# Making Access Control Easy in IoT

Vafa Andalibi, Jayati Dev, Donginn Kim, Eliot Lear, L. Jean Camp

**Noname manuscript No.**
(will be inserted by the editor)

# Making Access Control Easy in IoT

**Vafa Andalibi · Jayati Dev · DongInn Kim · Eliot Lear · L. Jean Camp**

Received: date / Accepted: date

**Abstract** Secure installation of Internet of Things (IoT) devices requires configuring access control correctly for each device. In order to enable correct configuration the Manufacturer Usage Description (MUD) has been developed by Internet Engineering Task Force (IETF) to automate the protection of IoT devices by micro-segmentation using dynamic access control lists. The protocol defines a conceptually straightforward method to implement access control upon installation by providing a list of every authorized access for each device. This access control list may contain a few rules or hundreds of rules for each device. As a result, validating these rules is a challenge. In order to make the MUD standard more usable for developers, system integrators, and network operators, we report on an interactive system called MUD-Visualizer that visualizes the files containing these access control rules. We show that, unlike manual analysis, the level of the knowledge and experience does not affect the accuracy of the analysis when MUD-Visualizer is used, indicating that the tool is effective for all participants in our study across knowledge and experience levels.

**Keywords** Usable Security · Internet of Things · Network Security · Usable Access Control · IoT · MUD · Manufacturer Usage Description

## 1 Introduction

The forecast for the number of connected IoT devices in 2025 is now raised to 30.9 billion [13], yet their (in)security is still a major concern [16]. There is a need for secure onboarding meaning that the device is secured as soon as it is connected to the network. One major component of secure onboarding both for cyber-physical systems and IoT is firewall configuration. Without access control,

V. Andalibi, J. Dev, D. Kim, L. J. Camp
Indiana University, Bloomington, IN, USA
E-mail: {vafandal, jdev, dikim, ljcamp}@indiana.edu

E. Lear
Cisco Systems, Zurich, Switzerland
E-mail: lear@cisco.com

IoT devices are susceptible to participate in DDoS attacks [10], are vulnerable to ransomware [24], and enable information exfiltration from within networks [6]. It is the nature of botnets that the subverted devices need to be controlled by the attackers' command and control (C2) infrastructure [3]. Secure onboarding that implements allow-list access control limits exposure of devices to attacks and prevents any subverted device from connecting to the attackers' C2 points. Unlike traditional botnets, the control servers in IoT are highly dynamic so the typical response of identifying then block-listing is infeasible [21].

To this end, the Internet Engineering Task Force (IETF) has developed the Manufacturer Usage Description (MUD); a standard that provides an isolation-based defense for IoT devices using dynamic access control [12]. The urgency and scale of the need for such a solution are shown by the fact that MUD is also a part of the National Institute of Standard and Technology (NIST) security for IoT initiatives [5]. In addition, the Department of Commerce has a working group to integrate the Software Bill of Materials (SBoM) initiative with MUD[1] and the IETF has a proposed standard integrating SBoM with MUD[2]. MUD can also be used for mitigating DDoS attacks in the Fog [1].

MUD relies on manufacturers for an Access Control List (ACL) in the form of a MUD-File. A MUD-File defines the allowed and expected behaviors of the associated device. The clear implication is that developers must be able to write clear and correct MUD-Files and network operators must be able to read and validate the MUD-Files to ensure that unnecessary communications, either locally or over the Internet, are not allowed. These are difficult problems, and like many security tasks, are not well aligned with human cognitive abilities [15].

In this work, we report on the usability analysis of the MUD-Visualizer [2]; a tool that is intended to support developers and network operators in evaluating overlaps, duplication, and possible conflicts in MUD-Files. We report on the design and results of our human subjects research that we conducted to investigate the following research questions:

**RQ1**: How does **security knowledge** affect the accuracy of the analysis of the MUD-Files?
**RQ2**: How does **security experience** affect the accuracy of the analysis of the MUD-Files?
**RQ3**: To what extent does **level of knowledge and experience** affect the accuracy of the analysis of the MUD-Files?

