192.168.1.242	52.86.200.176	upload1.hubble.in	39110	80	TCP	Unknown	US	8.2 M	716
ⓘ	2020-07-29	16:40:55	192.168.1.242	3.212.59.137	upload1.hubble.in	34874	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:42:34	192.168.1.242	3.225.140.9	upload1.hubble.in	53951	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:40:21	192.168.1.242	3.232.128.34	upload1.hubble.in	33659	80	TCP	Unknown	US	8.2 M	716
ⓘ	2020-07-29	16:42:16	192.168.1.223	239.255.255.250		47674	1900	UDP	Unknown	---	668	0
ⓘ	2020-07-29	16:40:22	192.168.1.242	3.225.140.9	upload1.hubble.in	52068	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:39:14	192.168.1.242	3.225.140.9	upload1.hubble.in	52062	80	TCP	Unknown	US	8.2 M	716
ⓘ	2020-07-29	16:39:49	192.168.1.242	3.225.140.9	upload1.hubble.in	52065	80	TCP	Unknown	US	52 K	716
ⓘ	2020-07-29	16:39:16	192.168.1.242	52.86.200.176	upload1.hubble.in	39100	80	TCP	Unknown	US	52 K	716

Figure 3.10: A part of flow table for Motorola baby care camera

We selected this particular snapshot from the flow table to show that unknown pro-

tocols are three different usages of the network. The first one, as mentioned before, is for SSDP, distinguishable by its port number 1900. The others use to connect to the ports that majorly use HTTP and HTTPS, mean port 80 and 443. From the destinations, it can be concluded that the connection on port 443 to the API can be for the authentication purpose and the other which is on port 80, for the upload purpose. However, the later one shows that uploading data, including the pictures and sensitive data to the Hubble server is done by using the HTTP protocol. As the security of this device is not in the scope of this thesis, we skip further investigation on this topic.

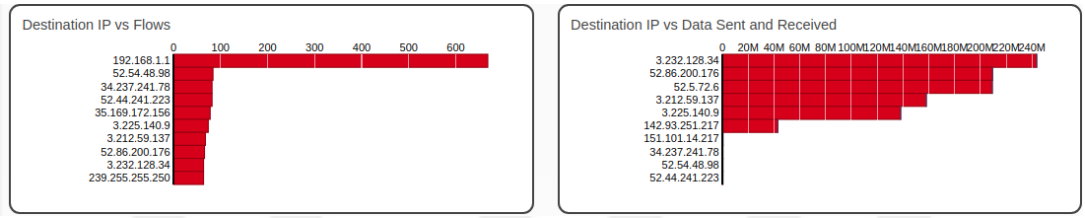

Figure 3.11: Destinations vs. flows in Motorola baby care camera

Figure 3.11 shows us that different functionalities of the Motorola baby care camera are synchronized with a different destination in the cloud. Although these subnets are more than what we observed in the Google voice kit and they can be different in other experiments, but they are still low in an amount enough to be included in the whitelist. Also, it is good to mention that the whitelisting process can be done on DNS results, which can be more static records as a domain name. Also, the below picture shows the relation of flow amount with the data amount in this capture. It can vary in different situations and scenarios, but with further experiments, there can be found a relation, especially for the DNS protocol to be used for anomaly detection purposes.

3.5 Implementation of MUD on Network

We implemented a specific built of MUD manager in our network in order to evaluate the proposed security features by IETF and test if our selected software stack does work in our lab environment. Also, this implementation shows the proof of work for our generated MUD files for Google voice kit and Motorola baby care, which is

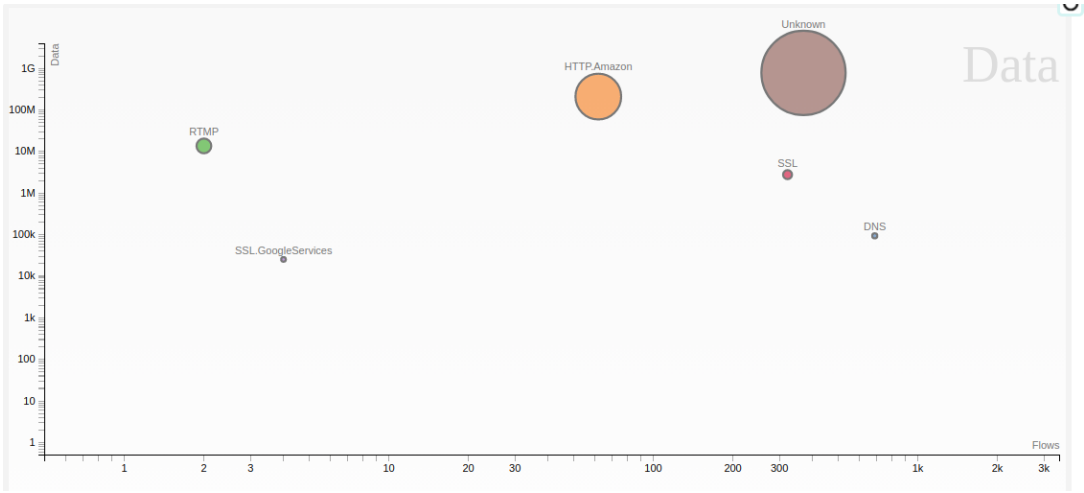

Figure 3.12: Data volumes and flows in Motorola baby care camera

explained in the following.

3.5.1 3.5.1 Background on the Manufacturer Usage Description (MUD)

Internet Engineering Task Force and industry collaboration from Cisco networks, with the support of other foundations and organizations, proposed an RFC to address the need for a standard for IoT networks (RFC 8520). Manufacturer usage description (henceforth, MUD) ensures that when an IoT device joins a network, it will access the required resources, based on the description that its manufacturer defined and not more than that (cf. The concept of confidentiality in cyber security). With this insurance, IoT devices are expected to function appropriately, and their attack surfaces will be significantly decreased.

Besides all its technical features and advantages, there is key progress that comes with MUD, and it is that the manufacturers are involved in contributing to the mitigation of cyber attacks against IoT devices. Thus they become more involved regarding the security of their products. They need to emphasize the functionality of their device in the concept of network communications and even host a MUD file server. These steps are beginning steps that can lead companies to get more intense about security evaluations, requirements, updates and etc.

3.5.2 The Infrastructure and Implementation of MUD Standard

The MUD file is structured by the YANG model (RFC 6020). A valid MUD file contains two root objects: a "MUD" container and an "ACL" container. The arguments can be matched to make policies in the container object. Also, some augmentation, like the ones that relate to the software version or last update, are handled in the MUD container. There is a sample MUD file in Appendix A, presented by IETF in its draft. This MUD file intends to connect to port 443 on TCP in order to get service from the cloud. The point that can be mentioned here is that the direction and initiator of the connection can be filtered. These two parameters are some of the most important filters to rely on for creating mitigation policies.

