Reichman University Efi Arazi School of Computer Science

Reichman University

Efi Arazi School of Computer Science

M.Sc. program - Research Track

MUDIS: MUD Inspection System

by

Ran Shister

M.Sc. dissertation, submitted in partial fulfillment of the requirements
for the M.Sc. degree, research track, School of Computer Science
Reichman University (The Interdisciplinary Center, Herzeliya)

March, 2022

This work was carried out under the supervision of **Prof. Anat Bremler-Barr** from the Efi Arazi School of Computer Science, Reichman University.

Abstract

Analyzing the network behavior of IoT devices, including which domains, protocols, and ports the device communicates with, is a fundamental challenge for IoT security and identification. Solutions that analyze and manage these areas must be able to learn what constitutes normal device behavior and then extract rules and features to permit only legitimate behavior or identify the device.

The Manufacturer Usage Description (MUD) is an IETF white-list protection scheme that formalizes the authorized network behavior in a MUD file; this MUD file can then be used as a type of firewall mechanism.

This thesis introduces MUDIS, a MUD Inspection System that inspects the network behavior of devices, based on their formal description in the MUD file. We use MUDIS to examine several use-cases that demonstrate why learning what is normal behavior for an IoT device is more challenging than expected. For example, (i) how the same IoT device, with the same firmware, can exhibit different behavior or connect to different domains with different protocols, depending on the device's geographical location; (ii) the impact of a firmware update; (iii) the correlation of network behavior between different devices of the same manufacture, and more.

MUDIS inspects two MUD files, clusters together and graphically visualize identical, similar, and dissimilar rules. It then calculates a similarity score that measures the similarity between them both. It also generalizes the two MUD files where possible, such that the resulting generalized MUD covers all the permitted (Allowed List) network behavior for both MUDs.

We demonstrate MUDIS comparison and generalization features, by processing MUD files that originate in different locations, compare their rules to learn the impact of the location over devices network behavior and generalize them to create a comprehensive MUD file that is applicable for all locations.

Our open-source MUDIS tool and proof-of-concept dataset are available for researchers and IoT manufacturers, allowing anyone to gain meaningful insights over the network behavior of IoT devices.

The research and its results were published and presented at the IEEE/IFIP Network Operations and Management Symposium (NOMS) 2022 conference in Budapest, Hungary [3], [4]

Contents

List of Figures

List of Tables

1 Introduction

The Manufacturer Usage Description (MUD) is an IETF white-list protection scheme [10] that formalizes the authorized network behavior in a MUD file. This MUD file can then be used as a type of firewall mechanism that provides security for the highly diverse IoT devices. MUD files consist of Access Control Lists (ACLs), each with several Access Control Entries (ACEs). Each ACE is defined as a 5-tuple:

$$\begin{gathered} ACE = (legitimate\_endpoints, protocol, \\ source\_port, destination\_port, direction) \end{gathered} \tag{1}$$

The MUD file is fetched by the IoT device using DHCP or LLDP, and thus there is a single MUD file for each firmware version, regardless of any other device or network factors.

The MUD can be provided by the manufacture or learned based on information captured from the device network traffic (PCAP) using a MUD generator tool such as MUDGEE [8] or MUD-PD [14]. Environmental variables can influence the network behavior of an IoT device and hence, have a direct impact on IoT security, including the MUD file that is learned [13, 15]. For example, a device's location impacts its behavior [3], which makes learning what is normal and secure behavior for an IoT device more challenging than expected. In many cases, the same IoT device, with the same firmware, can exhibit different behavior or connect to different domains/IPs with different ports and protocols, depending on the device's environment variables. This is even more challenging when learning the behavior of IoT devices that have more than one different environment variable (e.g., internet connection, DNS blocking, human interaction).

We present a novel and unique tool called MUD Inspection System or MUDIS for short. MUDIS inspects and analyzes two MUD files by comparing their rules, and produces a single generalized MUD file that is comprehensive, tight, and secure for both MUDs. MUDIS is useful for many cases, including analyzing MUDs that were generated from different network traffic due to different environmental factors, analyzing the differences in MUDs between different firmware versions, identifying anomalies based on their network behavior to spot rare actions like firmware updates, find malware infected devices, and more.

MUDIS is a web application with a RESTful web service that is written in Python and uses MongoDB for storage, following the Object-Oriented Programming approach. All the code is open source and available for the use of other researchers and IoT manufacturers at [21], together with our POC dataset [12] and an easy-to-use setup guide based on Docker.

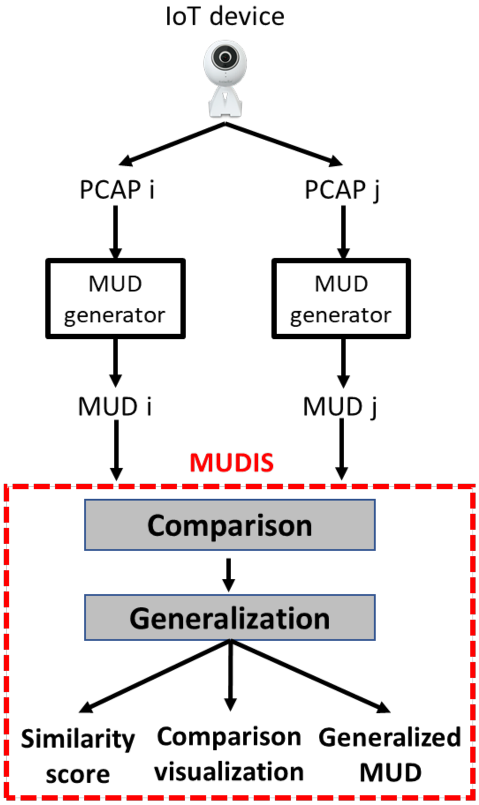


Figure 1: MUDIS architecture

The MUDIS architecture, depicted in Figure 1, receives two MUD files as input and performs four tasks, using a set of algorithms: parsing the input MUDs; comparing their rules; generalizing them into one MUD file; and then graphically visualizing the results. The MUDIS comparison task visualizes the differences between the two files and highlights identical ACEs, similar ACEs, clusters of ACEs, and dissimilar ones. The number of ACEs may differ between the devices and the network behavior captured, and can range from just a few ACEs to a few dozens of them. This emphasizes the importance of comparing and visualizing the ACEs so we can easily spot similarities and differences between the two MUDs. MUDIS also calculates a similarity score that measures and numerically represent the similarity between the two MUD files. The comparison task helps us drill down and gain insights about the origin of the differences, and analyze and emphasize their impact on the device's network behavior. For example, these may include domain differences, encrypted vs. plain communication, different ports and protocols for the same endpoint, the use of cloud services, and much more. The MUDIS generalization task outputs a generalized and comprehensive MUD file that can white-list the network behavior of both MUD files, in a tight and secure manner. The naive method would be to add both sets of rules to form a single unified MUD. However, in MUDIS we use ranges in the domain (e.g. *.iotvendor.com) field to create a generalized MUD with fewer rules. By doing this, we increase the explainability of the resulting MUD and reduce implementation costs in the firewall, which is crucial for network administrators and device manufacturers who need to support the devices' MUDs.

We used MUDIS features to examines how a device's location can influence its network behavior. We found that, *depending on its location, the same IoT device with the same firmware behaves differently and communicates with different domains, protocols, and ports*. To the extent of our knowledge, this is the first work that defines device location as a factor impacting device behavior.

Our dataset contains measurements for devices in our lab that were virtually connected to different locations using VPN, or logically connected to different locations by registering the device in the IoT application in different countries; this data was analyzed along with information from Ren et al. [19] who captured devices that were both physically positioned and logically connected in two locations. We show that, in many cases, the device location of the IoT device will impact its network behavior for various reasons. These can range from marketing reasons where the same IoT has different features while operating in different locations, to country requirements, to weak encryption, privacy regulations, CDN-like solutions, and more. The only related work we are aware of deals with the influence of privacy regulations (GDPR, FTC) on the network behavior of IoT devices in the United Kingdom and the United States [19]. In contrast, our work investigates the impact of location in many different countries and demonstrates that there exist other reasons for the differences.

2 Previous Works

A few tools were developed recently to help manufacturers and network administrators handle MUD standards. [8] presents a tool, MUDGEE, that creates a MUD out of a network capture (PCAP). [14] allows the characterization of IoT device network behavior and the creation and definition of appropriate MUD files using a graphic interface. Andalibi et al. [2] introduces a MUD visualizer tool for convenient viewing of MUD files. Our MUDIS tool introduces comparison and generalization features, allowing users to investigate MUD files differences.

We use the MUDIS tool to check the impact of IoT location on... In addition, to the extent of our knowledge, this is the first work that defines device location as a factor that impacts a device's network behavior. The only related work we are aware of deals with the influence of privacy regulations (GPDR, FTC) on the network behavior of IoT in the United Kingdom and the United States [19]. In contrast, our work investigates the impact of location in many different countries and demonstrates that there exist other reasons for the differences, such as cloud regions, country encryption policies, and more.