## 2 The MUD Standard

In this section, we briefly review the MUD standard for those readers who are unfamiliar with MUD. MUD is comprised of six main components: **MUD-File** which is a YANG-based JSON file (RFC 7951) created and digitally signed by the manufacturer. It embeds the behavioral profile of the IoT device in an access control list. MUD-Files should be hosted on manufacturer's **MUD file server**. The location of these files on the Internet is the **MUD-URI** which is stored on

[1] https://www.ntia.doc.gov/files/ntia/publications/ntia_practices_model_and_summary_19-02-20_0.pdf
[2] https://tools.ietf.org/html/draft-lear-opsawg-mud-sbom-00

the IoT device. Upon connection of the device to a MUD-compliant network, the device sends the embedded MUD-URI to the Authentication, Authorization, and Accounting, i.e., **AAA server**. The **MUD-Manager** is the core of MUD architecture. After receiving the MUD-URI, it will retrieve the MUD-File from the manufacturer's MUD file server and communicates the MUD-File rules to the AAA server [12]. The **Network Access Device (NAD)** (i.e., the router) is equipped with an internal firewall that is configured by the AAA server. MUD provides seven abstractions that can be used to define the behavior of and constraints on an IoT device in a MUD-File. The **domain-name** abstraction is used to enforce restrictions on cloud access. The **local-networks** abstraction defines the communication of a device with other devices on the network. With the **manufacturer** abstraction, the authority component (i.e., domain name) of a device is matched against the MUD-URI of another node which restricts devices' access to specific manufacturers. Similarly, the **same-manufacturer** abstraction defines when devices built by one manufacturer can communicate with each other but not with devices built by other manufacturers. Both of the **controller** and **my-controller** abstraction are used when devices use a controller to communicate. Lastly, the **model** abstraction constrains a device to communicate only with other instances of the same device (e.g., only lightbulbs interact) [12].

To address the human factors challenges in the analysis of the MUD-Files, Andalibi et al. [2] proposed and implemented MUD-Visualizer with the goal of 1) protocol checking to avoid formatting errors in the MUD-File to prevent coding errors 2) identifying internal inconsistencies and inefficiencies to prevent logic errors 3) enabling both manufacturers and sysadmins to review and validate the MUD-Files by processing the abstractions' access control rules and visualizing them. This processing is performed by encoding the merged Access Control Entries (ACEs) into a tree (i.e., ACE Tree) followed by pruning that tree to remove the duplicate ACEs that are generated by merging the MUD abstractions in two or more MUD-Files [2]. MUD-Visualizer can be deployed either as a stand-alone app or as a web app. It is scalable, open-source, and publicly available online on GitHub [2].

## 3 Related Work

Currently there are five implementations of MUD: Cisco MUD[3], NIST MUD[4], osMUD[5], Masterpeace MUD (closed-source), and CableLabs Micronets MUD[6]. NIST details the efficacy of these implementations against network-based attacks [5]. Regarding the MUD-Files, mudmaker[7] is a web app specifically for creating MUD-Files. For devices that are not MUD-compliant, Hamza et al. created MUDgee that uses the network traffic of the target IoT device to generate its MUD-File [8]. Beside MUD-Visualizer, which is the focus of this paper, mudpp[8] (MUD Pretty Printer) is another tool that is developed for summarizing the ACL in the

[3] https://github.com/CiscoDevNet/MUD-Manager
[4] https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927289
[5] https://github.com/osmud/osmud
[6] https://github.com/cablelabs/micronets-mud-tools
[7] https://www.mudmaker.org
[8] https://github.com/iot-onboarding/mudpp

MUD-File. However, since it does not perform any analysis on the interaction between the MUD-Files we did not consider it for our study.