The augmentation and the syntax to use for MUD files based on the YANG model can be found on the RFC 8520. As the MUD file is a sensitive file that defines the policy of the network security in its core, it needs to be protected from adversaries. Signing this file will prevent some of the problems that may occur in the future. For this purpose, Cryptographic message syntax or CMS can be used. For more information on this part, RFC 5652 [96] can be useful.

MUD files must be verified by the MUD manager, and in order to do this process, the MUD-signature file needs to be retrieved and compared to the signature in the MUD file. After the matching, the MUD file is verified and ready to use by the MUD manager. This solution increases the accountability of the MUD file based on the reputation of the signer of the MUD file that mostly should be the manufacturer of the IoT device.

Currently, there are three different methods to develop a MUD file for IoT devices. It depends on the conditions of the developer to select one of them. However, our observation shows that in an accurate operation, the result of these methods will be the same. The first method is the base of the other two, and it is to follow the syntax of the MUD file with the YANG model. By reading the syntax from the RFC draft of the IETF, this method can be accomplished.

The second method is to use mudmaker.org, which is an online platform that helps people to make MUD files by the wizard. For making use of this method, the devel-

oper of the MUD file must be aware of the connections that IoT devices need to have. Including the source and destination port, protocols that it uses, the domain name or IP addresses that the device wants to connect, and so on. The third method may be useful for the conditions that the developer doesn't have enough knowledge about the communication pattern of the IoT device. For this purpose, the UNSW research group has produced a tool called Mudgee. The Mudgee gets an input of a pcap file and gives the output of the MUD file in JSON and CSV. This tool has written in Java, and it is cross-platform. In order to achieve an accurate MUD file that comprehends all the needs of the IoT device to service, the developer needs to test all of the possible functionalities of the IoT device, from the general functions to updating firmware or etc.

3.5.3 The Architecture of the MUD-Enabled Network

The map of the network working with MUD can be described in different scenarios based on the component and technologies that support this standard. The simplest image of this network structure can be described in this shape. In this architecture [9],

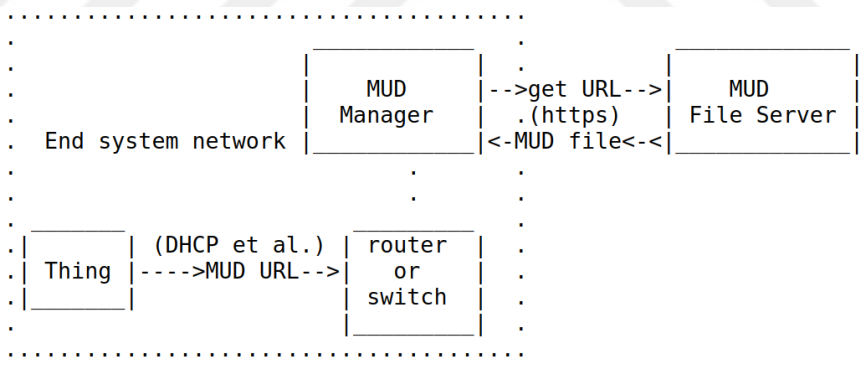

Figure 3.13: Architecture of MUD-enabled network

the thing sends the MUD URL to the gateway, which can be a software or hardware switch or even a router. The switch then sends the packets to the MUD manager, which can be on the same device as the switch is or a separate device. The protocol of sending the MUD URL also can be different based on the implementation. It can be sent through DHCP packets or by following the 802.1X standards [97] of IEEE by utilizing the EAP over Radius. The MUD manager extracts the MUD URL and asks

for the MUD file with a GET request from the MUD file server, which can be a web server hosting in the cloud. After the reception of the MUD file, the MUD manager translates it to specific network configurations and updates the configurations in the network gateway or any device responsible for the access control mechanism in the network.It is necessary to mention that the configurations and the MUD file itself will be purged after the device disconnects from the network.

3.5.4 Produced MUD Files for Google Voice Kit and Motorola Baby Care

As presented in the previous sections, we have generated a MUD file for the Google voice kit and Motorola baby care. As we are not the manufacturers of the devices, it was necessary to generate the first version based on the output of the Mudgee. Also, analyzing the pcap files helped us to get insight into the communication patterns. For example, some of the rules, especially for the camera devices or remote access enabled devices, might be wrong by generating only from pcap files. Consider this example that a parent wants to monitor a camera from different locations and different IP addresses. If the produced MUD file is only whitelisting the incoming connection from a specific IP address that had been accessed remotely to the device, then the other IP address wouldn't be able to access the device. For this purpose, the pcap should be analyzed for the different requests and how the remote access controls work. If these accesses are controlled through an API or a service from the cloud, then only the IP address that needs to communicate for monitoring purposes is the one that belongs to the cloud.As our generated MUD files are not short in length, we presented the MUD file contents for the Google voice kit and the Motorola baby care in Appendix A.

3.5.5 Implementing MUD Manager

For the purpose of evaluation, we needed to enable MUD support in our network. For implementing such a new standard, we needed a technical report on how the implementation is possible with software stacks.
As the MUD concept is new, there are not many experiments with this technology

neither in academia nor in the industry. However, one of the major contributors and supporters of the MUD, NIST, has developed guidance in three different levels for MUD implementation that includes four different types of builds. For executing and implementing these builds, NIST has asked for the contributions of different involved companies like Cisco, Digicert, Yikes, Molex and etc. Three of these four builds rely on proprietary software and technologies. The last build is based on the open source software, but it still needs some level of involvement with commercial services for the signature part. Due to our limitations from the hardware logistics and the signature part, we decided to try a new build on top of an alternative software as a MUD manager called osMUD or, better say, open source MUD. It might help to mention that there was no technical guidance for building this network at the time of our experiment.

For the purpose of implementing MUD, we compiled the osMUD [98] for the TP-Link Archer c7 v2 model which is running on the MIPS architecture. Next, we installed requirements and the osMUD, and after some configurations of the DNSMasq for reading the DHCP packets to extract the MUD URLs, the service started to work.

In this chapter, we presented our motivation and methodology to capture the network traffics of Google voice kit and Motorola baby care camera. Also, we provided useful report files that can be used for more specific investigations and also learning methods. Besides that, we introduced two other alternative datasets to be investigated and used for different purposes of researches. Next, we analyzed our dataset in order to see if we can find distinctive patterns to recognize our devices and also control them with whitelisting methods.

Furthermore, we explained how the manufacturer usage descriptions work, how we implemented MUD in our network, and how we generated MUD files based on these analysis. In the next chapter, we explain the evaluation procedure of the experiments in order to verify that MUD based network can preserve the communication needs of the devices and additionally add a layer of security on the network stack.

CHAPTER 4

RESULTS AND EVALUATION

After analyzing the collected data from IoT devices and concluding that whitelisting methods can be implementable on these two devices, we generated MUD files to implement the whitelisting method and enforce the access control entries by building our network to be compatible with MUD. In this chapter, we mention our results from manual analysis, the generated MUD file description, and the findings in the process of implementing MUD components in our network. Also, we evaluate each MUD file in order to see its proof of work and reliability in the aspects of security.

