USING MANUFACTURER USAGE DESCRIPTIONS FOR IOT NETWORK SECURITY: AN EXPERIMENTAL INVESTIGATION OF SMART HOME NETWORK DEVICES

Milad Kazemi Darazam[1], Pelin Angin[2], Cengiz Acartürk[3]
[1]METU Informatics Institute, Turkey, milad.darazam@metu.edu.tr, [2]METU Computer Engineering, Turkey, pangin@ceng.metu.edu.tr, [3]METU Informatics Institute, Turkey, acarturk@acm.org

NOTE: Corresponding author: Milad Kazemi Darazam, milad.darazam@metu.edu.tr

Abstract – *The Internet of Things (IoT) has shown significant growth in the past decades. Recently, IoT networks have been subject to cybersecurity threats on multiple fronts. A lack of improvement in IoT infrastructures' cybersecurity may result in challenges with a broad impact, such as DDoS attacks that target global DNS services. This paper reviews existing solutions to mitigate attacks on and from IoT networks. As a specific mitigation approach, we propose the use of a standardized whitelisting method, namely Manufacturer Usage Description (MUD), which provides enhanced explainability over machine learning-based approaches and is complementary to them. For evaluating the use of MUD in IoT networks, we report a case study, which we conducted through traffic analysis of two IoT devices by detecting recognizable and distinctive traffic patterns, which demonstrate the feasibility of MUD-based intrusion prevention.*

Keywords – IoT networks, manufacturer usage description, smart home security, traffic patterns

1. INTRODUCTION

Internet of Things (IoT) is a term coined by Kevin Ashton in 1999, to refer to the connection between the Internet and the physical world through ubiquitous sensors [1]. Recently, the term has mainly been used for embedded devices in various computation-enabled environments that sense, communicate, and provide services to human users or other devices. The IoT market size has been growing fast over the past decade due to the increasing use of IoT devices in daily settings. For instance, Cisco Systems Inc. predicts that 48 percent of Machine-to-Machine (M2M) traffic will be generated by smart home networks by the end of 2023 [2]. According to IoT Analytics, 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].

A significant challenge in using IoT devices in daily settings is the presence of cybersecurity threats. Due to the broadening use in smart home technologies, IoT devices have been subject to advanced security threats. Numerous IoT supply chain stakeholders, including IoT hardware manufacturers and end-user consumers, have faced threats at various fronts. These cover a broad spectrum of issues, such as hardware-embedded malware, operating system-level vulnerabilities, and application-level exposures. Furthermore, the limited grasp of the technology by the end users, in particular their lack of knowledge about the configuration of the devices for better cybersecurity, contributes to the adverse effects of attacks on IoT devices [4]. In smart home networks, device misconfiguration has been one of the top OWASP vulnerability categories [5]. Consequently, IoT networks constitute a primary target domain of attack for adversaries. As a result, the growth of the smart home market has been accompanied by growing security threats.

A significant characteristic of IoT devices is that their vulnerabilities threaten the security of the local networks they reside in, allowing the execution of attacks such as Distributed Denial of Service (DDoS) attacks. The adversaries aim at gaining unauthorized access to IoT devices primarily for enslaving them as part of botnets. An IoT end user in most cases is unaware of being part of a botnet. Nevertheless, they unwillingly take part in an attack targeting a third-party service provider, which sometimes results in loss of reputation, or financial losses [6] for that provider. Therefore, the intrusion detection and prevention solutions for IoT devices need to aim at ensuring security not only for IoT end users but also third-party service providers. Moreover, today's cybersecurity solutions have to address multiple facets of IoT security, such as the richness in the variety of IoT infrastructure components in smart homes.

IoT devices are produced for specific tasks. Therefore, they are expected to exhibit predictable network traffic patterns. Identifying traffic patterns in a network facilitates anomaly detection by comparison with the standard, expected behavior in the network. In this paper, we aim to answer the following questions to guide the design of robust Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) for IoT networks: (1) Is it possible to extract distinctive and recognizable traffic patterns from IoT devices, which allow the creation of rules to be used in the network gateways? (2) Can we achieve this type of IoT network traffic profiling with lightweight, explainable and easy-to-integrate methods applicable to a variety of IoT device types and brands?

If we are able to extract traffic patterns for specific IoT devices in a network, we can justify using whitelisting methods for securing communication with those devices. The ability to accurately extract the expected traffic patterns for a device will enable achieving low false-positive rates by IDSs. This is a significant step forward in developing secure IoT networks, given that a high rate of false positives is a significant issue in anomaly-based IDSs[8]. A Manufacturer Usage Description (MUD), which is a standard proposed by the Internet Engineering Task Force (IETF) [9] and technically reviewed, documented, and implemented by NIST [10], is used by IoT device manufacturers to advertise the intended communication patterns of their devices. MUD is of great help in defining the traffic patterns for IoT devices, hence it has significant potential for use in whitelisting approaches.

In this paper, we investigate network traffic profiling and whitelisting methods and propose a MUD-based approach for IoT network security. The contributions of this paper to the literature are as follows:

1. We propose a general architecture for a smart home network, which utilizes open-source approaches that employ MUD-based traffic profiling and whitelisting. Our approach is based on the implementation of MUD in a laboratory setting, and its evaluation for security. For this, we use the Yang model by IETF [9]. We build a MUD-enabled network to test our generated MUD files.

2. We demonstrate the efficiency of the methodology by analyzing IoT traffic on the network. For this, we captured traffic flow from real IoT devices while they generated traffic in a local area network. In order to reach the maximum coverage of variation, we recorded multiple capture sets. Following initial data analysis, we identified the capture sets that included the features needed for further analysis.

3. This study provides a Verification of Concept (VOC), aka Proof of Concept (PoC) approach to providing cybersecurity for smart home IoT networks. The approach presented in the study exhibits an open-source motivated approach, which is generalizable to a wide spectrum of IoT devices. We present two implementations to demonstrate supporting evidence that the concept is an applicable one (i.e., MUD as an implementable approach), therefore it is able to go beyond architectural level descriptions and run in a demo environment.

4. The demonstrations with MUD implementation on two devices show that the open source version of MUD works for small networks. In contrast to black-box machine learning models, such as artificial neural network approaches, the approach is also ”explainable” in that it is possible to track the data processes.

The remainder of this paper is organized as follows: Section 2 provides background information on IoT, smart environments, IoT security challenges and MUD. Section 3 provides an overview of related work in intrusion detection and prevention approaches. Section 4 describes our methodology for utilizing MUD files for intrusion prevention in a smart home network, as well as our network setup and data collection details. Section 5 provides an evaluation of the MUD-based approach for two real home IoT devices. Section 6 discusses the important findings of the study and presents conclusions. Section 7 lists future study directions.

2. BACKGROUND

2.1 IoT definition and domains of application

The term IoT was first coined by Kevin Ashton at the MIT Auto-ID Center[1], to refer to the connection between the Internet and the physical world through ubiquitous sensors. Since then, several definitions have been proposed. For example, International Telecommunication Union (ITU) used it for describing ”a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on the current and evolving inter-operable information and communication technologies or a network which is available anywhere, anytime, by anything and anyone” [12]. Similarly, according to IEEE, IoT is ”a network of items, each enabled with sensors connected to the Internet” [13]. ETSI, officially recognized by the EU as a European standards organization, uses the term ”machine-to-machine (M2M)” instead of IoT. It describes M2M as ”communication between two or more entities that do not necessarily need any human intervention” [14]. NIST also uses ”Cyber-Physical System” to describe IoT as a new way for connecting objects in daily use, enhancing efficiency and sustainability, and improving the quality of life [80].

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 will reach nearly 16 billion by the end of 2023 [3]. Cisco reports that by the end of 2023, 48% of the M2M traffic will be generated by smart home networks [2]. These figures suggest that IoT devices will have a significant role in shaping how we live, communicate, work, and learn shortly. The potential impact of IoT on daily operations has already been addressed by numerous studies [15], [16],[17] and [18]). Given its major role in daily life, the security of IoT devices and their connections will also be of utmost importance. Below, we describe IoT environments in terms of their common context of use.

2.2 Smart environments

The term ”smart” has become a prefix to numerous objects and services used in daily settings, at homes, offices, and living environments, as well as hospitals and cities. The concept of smartness is related to the ability of an object to connect and interact with other objects and humans. Many household devices are manufactured with integrated processors, sensors, and software that provide the necessary infrastructure. Our environments have changed to become smarter in the past decade, leading to revolutionary lifestyle changes. The primary environment that has changed rapidly with IoT is our homes. Temperature sensors and remote door locking mechanisms exemplify the early use of IoT in home environments. Domestic appliances, such as refrigerators, have been the pioneers of smart home devices. More recently, kitchen utensils, such as coffee machines have also become smart. These changes in end-user IoT technologies require appropriate infrastructures to provide interaction with humans and provide connectivity and cybersecurity. Nevertheless, the cybersecurity of IT systems usually lags behind their functionality. Given that the end users are not expected to be domain experts, misconfiguration and the lack of hardened security are frequently observed. Those issues lead to an increased interest of adversaries, thus increasing the urgency of the need for more secure IoT networks at smart homes.

Cities are also getting smart. The term ”smart city” is defined in various forms sharing some commonalities in the description, besides the discrepancies. A common aspect is the use of IoT as an embedded, integrated component of an urban-wide computer network infrastructure. Early uses of IoT technologies included transportation systems [19] and traffic management [20]. Since the scope of the IoT technology in those systems is usually the whole city, a cybersecurity threat may have an enormous impact. Healthcare is another domain that has been subject to IoT influence for the past decade. IoT devices are promising candidates to facilitate health tracking, thus supporting decision-making in the medical context [21].

In summary, IoT has been developed as an integrated part of ”smart” things, such as smart objects, smart environments, and smart systems in general, having a significant impact on our daily lives, as well as promising novel domains for living, working and learning in the future. A significant challenge, as in the development of all new technologies, is to ensure the cybersecurity of IoT networks. IEEE provides a simple approach for describing the layers of the functional components of IoT devices [13]. In this design, the bottom layer is the perception layer (or the sensing layer), also referred to as the hardware layer. This layer, consisting of elements such as sensors, is responsible for physical interaction with the environment. The upper layer is the network layer. This layer is responsible for networking and data communications, such as appropriate data transmission through various protocols and technologies, such as 5G, 4G, WiFi, ZigBee, and the infrared. Finally, the topmost layer is the application layer due to its primary responsibility of establishing connections between services. In the following section, we present major IoT security challenges that are at these different layers of functionality in IoT.