Usable access control has long been a challenge in usable security. An early study on the mitigation of human error in access control management was done by Maxion and Reeder [14]. They found that visualization improves the rate of completing the assigned task by a factor of three. The error in these completed tasks was also reduced by up to 94%. This study is particularly relevant to our work here because, like Maxion and Reeder, we selected computer and network science students with significant expertise.

The study conducted by Vaniea et al. [22] also investigated the difficulty of translating policy rules into access control rules where they recommend visual feedback. They implemented SPARCLE [22] to present the data in a table as a commonly used method of information visualization. The ExpandableGrid developed by Reeder et al. [18] for improving file permissions in Windows XP is another example in this category.

Graph Visualization was previously used by [11] which is more similar to MUD-Visualizer's flow-based visualization [2]. Another study that concludes the importance of visualization is the work by Xu and colleagues [23]. They investigate the uncertainties in access control decisions and found that a lack of feedback forced the administrators who intend to resolve access control conflicts into a trial and error mode. Moreover, Smetters et al. [20] found that limitations in the UI would lead to the reluctance to change the access control settings which applies to MUD deployment as well; manual evaluation of the interaction between multiple MUD-Files is a difficult and time-consuming task for system administrators.

Erbenich et al. [7] studied the efficacy of the link visualization to better protect the end-users against phishing. They break down the URL and only visualize the most critical part of it for successful phishing detection. The same concept was used in MUD-Visualizer where only the summary of the MUD-Files was presented to the users. In another work, Scott and Ophoff [19] conducted a user study to study the effectiveness of information security knowledge in decision making. By analyzing the knowledge-behavior gap, they found that a deeper technical understanding of cyber threats will help the user to effectively derive a more cautious and preventing behavior. This motivates one of our goals; to find out whether MUD-Visualizer can help the users with higher knowledge and expertise in the analysis of the MUD-Files.

## 4 Method

Our survey incorporated two groups of participants: the first group used MUD-Visualizer for the analysis and the second group directly analyzed plain-text MUD-Files (hereinafter referred to as mudviz and plain groups respectively). The `plain` group acted as a control group to measure the efficacy of the `mudviz` group. We asked a total of 81 questions, including three screening questions, five demographic questions, twenty-three questions related to the analysis of the MUD-Files (main experiment), forty expertise questions, and ten usability questions from the participants.

Our **Screening Questionnaire and recruitment** were designed to ensure that the participants have the required knowledge for analyzing a MUD-File. Be-

fore inclusion, participants had to show the knowledge of fundamentals of computer networking (i.e., understanding IP, Port, and access control) through manual parsing of components of a MUD-File. We focused on recruitment in an advanced computer networking course.

The **demographic questions** contained questions about age, gender, education, employment status, and income motivated from the study about the privacy for WEIRD populations [9].

The core of the **experiment design** was 23 questions about the analysis of the MUD-Files. We first asked the participants about the remote servers or local devices allowed for a specific device given its MUD-File. This included two questions about the number of nodes devices allow-listed, seven questions about the name of these allowed nodes, and one question about between-node communication. We also included thirteen questions about the Transport and Network layer protocols that are allow-listed for use, e.g. IP version, Port number, TCP vs UDP.

The **post-experiment** questions comprised 50 questions in two categories: forty expertise questions incorporating a set of computer expertise questions from [17] and ten usability questions from the System Usability Scale (SUS) [4].

## 5 Results

31% of our screening survey respondents (24 out of 76) failed to answer one or more of the screening questions and were not considered for the main study. The **participants** in our study were skewed with respect to gender (84.6% male, 15.4% female). Out of the total of 52 participants, 41 were below the age of 30 years. Over 70% were students, with 50 participants having at least a technical Bachelors's degree. This includes only the participants who passed the screening questions. Participants were split equally between the two groups, `mudviz` and `plain`.