4.1 Results of Data Analysis

The analysis of data shows that the Google voice kit uses only three different protocols in order to serve the user. These protocols are SSL, SSDP, and DNS. Also, the destinations for Google voice kit are limited to a couple of destinations that all of them are belong to Google services. These results can help to conclude that whitelisting the network communication for the Google voice kit is a reasonable approach. Results of analyzing Motorola baby care traffic pattern also shows that it uses five protocols. SSL for Google and also the Hubble API, HTTP for Amazon cloud, RTMP for streaming, SSDP, and DNS. Also, the destinations are limited to less than ten destinations, but if we use domain names, the destinations can be count for less than four different destinations. similarly, these results show whitelisting the network communication for Motorola baby care is also an implementable approach.

Another result of the analysis is lead to generating MUD files for the Google voice kit and Motorola baby care camera. As predicted before, by manual analysis, there

is no difference between different approaches for generating MUD files, and it is acceptable due to the nature of the file. These MUD files are supported by METU IoT lab and can be implemented in any IoT network which has these two devices in it. Also, by modifying MUD files to not use specific protocols like SSDP, the attack surface can be reduced more.

4.2 Network Implementation Results

As mentioned before, we used open source MUD (osMUD) as the MUD manager. As there were no detailed builds of this software as documentation, we did it experimentally. This implementation showed us for building the installation file that, in our case, were ipk format, the best possible approach is to use containers. During configurations, the promised folder structure can vary on different platforms. Although the authors mentioned that DNSMasq configurations could be skipped, our experiment shows that in order to have a working MUD service, the researcher needs to check the configurations to be sure that the MUD URL can be fetched from DHCP requests. We don't suggest this build for enterprise and operational implementations due to the version of the MUD manager, which is 0.2.0. But it can be a platform for testing new features for developers and researchers.

4.3 Evaluation

For evaluating MUD files and their reliability, there needs a working network environment that supports MUD as a standard. As we implemented this compatibility by using osMUD manager and modifying configurations on the gateway, which runs OpenWRT as a firmware, we observe that the Google voice kit and Motorola baby care camera can function without any problem in serving the user in the MUD enabled network. This is proof of work for MUD files that are containing access control entries and also for MUD managers that enforce those entries in the access control list on the OpenWRT's firewall. Also, it's necessary to mention that the MUD manager blocks every communication from any source to any destination as a default at the starting point. So the rules that are added to the access list are using the ACCEPT

keyword in every entry.

During the evaluation process, there was no sign of low performance. However, the comparison between the normal working network and MUD-enabled network must be made based on different parameters in more crowded intranet networks.

The osMUD had some problems in changing the MUD file entries to rules, but it was solved manually. Due to the developing stage of this MUD manager, it is acceptable to have such problems.

The MUD enabled network and MUD standard itself, and our experiment replication needs some security evaluation in general. There might be some questions about the security of architecture, the security of each component, and possible scenarios for bypassing MUD-enabled networks. In this part, we explain possible use cases that can encounter problems or security vulnerabilities in these networks.

4.3.1 Use case 1: Using Not Approved and Unverified MUD file

The whole architecture of the MUD enabled network is designed to secure the integrity and verification of the MUD file. But still, there can be methods to bypass those security mechanisms. Also, there can be some cases that there isn't any provided and verified MUD file from manufacturers, like our case in the experiment. In this case, there might be some problems that are needed to be explained. If the MUD file is created based on the traffic collection, it might not be comprehensive. Objectively, some functions and features of the IoT devices might not have been executed during the data collection procedure. This may result in some missing patterns in the generated MUD file. For example, consider that we didn't execute the firmware update during the network traffic collection of the Motorola baby care camera. In this situation, if the update server was a separate server from the previously accessed server via camera, then we are missing the destination address to include in the MUD file. As a result, our generated MUD file wouldn't have the entries for that destination, and the camera cannot connect to that server in the MUD-enabled network, which is using that specific the MUD file.

Also, in some cases, the network administrator may want to restrict some protocols

or ports more than what the MUD file originally intended to use. For example, the SSDP protocol may be blacklisted due to some security policies in the network of organizations. In this case, the administrator either needs to use a customized version of the MUD file or obligate the rules or flows from a security layer lower than what the MUD manager executes.

4.3.2 Use case 2: MUD URL Spoofing

One of the possible theoretical ways to compromise the devices is to address a wrong MUD file by sending a modified request that contains a MUD URL to the MUD manager. If an attacker in a scenario is able to modify the DHCP request (or any other protocol that the network support for initiator request) and change the MUD URL to a fake URL that addresses a fake MUD file, then the layer of security added by MUD is bypassed. This modification can be implemented in two ways.
First, the man in the middle attack, which is a known attack type that sniffs, interprets, modify, and forward the packets in the transmission procedure. Second, another device can spoof the newly added device's mac address and send the initiator request containing the MUD URL. In both cases, the result is the obligation of fake access control entries, which can denial the service of IoT devices or make it available to communicate with the destinations that are not acceptable by manufacturer or network administrators.

4.3.3 Use case 3: DNS Spoofing

Another approach to bypass MUD enabled network and change the MUD file with a fake one is to spoof DNS query results or DNS spoofing. In this scenario, after the MUD manager extracts the MUD URL from the DHCP packet, it needs to resolve the IP address of that domain. If, in any case, the adversaries can spoof this result and send a fake response, then a fake web server will serve the fake MUD file for the MUD manager, and as a result, the MUD manager will obligate fake access control rules to the IoT device.

4.3.4 Use case 4: Attacks on MUD Manager

The last approach that we mention as a threat for MUD enabled networks is to have a vulnerable MUD manager platform. If an attacker from inside of the network or possibly outside of the network can get access to the MUD manager by exploiting any vulnerability in the MUD manager itself or the firmware or operating system that serves the MUD manager, any modification in the policies and security mechanism of the network is possible. Of course, this possibility always remains for centralized security solutions, but the reason that we mention this particular threat is that as MUD managers are still in the developing stage, finding critical vulnerabilities in them should be considered.

In this chapter, we summarized the results of the analysis and our implementation of the IoT network. There are some limitations, open questions, and discussions that are explained in the next chapter.

CHAPTER 5

DISCUSSION AND CONCLUSION

In this chapter, we summarize the findings of the study and also the usages and importance of them to improve the security of IoT networks. Also, we mention our basic assumptions and limitations in the implementation. Finally, the part of the future work includes the possible research topics that can help to develop new research questions from the perspective of security.

5.1 Summary of Findings

During the research, we surveyed different approaches and solutions to smart home security issues. The motivation was to detect and prevent DDoS attacks generated from the IoT devices that are got infected by different types of malware. Then we checked if whitelisting policies are effective from the perspective of operation and security on a smart home scale IoT network. We selected MUD as a new, open to developing and simple in concept, architecture, implementation, and customization standard to use to enforce whitelisting policies in the network. Our generated MUD files for each of the IoT devices based on the syntax of the Yang model are published openly for use and customization. For evaluating the MUD files, we implemented a unique build of the MUD manager in our network and tested the proof of work for the MUD files and the promised working proof of available open source approaches for building the MUD enabled network.