2.3 IoT security challenges

Cybersecurity is usually a follow-up concern for a novel technology rather than an integrated part of it. This situation also applies to today’s IoT devices in the fast-growing IoT market, which mostly skip integrated security in connected devices. The usability goals that provide ease of configuration for non-technical end users also lead to many IoT devices being used with default configuration credentials, also running with unpatched and outdated software [22].

A review of the literature reveals that IoT security challenges may be categorized into four major groups [17]. The first group includes authentication-related problems. The second is related to the security of data transmission in the network. Integrity and availability are central security issues, particularly when the data is modified in transmission or interrupted due to attacks like DDoS. The third group includes privacy challenges that have attracted significant attention recently. IoT devices may process confidential data, and the use of data may lead to privacy violations. Therefore, security and usage policies are needed to regulate the processing and storage of data without any violation of individual users’ privacy. Besides those major categories, relevant security challenges exist which are specific to IoT devices, such as a limited amount of resources and processing capacity, especially for data encryption, decryption and intrusion detection. Another important challenge is the lack of common security standards [17].

In some of the attacks on IoT, the IoT devices themselves are not the direct target of the attack, but are used as instruments in the coordination of a large-scale attack on another target. DDoS is one of the well-known examples of such attacks. A DDoS is a joint attack, where attackers aim at interrupting the expected behavior of a system, usually a public service. Attackers expand the attack surface through distributed attacker devices, which unknowingly take part in the DDoS attack [23]. DDoS is a significant threat for IoT environments, since IoT networks have the potential to provide a large attack surface. A specific example of a DDoS capable malware is Mirai, which is an IoT malware with published source code [24]. Mirai uses a short dictionary of default usernames and passwords for brute-forcing the authentication of IoT devices that use default credentials. Mirai was used to attack Krebs on the Security blog, OVH data centers, and also Dyn name servers [25]. The latter caused interruption on accessing popular websites like Twitter. Although Mirai was not the only DDoS-capable malware, the Mirai attacks were a crucial point in the history of IoT security when researchers began to propose solutions to mitigate them [26]. Mirai revealed the potential problems of insecure IoT devices, leading to numerous proposals as mitigation, such as a white worm named AntibIoTic [27].

2.4 Manufacturer usage description

In this work, we use a MUD-based approach for intrusion detection and prevention in smart home networks. MUD is a standard proposed by IETF in RFC 8520, which allows IoT device manufacturers to advertise expected communication patterns and device specifications. MUD ensures that an IoT device will access the required resources after joining a network, based on the description that its manufacturer specified, and nothing more. IoT devices are expected to function appropriately with this insurance, and their attack surface is significantly decreased when they do. MUD enabled IoT device manufacturers to become more involved in the security of their products at the onset of production rather than implementing cybersecurity as a follow-up task.

A MUD file is structured by the YANG model (RFC 6020) [72]. A valid MUD file contains two root objects: A “MUD” container and an “ACL” container. The arguments can be matched to create policies in the container object. Further information, such as the software version or the last update, are included in the MUD container. The YANG model’s augmentation and syntax are available in RFC 8520. The MUD file is a sensitive file that defines the network security policy at its core. Therefore, it needs to be protected from adversaries. For this, Cryptographic Message Syntax (CMS) can be used (We point the reader to RFC 5652 [77] for more information on this). The MUD manager must also verify MUD files. 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 primarily should be the manufacturer of the IoT device.

A simple line of a MUD file usually consists of destination and source IP addresses, destination and source ports, protocols, and action containers. Therefore, a translation of the Yang model to access control list entries is needed by the MUD manager to add these entries as rules to firewalls. For example, based on the code snippet below, the MUD manager will allow outgoing traffic with UDP protocol on port 1900 for the Google Voice Kit:

”udp” : { ”destination-port” : { ”operator” : ”eq”, ”port” : 1900 } } }, ”actions” : { ”forwarding” : ”accept” } }

Besides these containers, the MUD file includes other containers like version, URL, to-device-policy, from-device-policy, last-update, cache-validity, is-supported, system-info, and other containers. Fig. 1 shows part of a sample MUD file. The URL container is useful for when the file and the signature are manually uploaded, and it points to the address of the MUD file. The controller container specifies a value that a controller will register with the MUD manager. The to-device-policy and from-device-policy containers are used to explicitly define the direction of traffic flow in the MUD files. By allowing the permitted traffic from and to the particular device, the potentially malicious traffic gets dropped, hence the attack surface is reduced. In order to create a MUD file, the manufacturer should list all possible connections from and to the device, including the destination addresses, port and protocols, and the direction of the connections. In the absence of a MUD file generated and verified by manufacturers, a network administrator can try to store all valid traffic and extract the network patterns and create a MUD file as we did in this work.

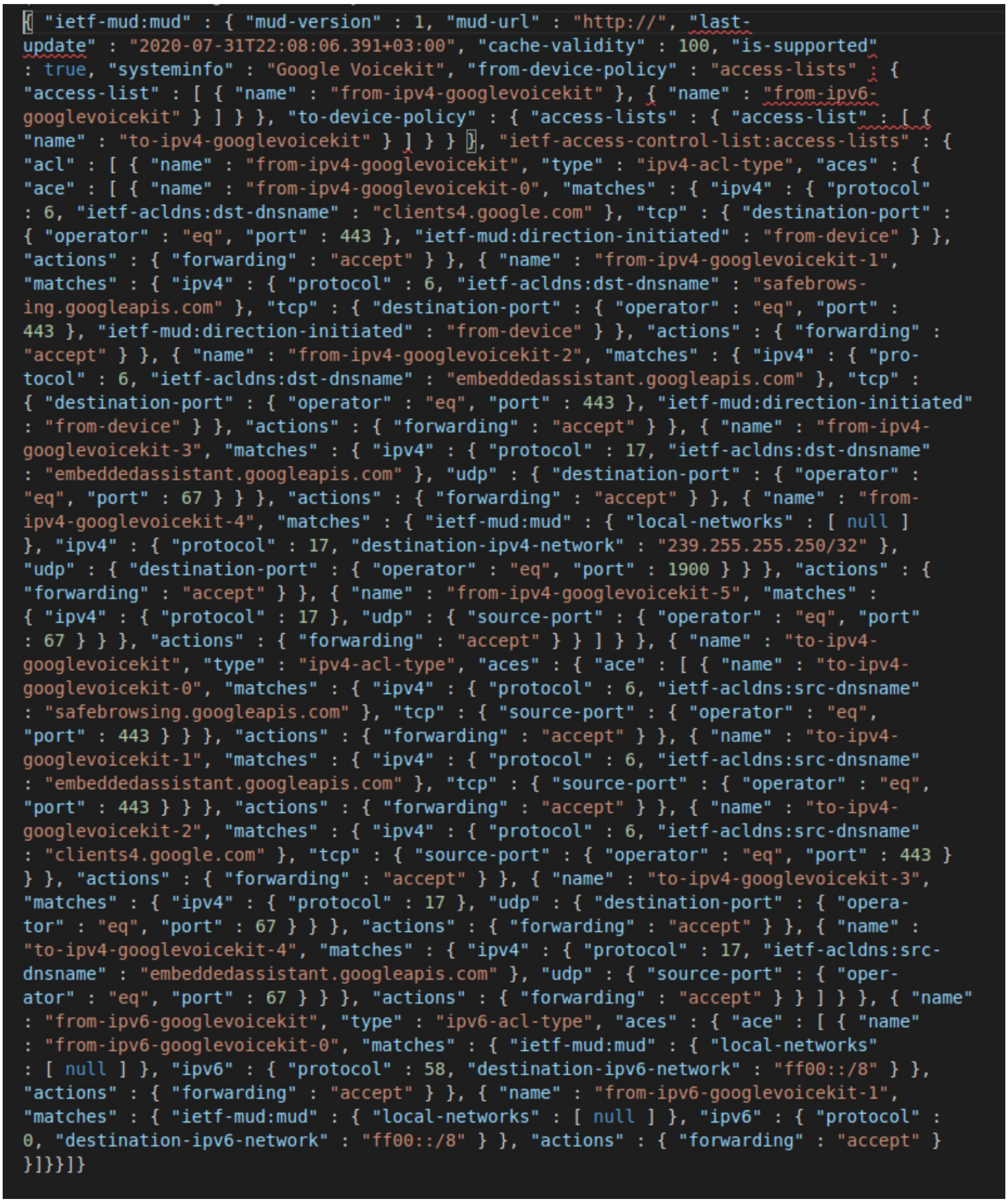
3. RELATED WORK

In this section, we provide an overview of related work in the fields of Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) for IoT.

Computer systems under cyberattacks may exhibit different characteristics than their regular activities. Those differences provide the basis of detection by an IDS [28]. There exist three main approaches to IDS [29]. One is ”misuse detection”. The idea here is to identify the patterns from intrusion attempts, specify them as signatures, and compare them with data gathered from the system. An IDS sends an alarm in case of a match between an attack signature and the data gathered from system monitors. The second approach to intrusion detection is ”anomaly detection”. The idea in anomaly detection is to create a model that builds a profile of the normal behavior of the host/network and rule significant deviations from the normal behavior as anomalies. The third approach is called ”specification-based detection.” As its name suggests, it is based on a collection of predefined specifications for intrusions and regular activities. In case of a match between an activity and the intrusion specifications, the system reports the activity as malicious.

An efficient IDS/IPS solution for IoT networks should be able to operate in real time [17]. The protection should be interoperable with diverse protocols for compatibility among different vendors. The topology and architecture of the network components also influence the design of intrusion detection in an IoT network.

With the rising number of threats against IoT systems, many legacy IDS/IPS tools were adapted for operation in an IoT setting. For example, Suricata, a widespread intrusion detection and prevention system, was adapted to work in a 6LoWPAN network to detect DDoS attacks and reduce false alarm rates in [30]. A lightweight model was proposed to detect DDoS attacks based on the energy consumption of the devices in a 6LoWPAN network in [31]. In [32] the authors used Snort [33], a popular IDS, to monitor, detect, and prevent the incoming malicious traffic for an IoT network. Their approach achieved reasonable detection rates. Another example of utilizing Snort can be found at [34]. A two-layer architecture was proposed for intrusion detection, where one of the agents resides in the home network, designed to collect data and send them to the others, which reside in the ISP to perform data analysis [35]. After analysis, the expert system generates rules and policies for the home network and enforces these policies at the home network gateway agent.



```
{ "ietf-mud:mud" : { "mud-version" : 1, "mud-url" : "http://", "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" : "safebrows-
ing.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" : { "pro-
tocol" : 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" : { "opera-
tor" : "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" : { "oper-
ator" : "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" }
}]}}]}
```

Fig. 1 – MUD file structure in JSON format for Google Voice Kit

This solution utilizes a combination of machine learning methods and signature-based methods. These kinds of solutions are also called hybrid systems, as exemplified by [36], which aim to achieve fast pattern matching to scale up the performance.

