One MUD to Rule Them All: IoT Location Impact

Anat Bremler-Barr
Reichman University (IDC)
Israel

Bar Meyuhas
Reichman University (IDC)
Israel

Ran Shister
Reichman University (IDC)
Israel

***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.**

We demonstrate that learning what is normal behavior for an IoT device is more challenging than expected. In many cases, 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.

We analyze and explain use-cases in which the location impacts device behavior. Then, we present a technique to generalize MUD files. By processing MUD files that originate in different locations, we can generalize and create a comprehensive MUD file that is applicable for all locations. To conduct the research, we created MUDIS, a MUD Inspection System tool, that compares and generalizes MUD files. Our open-source MUDIS tool and dataset are available online to researchers and IoT manufacturers, allowing anyone to visualize, compare, and generalize MUD files.

I. Introduction

The field of IoT is highly diverse, with many vendors but no leading paradigm for how devices should be designed, secured, and identified in the network. This work 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. [1] 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 [1]. In contrast, our work investigates the impact of location in many different countries and demonstrates that there exist other reasons for the differences.

978-1-6654-0601-7/22$31.00 © 2022 IEEE

This phenomenon has a direct impact on IoT security, including the Manufacturer Usage Description (MUD). The MUD is an IETF standard [2] that enables us to formalize the legitimate network behavior of IoT devices. In this way, the MUD file serves as a sort of Access Control List (ACL) or network firewall to verify that the device is not being compromised. The MUD file is fetched by the IoT device using DHCP or LLDP, and thus a single MUD file is required for each firmware version, regardless of where the device is located geographically. Nevertheless, our work shows that in 90% percent of tested devices, the same device with the same firmware demonstrates different network behavior in different geographic locations. We found that in many cases the device network behavior depends on its logical geographic location, chosen by the user in the account registration process and not by the current physical location of the device (i.e., using geo-IP).

The MUD can be learned based on information captured from device network traffic using a MUD generator tool such as MUDGEE [3]. To be comprehensive, the capture should consist of all potential network behaviors. The National Institute of Standards and Technology (NIST) [4] defined a list of environmental variables that can influence the network behavior of an IoT device [5] (i.e., internet connection, DNS blocking, human interaction); but, they did not address device location as a factor. We argue that location is another important environmental variable. Because the MUD framework is a white-list technique, learning the MUD file in one location and applying it in another location can cause the device to malfunction.

We calculate a similarity score that measures how similar the two MUD files are. We show that the similarity measure tends to be higher when we compare MUD files captured from the same device at different locations, provided these two locations are closer to each other geographically.

We also created an algorithm to create a generalized MUD that can white-list the network behavior of two locations, i.e., cover both MUDs. A naive algorithm would simply unify both MUDs. However, our generalized MUD reduces the number of rules, which decreases implementation costs in the firewall, and increases the explainability of the resulting MUD. We observed that in most cases, the IoT vendor uses different sub-domains for different locations. Our algorithm takes advantage of this fact and uses ranges in the domain field (e.g., $*.iotvendor.com$) to create a generalized access control entry (ACE) [6] from two similar ACEs [1] (see an example in Figure 3). Using ranges in the generalized MUD, we receive a comprehensive, tight and secure MUD. We also suggest a generalization algorithm for n MUD files, and show its convergence after processing fewer MUDs than the naive algorithm.

Motivated by the need to compare and generalize different MUDs from different locations, we present a novel tool called MUD Inspection System (MUDIS). Our open-source MUDIS tool and dataset are available online for researchers and IoT manufacturers [7], [8]. A few tools have been developed recently to help manufacturers and network administrators analyze devices network behavior and handle MUD standards [3], [9]–[12]. MUDIS is a generic tool for comparing and analyzing MUDs. It can also be useful in other use cases, for example, to analyze the differences in MUDs for different firmware versions. Moreover, because a MUD file is a formalization of the network behavior, the generalized MUD can also be used to extract generalized features for IoT identification [13]–[17].