During the research, we surveyed different approaches to smart home security issues. The motivation was to detect and prevent DDoS attacks which are generated from the IoT devices that are infected by different types of malware. We found that majority

of the proposed solutions are based on the software defined network architecture and machine learning methods.
The analysis of the collected dataset showed us that IoT devices have distinctive and recognizable traffic patterns with a limited variety of destinations, protocols, and used ports. This can help to conclude that normal behavior profiling and also whitelisting methods can be effective in IoT network security. For the purpose of following whitelisting methods, we produced a Manufacturer usage description for each of our IoT devices. These MUD files can be used to control and limit the communication of Google voice kit and Motorola baby care camera, to what these devices need, and not more than that.

Also, for the purpose of evaluation of our produced MUD files, we build a MUD enabled network. This implementation showed us that the open source MUD manager worked on budgetary access points for the smart home network gateway purpose and the MUD manager can integrate well with other services on the gateway.

5.2 Contribution of the MUD Approach to Cyber Security

In order to understand the benefits of the MUD approach, we should have an in-depth analysis of the other approaches to compare them. The earlier approaches for securing small to medium size networks are the researches that focus on minimizing the available misuse based intrusion detection systems to make them compatible with the new era of the network, which is an IoT network. In general, the goal is to disassemble a working IDS and make it cost-effective and lighter in the aspect of performance to be a reasonable solution to install in the IoT network. The problem with these types of approaches is that the architecture of available IDSs is not designed for this purpose. So by modification and improvements, it cannot function as it does in a regular way. The other critic to these type of approaches is that signature-based IDSs is losing the interest even in the industry due to the diversity of the new type of malware and the problems of its dependencies on the rule updates procedure.

The newer approaches benefit from machine learning to generate a model based on normal behavior profiles to detect anomalies. Although these approaches seem rea-

sonable due to the distinctive and recognizable patterns in the traffic of IoT devices, the platform of execution for these models is a problem. In the newer structure of networks that are following a software-defined networking technology, these models can be executed on the controllers. But in the traditional networks, there is no alternative execution platform for these models. Also, the performance of these models is a matter of concern, and the research to find minimal features to rely on is still an ongoing topic.

For criticizing other methods, there is a need for published proof of works in order to evaluate them. But until the time of writing this thesis, we could not find another type of approach that are tested solutions in a real-world environment.
MUD differently is a standard, developed and contributed by known companies and organizations. It is not only enforcing manufacturers to take security as a concern but also help them to be recognized for producing reliable and secure devices. One of the features that MUD has is that it is easy to implement in comparison with the other approaches. This easiness means that it is compatible with different hardware and operating systems, and the implementation needs the contribution of different chains in the industry. Also, the structure of architecture shows that logically the penalty of performance cost is low in comparison with the other approaches.

The main feature of the MUD enabled networks is that it is promises a high level of security by adding a layer on top of the security stack in the network. Whitelisting methods only enable communication with a trusted destination in a trusted direction by utilizing trusted ports and protocols. This feature is comparable with the normal behavior profiling with machine learning models, but the MUD has its own advantages that we mentioned.

5.3 Contribution of Our Research to Cyber Security

There are few surveys of security on smart homes, and also, the number of surveys on solutions for the detection of DDoS attacks sourced from IoT devices are few, too. Our literature review section can be a good example for researchers to find different categories of solutions on this topic.

As mentioned before, there are only two other datasets for IoT network traffics, and the majority of the researches on IoT networks is based on simulation environments. The dataset we provided can help researchers to access real-world traffic generated from two popular devices. Besides that, the reports that we have generated for these raw files can guide researchers in selecting their preferred features to use on analysis and learning methods. Manual analysis of the Google voice kit and Motorola baby care showed less amount of variation on the protocols and destinations. This can help to conclude about the distinctiveness of patterns belong to the IoT network traffic and help researchers to design algorithms for different purposes as well as the identification mechanisms based on the incoming traffic. Also, during the manual analysis, we found some flows in the Motorola baby care camera that are using HTTP protocols, which is not a safe method to communicate these days. However, we didn't investigate those flows to see if sensitive data is in transmission or not, due to the limitation of the scope of the thesis.

Implementing MUD enabled network is still challenging due to the lack of documentation and experience about this technology. However, we were successful in developing our build that does not follow the previous type of builds and is more compatible with the devices that are used in home networks. The features of this build can be helpful for other researchers to follow for their testing or developing purposes.

5.4 Limitations

During the experiments, we had different limitations. From the perspective of manufacture, we didn't have any other MUD files for Google voice kit or Motorola baby care devices to compare our produced MUD with them. So we must have proof of work for them to evaluate our results. Also, for verifying the signatures of the MUD files, which are not necessary for our case, we couldn't find any free intermediate certificates generated for the purpose of signing files. Besides that, our limitations in hardware limited us from testing different builds of MUD networks. Although this was not in the scope of this thesis, the functionality of MUD files on different builds needs to be investigated. Also, a complex network from the hardware perspective could help us to measure the performance of open source MUD manager and

inter-domain communications could be analyzed too.

5.5 Future Work

There are open questions related to the earlier versions of security cameras that do not support the communication through API. We don't know if MUD can be effective on the security of those cameras that have a diversity of remote requests. The solution can be searched by whitelisting the protocols and ports and generating MUD files based on these features. Also, the performance comparison of different builds of the MUD enabled network is another question that can be investigated with reasonable logistics. Possible bypass mechanisms of MUD enabled networks can be a question to research in the early days of developing MUD. This can help for early upgrades on MUD to promise more security.

Large scale implementation of MUD enabled network is another open to developing topic. This needs an evaluation of different available builds in the real world. It might explain the advantages and disadvantages of different builds. This can lead to changes in the components of the architecture. For example, maybe integration of MUD manager with SDN controller in the builds published by NIST can perform better in large scale architecture than other available ones. Also, the technologies of communication in larger-scale networks can be different, so it is questionable that if MUD can operate in crowded networks like the ones that are using, for example, 5G technology to communicate.

The willingness of smart home users, network admins, and manufacturers also need to be measured to present the MUD enabled network builds, based on their expectation and potential resources. Although there are a couple of proposed builds, it still cannot satisfy the wide variety of customers and there need for innovation in new builds.

REFERENCES

[1] Kevin Ashton, “That ’ Internet of Things ’ Thing,” RFID J., p. 4986, 2010, doi: 10.1038/nature03475.

[2] Cisco, “Cisco Annual Internet Report (2018–2023),” Cisco, pp. 1–41, 2020.

[3] IoT-Analytics, “State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating.” [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/. [Accessed: 06-May-2020].

[4] D. T. Nguyen and T. Kim, “An SDN-Based Connectivity Control System for Wi-Fi Devices,” Wirel. Commun. Mob. Comput., vol. 2018, 2018, doi: 10.1155/2018/9359878.