3 MUD Background

In our approach, MUD plays two roles. First, the MUD file formalizes network behavior at the flow level, enabling us to analyze it. Thus, the results and insights of MUDIS can help here and in other use cases such as IoT identification (see Section 6). Second, MUD methodology serves as a security solution and improving it is one of the basic motivations for this work.

MUD is an Internet standard [10] that aims to reduce the attack surface for IoT devices by describing their appropriate traffic patterns. Any traffic that does not comply with this description is considered malicious and can be, for example, blocked. These descriptions are provided by the IoT manufacturers in *MUD files*.

MUD files consist of Access Control Lists (ACLs), each with several Access Control Entries (ACEs). Each ACE is defined as a 5-tuple, as depicted in Figure 5

$$\begin{aligned} ACE = (&legitimate\_endpoints, protocol, source\_port, \\ &destination\_port, direction) \end{aligned} \tag{2}$$

The legitimate endpoints are the endpoints with which the IoT connects; they are commonly defined by domain name or by a range of domains[10, 5] (e.g., *.iotvendor.com), IP subnet (including *), or MAC for intra-LAN scenarios. We note that MUD [10] standardization highly recommends avoiding the use of IP addresses and uses domains instead.

The corresponding action of the ACE is typically to either "accept" or "drop". Because the MUD file specifies a white-list, the default rule is to drop traffic that does not correspond to any ACE. Throughout this paper, we write that a flow is matched by a MUD if there is an ACE in the MUD that matches the flow.

The MUD framework itself consists of several components. A *MUD manager*, also known as the *MUD controller*, is responsible for obtaining and processing the MUD information. For each IoT device, the MUD manager first obtains the MUD file from its manufacturer's *MUD server*. The MUD server's address for the IoT device is stored as a *MUD URI* in the device's firmware. This URI can be obtained by the MUD manager in a variety of ways as specified in [10]. Nevertheless, it is most commonly obtained through a dedicated option in the DHCP protocol, which the IoT device executes to connect to the network. Thus, the MUD file should be applicable to all possible locations in which the IoT device can be situated. With the MUD file at hand, the MUD manager parses the file and installs the corresponding ACL rules on a network security device, such as a firewall or AAA server, to reduce the attack surface of the device.

Manufacturers are faced with the challenging task of creating a comprehensive and representative MUD that takes into account many parameters, such as the use of third-party libraries, the OS network behavior, the entire device's operational functions, and more. To overcome these challenges, there are tools that generate MUD files from network captures [8, 14].

Another approach, is that a network security component [1] would acquired and learned the MUD file from wild-traffic using big-data information. This helps cope with the situation where IoT vendors lack the incentive or knowledge to create a MUD file.

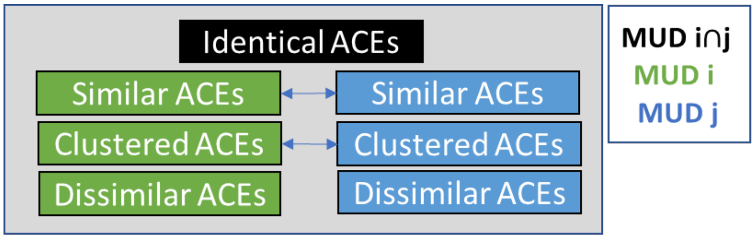


Figure 2: MUDIS comparison visualization

4 Main features

In the following subsections we present two main features of MUDIS: Comparison and Generalization

4.1 MUD Comparison

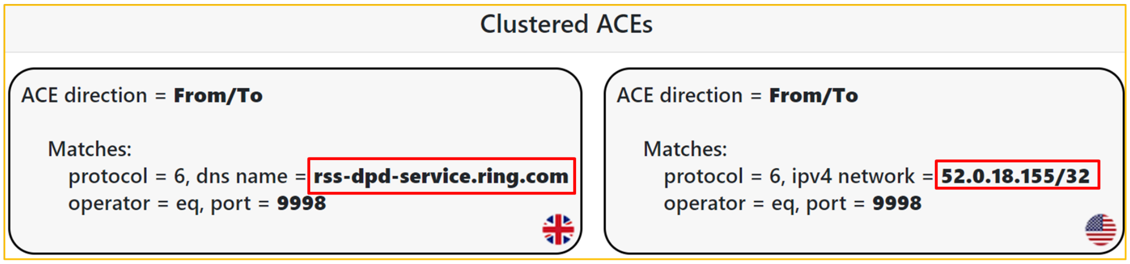
Given two MUD files, MUD_i and MUD_j, our tool compares them to find their differences and similarities. Initially, it outputs a similarity score to measure and numerically represent the similarity between the two MUDs. The similarity scale ranges from 0 to 1, where 0 means that there are no similar ACEs among the MUD files and 1 means the two MUDs are identical. We define MUD similarity as the Jaccard similarity coefficient of the two MUDs and divide the number of equal ACEs in both MUDs by their total number of ACEs. The similarity measure of two MUDs is defined formally as:

$$Similarity(MUD_i, MUD_j) = \frac{|MUD_i \cap MUD_j|}{|MUD_i \cup MUD_j|} \tag{3}$$

MUDIS then divides the ACEs of the two MUDs into four groups: identical ACEs, similar ACEs, clustered ACES, and dissimilar ones. This separation highlights valuable connections and patterns between the ACEs, enabling MUDIS users to gain meaningful insights. The algorithm uses the following four steps, where each step output is visualized in a different frame on the MUDIS visualization screen (see Figure 2):

1. **Find identical ACEs**. ACEs in which all of their fields are identical.

2. **Find similar ACEs**. MUDIS marks two ACEs of two MUDs as similar if they have similar domain names and all other fields in the ACEs are identical (port, protocol, etc.). A similar domain name is defined as follows: let the domain name be in the format subDomain-part.mainDomain.suffixTLD. The suffixTLD is the top level domain (e.g., .com, .net or .us) or a combination of top level domains (e.g.,



Clustered ACEs

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **rss-dpd-service.ring.com**
operator = eq, port = **9998**

ACE direction = **From/To**

Matches:
protocol = 6, ipv4 network = **52.0.18.155/32**
operator = eq, port = **9998**

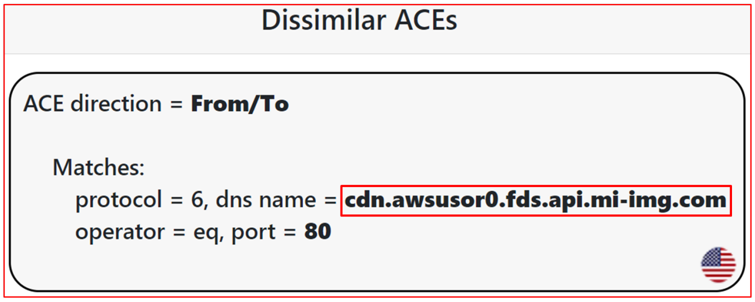
Figure 3: Ring Doorbell clustered ACEs

.com.tw). The subDomain part can be empty or include multiple sub-domains (i.e., sub1.sub2.sub3.mainDomain.suffixTLD). Two domain names are similar if and only if their mainDomain part is equal. For example, In Figure 5 we compared two different devices, the Bulb and Plug, of the same manufacture (Tp-link). MUDIS found similar ACEs that differ only in their sub-domain parts: the Bulb uses **n-devs**.tplinkcloud.com whereas the Plug uses **devs**.tplinkcloud.com.

3. **Find Clustered ACEs**. After the first two steps, if there are still unmatched ACEs remaining, we cluster the ACEs with the same traffic directions into two types of clusters: (1) ACEs with similar or equal endpoints but with different ports or protocols (2) ACEs with the same ports and protocol but with different endpoints. Note that each cluster type may contain several clustered ACEs and the same ACE can be clustered with multiple ACEs of the other MUD.

   For example, In Figure 3 we compared the Ring Doorbell device behavior in two different locations (UK and US). MUDIS managed to automatically spot a difference and clustered two ACEs that communicate with the same protocol and unique high port, but use two different endpoints.

4. **Find all dissimilar ACEs** Finally, we gather all the non-clustered ACEs into the dissimilar ACEs section. For example, in Figure 4 we compared two MUDS of the same Xiaomi device, where one of the MUDs was generated out of a PCAP with a rare action such as firmware update. MUDIS found a unique domain of Xiaomi that downloads a new firmware version over port 80.



Dissimilar ACEs

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **cdn.awsusor0.fds.api.mi-img.com**
operator = eq, port = **80**

Figure 4: The Xiaomi Bulb's unique domain when downloading a new firmware version

4.2 MUD Generalization

The goal of this feature is to create a generalized MUD that is comprehensive, tight, and secure. **Comprehensive** means that the generalized MUD should be applicable to the two MUDs presented. It must also be **tight and secure** because a MUD's main goal is to whitelist only legitimate flows of the IoT and thereby reduce the device attack surface.