II. 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. 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 [2] 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 3:

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

The legitimate endpoints are the destinations with which the IoT connects. These are commonly defined by domain name or by a range of domains [2], [18] (e.g., *.iotvendor.com), IP subnet (including *), or MAC for intra-LAN scenarios. We note that MUD [2] standardization highly recommends avoiding the use of IP addresses and proposes using 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. The MUD manager is a component that parses the MUD 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 [3], [10]. In another approach, a network security component [19] will acquire and learn 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.

III. Device Location Impact Analysis

Our dataset consists of network traffic data (pcap files) captured from the router in our lab, and log files from Ren et al. [1]. 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 [20], 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.

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 (see full technical report for more details [21]). Next, we generated MUD files from the pcaps using MUDGEE [3]. The resulting MUD files per country and the full list of tested devices are available at [8][2].

To compare and find the similarities between two MUDs, we define **MUD similarity** as the Jaccard similarity coefficient of the two MUDs, and divide the number of equal ACEs in two MUDs by the total number of ACEs in both MUDs.

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:

$$Similarity_d(MUD_i^d, MUD_j^d) = \frac{|MUD_i^d \cap MUD_j^d|}{|MUD_i^d \cup MUD_j^d|} \tag{2}$$

Figure 1 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] An ACL is a user-ordered set of rules that is used to filter traffic on a networking device. Each rule is represented by an Access Control Entry (ACE).

[2] In several cases such as cameras, the devices also use peer-to-peer protocols such as STUN [22] 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).

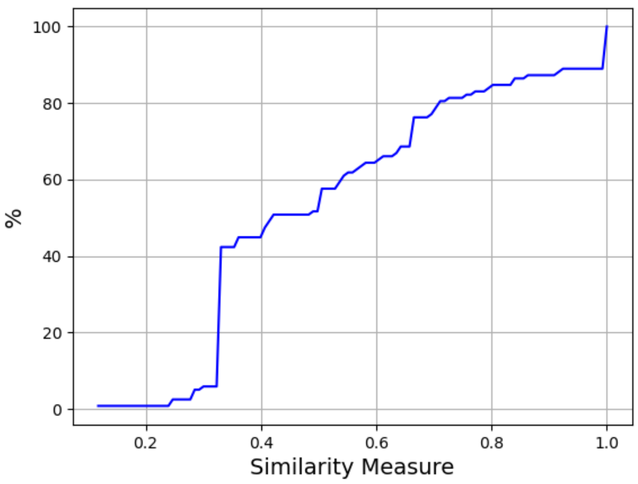


Figure 1: Cumulative Distribution Function (CDF) of MUD files similarity scores for all the devices in the dataset. Each similarity score is calculated by comparing two different locations MUD files of a device. Each device was captured in up to 14 locations.

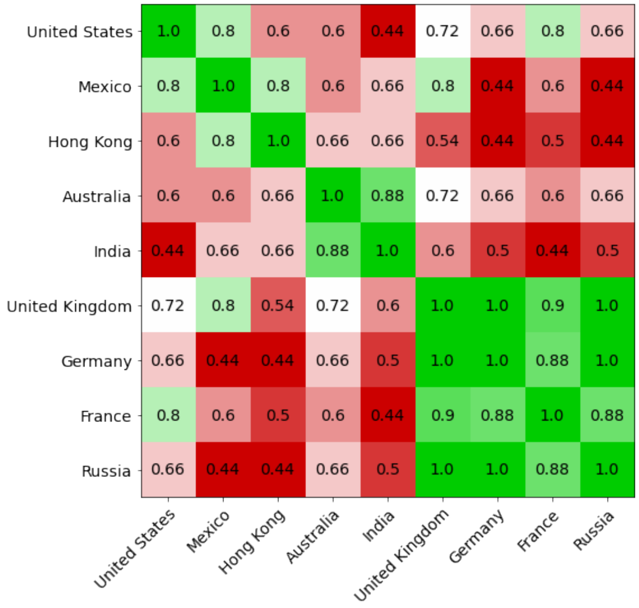


Figure 2: Heat map of similarity measure for the Yi camera, across ten different logical locations. The heat-map highlights that cross-region locations have lower similarity scores.