Machine Learning (ML) methods have become a popular choice for building effective IDS in recent years. They have been employed to build models for representing the expected behavior of a system in the case of anomaly detection. Despite their effectiveness in detecting zero-day attacks, anomaly detection models may suffer from high false-positive rates. When detection of specific attack types is also needed, ML models for multi-class classification are employed. One issue with ML-based IDS is the need for up-to-date, labeled datasets that include the normal behavior of common IoT devices in the case of anomaly detection and datasets with sample attack instances in the case of misuse detection.

IoT devices may exhibit specific characteristics in network traffic, thus providing the opportunity for classification based on the traffic behavior, especially in smart environments [37]. Classical ML algorithms such as Naive Bayes, K-Nearest Neighbor (KNN), decision trees, and random forest may be employed to detect malware in the early stages when the attacker starts to scan a home network [38, 41, 42]. Combining ML models with Software-Defined Networks (SDN) may help reduce the impact of a cyberattack, as in the case of [39].

SDN architectures have a set of advantages, such as benefiting from virtualization, central management, and offering APIs for external services [43, 47]. A primary advantage of employing SDN is that it makes it possible to provide security as a service [44]. An orchestrator in the cloud provides a Restful API to the customer network and the ISP. Security providers may implement security mechanisms, which are enabled or disabled through web interfaces by customers and the ISP. A smart home SDN network was designed by [48], where a controller is integrated with an IDS to detect DDoS attacks near the source. The soft switch of the network mirrors all the traffic to the IDS. In case of malicious behavior in the traffic, the IDS sends an alert to the controller, which generates a flow for blocking further traffic from that device. The solution seems to guarantee reducing delay in the network and assure a high Quality of Service (QoS). Similarly, [46] shows that the cost of flow-based traffic analysis is much lower than packet analysis. Another near-source detection approach employs an entropy formula to detect cyberattacks, based on the observation that a network under attack would result in a decrease in the entropy of the network packets due to reduced variation in the payload [45]. The main concern in these solutions is the central unit of decision, which contains user activity details. The centralization attracts the attention of adversaries since a security breach may threaten the privacy and security of the whole home network and its users at a critical level. A large number of IDS alerts may also interrupt the controller's service, making it a single point of failure.

Besides the above-mentioned solutions, numerous IDS/IPS solutions exist, such as blockchain-based solutions and benign-worm solutions, to detect and mitigate attacks in IoT. Blockchain is a distributed, shared ledger, characterized by consensus in a peer-to-peer network. It consists of sequences of blocks, where each block is linked to the previous block in the shared ledger and the verification of a block depends on the agreement in the community of the network nodes [49]. Blockchain solutions for IoT security mainly focus on the integrity of network intelligence data stored in the ledger [50]. For instance, a Byzantine Fault Tolerant blockchain was utilized to find the mutual contacts in communication and mark them in the shared blacklist in [51]. Another approach was proposed in [52] to defend against DDoS attacks. The framework's security relies on generating addresses for nodes that are IoT devices and custom-coded smart contracts.

A related community-based approach, which did not utilize blockchain is the Community Guard [53]. This system has two major components, the guardian node, which serves between the home router and the modem. The second is a community outpost that runs in the cloud server and is a central managing unit. The nodes pull part of rules and blacklisted IP(s) from the IDS, which runs on an outpost and pushes the suspicious traffic to the outpost. By the collaboration of the nodes in a home network, the outpost is kept updated against novel attacks and can generate new rules for attack detection. ShadowNet [54] is another solution proposed 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 will handle the requests correlated with web services. With this part of the solution, blocking specific blacklisted IP(s) is possible without any effort at the gateways.

An example of benign-worm solutions is AntibIoTic, a solution for securing IoT devices automatically using a white worm [26, 27]. The worm exploits the spreading capability of IoT malware, such as Mirai, to compete for infecting the IoT devices. However, instead of harming the network by DDoS attacks, AntibIoTic aims to patch the security vulnerabilities and inform system administrators for further security operations. Another approach to detecting application-layer DDoS attacks is the utilization of the Chaos theory [55]. The approach is based on the statistical analysis of features, including the request and packet numbers, the data rate, the average packet size, the combination of request and response time, and parallel requests. A key advantage of this approach is its realistic approximation to real-world settings.

A major limitation of the available IoT security solutions, especially those at the network layer, is the presence of numerous types of devices from numerous manufacturers. This is a significant challenge, since the proposed solution must be compatible with all the available devices under specific environments and conditions. Moreover, the limitations in the available resources, such as power, memory, and process units, need to be addressed by the lightweight solutions for detection mechanisms. Also, usually, the gateways of smart homes have limited resources as well, and this is one of the most important reasons why security of smart homes have traditionally been outsourced to more powerful processing centers like ISPs [7].

4. METHODOLOGY

In this section, we present the methodology of our study. We state our threat model, discuss our data capturing methodology, data analysis, and two case studies that demonstrate real-world applicability of the MUD-based approach.

To investigate the applicability of this approach, we created a dataset by collecting data in an IoT network. We then analyzed the data to identify the features that allow us to recognize patterns in the traffic flow. For evaluation, we generated MUD files and investigated the feasibility of using MUD as a solution for the attack mitigation problem. In particular, we investigated the effectiveness of the security mechanism that it provides [56]. Fig. 2 shows the procedure for creating the MUD files.

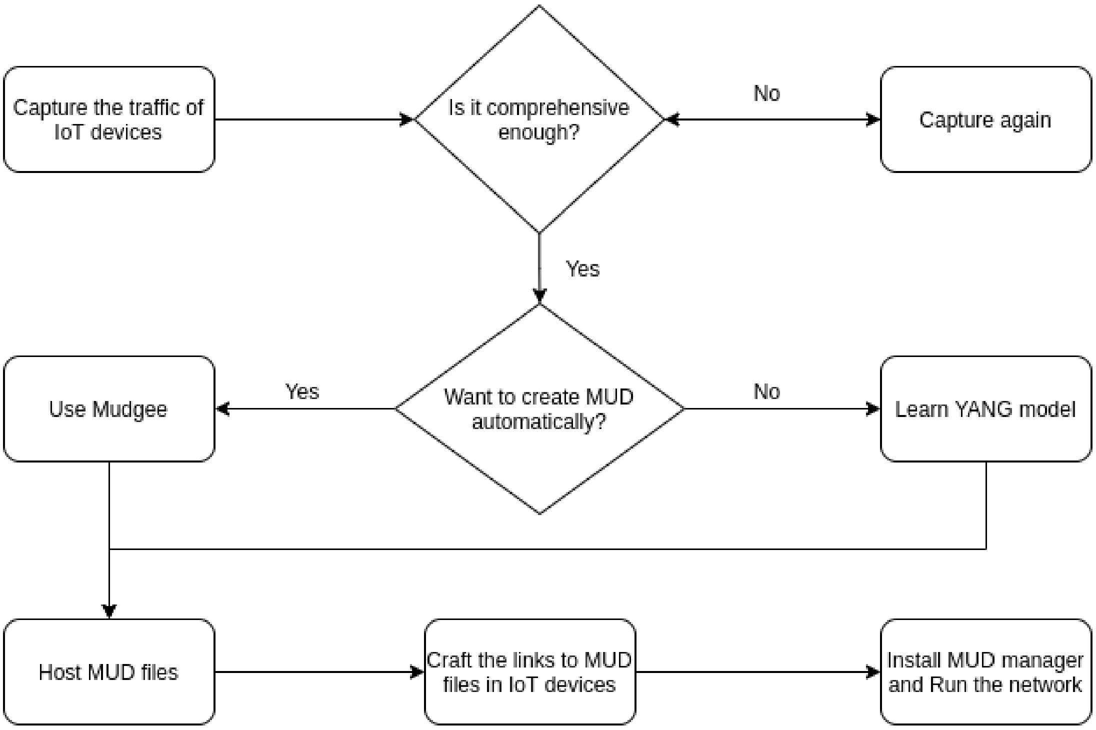

Capture the traffic of IoT devices
Is it comprehensive enough?
No
Capture again
Yes
Use Mudgee
Yes
Want to create MUD automatically?
No
Learn YANG model
Host MUD files
Craft the links to MUD files in IoT devices
Install MUD manager and Run the network

Fig. 2 – Procedure for creating a MUD file

4.1 Threat model

In this work, our focus is a smart home network, where multiple IoT devices such as IP cameras, voice kits, smart TVs, refrigerators, thermostat controllers, lights, etc. are connected to the home Wi-Fi router, which connects them to the modem connecting to the ISP.

IoT devices are usually added to networks together with other computing devices, such as end user computers and the current common practice allows remote access to IoT devices. The spread of IoT malware, such as the Mirai has shown that most IoT devices are accessible remotely. For high security, communication among IoT devices should be limited to registered and identified devices. Remotely accessible IoT devices may grant unauthorized access to attackers, which may then be followed by dictionary attacks against the authentication mechanisms of IoT devices that serve publicly. In case of a successful attack, the IoT devices may function as zombie devices, which take part in DDoS attacks. In this study, we focus on smart home IoT devices that fall victim to these attacks through communication with resources not needed for their normal operation.

4.2 Equipment and network topology

A review of publicly available IoT network traffic datasets reveals a frequent use of baby monitoring cameras [57]. These are sometimes partially compatible with MUD concepts, especially when using the Session Traversal Utilities for NAT (STUN) protocol. STUN is a protocol that collaborates with other protocols and tools to discover the presence of a Network Address Translation (NAT) service. On the other hand, the incompatibility is due to the nature of the monitoring systems, where system administrators may want to monitor different places and IP addresses. A similar situation applies to general-purpose smart home voice assistants since they also have a wide range of diversity in their network traffic patterns. In this study we investigate baby monitoring cameras and smart home voice assistants to approximate the real-world diversity in their traffic patterns. For this, we selected the Motorola Baby Care camera [58], which has a wider variety of functionalities than other commercially available, common baby monitoring cameras for end users. As for the smart home voice assistant, we selected Google Voice Kit [59] as a ready-to-develop home assistant. IoT networks usually require a gateway to connect to the Internet, either directly or through a subnet of a larger network. We utilized a TP-Link Archer C7 access point as the gateway [60], which is an affordable access point compatible with OpenWRT as the firmware [61]. Fig. 3 shows the architecture of the established IoT network.