The generalization algorithm has three steps:

1. **Add equal ACEs.** All the identical ACEs that appear in both MUDs are added only once to the generalized MUD.

2. **Generalize similar ACEs by generalizing similar domains.** As mentioned previously, MUDIS marks two ACEs as similar if they have similar domain names, and all other fields are identical. MUDIS use ranges in domain (e.g., *.iotvendor.com) to create a generalized ACE from two similar ACEs, as shown in Figure 5(b). MUDIS only generalizes sub-domains where the whole domain is in the control of the main domain owner i.e., the IoT manufacturer or the exact IoT service that the manufacturer uses. Moreover, to keep the generalized MUD tight and secure, MUDIS automatically identifies problematic scenarios and does not generalize ACEs with different domain suffixes (TLDs) and known cloud services that are shared across clients (e.g., *.s3.amazonaws.com). This is aligned with the IETF Operational Consideration for the use of DNS in IoT [20].

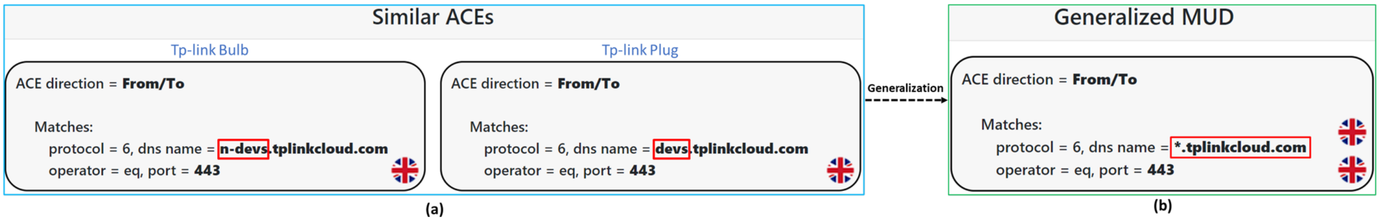

Similar ACEs
Tp-link Bulb
ACE direction = From/To
Matches:
protocol = 6, dns name = n-devs.tplinkcloud.com
operator = eq, port = 443
Tp-link Plug
ACE direction = From/To
Matches:
protocol = 6, dns name = devs.tplinkcloud.com
operator = eq, port = 443
Generalization
Generalized MUD
ACE direction = From/To
Matches:
protocol = 6, dns name = *.tplinkcloud.com
operator = eq, port = 443
(a)
(b)

Figure 5: MUDs Comparison (a) and generalization (b) of two different devices of the same manufacturer

3. **Adding dissimilar and clustered ACEs.** Following previous steps, we are left with any ACEs in both MUDs that are neither identical nor similar. These are added to the generalized MUD as-is. However, to ensure fast convergence, if some ACEs share a domain that can be safely generalized, we generalize it for all the ACEs in which it appears to support future differences that we have not yet encountered.

A naive generalization algorithm that simply unifies all available MUDs, would also be both comprehensive and tight. However, MUDIS generalization algorithm demonstrates superior performance compared to the naıve algorithm in terms of converging velocity and ACEs cardinality [3]. The generalized MUD is also more explainable and easier to implement in a firewall, due to the reduced amount of rules.

5 MUDIS usage - device's location impact analysis

In this section we will closely explore how MUDIS has helped us with the complicated task of examine (our device location oriented dataset) how a device's location can influence its network behavior and how we have used the generalization feature that was presented in the previous section to generalize different MUDs that were originated from different geolocations.

Our dataset consists of network traffic data (pcap files) captured from the router in our lab, and log files from Ren et al. [19]. Our captures comprise 31 IoT devices (e.g., plugs, cameras, bulbs, and so on) that are physically or virtually located in up to 14 countries using VPN [16], and use all of their device functionalities. We chose the countries in which the devices were activated according to those countries available for the registration and provisioning process in the IoT user's application [1] (see sub-section 5.3 for more details).

We found that the device network behavior in most cases does not depend on the physical location (i.e., IP of the device as seen in the VPN) but rather on the device's logical location, which is the country chosen in the device provisioning process. Next, we generated MUD files from the pcaps using MUDGEE [8]. The resulting MUD files per country and the full list of tested devices are available at [12][2].

To compare and find the similarities between two MUDs, we used **MUDIS similarity measure** that was explained earlier.

Let MUD_i^d be the MUD of device d at location i, the **similarity measure** of two MUDs for the same device d, at location i and location j is defined formally as:

$$Similarity_d(MUD_i^d, MUD_j^d) = \frac{|MUD_i^d \cap MUD_j^d|}{|MUD_i^d \cup MUD_j^d|} \tag{4}$$

Figure 6 shows the cumulative distribution function (CDF) of MUD similarity values for the devices in our dataset, and compares their resulting MUD files for different locations. It is clear that device location has a significant impact on the MUD, since 80% of the MUD comparisons show similarity measure lowers than ~ 0.7 .

[1] The IoT application is the user's application that activate the IoT, and it is commonly installed on the user's mobile device

[2] In several cases such as cameras, the devices also use peer-to-peer protocols such as STUN [18] to allow client connections. We omitted the ACEs of peer-to-peer flows that would show a synthetic difference between MUDs that originated in client parameters (e.g., client device's IP/MAC).

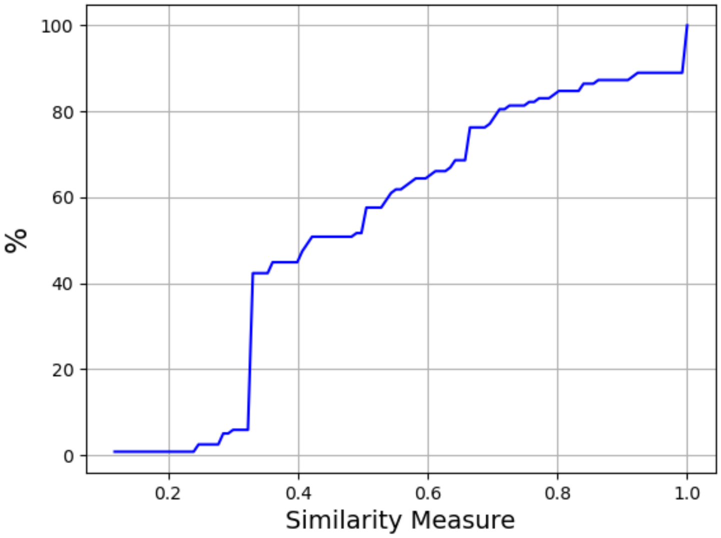


Figure 6: Cumulative Distribution Function (CDF) of MUD files similarity scores for all the devices in the dataset. Each similarity score is calculated by comparing two different locations MUD files of a device. Each device was captured in up to 14 locations.

In Figure 7, we take a deep dive and focus on an individual device, investigate its MUD similarity scores as a function of the geographical location. Figure 7 shows the MUD similarity heat-map of the Yi camera MUD files as measured in ten countries. We ordered the countries according to region. As can be observed, locations further away from each other (cross-regions) have lower MUD similarity values.

Throughout our experiments, we observed that some device functionalities were not supported in all locations. For example, the Xiaomi camera face recognition features were supported only in the Chinese region. The reasons range from local regulations to manufacturer marketing strategies. It is common that a manufacturer creates different versions of a product, with each version having variants according to the regions in which it is sold, (e.g., [22]).

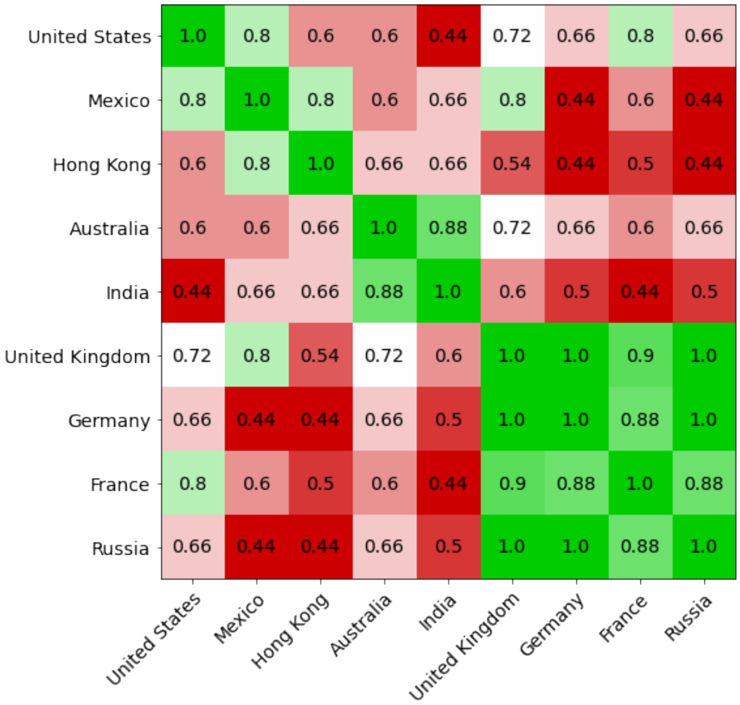


Figure 7: Heat map of similarity measure for the Yi camera, across ten different logical locations. The heat-map highlights that cross-region locations have lower similarity scores.