In order to evaluate **participants' security and computer expertise**, they were presented with a set of 13 question categories. These questions were obtained from the set of computer expertise questions from [17]. For measures of `knowledge`, these were knowledge-based questions on (i) phishing (`Kphish`) (ii) certificates (`Kcert`) (iii) SQL commands (`Ksql`) (iv) intrusion detection systems (`Kids`) (v) port 80 (`K80`) (vi) Website markers for security (`Kweb`) (vii) defining IoT (`Kiot`) and (viii) access control (`Kac`). For single response questions, if the participants' answers matched the correct responses, these variables were coded as 1, otherwise 0. For multiple response questions (`Kphish` and `Kcert`), if the participants' got a sum of correct values above the median in each category, the variables were coded as 1, otherwise 0. Since, all participants got responses to `Kiot` correct, these responses were removed in calculating the covariance matrix for factor analysis.

We then performed a factor analysis on the remaining seven variables to create a `TotalKnowledge` variable. A scree plot and a test of hypothesis showed that a factor of one was sufficient to measure knowledge. This factor, `TotalKnowledge`, was a combination of four factors, calculated by the equation below:

$$TotalKnowledge \leftarrow (-0.5 * Kcert) + (0.6 * Ksql) + (0.6 * Kids) + (0.7 * K80)$$

`TotalExperience` was similarly a combination of weighted factors, given by the equation below:

$$TotalExperience \leftarrow (0.5 * Eyears) + (0.4 * Elang) + (0.4 * Efreq)$$

That is, for the measure of experience, the remaining five questions on experience were evaluated - (i) prior computer expertise (`Eexp`) (ii) prior security expertise (`Etech`) (iii) programming languages known (`Elang`) (iv) years of experience working in security (`Eyears`) and (v) frequency of dealing with security problems (`Efreq`). Since the answers to these questions cannot be evaluated as correct/incorrect, we normalized each of the five variables and performed a second-factor analysis to create a `TotalExperience` variable. A scree plot and a test of hypothesis showed that a factor of one was sufficient to measure knowledge.

We then evaluated the **Effect of Knowledge on Accuracy** by first calculating `TotalKnowledge` and `TotalExperience`. `Accuracy` was measured as a summation of the correct answer to the 23 questions in the experiment, providing a raw accuracy percentage for each participant.

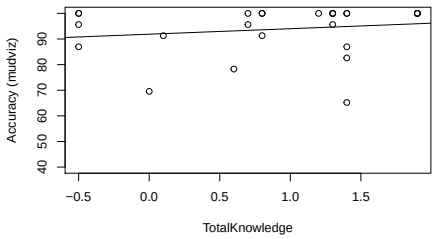
In order to answer **RQ1**, we first performed a linear regression to measure the effect of the independent variable `TotalKnowledge` on the dependent variable `Accuracy` for both groups (Fig. 1a and 1b). Unsurprisingly, knowledge has a positive effect on the accuracy of the analysis of the MUD-Files. We also found that the effect of `TotalKnowledge` on `Accuracy` is significant in the `plain` group ($b = 7.689, p-value = 0.0164$) but not for the `mudviz` group ($b = 2.148, p-value = 0.406$). Thus, participants in the `mudviz` group seemed to have the same level of accuracy across computer and security knowledge levels. However, this is not the case for plain text files. Participants with greater `TotalKnowledge` seemed to have significantly high `Accuracy` in the `plain` group. This suggests that normally a high level of security expertise is needed to understand textual MUD-Files, but that an effective visualization can result in accuracy by moderate experts indistinguishable from that of the most expert.

The results of a linear regression conducted on each of the factors indicate that none of the factors in the `mudviz` have a significant effect on `Accuracy`, but some factors in the `plain` group are significant. Table 1 shows the regression of individual knowledge factors for both groups. We see that the `Kphish`, `Kids`, `K80`, and, `Ksql` are more strongly significant than the other factors in contributing to `Accuracy`.