Below are further characteristics of the configuration and assembling procedure for the aforementioned components:

- The Google Voice Kit is physically assembled with a Raspberry Pi [62] as a processor unit. The installation includes an operating system and configuration of the software and scripts recommended by Google.
- The Motorola Baby Care is configured through a synchronization process with a smart mobile phone with the Motorola Hubble application installed.

- The TP-Link access point is flashed with OpenWRT (v19.3). The installation includes the required packages for storing, configuring, and capturing traffic.

These devices were used for creating the dataset at the IoT laboratory by capturing benign traffic. In the following section, we describe the data collection procedure.

4.3 Data collection

In order to capture network traffic, `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. A USB storage was attached to the access point for storing the PCAP files. An access point was connected to the Internet via a WAN port. The IoT devices were connected to the WLAN. No other devices were connected to the access point.

The devices were obtained for the purposes of the present study and they were run for the first time in the established infrastructure. Therefore, they were expected to have no malware infection. To capture traffic from Motorola Baby Care, we monitored and executed all of the application features, such as capturing images, playing music, and sending voice commands. We captured two files, one for 24 hours and the other about 1.5 hours, to make sure that the devices complete their data transaction with manufacturer services for likely updates and reporting. Nevertheless, our analysis of the PCAPs didn't show evidence for such an activity. Since the two PCAP files were similar, we used the smaller PCAP file for the analysis. The remote connections to the devices were established both from from the local premises on campus and through a cellular network. The frequency of command execution did not have an impact on the features analyzed.

We used voice commands in two sessions to capture traffic from Google Voice Kit, each about 1.5 hours of traffic capture. The commands were retrieved from the official support website. We covered virtually all categories of voice commands reported in the manuals. The devices were isolated from the rest of the network during the data capture. The devices were not in communication with other IoT devices either. The reason for the isolated traffic is that we aimed to generate the MUD files for each device with this traffic in this phase. During the testing phase, traffic was not isolated. The packet destinations were only on the WAN side since our goal was to analyze the communication pattern with the WAN side.

4.4 The dataset

In this section, we describe our dataset and its features. Also, we report two publicly available datasets for comparison. After collecting the benign data, we generated reports and log files by following the previous work, e.g., [63]. We used `Capinfos` [64], `PassiveDNS` [65], `TCPdstat` [66] , `DNStop` [67], and `Zeek`[68] for the analysis. Moreover, the traffic flow was bidirectionally mapped using `Argus` [69], 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: A software tool to generate statistical reports from a traffic capture. The reports can be in a long format or table format. The long reports are suitable for human readability, and the table format is used for analysis with other tools.
- DNStop: A tool built on libpcap. It provides statistics about DNS queries. It provides a map of destinations and their popularity in the local network.
- Passivedns: It filters DNS queries and returns information about traffic flow direction. It is used in interfaces or for digital forensic operations.
- TCPdStat: A classical network forensic analysis tool.
- 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.

We used the PCAP files to support manual analysis as input for generating the MUD files. For this, we used Mudgee [70] to create MUD profiles for the IoT devices that do not have a MUD profile. We also used mudmaker.org [71], a website helping manufacturers and network administrators to produce their MUD files. It is also possible to consult the YANG model [72] by IETF, to generate the MUD files with the correct syntax.

Our dataset resembles the one published by the UNSW research group. They reported an experimental investigation, in which they collected traffic of 28 distinct IoT devices [73]. They proposed solutions for the security of IoT networks, including manufacturer usage descriptions, anomaly detection with machine learning methods, and SDN. The collected dataset was published in PCAP and CSV file formats. Also, using the Mudgee software, they generated MUD profiles for 20 of their IoT devices. Briefly, the goal of the study was to classify a large set of IoT devices based on their traffic patterns. Instead, our focus is to gather distinctive traffic patterns for specific IoT devices, not necessarily classified. For this, we focus on two IoT devices that provide assistance services. Another similar dataset (CTU) is the one published by the Stratosphere research lab in 2020 [74]. They captured traffic from 20 IoT malware in the lab and analyzed them manually using the Zeek anomaly detector. They also captured benign traffic of three IoT devices with normal behavior.

There are differences, besides similarities between the two datasets and our dataset. All three datasets captured benign network traffic to present the expected behavior of the target IoT devices. Also, the datasets employed 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 our dataset,

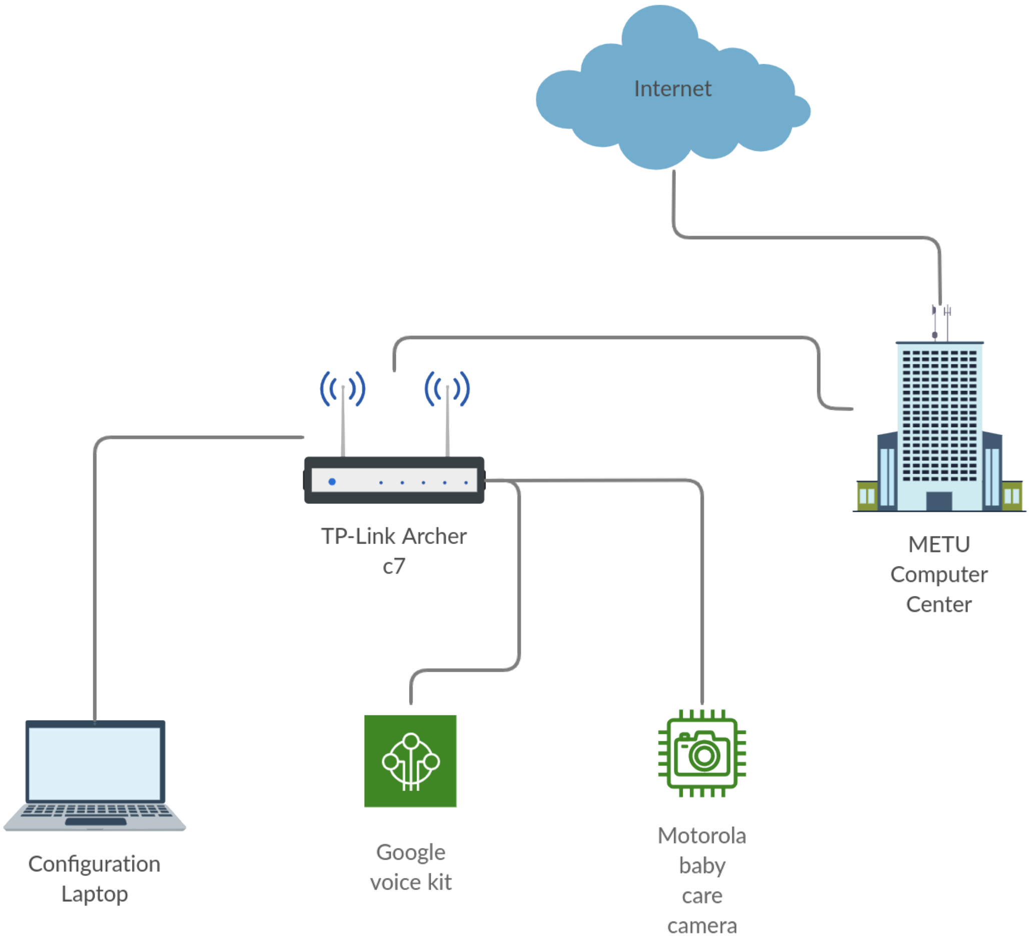


Fig. 3 – IoT lab Infrastructure

the target was IoT devices solely. However, the UNSW dataset includes the traces of other devices as well. Finally, we have no attack traces in the dataset, whereas the CTU dataset includes traces of IoT malware, and the UNSW dataset includes traces from general attack types. The following section presents the data analysis of our data.

4.5 Data analysis

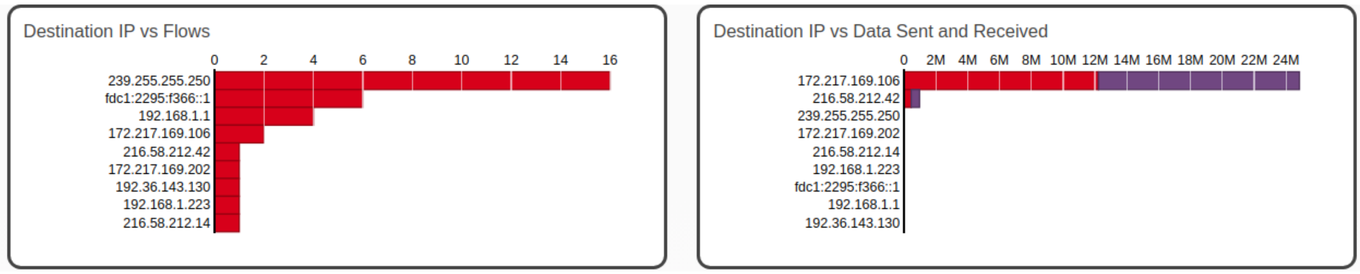
The traffic from the Google Voice Kit and the Motorola Baby Care was analyzed to find the variation of the protocols and packet destinations. Our goal was to investigate if it was possible to detect regular communication patterns for each device by conducting traffic analysis. The ultimate goal was to understand if the whitelisting methods could control the flow of communication in IoT networks.

4.5.1 Google Voice Kit

We used the CapAnalysis tool [75] for visualizing traffic flow and the data volume transmitted between the Google Voice Kit and the rest of the network. A manual inspection revealed that three specific and known protocols were involved during the Google Voice Kit traffic capture, besides one or more unknown protocols. The inspection also showed that most traffic flows were dedicated to the unknown protocol(s).

To find out the unknown protocol(s) and how they communicate, a closer look at the flows is needed. Fig. 4 presents a partial view of the flow table for Google Voice Kit. The red color in the right part shows the amount of data sent and the purple shows the amount of data received. The flows that belong to the unknown protocol started to communicate periodically within two minutes. Furthermore, the destination IP address suggests that those flows belong to the Simple Service Discovery Protocol (SSDP) [76]. Besides the SSDP, the other three protocols used for the communication of Google Voice Kit were DNS, SSL.Google and Google service protocol.

For analyzing the specific destinations of the Google service protocol, we need to find out the IP addresses involved. We used Passivedns and DNStop tools for a further investigation of the data. For visualization, we used the statistics section of CAPAnalysis. The inspection of the PCAPs showed that the majority of the sent data and the received data targeted a specific IP address that be-


Destination IP vs Flows

Destination IP vs Data Sent and Received

Fig. 4 – Destinations of Google Voice Kit

longed to Google servers. The available data allowed us to calculate a ratio (sent/received) that characterized the communication of the device, besides its temporal properties, such as a threshold for the volume of data sent and received in repeating the time loops.