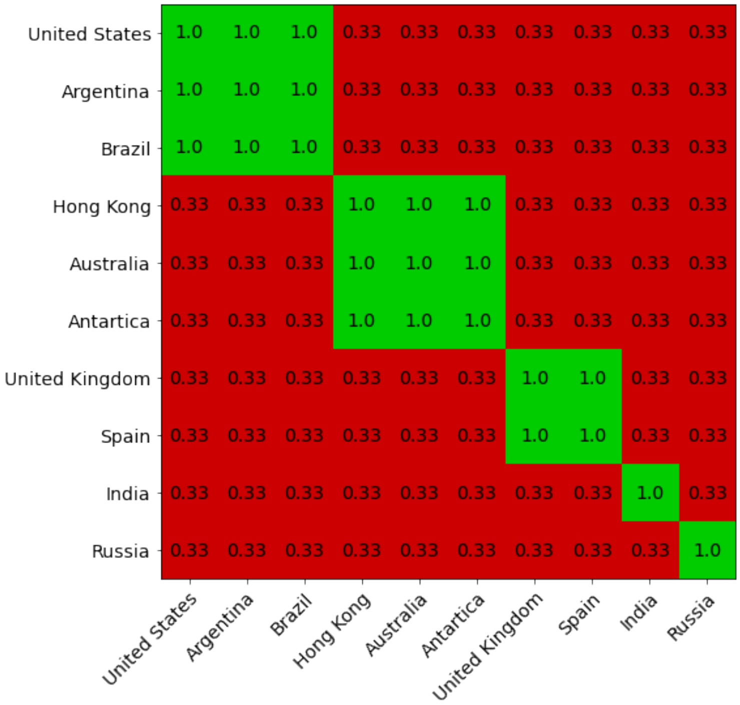


Figure 8: Heat map of similarity measure for the Xiaomi light bulb. The heat-map clearly highlights that cross-region locations have lower similarity scores.

5.1 MUDIS - MUD location Comparison

MUDIS comparison feature helps us to compare the different MUD files that was originated for the same device but in different location and gain meaningful insights over those differences and their effect on the devices network behaviour.

By comparing different locations MUD files from our broad dataset of different devices, we observed that the most common changes in ACEs involve the domain names of the allowed endpoints. For 80% of the devices in our datasets, there are differences in the domains that appear in the sub-domain. For example, the Samsung SmartThings Hub (see Figure 9a) works with two different domains in the UK and US: **dc-eu01-euwest1**.connect.smartthing.com and **dc-na04-useast2**.connect.smartthing.com, respectively. Nonetheless, 9% of the devices in the dataset exhibited a difference in the top level domain (TLD). For example, the Yi camera communicates with two different TLDs in Hong Kong and Germany: api.xiaoyi**.com.tw** and api.eu.xiaoyi**.com**, respectively.

We assume that the usage of a few domain identifiers allows the manufacturer to support different features and policies based on the logical location of the device, which was chosen by the user in registration process. Note that the manufacturer can have physical location-based decisions made by using a standard DNS server that is capable of connecting a single domain to different servers, according to the geo-locations; but, in this case, the user would not be able to choose a different logical location.

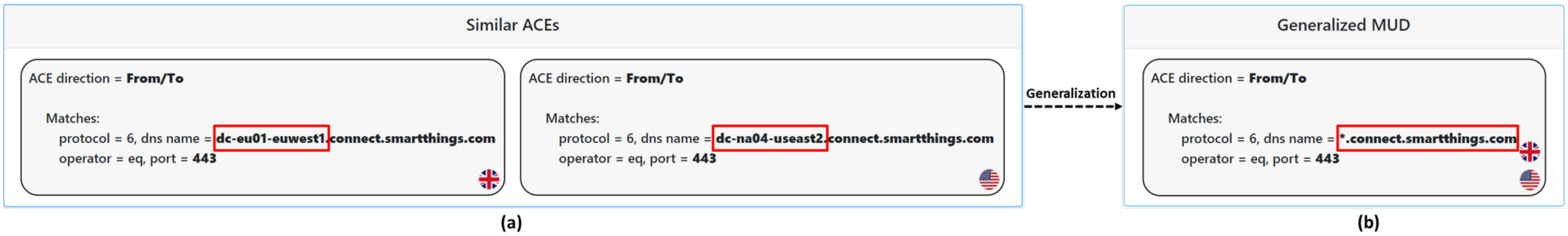
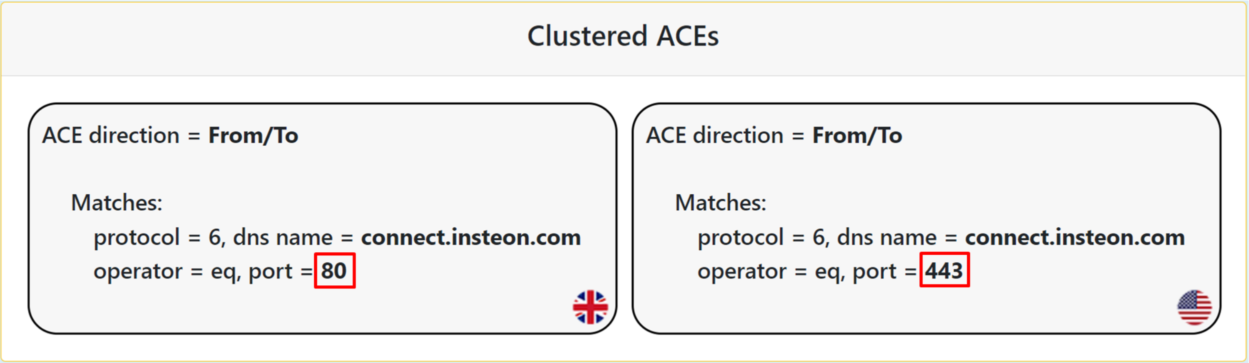

Similar ACEs
ACE direction = From/To
Matches:
protocol = 6, dns name = dc-eu01-euwest1.connect.smartthings.com
operator = eq, port = 443
ACE direction = From/To
Matches:
protocol = 6, dns name = dc-na04-useast2.connect.smartthings.com
operator = eq, port = 443
(a)
Generalization
Generalized MUD
ACE direction = From/To
Matches:
protocol = 6, dns name = *.connect.smartthings.com
operator = eq, port = 443
(b)

Figure 9: Two similar ACEs from the MUDs of SmartThings hub in two different locations: US and UK. MUDIS created a generalized ACE in which the endpoint is $*$*.connect.smartthings.com*, all other parameters remain the same.



Clustered ACEs

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **connect.insteon.com**
operator = eq, port = **80**

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **connect.insteon.com**
operator = eq, port = **443**

Figure 10: The Insteon Hub works with two different ports, depending on the device location (US and UL)

Using MUDIS ACEs clustering method, We also found that 9% of the devices in the dataset use different ports and protocols at different device locations. For example as shown in Figure 10, MUDIS clusters two ACEs in which the Insteon Hub device exhibits similar behavior using HTTP (unencrypted) or HTTPS (encrypted), depending on its location.

Another example is shown in Table 1 that presents the case of the Xiaomi camera, where the location affects not only which port and protocols are used but also the IP resolution and encryption methods that are used by the device.

	China	Israel
Domain Names	Fixed IP	sg.ots.io.mi.com
Port	HTTP (80)	HTTPS (443)
IP Resolution	HTTP Request	DNS
Encryption	Self-signature	Standard TLS

Table 1: Comparison of Xiaomi Camera network behavior (domains, ports, and protocols) in two different logical locations.

5.2 MUDIS - MUD location Generalization

In this section we are using MUDIS generalization feature to create a generalized MUD. The generalized MUD should be comprehensive (applicable to each of the device locations), tight, and secure.

The basic generalization algorithm works on two MUDs at a time. We can use the algorithm to generalize n MUD files by using an iterative process, where we take the generalization algorithm output from iteration $n-1$ and process it with the $n-th$ MUD file. We aim to create a generalized and comprehensive MUD using a minimal number of iterations. We show how our algorithm converges more quickly than the naive algorithm. Namely, adding more MUDs from more locations will not change the generalized MUD.

As explained in the MUDIS generalization section, The naive generalization algorithm would be comprehensive for all of the device location and it will also generate a tight and secure MUD. However, when applying MUDIS generalization algorithm on MUDs that were originated from different locations, MUDIS was able to create a faster convergence process with a significant lower rules cardinality in comparison to the naive one.