[5] “OWASP Internet of Things Project - OWASP.” [Online]. Available: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project. [Accessed: 06-May-2020].

[6] "Was Mirai malware behind Dyn DDoS attack?" [Online]. Available: https://www.itpro.co.uk/hacking/27449/was-mirai-malware-behind-dyn-ddos-attack. [Accessed: 06-May-2020].

[7] Feamster, N. (2010, September). Outsourcing home network security. In Proceedings of the 2010 ACM SIGCOMM workshop on Home networks (pp. 37-42).

[8] M. Sabhnani and G. Serpen, “Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set,” Intell. Data Anal., vol. 8, no. 4, pp. 403–415, 2004, doi: 10.3233/ida-2004-8406.

[9] E. Lear and D. Romascanu, “RFC 8520 - Manufacturer Usage Description Specification,” 2019.

[10] “Executive Summary — NIST SP 1800-15 documentation.” [Online]. Available: https://www.nccoe.nist.gov/publication/1800-15/VolA/index.html. [Accessed: 04-Sep-2020].

[11] F. Xia, L. T. Yang, L. Wang, and A. Vinel, “Internet of things,” International Journal of Communication Systems, vol. 25, no. 9. pp. 1101–1102, 2012, doi: 10.1002/dac.2417.

[12] T. The, T. Itu, T. Standardization, and E. Internet-based, “5 The Internet of Things: data for development,” pp. 147–171, 2005.

[13] Minerva, R., Biru, A., Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). IEEE Internet Initiative, 1(1), 1-86.

[14] ETSI, Technical Specification, “Machine-to-Machine communications (M2M); M2M service requirements ” vol. 1, pp. 1–34, 2010. RTS/M2M-00001ed121

[15] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in Internet of Things,” Journal of Network and Computer Applications, vol. 84. Academic Press, pp. 25–37, 15-Apr-2017, doi: 10.1016/j.jnca.2017.02.009.

[16] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network Intrusion Detection for IoT Security Based on Learning Techniques,” IEEE Commun. Surv. Tutorials, vol. 21, no. 3, pp. 2671–2701, Jul. 2019, doi: 10.1109/COMST.2019.2896380.

[17] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” Journal of Cloud Computing, vol. 7, no. 1. Springer Verlag, 01-Dec-2018, doi: 10.1186/s13677-018-0123-6.

[18] K. Patel and H. Upadhyay, “A Survey: Mitigation of DDoS attack on IoT Environment,” Volume 6, Issue I, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 94-96, ISSN:2321-9653, 2018.

[19] Sherly, J., Somasundareswari, D. (2015). Internet of things based smart transportation systems. International Research Journal of Engineering and Technology, 2(7), 1207-1210.

[20] P. Rizwan, K. Suresh, and M. Rajasekhara Babu, “Real-time smart traffic management system for smart cities by using Internet of Things and big data,” in Proceedings of IEEE International Conference on Emerging Technological Trends in

Computing, Communications and Electrical Engineering, ICETT 2016, 2017, doi: 10.1109/ICETT.2016.7873660.

[21] Gross, T. M., Bode, B. W., Einhorn, D., Kayne, D. M., Reed, J. H., White, N. H., Mastrototaro, J. J. (2000). Performance evaluation of the MiniMed® continuous glucose monitoring system during patient home use. Diabetes technology therapeutics, 2(1), 49-56.

[22] C. Tankard, "The security issues of the Internet of Things," Comput. Fraud Secur., vol. 2015, no. 9, pp. 11–14, Sep. 2015, doi: 10.1016/S1361-3723(15)30084-1.

[23] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," Int. J. Distrib. Sens. Networks, vol. 13, no. 12, Dec. 2017, doi: 10.1177/1550147717741463.

[24] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," IEEE Commun. Surv. Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[25] "GitHub - jgamblin/Mirai-Source-Code: Leaked Mirai Source Code for Research/IoC Development Purposes." [Online]. Available: https://github.com/jgamblin/Mirai-Source-Code. [Accessed: 07-May-2020].

[26] Angrishi, K. (2017). Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. arXiv preprint arXiv:1702.03681.

[27] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "Analysis of DDoS-capable IoT malwares," in Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, 2017, pp. 807–816, doi: 10.15439/2017F288.

[28] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation," Security and Communication Networks, vol. 2018. Hindawi Limited, 2018, doi:10.1155/2018/7178164.

[29] M. De Donno, N. Dragoni, A. Giaretta, and M. Mazzara, “AntibIoTic: Protecting IoT Devices Against DDoS Attacks,” Jun. 2017, doi:10.1007/978-3-319-70578-1_7.

[30] P. Manso, J. Moura, and C. Serrão, “SDN-based intrusion detection system for early detection and mitigation of DDoS attacks,” Inf., vol. 10, no. 3, 2019, doi:10.3390/info10030106.

[31] Mirkovic, J., Prier, G., Reiher, P. (2002, November). Attacking DDoS at the source. In 10th IEEE International Conference on Network Protocols, 2002. Proceedings. (pp. 312-321). IEEE.

[32] Gil, T. M., Poletto, M. (2001, August). MULTOPS: A Data-Structure for Bandwidth Attack Detection. In USENIX Security Symposium (pp. 23-38).

[33] Kerravala, Z. (2004). As the value of enterprise networks escalates, so does the need for configuration management. The Yankee Group, 4.

[34] Denning, D. E. (1987). An intrusion-detection model. IEEE Transactions on software engineering, (2), 222-232.

[35] Ghorbani, A. A., Lu, W., Tavallaee, M. (2009). Network intrusion detection and prevention: concepts and techniques (Vol. 47). Springer Science Business Media.

[36] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, “Denial-of-Service detection in 6LoWPAN based Internet of Things,” Int. Conf. Wirel. Mob. Comput. Netw. Commun., pp. 600–607, 2013, doi: 10.1109/WiMOB.2013.6673419.

[37] “Suricata | Open Source IDS / IPS / NSM engine.” [Online]. Available: https://suricata-ids.org/. [Accessed: 12-May-2020].

[38] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, and M.-C. Hsieh, “A Lightweight Intrusion Detection Scheme Based on Energy Consumption Analysis in 6LowPAN,” Springer, Dordrecht, 2014, pp. 1205–1213.

[39] Patel, K., Upadhyay, H. A Rule based Approach to Mitigate DDoS attack in IoT Environment, IJARIIE-ISSN(O)-2395-4396, Vol-4 Issue-3 2018

[40] “Snort - Network Intrusion Detection Prevention System.” [Online]. Available: https://www.snort.org/. [Accessed: 13-May-2020].

[41] M. N. Omar, G. Yusuf, M. Guled, and H. Zakaria, “Home-Based Intrusion Detection System,” vol. 9, no. 2, pp. 107–111, 2004.

[42] M. Gajewski, J. M. Batalla, G. Mastorakis, and C. X. Mavromoustakis, “A distributed IDS architecture model for Smart Home systems,” Cluster Comput., vol. 22, pp. 1739–1749, Jan. 2019, doi: 10.1007/s10586-017-1105-z.