4.5.2 Motorola Baby Care camera

The report generation and PCAP processing methodology were repeated for Motorola Baby Care. Motorola Baby Care uses SSL, HTTP, DNS, RTMP, and a set of protocols not detected by CapAnalysis. An inspection of the captured data revealed that this device used cloud-based services from Google and Amazon. The presence of the plain HTTP connection may indicate a vulnerability from the perspective of cybersecurity, since it allowed plain-text data transfer. Nevertheless, a further inspection is beyond the scope of the present study.

The flow table indicates that unknown protocols targeted three destination ports. The first one was SSDP, distinguishable by its port number 1900. The other two (port 80 and 443) were HTTP and HTTPS ports. The connection on port 443 to the API may serve for authentication, whereas port 80 may serve for uploading.

The Motorola Baby Care camera functionalities are synchronized with different destination subnets in the cloud, as shown in Fig. 5 and Fig. 6. The number of the subnets in the Motorola device was larger than those in the Google Voice Kit. Although this observation is limited to the small scale of the collected data, the available captures may indicate that the number of subnets was small enough to be included in a whitelist.

4.6 Implementation of MUD on the network

We implemented a custom-built MUD manager in our network to evaluate the proposed security features and test if our selected software stack works as expected in the established lab environment. The implementation also allowed us to check the proof of work developed by the generated MUD files for Google Voice Kit and Motorola Baby Care. In this section, we first present the technical background on the MUD concept and then report the details of the implementation.[1]

Recently, there have been three methods to develop a MUD file for IoT devices. The common practice suggests that those methods have the same outcome in an accurate operation. The first method is basically to follow the syntax of the MUD file with the YANG model, also the basis of the other two methods. The second is to use mudmaker.org, an online platform for making MUD files using a wizard. The wizard requires the connections of the IoT device from the user, including the source and destination ports, protocols, the domain name, or IP addresses that the device connects. The third may be useful if the developer does not know the communication pattern of the IoT device. For this, the Mudgee tool may be used, which receives input of a PCAP file and returns the output of the MUD file in JSON or CSV format. In order to achieve an accurate MUD file that includes necessary information for the IoT device to service, the developer needs to test all possible functionalities of the device, from general functions to specific ones, such as updating firmware.

Fig. 7 presents a simplified architecture of a MUD-integrated network. In this architecture, the ”Thing” component represents the IoT device that sends the MUD URL to the gateway, as suggested by [9]. The gateway may be software, a hardware switch, or even a router. The switch then sends the packets to the MUD manager. The MUD manager may be installed on the same device as the gateway or a separate device. The protocol for sending the MUD URL may vary, depending on the implementation. For example, the data may be sent via DHCP packets, or by following the IEEE 802.1X standards [78] to utilize EAP over Radius. The MUD manager extracts the MUD URL, and it requests the MUD file from the MUD file server with a GET command. The MUD file server may be a Web server in the cloud. After the receipt of the MUD file, the MUD manager converts it into specific network configurations and updates the configurations in the network gateway or any device responsible for the access control mechanism in the network. The configurations and the MUD file are deleted after the device disconnects from the network.

4.6.1 The MUD files for Google Voice Kit and Motorola Baby Care

We generated an initial version of the MUD file for Google Voice Kit and Motorola Baby Care by using the output of

[1] The MUD files and PCAP files are accessible at the Open Science Framework (OSF) repository: https://osf.io/gtw2h/

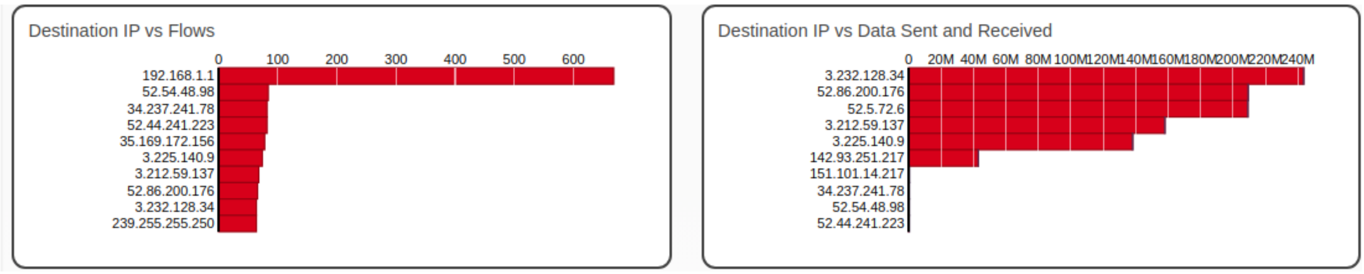


Fig. 5 – Destinations vs. flows in Motorola Baby Care camera

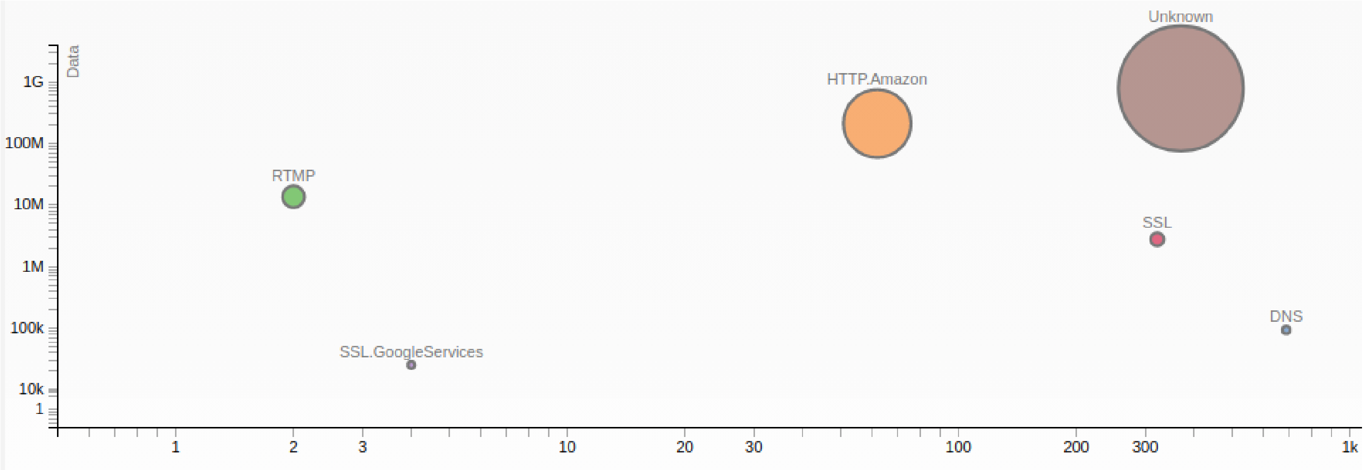


Fig. 6 – Data volumes and flows in Motorola Baby Care camera

```
.......................................
.                      ____________   .   _____________
.                     |            |  .  |             |
.                     |    MUD     |-->get URL-->|    MUD      |
.                     |  Manager   |  .(https)   | File Server |
.  End system network |____________|<-MUD file<-<|_____________|
.                             .       .
.                             .       .
. _______                 _________   .
.|       | (DHCP et al.) | router  |  .
.| Thing |---->MUD URL-->|   or    |  .
.|_______|               | switch  |  .
.                        |_________|  .
.......................................
```

Fig. 7 – Architecture of MUD-enabled network

the Mudgee software tool. The PCAP files were investigated to reveal the communication patterns in the network. A manual check was performed for the MUD files since specific MUD rules may be incorrect when fully automatically generated from PCAP files, especially for the camera devices and the devices with remote access. For example, suppose a parent uses a camera to monitor various locations through different IP addresses. If the MUD file is limited by whitelisting for a specific IP address, it will lose its capacity to allow multiple IP addresses. Accordingly, the PCAP should be manually analyzed for the requests and the remote access controls. When controlled through an API or service from the cloud, the IP addresses must be specified for remote access.

Manual traffic inspection and analysis, and MUD file generation may be time consuming for network administrators, for each and every IoT device in the network. As mentioned before, our study aims to make network administrators who are concerned about automatic MUD generation methods, more confident about the usability and effectiveness of these methods. It is necessary to mention that all these alternatives are just temporary solutions until the manufacturers adopt the MUD standard.

Due to the novelty of the MUD concept, not many experiments are available in the literature. The NIST, a major contributor and supporter of MUD, developed guidances for MUD implementation, including different types of builds. Three builds were for proprietary software and technologies. The other was open source, though being limited to supporting hardware signatures. Therefore, we developed a new build based on an open-source MUD manager, namely osMUD [79]. We compiled the osMUD for the TP-Link Archer c7 v2 model running on the MIPS architecture. After configuring the DNSMasq for reading the DHCP packets to extract the MUD URLs, the service started to work.

In the next section, we discuss our experiments, which were conducted to verify that the MUD-based network can meet the communication needs of the devices and add a layer of security on the network stack.

5. EVALUATION

MUD-enabled networks are open to potential threats that target them. The analysis of the collected data from the two IoT devices showed that whitelisting methods are possible to implement through generated MUD files. In this section, we report the results of our data analysis and present an evaluation of MUD-enabled networks at a conceptual level.

5.1 Results of traffic data analysis

The traffic patterns reveal that the Google Voice Kit uses three commonly used protocols to serve the user: SSL, SSDP, and DNS. Its destinations are also limited to a couple of Google services, suggesting that whitelisting the network communication is reasonable for the Google Voice Kit. As for Motorola Baby Care, the traffic patterns show that it uses five protocols: SSL for Google and the Hubble API, HTTP for Amazon cloud, RTMP for streaming, SSDP, and DNS. Furthermore, the number of protocol destinations is less than ten. It is even less than four if domain names are used. Those results suggest that whitelisting the network communication for Motorola Baby Care is also a feasible approach. The findings also suggest that our MUD files can be implemented in any IoT network having those two devices. It is also possible to reduce the attack surface by modifying the MUD files so that specific protocols, such as SSDP, are not used.

5.2 Threats against MUD-enabled network

IoT devices are usually added to networks together with other computing devices, such as end-user computers. In principle, the IoT devices should serve in separated and isolated LANs or VLANs, whereas the common practice allows remote access to IoT devices. The spread of IoT malware, such as the Mirai indicates that most IoT devices are accessible remotely. In addition, communication among the IoT devices should be limited to registered and identified devices. Remotely accessible IoT devices may also grant unauthorized access to attackers, which may then be followed by dictionary attacks against the authentication mechanisms of IoT devices that serve publicly. In case of a successful attack, the IoT devices may function as zombie devices, which take part in DDoS attacks. This section presents four cases that describe attack surfaces as such.