In Figure 11 we present a convergence analysis of MUD files for the Yi Camera, while using

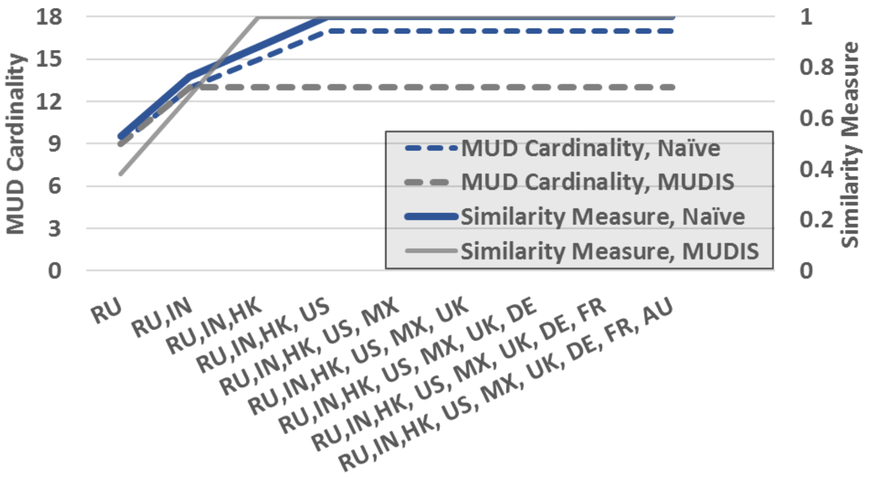


Figure 11: Performance comparison of generalized MUD and naïve unifying MUD files of the Yi Camera. Each point on the x-axis corresponds to the unified or generalized MUD at the specified locations. To evaluate a similarity score, each MUD is compared to the correlated global MUD, consisting of all available locations.

MUDs from 10 different locations. We order the locations, such that we first pick locations from different regions, aiming to achieve fast convergence. As shown in Figure 7, cross-regions locations have lower similarity scores and thus add more information to the generalized MUD. We compared our MUDIS generalization algorithm with a naive algorithm that simply unifies all available MUDs. Each point on the x-axis corresponds to the MUD generalization at the specified locations. For example, the RU, IN point corresponds to the generalized MUD after generalization of the RU (Russia) and IN (India) MUDs. For each generalized MUD, we output its cardinality (number of ACEs) and its similarity score in comparison to the correlated global MUD; this global MUD is defined as the output of the algorithms (naive, or MUDIS) after processing all available locations. Our generalization MUD algorithm shows superior performance compared to the naïve algorithm both in cardinality and convergence time.

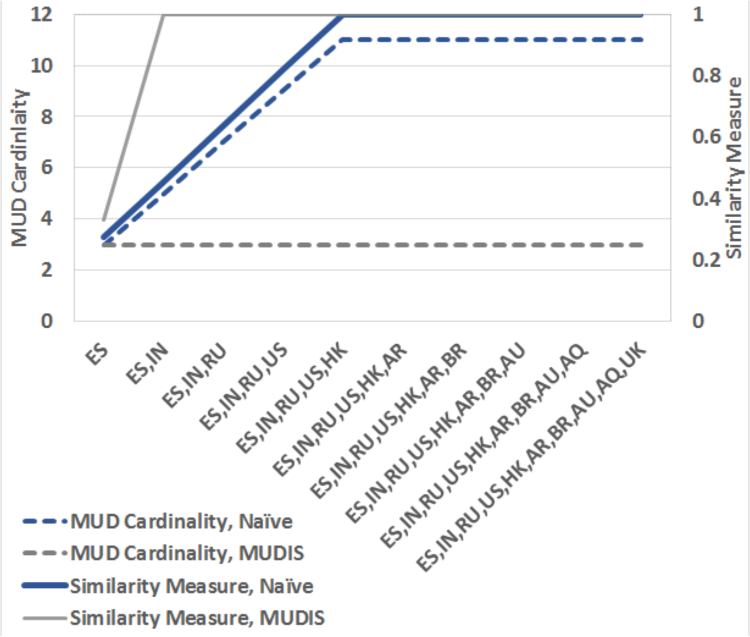


Figure 12: Performance comparison of generalized MUD and naïve unifying MUD files of the Xiaomi light bulb.

5.3 Device Geo-IP vs User Location Decision

To further explore and measure the impact of the device IP geolocation on its network behavior, we made the following measurement: we used 26 different devices network captures that were captured with the same user-account [3] under two scenarios: (1) using local ISP (preserving the network geo-IP) (2) using a VPN to a different country than the chosen country in the registration process (actively changing the geo-IP of the device). This measurement was repeated twice in two geo-locations (UK and US), for all devices, Then, we formalized their network behavior by generating their MUDs (using the 52 captures we had) and comparing them by using MUDIS similarity measure. Figure 13 presents the two graphs side by side; the red graph is the CDF of the similarity measure for different geolocations that was set manually by a user input at the registration process and the blue graph is the measurement we just presented using a different geolocations that is determined automatically by the device IP (given by the VPN connection).

As depicted in the figure, 80% of the device Geo-IP MUD comparisons (VPN, blue line) show similarity measure **higher** than 0.8 (and rising fast) in contrast to the user location decision (Registration, red line) that shows the opposite results where 80% of the MUD comparisons show similarity measure **lower** than 0.7. The explanation for this behavior is related to to the need of manufactures that regulatory obligated to support user location decisions, which forces the manufactures to use different domains and no other mechanized means such as Gro-IP, DNS, and client-subnet (eDNS). This allows the manufacture to comply with privacy policies and other regulations according to the user location decision and not by identifying its real location (using the device Geo-IP)

These measurements show some clear results and support our initial finding that the device network behavior in most cases does not depend on the physical location but rather on the device's logical location, which is the country chosen in the device provisioning process.

[3] User-accounts for all devices were created in the same country in which they were deployed.

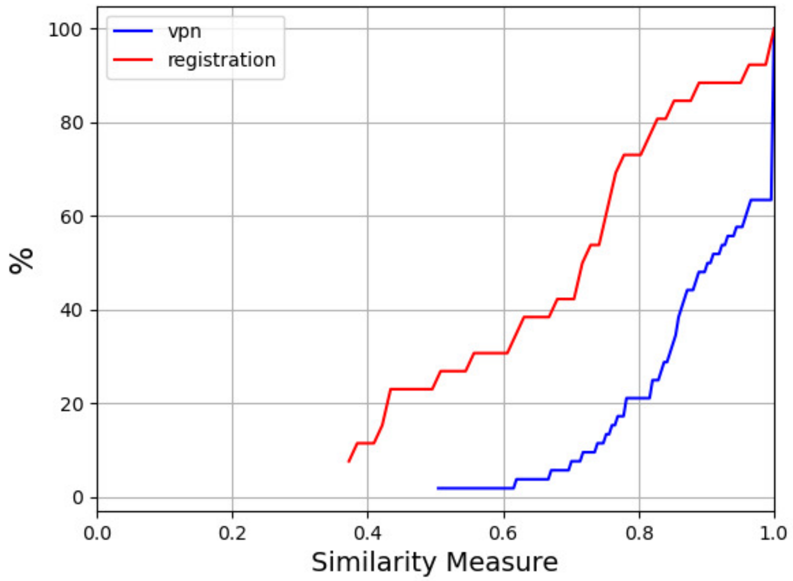


Figure 13: Cumulative Distribution Function (CDF) of MUD files similarity scores for 26 devices from our dataset under two different geolocation changing methods. The red graph is the CDF of the similarity measure for different geolocations that was set manually by a user input at the registration process and the blue graph is the measurement of different geolocation that is determined automatically by the device IP (given by the VPN connection)

6 Applications of MUD Generalization

In this section we describe the impact of the generalized MUD on the MUD and IoT identification.

Generalized domains require that the MUD manager know how to process them and then insert them as firewall rules. We note that this is aligned with the RFC [10] and the configuration of some routers [5].