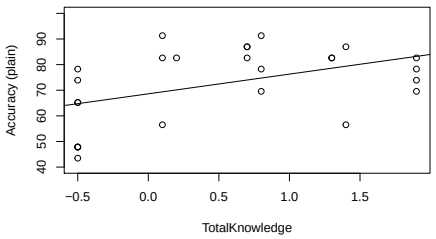
To answer the first part of **RQ3**, we analyzed whether `TotalKnowledge` can be divided into sub-groups of knowledge and expertise respectively; and how these interact with `Accuracy`. We sorted the participants from each of the `mudviz` and `plain` groups in ascending order based on their `TotalKnowledge` with 13 participants in each sub-group. A signed Wilcoxon Rank-sum test indicated significant difference between the four sub-group categories, with p-values between the

Table 1: Regression analysis for individual knowledge factors versus accuracy in MUD analysis (showing significant components only).

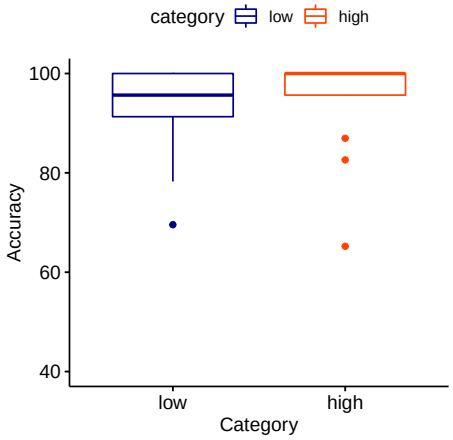
| Factors | mudviz | | plain | |
|---|---|---|---|---|
| | co-efficient | p-value | co-efficient | p-value |
| Kphish | 7.412 | 0.136 | **12.847** | **0.0445 *** |
| Kids | 1.967 | 0.624 | **11.594** | **0.0413 *** |
| K80 | 3.370 | 0.411 | **9.576** | **0.0968 .** |
| Ksql | 1.733 | 0.701 | **11.957** | **0.0348 *** |
| TotalKnowledge | 2.148 | 0.406 | **7.689** | **0.0164 *** |



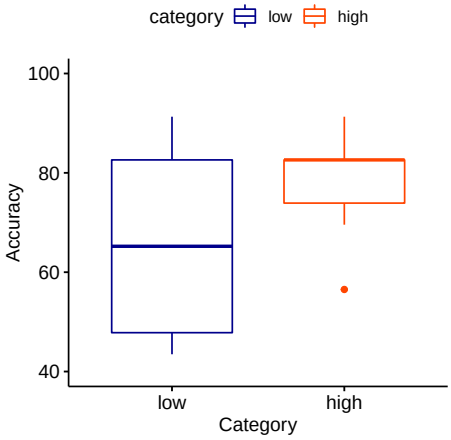

(a) Regression for `mudviz` group




(b) Regression for `plain` group




(c) `Accuracy` v.`TotalKnowledge`(`mudviz`)

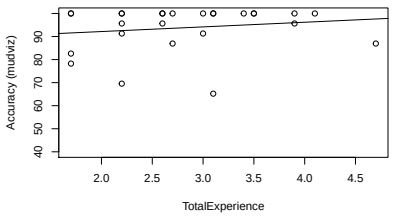



(d) `Accuracy` v.`TotalKnowledge`(`plain`)

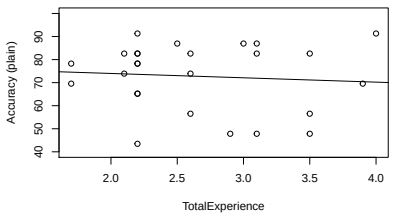
Fig. 1: (a) and (b) show the scatter plot of `Accuracy` against `TotalKnowledge`. (c) and (d) show `Accuracy` for four groups, indicating that the effect of the MUD-Visualizer is consistently positive across knowledge groups.

low and high groups of less than 0.001. We conducted an ordinal logistic regression between the two categories (low and high) for each of the two groups for `TotalKnowledge` against `Accuracy`, (a) `Mudviz` and (b) `Plain`. As seen in Fig. 1c, the accuracy in correct interpretation of the MUD-Files did not vary significantly between high and low knowledge categories in the `Mudviz` group ($b = -0.018, p - value = 0.663$). However, in case of the `plain group` (Fig. 1d) `TotalKnowledge` played a significant role in increasing the accuracy ($b = -0.066, p - value = 0.054$). The accuracy was consistently higher in the `mudviz` group compared to the `plain` group in all cases.