A compatible network environment is needed to evaluate the MUD files for reliability. We achieved this compatibility by implementing osMUD manager and modifying the gateway configurations that run OpenWRT as firmware. After building this infrastructure, Google Voice Kit and Motorola Baby Care camera ran without problems in the MUD-enabled network. In this network, the MUD files specify access control through their entries, and the MUD manager enforces those entries in the access control list on the OpenWRT's firewall, which in our case is iptables. Moreover, the MUD manager initially blocks all communication from any source to any destination by default. However, by adding rules to the access list, devices can communicate with whitelisted destinations and protocols. Therefore, the firewall rules are the mechanism of prevention. As a result, the attacks which can bypass the firewall rules can also bypass MUD-enabled networks. Below we present a set of cases for explaining potential attacks on the MUD-enabled network and the MUD standard from the perspective of cybersecurity. We assume that the attacker has understanding of MUD-enabled networks, how DNS and firewalls work and in some cases, has access to the local network or is an insider. This general threat model where insiders and outsiders can be involved is shown in Fig. 8. Usually, an insider tries to exploit vulnerabilities in protocols and an outsider tries to brute force and do dictionary attacks against authentication panels of IoT devices. As previously explained, in major cases, these devices are used to launch DDoS attacks against public services. The outsiders are usually Advanced Persistent Threat (APT) groups who aim to damage governments or international organizations. The insiders are usually the hackers who aim to steal information and publish or sell them.

5.2.1 *Case 1: Using non-vendor MUD files*

The architecture of the MUD-enabled network is designed for securing the integrity and verification of the MUD file. On the other hand, specific methods may be developed to bypass the available security mechanisms. Moreover, vendor-provided, thus verified MUD files may not be available. In particular, if the MUD file is generated by analyzing the traffic collection, it may be limited in scope since certain functions and features of the IoT devices might not have been executed during data collection from the network traffic. This may lead to missing patterns in the generated MUD file. For instance, suppose we did not 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 the camera, then we miss the destination address that has to be included in the MUD file.

As a result, our generated MUD file would not cover the entries for that destination, and the camera would not be able to connect to that server in the MUD-enabled network. Thus, a network administrator should either re-

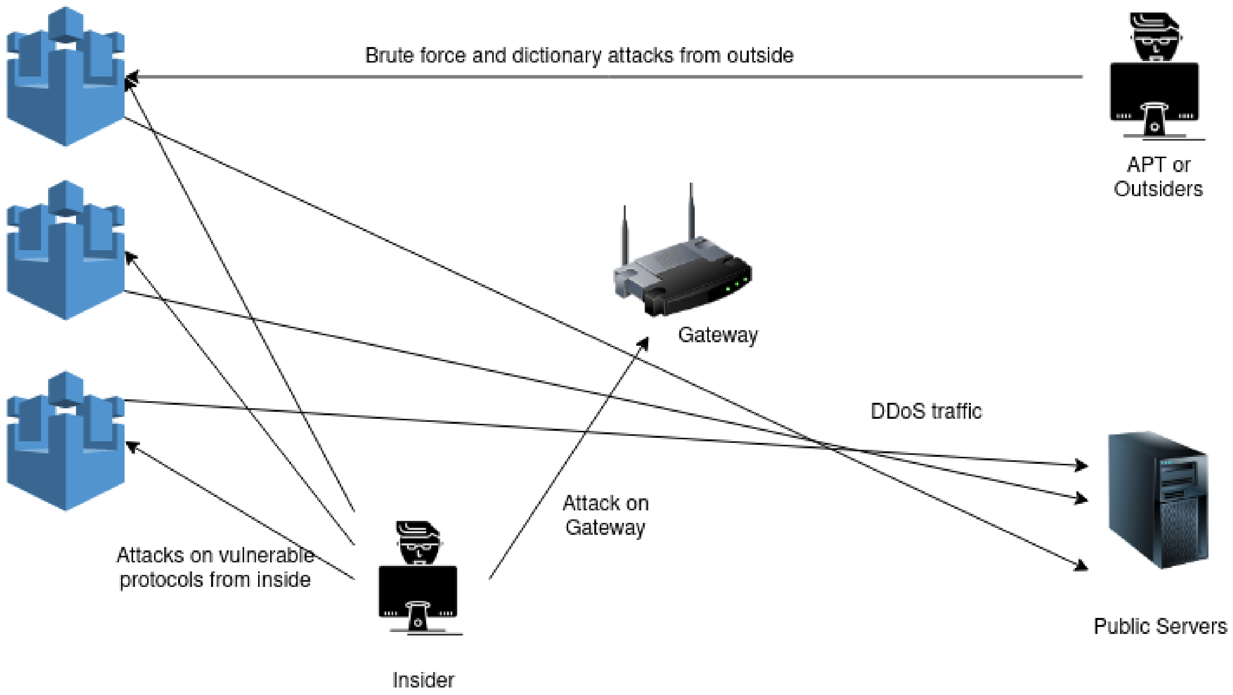

Brute force and dictionary attacks from outside
APT or Outsiders
Gateway
DDoS traffic
Attack on Gateway
Attacks on vulnerable protocols from inside
Insider
Public Servers

Fig. 8 – General threat model on MUD-enabled network

quest a verified MUD file from the vendor or execute all the functionalities of the IoT device and capture its communication patterns to maximize the scope of the MUD file. Also, the network administrator may want to restrict specific protocols or ports not specified by the MUD file in some cases. For instance, the SSDP protocol may be blacklisted due to security policies in an organization's network. In this case, the administrator either needs to use a customized version of the MUD file or enforce the rules or flows from a security layer lower than the MUD manager executes.

5.2.2 Case 2: MUD URL spoofing

A likely method of compromising the devices in an IoT network is to point at a compromised MUD file by sending a modified request containing a malicious MUD URL to the MUD manager. Suppose an attacker can modify the DHCP request (or any other protocol that the network supports for initiator request) and change the MUD URL to a fake URL pointing to a bogus MUD file. In that case, the layer of security added by MUD is bypassed. This modification can be implemented in two ways. The first is the Man-in-the-Middle (MITM) attack, a known attack type that aims to sniff, interpret, modify, and forward the packets in transmission. Second, another device may spoof the newly added device's Media Access Control (MAC) address and send the initiator request containing the MUD URL. In both cases, the result is fake access control entries, which can deny the service of IoT devices or make it available to communicate with destinations not acceptable by manufacturer or network administrators. Fig. 9 shows a potential model of this attack type.

5.2.3 Case 3: DNS spoofing

Another approach to bypass the MUD-enabled network policies and update the MUD file with a fake one is to spoof DNS query results. 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 enforce fake access control rules on the IoT device.

5.2.4 Case 4: Attacks on MUD manager

The last scenario that we present as a case study, which aims to exemplify threats for MUD-enabled networks, is to have a vulnerable MUD manager software. If an attacker gains access to the MUD manager by exploiting a vulnerability in the MUD manager, the firmware, or the operating system that serves the MUD manager, modification in the policies and security mechanism of the network may become possible. This scenario is not specific to MUD-enabled networks. Nevertheless, MUD managers are still at an early stage in development. Therefore, we should consider the presence of critical vulnerabilities in MUD-enabled networks.

6. DISCUSSION AND CONCLUSIONS

In this section, we summarize the findings of the study, present a discussion of the MUD concept for improving the security of IoT networks and the operational assumptions and limitations in the implementation.