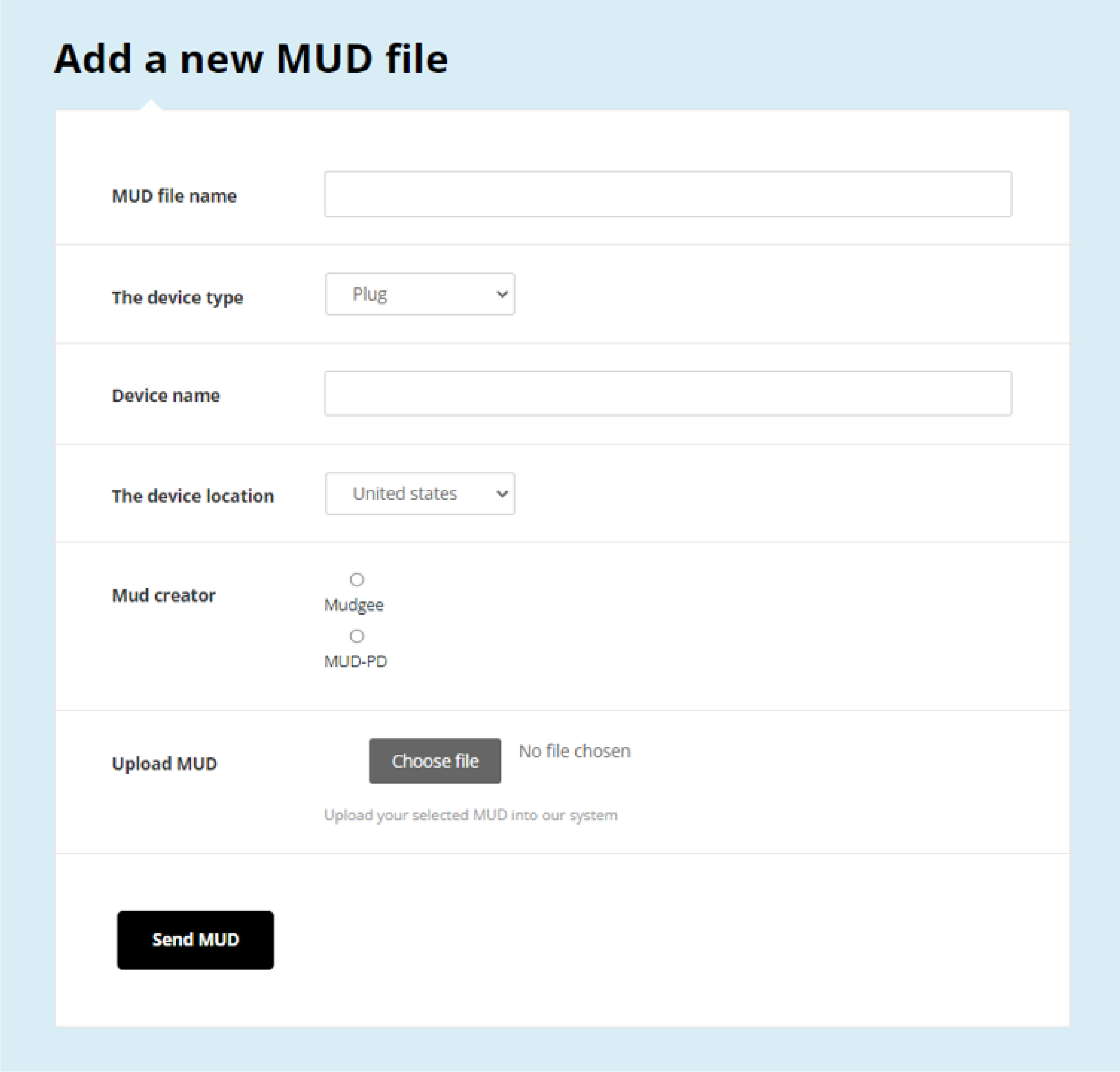
Generalized domains are also important when building applications for IoT identification, since many of them [6, 11, 9, 7, 17] also rely on the domain names. For example, [17] uses domain information based on DNS network traffic that originated in one location (USA). However, many of the devices in their dataset (including Echo Dot, FireTV, SmartThings Hub, TP-Link Bulb, TP-Link Plug, and more) also appear in our dataset. These devices showed major differences across locations, thus harming the accuracy of the IoT identification algorithm. We suggest an approach such as MUDIS to use a generalized domain for more accurate device identification across locations.

7 Conclusions

This work introduces MUDIS, a tool for MUD inspection, comparison, and generalization. Using MUDIS we demonstrate that device location has an impact on the network behavior of IoT devices and their corresponding MUD values. Additionally, we present an efficient generalization algorithm to create a single MUD that can work in all locations. We strongly encourage the use of MUDIS to achieve better and deeper understanding over the network behavior of IoT devices.

Appendices

A MUDIS screens



Add a new MUD file

MUD file name

The device type: Plug

Device name

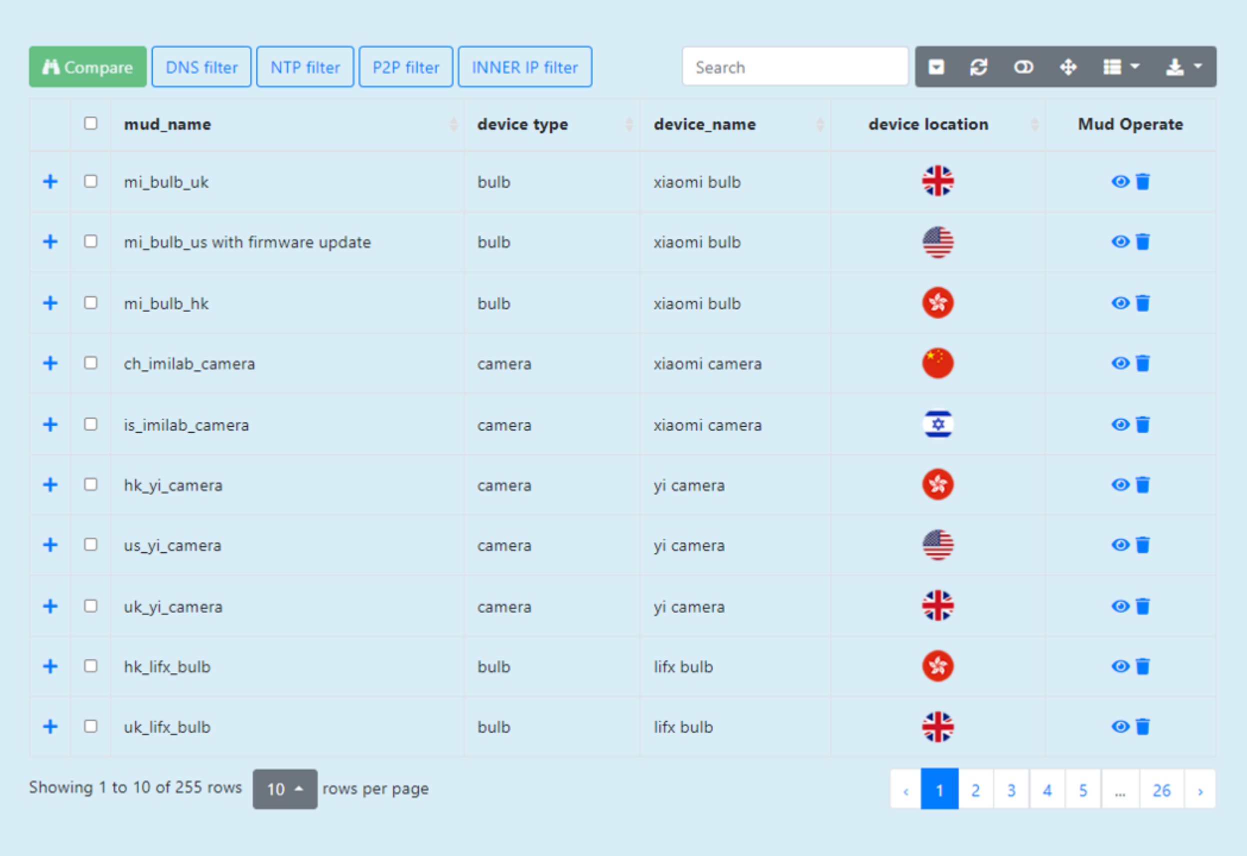
The device location: United states

Mud creator
○ Mudgee
○ MUD-PD

Upload MUD: No file chosen

Upload your selected MUD into our system

Figure 14: Add a new MUD screen - gives users the ability to add new generated MUDs into the system for further investigation