In Figure 2, we take a deep dive and focus on an individual device, investigate its MUD similarity scores as a function of the geographical location. Figure 2 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., [23]).

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

IV. MUD Comparison

In this chapter, we examine the difference between two MUD files. 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 subdomain. For example, the Samsung SmartThings Hub (see Figure 3a) 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.

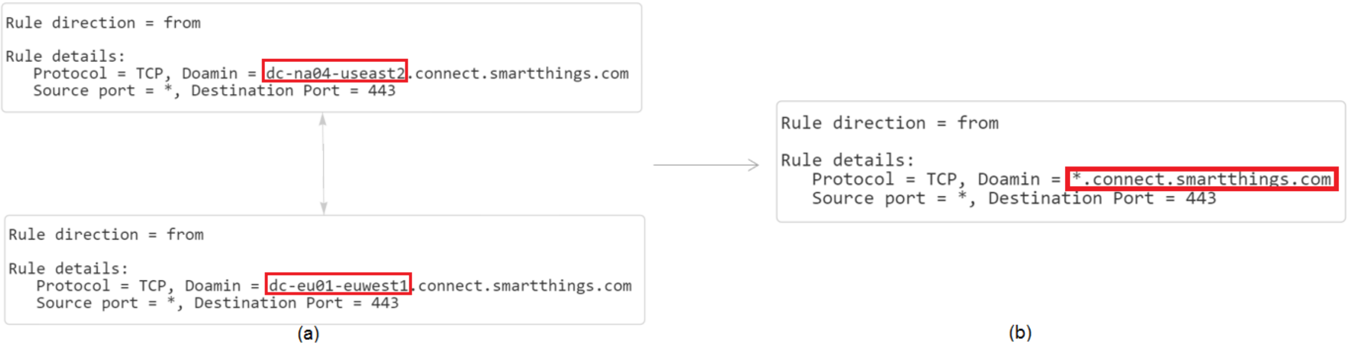
We define two ACEs from two MUDs as **similar ACEs** if they have similar domain names[3] and all other fields in the ACEs are identical. In the next section, we show that in some cases, we can generalize similar ACEs to a single ACE in the generalized MUD by using ranges in the sub-domain name.

In some of the cases, the device location has more impact. Table I presents the case of the Xiaomi camera, where the location also affects which port and protocols are used by the device.We perform further ACE comparisons, detailed in our technical report [21], by clustering ACEs with the same traffic direction into two 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.

V. MUD Generalization

In this section our goal is to create a generalized MUD that is comprehensive, tight, and secure. Comprehensive means the generalized MUD should be applicable to each of the locations. The MUD must be tight because its main goal is to

[3] Two domains are similar if their mainDomain part is equal.



```
Rule direction = from

Rule details:
   Protocol = TCP, Doamin = dc-na04-useast2.connect.smartthings.com
   Source port = *, Destination Port = 443
```

```
Rule direction = from

Rule details:
   Protocol = TCP, Doamin = dc-eu01-euwest1.connect.smartthings.com
   Source port = *, Destination Port = 443
```

(a)

```
Rule direction = from

Rule details:
   Protocol = TCP, Doamin = *.connect.smartthings.com
   Source port = *, Destination Port = 443
```

(b)

Figure 3: Two similar ACEs from the MUDs of SmartThings hub in two different locations: US (up) and UK (bottom). MUDIS created a generalized ACE (right) in which the endpoint is $*.connect.smartthings.com$, all other parameters remain the same.

white-list only legitimate flows of the IoT and thereby reduce the device attack surface. 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.