[43] Sheikh, N. U., Rahman, H., Vikram, S., AlQahtani, H. (2018). A Lightweight Signature-Based IDS for IoT Environment. arXiv preprint arXiv:1811.04582.

[44] I. Cvitić, D. Peraković, M. Periša, and M. Botica, “Novel approach for detection of IoT generated DDoS traffic,” Wirel. Networks, 2019, doi: 10.1007/s11276-019-02043-1.

[45] Kumar, A., Lim, T. J. (2019, April). EDIMA: Early detection of IoT malware network activity using machine learning techniques. In 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (pp. 289-294). IEEE.

[46] S. Y. Mehr and B. Ramamurthy, “An SVM based DDoS attack detection method for Ryu SDN controller,” Conex. 2019 Companion - Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol. Part Conex. 2019, pp. 72–73, 2019, doi: 10.1145/3360468.3368183.

[47] A. Hamza, H. H. Gharakheili, and V. Sivaraman, “Combining MUD policies with SDN for IoT intrusion detection,” in IoT S and P 2018 - Proceedings of the 2018 Workshop on IoT Security and Privacy, Part of SIGCOMM 2018, 2018, pp. 1–7, doi: 10.1145/3229565.3229571.

[48] Dao, N. N., Phan, T. V., Kim, J., Bauschert, T., Cho, S. (2017). Securing heterogeneous iot with intelligent ddos attack behavior learning. arXiv preprint arXiv:1711.06041.

[49] R. Doshi, N. Apthorpe, and N. Feamster, “Machine Learning DDoS Detection for Consumer Internet of Things Devices,” Apr. 2018, doi: 10.1109/SPW.2018.00013.

[50] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song, and K. M. Malik, “NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched

IDS for securing IoT against DDoS attacks," J. Supercomput., vol. 74, no. 10, pp. 5156–5170, Oct. 2018, doi: 10.1007/s11227-018-2413-7.

[51] S. Misra, P. Venkata Krishna, H. Agarwal, A. Saxena, and M. S. Obaidat, "A learning automata based solution for preventing distributed denial of service in internet of things," in Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, iThings/CPSCom 2011, 2011, pp. 114–122, doi: 10.1109/iThings/CPSCom.2011.84.

[52] A. M. Alshnta, M. F. Abdollah, and A. Al-Haiqi, "SDN in the home: A survey of home network solutions using Software Defined Networking," Cogent Engineering, vol. 5, no. 1. Cogent OA, pp. 1–40, 01-Jan-2018, doi: 10.1080/23311916.2018.1469949.

[53] Xu, K., Wang, X., Wei, W., Song, H., Mao, B. (2016). Toward software defined smart home. IEEE Communications Magazine, 54(5), 116-122.

[54] M. Ejaz Ahmed and H. Kim, "DDoS attack mitigation in internet of things using software defined networking," in Proceedings - 3rd IEEE International Conference on Big Data Computing Service and Applications, BigDataService 2017, 2017, pp. 271–276, doi: 10.1109/BigDataService.2017.41.

[55] Sivaraman, V., Gharakheili, H. H., Vishwanath, A., Boreli, R., Mehani, O. (2015, October). Network-level security and privacy control for smart-home IoT devices. In 2015 IEEE 11th International conference on wireless and mobile computing, networking and communications (WiMob) (pp. 163-167). IEEE.

[56] Luo, S., Wu, J., Li, J., Guo, L. (2016). A multi-stage attack mitigation mechanism for software-defined home networks. IEEE Transactions on Consumer Electronics, 62(2), 200-207.

[57] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow," in Proceedings - 2016 11th International Conference on Availability, Reliability and Security, ARES 2016, 2016, pp. 147–156, doi: 10.1109/ARES.2016.64.

[58] N. Sambandam, M. Hussein, N. Siddiqi, and C. H. Lung, "Network Security for IoT Using SDN: Timely DDoS Detection," DSC 2018 - 2018

IEEE Conf. Dependable Secur. Comput., no. February, pp. 1–2, 2019, doi: 10.1109/DESEC.2018.8625119.

[59] P. K. Sharma, J. H. Park, Y. S. Jeong, and J. H. Park, "SHSec: SDN based Secure Smart Home Network Architecture for Internet of Things," Mob. Networks Appl., vol. 24, no. 3, pp. 913–924, Jun. 2019, doi: 10.1007/s11036-018-1147-3.

[60] Sivanathan, A., Sherratt, D., Gharakheili, H. H., Sivaraman, V., Vishwanath, A. (2016, November). Low-cost flow-based security solutions for smart-home IoT devices. In 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (pp. 1-6). IEEE.

[61] D. Yin, L. Zhang, and K. Yang, "A DDoS Attack Detection and Mitigation with Software-Defined Internet of Things Framework," IEEE Access, vol. 6, pp. 24694–24705, Apr. 2018, doi: 10.1109/ACCESS.2018.2831284.

[62] S. Soni and B. Bhushan, "A Comprehensive survey on Blockchain: Working, security analysis, privacy threats and potential applications," 2019 2nd Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2019, pp. 922–926, 2019, doi: 10.1109/ICICICT46008.2019.8993210.

[63] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, "Blockchain for smart homes: Review of current trends and research challenges," Comput. Electr. Eng., vol. 83, no. January, p. 106585, 2020, doi: 10.1016/j.compeleceng.2020.106585.

[64] Sagirlar, G., Carminati, B., Ferrari, E. (2018, October). AutoBotCatcher: blockchain-based P2P botnet detection for the internet of things. In 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC) (pp. 1-8). IEEE.

[65] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, "Mitigating IoT device based DDoS attacks using blockchain," in CRYBLOCK 2018 - Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Part of MobiSys 2018, 2018, pp. 71–76, doi: 10.1145/3211933.3211946.

[66] W. M. Kang, S. Y. Moon, and J. H. Park, "An enhanced security framework for home appliances in smart home," Human-centric Comput. Inf. Sci., vol. 7, no. 1, Dec. 2017, doi: 10.1186/s13673-017-0087-4.

[67] K. Nagara, K. Aoki, Y. Matsubara, and H. Takada, "Portable DoS test tool for IoT devices," in IoT S and P 2017 - Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, co-located with CCS 2017, 2017, pp. 57–58, doi: 10.1145/3139937.3139950.

[68] T. Polk, M. Souppaya, B. Haag, and W. C. Barker, "MITIGATING IOT-BASED DISTRIBUTED DENIAL OF SERVICE (DDOS) mitigating-iot-ddos-nccoe@nist.gov," 2017.

[69] Procopiou, A., Komninos, N., Douligeris, C. (2019). ForChaos: real time application DDoS detection using forecasting and chaos theory in smart home IoT network. Wireless Communications and Mobile Computing, 2019.

[70] C. E. Stewart, A. M. Vasu, and E. Keller, "CommunityGuard: A crowdsourced home cyber-security system," in SDN-NFVSec 2017 - Proceedings of the ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, co-located with CODASPY 2017, 2017, pp. 1–6, doi: 10.1145/3040992.3040997.