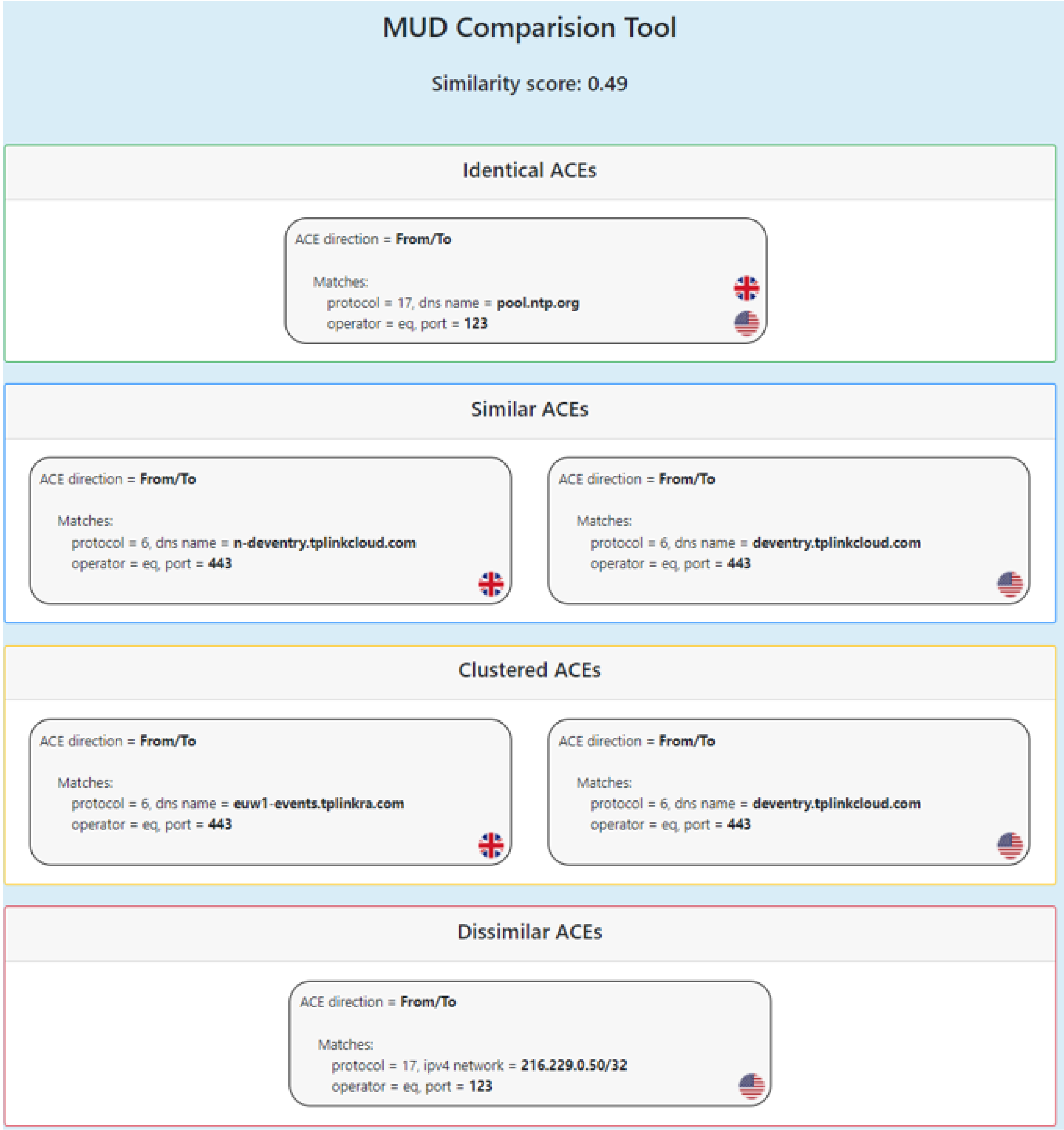


	□	mud_name	device type	device_name	device location	Mud Operate
+	□	mi_bulb_uk	bulb	xiaomi bulb	🇬🇧	👁 🗑
+	□	mi_bulb_us with firmware update	bulb	xiaomi bulb	🇺🇸	👁 🗑
+	□	mi_bulb_hk	bulb	xiaomi bulb	🇭🇰	👁 🗑
+	□	ch_imilab_camera	camera	xiaomi camera	🇨🇳	👁 🗑
+	□	is_imilab_camera	camera	xiaomi camera	🇮🇱	👁 🗑
+	□	hk_yi_camera	camera	yi camera	🇭🇰	👁 🗑
+	□	us_yi_camera	camera	yi camera	🇺🇸	👁 🗑
+	□	uk_yi_camera	camera	yi camera	🇬🇧	👁 🗑
+	□	hk_lifx_bulb	bulb	lifx bulb	🇭🇰	👁 🗑
+	□	uk_lifx_bulb	bulb	lifx bulb	🇬🇧	👁 🗑

Showing 1 to 10 of 255 rows 10 rows per page

‹ 1 2 3 4 5 ... 26 ›

Figure 15: MUDIS home screen - gives users the ability to choose two MUDs and to compare, generalize and filter ACEs using the system



MUD Comparision Tool

Similarity score: 0.49

Identical ACEs

ACE direction = **From/To**

Matches:
protocol = 17, dns name = **pool.ntp.org**
operator = eq, port = **123**
🇬🇧 🇺🇸

Similar ACEs

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **n-deventry.tplinkcloud.com**
operator = eq, port = **443**
🇬🇧

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **deventry.tplinkcloud.com**
operator = eq, port = **443**
🇺🇸

Clustered ACEs

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **euw1-events.tplinkra.com**
operator = eq, port = **443**
🇬🇧

ACE direction = **From/To**

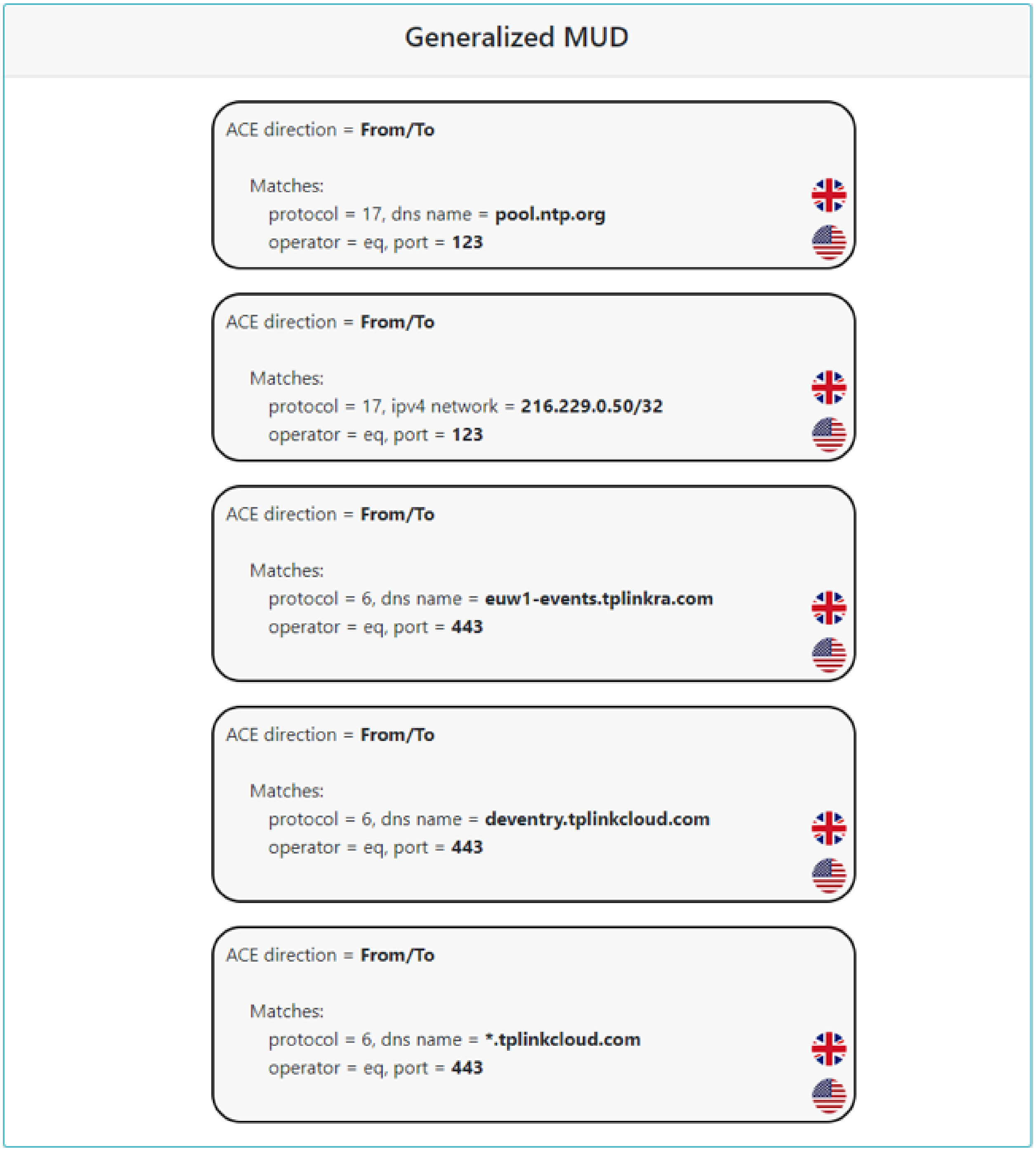
Matches:
protocol = 6, dns name = **deventry.tplinkcloud.com**
operator = eq, port = **443**
🇺🇸

Dissimilar ACEs

ACE direction = **From/To**

Matches:
protocol = 17, ipv4 network = **216.229.0.50/32**
operator = eq, port = **123**
🇺🇸

Figure 16: MUDIS comparison screen - graphically visualize the similarity score together with the identical, similar, clustered and dissimilar rules between both MUDs.



Generalized MUD

ACE direction = **From/To**

Matches:
protocol = 17, dns name = **pool.ntp.org**
operator = eq, port = **123**

ACE direction = **From/To**

Matches:
protocol = 17, ipv4 network = **216.229.0.50/32**
operator = eq, port = **123**

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **euw1-events.tplinkra.com**
operator = eq, port = **443**

ACE direction = **From/To**

Matches:
protocol = 6, dns name = **deventry.tplinkcloud.com**
operator = eq, port = **443**

ACE direction = **From/To**

Matches:
protocol = 6, dns name = ***.tplinkcloud.com**
operator = eq, port = **443**

Figure 17: MUDIS generalization screen - graphically visualize the generalized MUD that was produced by generalizing both MUDs.