A naive generalization algorithm that would simply unify all available MUDs would be both comprehensive and tight. However, we show that it has slow convergence, and results in a larger MUD file than our MUDIS generalized MUD. Having fewer rules in a MUD file is an advantage because it is more explainable to humans; this is important to administrators or manufacturers who need to maintain the MUD. Moreover, it offers a lightweight implementation in a firewall.

The main idea behind MUDIS generalization, is the generalization of two similar ACEs that differ only in the sub-domain part, by using $*$ in the sub-domain field, e.g., $*.connect.smartthings.com$ in Figure 3b. To keep the generalized MUD tight and secure, MUDIS does not generalize ACEs with different domain suffixes (TLDs) (e.g., $iotvendor.co.*$) and known cloud services that are shared across clients (e.g., $*.s3.amazonaws.com$). 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. This is aligned with IETF Operational Consideration for the use of DNS in IoT [24]. In order to ensure fast convergence, if some ACEs are sharing a domain that was generalized in some similar ACEs, we generalize it in all ACEs in which it appears. The key intuition behind this step is that such a domain probably has a segment that depends on the location, and hence we generalize it *to support future locations that we have not yet encountered.*

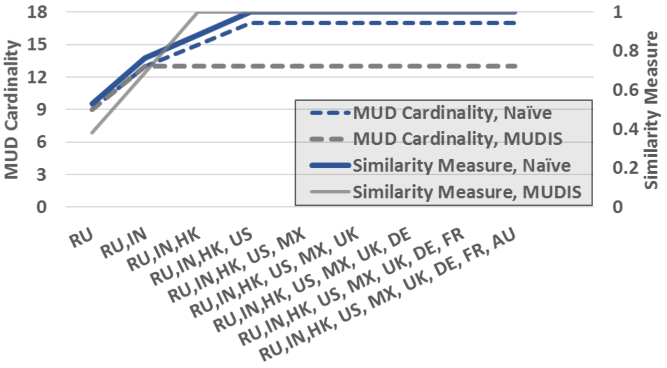


Figure 4: Performance comparison of generalized MUD and naïve unifying MUD files of the Yi Camera. Each point on the x-axis corresponds to the unified or generalized MUD at the specified locations. To evaluate a similarity score, each MUD is compared to the correlated global MUD, consisting of all available locations.

In Figure 4 we present a convergence analysis of MUD files for the Yi Camera, while using 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 2, 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.

VI. Conclusions and Future Work

In this work, we demonstrate that device location has an impact on the network behavior of IoT devices and their corresponding MUD values. We present an efficient generalization algorithm to create a single MUD that can work in all locations.

REFERENCES

[1] 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.

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

[3] 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.

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

[5] N. I. o. S. NIST and Technology, "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

[6] M. Jethanandani, S. Agarwal, L. Huang, and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)," RFC 8519, Mar. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8519

[7] A. Anonymous, "Mudis - mud inspection system," 2021. [Online]. Available: https://github.com/ransh93/MUDIS

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

[9] 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.

[10] N. I. o. S. NIST and Technology, "Mud-pd is a tool assist in the characterization of iot device network behavior and the creation and definition of appropriate mud files." [Online]. Available: https://github.com/usnistgov/MUD-PD

[11] A. M. Mandalari, R. Dubois, Daniel J.and Kolcun, M. T. Paracha, H. Haddadi, and D. Choffnes, "Blocking without breaking: Identification and mitigation of non-essential iot traffic," in *Proc. of the Privacy Enhancing Technologies Symposium (PETS)*, 2021.

[12] "Mud maker tool," 2021. [Online]. Available: https://mudmaker.org/

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

[14] 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.

[15] 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.

[16] 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.

[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] 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

[19] 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.

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

[21] A. Bremler-Barr, B. Meyuhas, and R. Shister, "Technical paper: One mud to rule them all," https://www.researchgate.net/publication/357974810_noms_sumbimssion_final1, 2022, [Online; accessed 14-January-2022].

[22] 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

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

[24] 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