[71] I. ul Sami, M. Bin Ahmad, M. Asif, and R. Ullah, "DoS/DDoS detection for E-healthcare in Internet of Things," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 1, pp. 297–300, 2018, doi: 10.14569/IJACSA.2018.090140.

[72] L. Zhou, H. Guo, and G. Deng, "A fog computing based approach to DDoS mitigation in IIoT systems," Comput. Secur., vol. 85, pp. 51–62, Aug. 2019, doi: 10.1016/j.cose.2019.04.017.

[73] Bhardwaj, K., Miranda, J. C., Gavrilovska, A. (2018). Towards iot-ddos prevention using edge computing. In USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18).

[74] Dodson, D., Montgomery, D., Polk, W., Ranganathan, M., Souppaya, M., Johnson, S., ... Lear, E. (2020). Securing Small Business and Home Internet of Things

(IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD) (No. NIST Special Publication (SP) 1800-15 (Draft)). National Institute of Standards and Technology.

[75] A. Hamza, D. Ranathunga, H. H. Gharakheili, M. Roughan, and V. Sivaraman, "Clear as MUD: Generating, validating and applying IoT behavioral profiles," in IoT S and P 2018 - Proceedings of the 2018 Workshop on IoT Security and Privacy, Part of SIGCOMM 2018, 2018, pp. 8–14, doi: 10.1145/3229565.3229566.

[76] "Buy Motorola Baby Monitors for Home - MotorolaStore.com." [Online]. Available: https://www.motorolastore.com/baby-monitors/video-monitoring. [Accessed: 04-Sep-2020].

[77] "Voice." [Online]. Available: https://aiyprojects.withgoogle.com/voice/. [Accessed: 04-Sep-2020].

[78] "Archer C7 | AC1750 Wireless Dual Band Gigabit Router | TP-Link." [Online]. Available: https://www.tp-link.com/us/home-networking/wifi-router/archer-c7/. [Accessed: 04-Sep-2020].

[79] "OpenWrt Project: Welcome to the OpenWrt Project." [Online]. Available: https://openwrt.org/. [Accessed: 04-Sep-2020].

[80] "Buy a Raspberry Pi 3 Model B – Raspberry Pi." [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 04-Sep-2020].

[81] "Datasets Overview — Stratosphere IPS." [Online]. Available: https://www.stratosphereips.org/datasets-overview. [Accessed: 04-Sep-2020].

[82] "capinfos - The Wireshark Network Analyzer 3.2.6." [Online]. Available: https://www.wireshark.org/docs/man-pages/capinfos.html. [Accessed: 04-Sep-2020].

[83] "GitHub - gamelinux/passivedns: A network sniffer that logs all DNS server replies for use in a passive DNS setup." [Online]. Available: https://github.com/gamelinux/passivedns. [Accessed: 04-Sep-2020].

[84] "GitHub - netik/tcpdstat: Get protocol statistics from tcpdump pcap files (fork)." [Online]. Available: https://github.com/netik/tcpdstat. [Accessed: 04-Sep-2020].

[85] "Dnstop: Stay on Top of Your DNS Traffic." [Online]. Available: http://dns.measurement-factory.com/tools/dnstop/. [Accessed: 04-Sep-2020].

[86] "The Zeek Network Security Monitor." [Online]. Available: https://zeek.org/. [Accessed: 04-Sep-2020].

[87] "openargus - Home." [Online]. Available: https://openargus.org/. [Accessed: 04-Sep-2020].

[88] "GitHub - ayyoob/mudgee: This tool is used for generating Manufacture Usage Description(MUD) from Device Traffic Trace(PCAP)." [Online]. Available: https://github.com/ayyoob/mudgee. [Accessed: 04-Sep-2020].

[89] "Welcome to MUD Maker." [Online]. Available: https://www.mudmaker.org/. [Accessed: 04-Sep-2020].

[90] Bjorklund, M. (2010). YANG-a data modeling language for the network configuration protocol (NETCONF).

[91] A. Sivanathan et al., "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," IEEE Trans. Mob. Comput., vol. 18, no. 8, pp. 1745–1759, Aug. 2019, doi: 10.1109/TMC.2018.2866249.

[92] "IoT-23 Dataset: A labeled dataset of Malware and Benign IoT Traffic. — Stratosphere IPS." [Online]. Available: https://www.stratosphereips.org/datasets-iot23. [Accessed: 04-Sep-2020].

[93] "Technology — Stratosphere IPS." [Online]. Available: https://www.stratosphereips.org/technology. [Accessed: 04-Sep-2020].

[94] "CapAnalysis | PCAP from another point of view." [Online]. Available: https://www.capanalysis.net/ca/. [Accessed: 04-Sep-2020].

[95] Y. Y. Goland, T. Cai, P. Leach, and Y. Gu, "Internet Engineering Task Force Expires April 2000 Simple Service Discovery Protocol/1.0 Operating without an Arbiter," 1999.

[96] R. Housley, “RFC 5652 - Cryptographic Message Syntax . . . CMS—,” 2009.

[97] “802.1X: Port-Based Network Access Control |.” [Online]. Available: https://1.ieee802.org/security/802-1x/. [Accessed: 04-Sep-2020].

[98] “osMUD.org – The Open Source MUD Manager.” [Online]. Available: https://osmud.org/. [Accessed: 04-Sep-2020].

APPENDIX A

APPENDIX A

1. The MUD File Example by IETF

{ "ietf-mud:mud": { "mud-version": 1, "mud-url": "https://lighting.example.com/lightbulb2000", "last-update": "2019-01-28T11:20:51+01:00", "cache-validity": 48, "is-supported": true, "systeminfo": "The BMS Example Light Bulb", "from-device-policy": { "access-lists": { "access-list": [{ "name": "mud-76100-v6fr" }] } }, "to-device-policy": { "access-lists": { "access-list": [{ "name": "mud-76100-v6to" }] } } }, "ietf-access-control-list:acls": { "acl": [{ "name": "mud-76100-v6to", "type": "ipv6-acl-type", "aces": { "ace": [{ "name": "cl0-todev", "matches": { "ipv6": { "ietf-acldns:src-dnsname": "test.example.com", "protocol": 6 }, "tcp": { "ietf-mud:direction-initiated": "from-device",

"source-port": { "operator": "eq", "port": 443 } } }, "actions": "forwarding": "accept" } }] } }, { "name": "mud-76100-v6fr", "type": "ipv6-acl-type", "aces": { "ace": [{ "name": "cl0-frdev", "matches": { "ipv6": { "ietf-acldns:dst-dnsname": "test.example.com", "protocol": 6 }, "tcp": { "ietf-mud:direction-initiated": "from-device", "destination-port": { "operator": "eq", "port": 443 } } }, "actions": { "forwarding": "accept" } }] } }] } }