References

[1] Y. Afek, A. Bremler-Barr, D. Hay, R. Goldschmidt, L. Shafir, G. Avraham, and A. Shalev, “Nfv-based iot security for home networks using mud,” in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium.* IEEE, 2020, pp. 1–9.

[2] V. Andalibi, E. Lear, D. Kim, and L. J. Camp, “On the analysis of mud-files’ interactions, conflicts, and configuration requirements before deployment,” *arXiv preprint arXiv:2107.06372*, 2021.

[3] A. Bremler-Barr, B. Meyuhas, and R. Shister, “One mud to rule them all: Iot location impact,” in *NOMS 2022 - short paper*, apr 2022, accepted for publication.

[4] A. Bremler-Barr, R. Shister, and B. Meyuhas, “Mudis: Mud inspection system,” in *NOMS 2022 - Demo paper*, apr 2022, accepted for publication.

[5] Cisco, Jan 2018. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/xe-3s/sec-data-acl-xe-3s-book/sec-cfg-fqdn-acl.html

[6] H. Guo and J. Heidemann, “Detecting iot devices in the internet,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2323–2336, 2020.

[7] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal, “Iot device identification via network-flow based fingerprinting and learning,” in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE).* IEEE, 2019, pp. 103–111.

[8] A. Hamza, D. Ranathunga, H. Gharakheili, M. Roughan, and V. Sivaraman, “Clear as mud: Generating, validating and applying iot behavioral profiles,” in *Workshop on IoT Security and Privacy.* USA: Association for Computing, 2018, pp. 8–14.

[9] G. Hu and K. Fukuda, “Toward detecting iot device traffic in transit networks,” in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 525–530.

[10] E. Lear, R. Droms, and D. Romascanu, “Manufacturer Usage Description Specification,” RFC 8520, Mar. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8520.txt

[11] M. H. Mazhar and Z. Shafiq, “Characterizing smart home iot traffic in the wild,” in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020, pp. 203–215.

[12] B. Meyuhas, Shister, “Mud files dataset in different locations.” 2021. [Online]. Available: https://github.com/barmey/IoT_mud_files_locations

[13] NIST, “National institute of standards and technology,” Sep 2021. [Online]. Available: https://www.nist.gov/

[14] N. I. o. S. NIST and Technology, “Methodology for characterizing network behavior of internet of things devices.” [Online]. Available: https://github.com/usnistgov/MUD-PD

[15] ——, “Methodology for characterizing network behavior of internet of things devices,” Apr 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04012020-draft.pdf

[16] NordVPN, “Leading vpn service. online security starts with a click.” 2021. [Online]. Available: https://nordvpn.com/

[17] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis, “Iotfinder: Efficient large-scale identification of iot devices via passive dns traffic analysis,” in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 474–489.

[18] M. Petit-Huguenin, G. Salgueiro, J. Rosenberg, D. Wing, R. Mahy, and P. Matthews,

"Session Traversal Utilities for NAT (STUN)," RFC 8489, Feb. 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc8489.txt

[19] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach," in *IMC Conference*. New York, NY, USA: ACM, 2019, pp. 267–279.

[20] M. Richardson and W. Pan, "Operational Considerations for use of DNS in IoT devices," Internet Engineering Task Force, Internet-Draft draft-ietf-opsawg-mud-iot-dns-considerations-02, Jul. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-mud-iot-dns-considerations-02

[21] R. Shister, B. Meyuhas, and A. Bremler-Barr, "Mudis - mud inspection system." [Online]. Available: https://github.com/ransh93/MUDIS

[22] Wikipedia contributors, "Miui — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=MIUI&oldid=1063553442, 2022, [Online; accessed 14-January-2022].

תקציר

ניתוח ההתנהגות התקשורתית של רכיבי IOT (אינטרנט של הדברים), אשר מורכבת מפרוטוקולים, דומיינים ופורטים בהם מתקשרים הרכיבים, הוא אתגר מרכזי ומשמעותי בהקשרי אבטחת הרכיבים והזיהוי שלהם. פתרונות אשר מנתחים ופועלים בהקשרים הללו, חייבים להיות מסוגלים ללמוד ממה מורכבת תעבורה רשתית המוגדרת כ-״תקינה ולגיטימית״ ועל ידי כך לייצר סט של חוקים ומאפיינים רשתיים נוספים אשר מאפשרים שימוש של תעבורה לגיטימית בלבד במקביל ליכולת לזהות סוגים שונים ומגוונים של רכיבים.

MUD הינה סכמה אשר נחקרת ומפותחת על ידי ארגון ה – IETF, שמטרתה העיקרית היא להוות סט של חוקים המיושמים בתצורה של white-list ומשמשים בתור הגדרה פורמלית וקשיחה של ההתנהגות התקשורתית המאושרת על ידי היצרן של רכיב ה - IOT. קובץ ה - MUD יכול למעשה לשמש כמעין סוג של חומת אש אשר מאפשרת מעבר בלעדי של תקשורת חוקית בלבד (ומניעה של כל תקשורת אחרת, המוגדרת לא חוקית לפי סט החוקים שהוגדרו).

תזה זו מציגה את MUDIS - Mud Inspection System, מערכת אשר מנתחת את ההתנהגות התקשורתית של רכיבי IOT בהתבססות על ההגדרה הפורמלית שלהם (סט החוקים) בקובץ ה MUD המשויך אליהם. אנו משתמשים ב MUDIS על מנת לבחון מספר מקרים אשר מדגימים מדוע תהליך הלמידה של מהי התנהגות תקשורתית ״תקינה ולגיטימית״ היא קשה ומאתגרת יותר מהמצופה. לדוגמא: (1) איך אותו רכיב IOT עם אותה הקשחה, מייצר תקשורת רשתית שונה המתבטאת בחיבור לדומיינים, פרוטוקולים ופורטים שונים כתלות במיקום הגאוגרפי של הרכיב. (2) ההשלכה המהותית על התקשרות של הרכיבים כתוצאה מעדכוני הקשחה של הרכיבים. (3) הקורלציה של ההתנהגות התקשורתית בין רכיבים שונים של אותו היצרן, ועוד.

MUDIS מנתחת שני קבצי MUD וכתוצאה מקבצת יחדיו ומציגה בצורה ויזואלית את כל החוקים הזהים, הדומים והשונים. בנוסף, היא ממחשבת את מדד הדמיון בין אותם ה-MUDים, מדד אשר מגדיר את המרחק בין ההתנהגות התקשורתית שלהם. לבסוף, המערכת מכלילה את שני קבצי ה – MUD (במידה ואפשר), כך שה - MUD המוכלל ״מכסה״ את כלל החוקים שיש לאפשר על מנת לשמור ולקיים את התקשורת התקינה של שני הרכיבים המוגדרים על ידי קבצי ה – MUD שהוכנסו למערכת.

אנו מדגימים את MUDIS ואת יכולות ההשוואה וההכללה שלה, על ידי ניתוח של קבצי MUD שונים אשר נוצרו ממספר רב של רכיבים ומדינות, השוואה של החוקים שלהם על מנת ללמוד ולאפיין את ההשלכות של מיקום הרכיב על ההתנהגות התקשורתית שלהם ולבסוף הכללה שלהם לכדי קובץ MUD מוכלל אשר מאפשר לרכיבים תקשורת תקינה בכלל המדינות המאופשרות על ידי הרכיבים.

הקוד של MUDIS וכל המידע שהוקלט מרכיבי ה IOT השונים, נמצאים כקוד פתוח ומאגר חופשי עבור חוקרים ומפתחי הרכיבים ובכך מאפשרים לכל העוסקים בנושא להסיק מסקנות משמעותיות על ההתנהגות התקשורתית של הרכיבים שלהם בצורה קלה, מהירה, ויזואלית, מעמיקה ומדויקת.

המחקר המלא ותוצאותיו פורסמו והוצגו בכנס NOMS 2022 אשר התקיים בבודפשט, הונגריה [3], [4].

עבודה זו בוצעה בהדרכתה של **פרופ׳ ענת ברמלר-בר** מבי״ס אפי ארזי למדעי המחשב, אוניברסיטת רייכמן.

אוניברסיטת רייכמן
בית ספר אפי ארזי
למדעי המחשב

אוניברסיטת רייכמן

בית-ספר אפי ארזי למדעי המחשב

התכנית לתואר שני (M.Sc.) - מסלול מחקרי

MUDIS:
מערכת לניתוח ואבחון של MUDים

מאת

רן שיסטר

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc.
במסלול המחקרי בבית ספר אפי ארזי למדעי המחשב, אוניברסיטת רייכמן

מרץ 2022