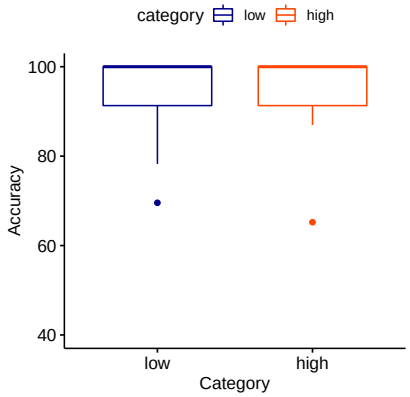
To investigate **Effect of Experience on Accuracy** (**RQ2**) we began with a linear regression to measure the effect of independent variable `TotalExperience` on `Accuracy` for the both groups. Fig. 2a and 2b show the scatterplot and the regression lines for each of the `mudviz` and `plain` groups respectively. Unsurprisingly, experience has a positive effect on `Accuracy` in case of `mudviz`. Yet there



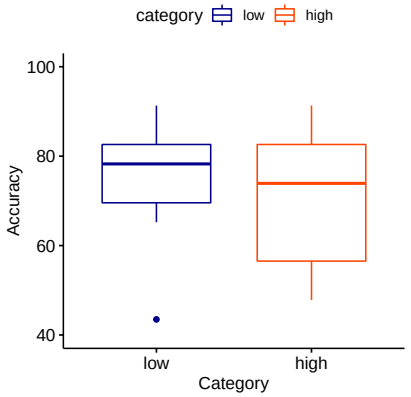

(a) Regression for `mudviz` group




(b) Regression for `plain` group




(c) `Accuracy` v.`TotalExperience` (`mudviz`)




(d) `Accuracy` v.`TotalExperience` (`plain`)

Fig. 2: (a) and (b) show the scatter plot of `Accuracy` against `TotalExperience`. (c) and (d) show `Accuracy` for four groups, indicating that the effect of the MUD-Visualizer is consistently positive across all experience groups.

appears to a weak negative effect on `Accuracy` in case of `plain` in Fig. 2d, which we delve into in Table 2 below.

`TotalExperience` is not significant for `Accuracy` in either case of the `plain` group ($b = -1.879, p-value = 0.687$) or the `mudviz` group ($b = 2.018, p-value = 0.425$); although differences in the distribution of the `plain` are apparent. Thus,

Table 2: Regression analysis for individual experience factors versus accuracy in MUD analysis.

| Factors | mudviz | | plain | |
|---|---|---|---|---|
| | co-efficient | p-value | co-efficient | p-value |
| Eexp | 1.187 | 0.393 | **6.505** | **0.00299 **** |
| Efreq | 1.789 | 0.259 | -4.797 | 0.18 |
| Eyears | 0.345 | 0.899 | -0.050 | 0.989 |
| TotalExperience | 2.018 | 0.425 | -1.879 | 0.687 |

participants in the group that were presented with the MUD-Visualizer seemed to have the same level of accuracy across computer and security experience levels.

Taking a closer look at the experience factors by conducting a linear regression for each of the factors, we see that none of the factors in the `mudviz` group affect `Accuracy` significantly, but the `Eexp` factor in the `plain` group does. In that case, `Eexp` is significant and positive. Table 2 shows the regression of individual experience factors for both groups. `Eexp` is a set of Booleans from querying if participants had experience with any of the following: designing a website, registering a domain name, using SSH, configuring a firewall, creating a database, installing a computer program, and writing a computing program. The intriguing but not significant negative effect on `Accuracy` is due to `Efreq` (frequency of handling security incidents) and `Eyears` (years of experience working in the security field). It is possible that this may result from less experienced people defining security incidents (e.g., spam vs. an intrusion) or being in the security field differently (e.g., total years in coursework vs. years in incident response not DevOps).