2. Google Voice Kit - MUD file

{ "ietf-mud:mud" : { "mud-version" : 1, "mud-url" : "http://www.milad.me", "last-update" : "2020-07-31T22:08:06.391+03:00", "cache-validity" : 100, "is-supported" : true, "systeminfo" : "Google Voicekit", "from-device-policy" : "access-lists" : {

"access-list" : [{ "name" : "from-ipv4-googlevoicekit" }, { "name" : "from-ipv6-googlevoicekit" }] } }, "to-device-policy" : { "access-lists" : { "access-list" : [{ "name" : "to-ipv4-googlevoicekit" }] } } }, "ietf-access-control-list:access-lists" : { "acl" : [{ "name" : "from-ipv4-googlevoicekit", "type" : "ipv4-acl-type", "aces" : { "ace" : [{ "name" : "from-ipv4-googlevoicekit-0", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:dst-dnsname" : "clients4.google.com" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 443 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-googlevoicekit-1", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:dst-dnsname" : "safebrowsing.googleapis.com" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 443 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-googlevoicekit-2", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:dst-dnsname" : "embeddedassistant.googleapis.com" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 443 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-googlevoicekit-3", "matches" : { "ipv4" : { "protocol" : 17, "ietf-acldns:dst-dnsname" : "embeddedassistant.googleapis.com" }, "udp" : { "destination-port" : { "operator" : "eq", "port" : 67 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-googlevoicekit-4", "matches" : { "ietf-mud:mud" : { "local-networks" : [null] }, "ipv4" : { "protocol" : 17, "destination-ipv4-network" : "239.255.255.250/32" }, "udp" : { "destination-port" : { "operator" : "eq", "port" : 1900 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-googlevoicekit-5", "matches" : { "ipv4" : { "protocol" : 17 }, "udp" : { "source-port" : { "operator" : "eq", "port" : 67 } } }, "actions" : { "forwarding" : "accept" } }] } }, { "name" : "to-ipv4-googlevoicekit", "type" : "ipv4-acl-type", "aces" : { "ace" : [{ "name" : "to-ipv4-googlevoicekit-0", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:src-dnsname" : "safebrowsing.googleapis.com" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 443 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-googlevoicekit-1", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:src-dnsname" : "embeddedassistant.googleapis.com" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 443 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-googlevoicekit-2", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:src-dnsname" : "clients4.google.com" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 443 }

} }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-googlevoicekit-3", "matches" : { "ipv4" : { "protocol" : 17 }, "udp" : { "destination-port" : { "operator" : "eq", "port" : 67 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-googlevoicekit-4", "matches" : { "ipv4" : { "protocol" : 17, "ietf-acldns:src-dnsname" : "embeddedassistant.googleapis.com" }, "udp" : { "source-port" : { "operator" : "eq", "port" : 67 } } }, "actions" : { "forwarding" : "accept" } }] } }, { "name" : "from-ipv6-googlevoicekit", "type" : "ipv6-acl-type", "aces" : { "ace" : [{ "name" : "from-ipv6-googlevoicekit-0", "matches" : { "ietf-mud:mud" : { "local-networks" : [null] }, "ipv6" : { "protocol" : 58, "destination-ipv6-network" : "ff00::/8" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv6-googlevoicekit-1", "matches" : { "ietf-mud:mud" : { "local-networks" : [null] }, "ipv6" : { "protocol" : 0, "destination-ipv6-network" : "ff00::/8" } }, "actions" : { "forwarding" : "accept" } }] } }] }

3. Motorola Baby Care Camera - MUD File

{ "ietf-mud:mud" : { "mud-version" : 1, "mud-url" : "https://milad.me", "last-update" : "2020-07-31T02:24:28.068+03:00", "cache-validity" : 100, "is-supported" : true, "systeminfo" : "Motorola baby care camera", "from-device-policy" : { "access-lists" : { "access-list" : [{ "name" : "from-ipv4-motorolababycarecamera" }] } }, "to-device-policy" : { "access-lists" : { "access-list" : [{ "name" : "to-ipv4-motorolababycarecamera" }] } } }, "ietf-access-control-list:access-lists" : { "acl" : [{ "name" : "from-ipv4-motorolababycarecamera", "type" : "ipv4-acl-type", "aces" : { "ace" : [{ "name" : "from-ipv4-motorolababycarecamera-0", "matches" : { "ietf-mud:mud" : { "controller" : "urn:ietf:params:mud:dns" }, "ipv4" : { "protocol" : 17 }, "udp" : { "destination-port" : { "operator" : "eq", "port" : 53 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-motorolababycarecamera-1", "matches" : { "ipv4" : { "protocol" : 17, "ietf-acldns:dst-dnsname" : "ntp.hubble.in" }, "udp" : { "destination-port" : { "operator" : "eq", "port" : 123 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-motorolababycarecamera-2", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:dst-dnsname" : "api.hubble.in" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 443 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-

motorolababycarecamera-3", "matches" : { "ipv4" : { "protocol" : 6, "destination-ipv4-network" : "142.93.251.217/32" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 1935 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-motorolababycarecamera-4", "matches" : { "ipv4" : { "protocol" : 17 }, "udp" : { } }, "actions" : { "forwarding" : "accept" } }, { "name" : "from-ipv4-motorolababycarecamera-5", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:dst-dnsname" : "upload1.hubble.in" }, "tcp" : { "destination-port" : { "operator" : "eq", "port" : 80 }, "ietf-mud:direction-initiated" : "from-device" } }, "actions" : { "forwarding" : "accept" } }] } }, { "name" : "to-ipv4-motorolababycarecamera", "type" : "ipv4-acl-type", "aces" : { "ace" : [{ "name" : "to-ipv4-motorolababycarecamera-0", "matches" : { "ipv4" : { "protocol" : 6, "source-ipv4-network" : "142.93.251.217/32" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 1935 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-1", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:src-dnsname" : "upload1.hubble.in" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 80 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-2", "matches" : { "ietf-mud:mud" : { "controller" : "urn:ietf:params:mud:dns" }, "ipv4" : { "protocol" : 17 }, "udp" : { "source-port" : { "operator" : "eq", "port" : 53 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-3", "matches" : { "ietf-mud:mud" : { "controller" : "urn:ietf:params:mud:gateway" }, "ipv4" : { "protocol" : 17 }, "udp" : { "source-port" : { "operator" : "eq", "port" : 67 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-4", "matches" : { "ipv4" : { "protocol" : 17 }, "udp" : { } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-5", "matches" : { "ipv4" : { "protocol" : 6, "ietf-acldns:src-dnsname" : "api.hubble.in" }, "tcp" : { "source-port" : { "operator" : "eq", "port" : 443 } } }, "actions" : { "forwarding" : "accept" } }, { "name" : "to-ipv4-motorolababycarecamera-6", "matches" : { "ipv4" : { "protocol" : 17, "ietf-acldns:src-dnsname" : "ntp.hubble.in" }, "udp" : { "source-port" : { "operator" : "eq", "port" : 123 } } }, "actions" : { "forwarding" : "accept" } }] } }] } }