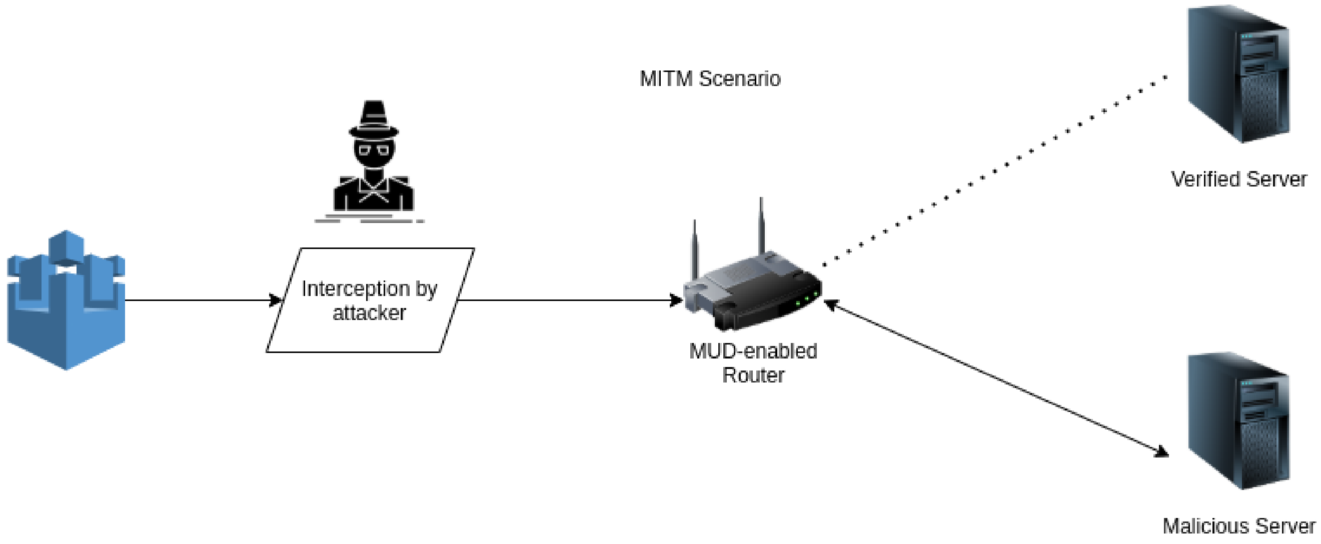


Fig. 9 – MUD URL spoofing

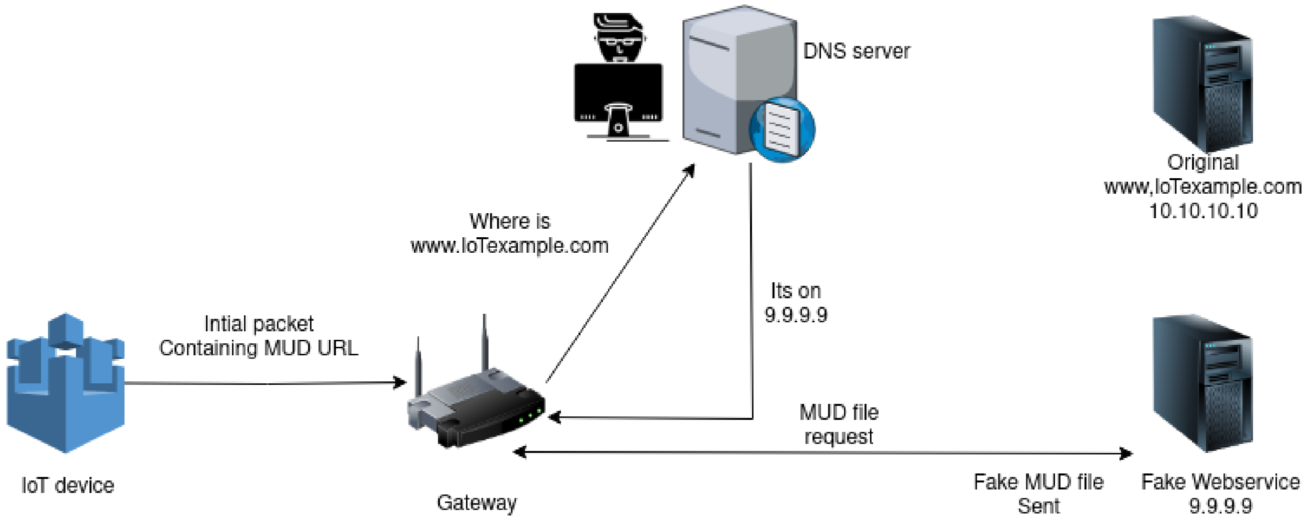

DNS server
Original www,IoTexample.com 10.10.10.10
Where is www.IoTexample.com
Its on 9.9.9.9
Intial packet Containing MUD URL
MUD file request
IoT device
Gateway
Fake MUD file Sent
Fake Webservice 9.9.9.9

Fig. 10 – DNS spoofing

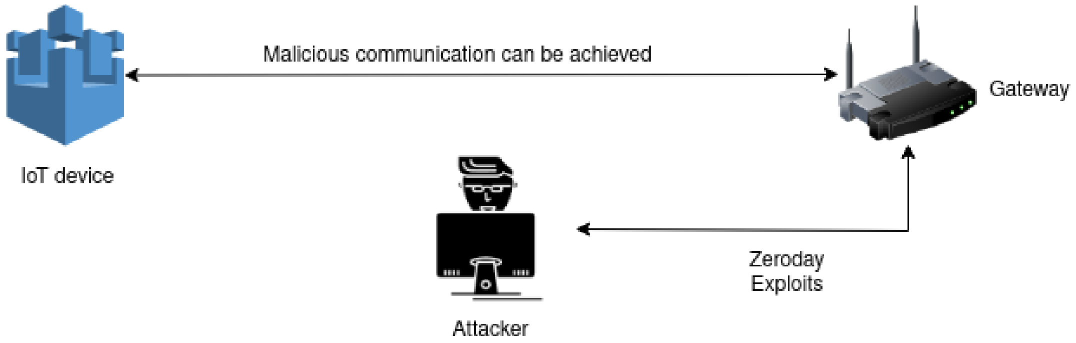


Fig. 11 – Attacks on MUD manager

The present study is a simplistic Verification of Concept (VoC), aka Proof of Concept (PoC) study, which demonstrates the use of MUD implementations using an open-source methodology on two specific devices, rather than being a performance evaluation study that aims to evaluate the performance of specific software over specific hardware. A practical contribution of the study is the presentation of explicit, step-by-step descriptions of the implementation on two devices through open-source methodology for MUD to improve the security of IoT networks, in particular, in smart home environments. Therefore, we started with an investigation of available approaches and solutions for improving the security of smart home networks. Recently, smart home networks have been most vulnerable to DDoS attacks, where infected IoT devices are used as botnets. We found that most of the available solutions are based on SDN architectures and ML methods. As an alternative, we proposed that whitelisting policies are effective given their ease of operation and applicability on smart home scale IoT networks. Therefore, we presented MUD as a concept compatible with customized standardization, having a simplistic architecture and straightforward implementation for enforcing whitelisting policies in the network. We tested the approach in two devices by generating MUD files from network traffic data. We used the syntax of the Yang model, 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 concept for the MUD files for building a MUD-enabled network.

Recent approaches for securing small-to-medium size networks have mostly focused on downsizing available misuse-based IDSs to make them compatible with IoT networks. In general, the goal is to disassemble a working IDS and make it cost-effective and lighter for improved performance in an IoT network. The main issue with this approach is that the architectures of the available IDSs are not designed for this purpose. Therefore, those devices may not function as expected even by modification and improvements. Another issue is that signature-based IDSs are disadvantageous against novel, polymorphic malware. More recent approaches benefit from ML to generate models of normal behavior profiles and then run the models to detect anomalies. Although those approaches are feasible due to their robustness in recognizing patterns in IoT networks, their training and execution may demand high processing power.

In contrast to the available approaches that aim at providing technical solutions for IoT network security, MUD is a standard developed and contributed by the manufacturers. The MUD concept enforces the manufacturers to consider security before presenting the product to the market rather than providing security afterward. In return, providing product-specific MUD files increases a manufacturer's reputation in producing reliable and secure devices. The MUD concept has further advantages, such as being easy to implement compared to the alternatives. Furthermore, it is compatible with different hardware and operating systems. Also, computational requirements are relatively low, thus bringing performance through a lightweight architecture.

The main feature of MUD-enabled networks is the high level of security by adding a layer on top of the security stack in the network. The whitelisting methods enable communication with a trusted destination in a trusted direction by utilizing trusted ports and protocols. This feature is comparable with the expected behavior profiling with machine learning models, yet MUD has its advantages.

Implementing a MUD-enabled network is still challenging due to the lack of documentation and experience about this technology. However, the present study demonstrated its potential for developing device-compatible builds that do not have to follow the previous ones. As for the implementation, the Google Voice Kit and the Motorola Baby Care did not have original MUD files to allow comparative analysis. These findings are valuable since they reflect the potential of MUD as an emerging technology for IoT cybersecurity.

7. FUTURE STUDIES

In this study, we did not observe any indication of low performance during service. Nevertheless, the comparison between the normal working and MUD-enabled networks must be made based on different parameters in more crowded intranet networks. More generally, the functionality of the MUD files on different builds needs to be investigated. In particular, a complex network from the hardware perspective may facilitate evaluating the performance of the open-source osMUD manager and the inter-domain communications.

There are open questions related to the earlier versions of security cameras that do not support communication through API. Further research is needed to check if the MUD approach would effectively improve 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 affordable logistics. Finally, other bypass mechanisms of MUD-enabled networks should be investigated in future research. This can help for early upgrades on MUD to promise enhanced security.

Another frontier of future research is integrating the MUD-enabled network in a large-scale implementation that includes multiple builds in a real environment. In particular, integrating the MUD manager with an SDN controller in the builds published by NIST may lead to a better performance in a large-scale architecture than its alternatives. At this stage, it is not certain if MUD is able to operate effectively in crowded networks, such as the ones that use 5G technology to communicate.

The presentation of the quantitative evaluation is limited in the present study. The primary goal has been to present a verification of concept approach and demonstrate its implementation on two devices, thus positioning itself as a conceptual approach without directly aiming at presenting a dataset and the performance of a methodology. Given this approach, a quantitative evaluation on the performance of a network of this scale would be incomplete since it involves two IoT devices. We have also discussed scenarios for possible attacks on the MUD-enabled network. In future work we plan to address the concept through rigorous quantitative evaluation with large-scale, MUD-utilized IoT networks. We also plan to propose a set of prevention methods for the discussed threats. Finally, we plan to compose a dataset with attacks performed on this testbed for the benefit of the academic community.

REFERENCES

[1] Ashton, K., 2009. That 'internet of things' thing. RFID journal, 22(7), pp.97-114.

[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] Nguyen, Duc-Thang, and Taehong Kim. "An SDN-based connectivity control system for Wi-Fi devices." Wireless Communications and Mobile Computing 2018 (2018)., doi: 10.1155/2018/9359878.

[5] "OWASP Internet of Things Project- OWASP." [Online]. Available: https://owasp.org/www-project-internet-of-things/. [Accessed: 06-May-2020].

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

[7] Feamster, Nick. "Outsourcing home network security." In Proceedings of the 2010 ACM SIGCOMM workshop on Home networks, pp. 37-42. 2010., doi: 10.1145/1851307.1851317.

[8] Sabhnani, Maheshkumar, and Gursel Serpen. "Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set." Intelligent data analysis 8, no. 4 (2004): 403-415., 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] Xia, Feng, Laurence T. Yang, Lizhe Wang, and Alexey Vinel. "Internet of things." International journal of communication systems 25, no. 9 (2012): 1101., doi: 10.1002/dac.2417.

[12] Ray, Partha Pratim. "A survey on Internet of Things architectures." Journal of King Saud University-Computer and Information Sciences 30, no. 3 (2018): 291-319., doi: 10.1016/j.jksuci.2016.10.003.

[13] Minerva, Roberto, Abyi Biru, and Domenico Rotondi. "Towards a definition of the Internet of Things (IoT)." IEEE Internet Initiative 1, no. 1 (2015): 1-86.

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

[15] Zarpelão, Bruno Bogaz, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. "A survey of intrusion detection in Internet of Things." Journal of Network and Computer Applications 84 (2017): 25-37., doi: 10.1016/j.jnca.2017.02.009.

[16] Chaabouni, Nadia, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac, and Parvez Faruki. "Network intrusion detection for IoT security based on learning techniques." IEEE Communications Surveys Tutorials 21, no. 3 (2019): 2671-2701., doi: 10.1109/COMST.2019.2896380.

[17] Elrawy, Mohamed Faisal, Ali Ismail Awad, and Hesham FA Hamed. "Intrusion detection systems for IoT-based smart environments: a survey." Journal of Cloud Computing 7, no. 1 (2018): 1-20., 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)(2018): 94-96, ISSN:2321-9653.

[19] Sherly, J., and D. Somasundareswari. "Internet of things based smart transportation systems." International Research Journal of Engineering and Technology 2, no. 7 (2015): 1207-1210.

[20] Rizwan, Patan, K. Suresh, and M. Rajasekhara Babu. "Real-time smart traffic management system for smart cities by using Internet of Things and big data." In 2016 international conference on emerging technological trends (ICETT), pp. 1-7. IEEE, 2016., doi: 10.1109/ICETT.2016.7873660.

[21] Gross, Todd M., Bruce W. Bode, Daniel Einhorn, David M. Kayne, John H. Reed, Neil H. White, and John J. Mastrototaro. "Performance evaluation of the MiniMed® continuous glucose monitoring system during patient home use." Diabetes technology & therapeutics 2, no. 1 (2000): 49-56.

[22] Tankard, Colin. "The security issues of the Internet of Things." Computer Fraud & Security 2015, no. 9 (2015): 11-14., doi: 10.1016/S1361-3723(15)30084-1.

[23] Mahjabin, Tasnuva, Yang Xiao, Guang Sun, and Wangdong Jiang. ”A survey of distributed denial-of-service attack, prevention, and mitigation techniques.” International Journal of Distributed Sensor Networks 13, no. 12 (2017): 1550147717741463., doi: 10.1177/1550147717741463.

[24] “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].

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

[26] De Donno, Michele, Nicola Dragoni, Alberto Giaretta, and Angelo Spognardi. ”DDoS-capable IoT malwares: Comparative analysis and Mirai investigation.” Security and Communication Networks 2018 (2018)., doi:10.1155/2018/7178164.

[27] De Donno, Michele, Nicola Dragoni, Alberto Giaretta, and Manuel Mazzara. ”AntibIoTic: protecting IoT devices against DDoS attacks.” In International Conference in Software Engineering for Defence Applications, pp. 59-72. Springer, Cham, 2016, doi:10.1007/978-3-319-70578.

[28] Denning, Dorothy E. ”An intrusion-detection model.” IEEE Transactions on software engineering 2 (1987): 222-232.

[29] Ghorbani, Ali A., Wei Lu, and Mahbod Tavallaee. Network intrusion detection and prevention: concepts and techniques. Vol. 47. Springer Science & Business Media, 2009.

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

[31] Lee, Tsung-Han, Chih-Hao Wen, Lin-Huang Chang, Hung-Shiou Chiang, and Ming-Chun Hsieh. ”A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN.” In Advanced Technologies, Embedded and Multimedia for Human-centric Computing, pp. 1205-1213. Springer, Dordrecht, 2014.

[32] 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

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

[34] Omar, Mohd Nizam, Guled Yusuf Mihile Guled, Haryani Zakaria, Angela Ampawan, and Roshidi Din. ”Home-Based Intrusion Detection System.” Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9, no. 2-4 (2017): 107-111.

[35] Gajewski, Mariusz, Jordi Mongay Batalla, George Mastorakis, and Constandinos X. Mavromoustakis. ”A distributed IDS architecture model for Smart Home systems.” Cluster Computing 22, no. 1 (2019): 1739-1749.

[36] Sheikh, Nazim Uddin, Hasina Rahman, Shashwat Vikram, and Hamed AlQahtani. ”A Lightweight Signature-Based IDS for IoT Environment.” arXiv preprint arXiv:1811.04582 (2018).

[37] Cvitić, Ivan, Dragan Peraković, Marko Periša, and Mate Botica. ”Novel approach for detection of IoT generated DDoS traffic.” Wireless Networks 27, no. 3 (2021): 1573-1586., doi: 10.1007/s11276-019-02043-1.

[38] Kumar, Ayush, and Teng Joon Lim. ”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, 2019.

[39] Mehr, Shideh Yavary, and Byrav Ramamurthy. ”An SVM based DDoS attack detection method for RYU SDN controller.” In Proceedings of the 15th international conference on emerging networking experiments and technologies, pp. 72-73. 2019., doi: 10.1145/3360468.3368183.

[40] Hamza, Ayyoob, Hassan Habibi Gharakheili, and Vijay Sivaraman. ”Combining MUD policies with SDN for IoT intrusion detection.” In Proceedings of the 2018 Workshop on IoT Security and Privacy, pp. 1-7. 2018.,doi:10.1145/3229565.3229571.

[41] Doshi, Rohan, Noah Apthorpe, and Nick Feamster. ”Machine learning ddos detection for consumer internet of things devices.” In 2018 IEEE Security and Privacy Workshops (SPW), pp. 29-35. IEEE, 2018., doi: 10.1109/SPW.2018.00013.

[42] Mehmood, Amjad, Mithun Mukherjee, Syed Hassan Ahmed, Houbing Song, and Khalid Mahmood Malik. ”NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks.” The Journal of Supercomputing 74, no. 10 (2018): 5156-5170., doi: 10.1007/s11227-018-2413-7.

[43] Misra, Sudip, P. Venkata Krishna, Harshit Agarwal, Antriksh Saxena, and Mohammad S. Obaidat. ”A learning automata based solution for preventing distributed denial of service in internet of things.” In 2011 international conference on internet of things and 4th international conference on cyber, physical and social computing, pp. 114-122. IEEE, 2011., doi: 10.1109/iThings/CPSCom.2011.84.

[44] Xu, Ke, Xiaoliang Wang, Wei Wei, Houbing Song, and Bo Mao. ”Toward software defined smart home.” IEEE Communications Magazine 54, no. 5 (2016): 116-122., doi: 10.1109/BigDataService.2017.41.

[45] Sivaraman, Vijay, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. ”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, 2015.

[46] Sambandam, Narmadha, Mourad Hussein, Noor Siddiqi, and Chung-Horng Lung. ”Network security for iot using sdn: Timely ddos detection.” In 2018 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1-2. IEEE, 2018., doi: 10.1109/DESEC.2018.8625119.

[47] Sivanathan, Arunan, Daniel Sherratt, Hassan Habibi Gharakheili, Vijay Sivaraman, and Arun Vishwanath. ”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, 2016.

[48] Yin, Da, Lianming Zhang, and Kun Yang. ”A DDoS attack detection and mitigation with software-defined Internet of Things framework.” IEEE Access 6 (2018): 24694-24705., doi: 10.1109/ACCESS.2018.2831284.

[49] Soni, Sumit, and Bharat Bhushan. ”A comprehensive survey on blockchain: Working, security analysis, privacy threats and potential applications.” In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), vol. 1, pp. 922-926. IEEE, 2019., doi: 10.1109/ICICICT46008.2019.8993210.

[50] Moniruzzaman, Md, Seyednima Khezr, Abdulsalam Yassine, and Rachid Benlamri. ”Blockchain for smart homes: Review of current trends and research challenges.” Computers & Electrical Engineering 83 (2020): 106585., doi: 10.1016/j.compeleceng.2020.106585.

[51] Sagirlar, Gokhan, Barbara Carminati, and Elena Ferrari. ”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, 2018.

[52] Javaid, Uzair, Ang Kiang Siang, Muhammad Naveed Aman, and Biplab Sikdar. ”Mitigating loT device based DDoS attacks using blockchain.” In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, pp. 71-76. 2018., doi: 10.1145/3211933.3211946.

[53] Stewart, Chase E., Anne Maria Vasu, and Eric Keller. ”CommunityGuard: A crowdsourced home cyber-security system.” In Proceedings of the ACM International workshop on security in software defined networks & network function virtualization, pp. 1-6. 2017., doi: 10.1145/3040992.3040997.

[54] Bhardwaj, Ketan, Joaquin Chung Miranda, and Ada Gavrilovska. ”Towards IoT-DDoS prevention using edge computing.” In USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). 2018.

[55] Procopiou, Andria, Nikos Komninos, and Christos Douligeris. ”ForChaos: Real time application DDoS detection using forecasting and chaos theory in smart home IoT network.” Wireless Communications and Mobile Computing 2019 (2019).

[56] Dodson, Donna, Douglas Montgomery, W. Polk, Mudumbai Ranganathan, Murugiah Souppaya, Steve Johnson, Ashwini Kadam et al. 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, 2020.

[57] Hamza, Ayyoob, Dinesha Ranathunga, Hassan Habibi Gharakheili, Matthew Roughan, and Vijay Sivaraman. ”Clear as MUD: Generating, validating and applying IoT behavioral profiles.” In Proceedings of the 2018 Workshop on IoT Security and Privacy, pp. 8-14. 2018, doi: 10.1145/3229565.3229566.

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

[59] “Voice.” [Online]. Available: https://aiyprojects.withgoogle.com/voice/. [Accessed: 04-Sep-2020].

[60] “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].

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

[62] ”Buy a Raspberry Pi3 Model B” [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 04-Sep-2020].

[63] “Dataset Overview - Stratosphere IPS.” [Online]. Available: https://www.stratosphereips.org/datasets-overview. [Accessed: 04-Sep-2020].

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

[65] “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].

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

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

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

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

[70] “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].

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

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

[73] 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.

[74] ”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].

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

[76] 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.

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

[78] “802.1X: Port-Based Network Access Control |.” [Online]. Available: https://1.ieee802.org/security/802-1x/. [Accessed: 04-Sep-2020].

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

[80] Greer, Christopher, Martin Burns, David Wollman, and Edward Griffor. ”Cyber-physical systems and internet of things.” (2019): 03-07. doi.org/10.6028/NIST.SP.1900-202

[81] Feraudo, Angelo, Poonam Yadav, Richard Mortier, Paolo Bellavista, and Jon Crowcroft. ”SoK: Beyond IoT MUD Deployments–Challenges and Future Directions.” arXiv preprint arXiv:2004.08003 (2020).

AUTHORS

Milad Kazemi Darazam is a Ph.D. student in the medical informatics program at Middle East Technical University (METU). He received his master’s degree in cybersecurity in 2020. His research interests are network security, IIoT, critical safety systems, and neurosecurity.

Pelin Angin is an assistant professor of computer engineering at Middle East Technical University (METU). She completed her B.S. in computer engineering at Bilkent University in 2007 and her Ph.D. in computer science at Purdue University, USA in 2013. Between 2014-2016, she worked as a visiting assistant professor and postdoctoral researcher at Purdue University. Her research interests lie in the fields of cloud computing and IoT security, distributed systems, 5G networks and blockchain. She is among the founding members of the Systems Security Research Laboratory and an affiliate of the Wireless Systems, Networks and Cybersecurity Laboratory at METU. She serves on the editorial boards of multiple journals on IoT and mobile computing. Her work in security has been published in high impact journals including IEEE Transactions on Dependable and Secure Computing, Computers Security and IEEE Access among others.

Cengiz Acartürk received a B.Sc. degree in mechanical engineering and an M.Sc. degree in cognitive sciences from the Informatics Institute, Middle East Technical University (METU), Turkey, in 1998 and 2005, respectively, and a Ph.D. degree in computer science from the Center for Intelligent Systems and Robotics (ISR), Department of Informatics, Knowledge and Language Processing Institute (WSV), University of Hamburg, Germany, in 2010. He worked at the Cognitive Science and Cybersecurity Graduate Programs, METU Informatics Institute. He is recently conducting research at Jagiellonian University, Poland. His research interests include interaction between cybersecurity and cybercognition, human-computer interaction, human factors in cybersecurity, natural language processing (NLP). He has been a member of IEEE, ACM and the Cognitive Science Society, as well as of local ICT NGOs.