To answer the second part of the **RQ3**, we analyzed whether `TotalExperience` can be divided into sub-groups of knowledge and expertise respectively, and how they affect the `Accuracy`. We sorted the participants from each of the `mudviz` and `plain` groups in ascending order based on their `TotalExperience`. Again, we considered 13 participants in each sub-group. A signed Wilcoxon Rank-sum test showed that the four sub-group categories are significantly different, with $p-values$ between each of the low versus high groups being less than 0.001. We conducted an ordinal logistic regression between the two categories (low and high) for each of the two groups of `TotalExperience` against `Accuracy`, (a) `Mudviz` (b) `Plain`. The results illustrated that for `Mudviz` ($b = 2.018, p-value = 0.425$), the accuracy in interpreting the MUD-File correctly was the nearly the same for low and high `TotalExperience` (Similar to `TotalKnowledge`).

## 6 Conclusions

In this work, we sought to evaluate the efficacy of MUD-Visualizer for correct evaluation of MUD-File by participants with some expertise. We report on the increase in efficacy among all participants, showing that the difference in the performance of network engineers with and without knowledge of security or security expertise was significant. More-so, accuracy of participants using the MUD-Visualizer showed knowledge of security to be insignificant (among these participants). Given the difficulty of providing network engineers with security expertise, having a visualization that decreases the cost of inexperience argues for the importance of human factors in standards. Beyond that we found evidence that interpretation of security questions may be having a subtle impact on the results; those with less experience may not be reporting experience with the same baseline as those with more. This phenomena is worthy of additional research, although in this case any impact would strengthen the results.

**Acknowledgements** This research was supported in part by the National Science Foundation awards CNS 1565375 and CNS 1814518, as well as the grant #H8230-19-1-0310, Cisco Research Support, Google Research, and the Comcast Innovation Fund. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do

not necessarily reflect the views of the National Science Foundation, Cisco, Comcast, Google, nor Indiana University.

## References

1. Andalibi, V., Kim, D., Camp, L.J.: Throwing MUD into the FOG: Defending IoT and Fog by Expanding MUD to Fog Network. In: 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19) (2019)
2. Andalibi, V., Lear, E., Kim, D., Camp, J.: On the Analysis of MUD-Files' Interactions, Conflicts, and Configuration Requirements Before Deployment. In: 5th EAI International Conference on Safety and Security in Internet of Things, SaSeIoT. Springer (2021)
3. Bailey, M., Cooke, E., Jahanian, F., Xu, Y., Karir, M.: A Survey of Botnet Technology and Defenses. In: 2009 Cybersecurity Applications & Technology Conference for Homeland Security, pp. 299–304. IEEE (2009)
4. Brooke, J.: SUS: A "Quick and Dirty" Usability. CRC Press (1996)
5. Dodson, D., Polk, W., Souppaya, M., Barker, W., Lear, E., Weis, B., Fashina, Y., Grayeli, P., Klosterman, J., Mulugeta, B., et al.: Securing Small Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD). Tech. rep., National Institute of Standards and Technology (2019)
6. D'Orazio, C.J., Choo, K.K.R., Yang, L.T.: Data Exfiltration from Internet of Things Devices: iOS Devices as Case Studies. IEEE Internet of Things Journal **4**(2), 524–535 (2016)
7. Erbenich, V.I.P., Träder, D., Heinemann, A., Nural, M.: Phishing Attack Recognition by End-Users: Concepts for URL Visualization and Implementation. In: HAISA, pp. 179–188 (2019)
8. Hamza, A., Ranathunga, D., Gharakheili, H.H., Roughan, M., Sivaraman, V.: Clear as MUD: Generating, Validating and Applying IoT Behavioral Profiles. In: Proceedings of the 2018 Workshop on IoT Security and Privacy, pp. 8–14. ACM (2018)
9. Henrich, J., Heine, S.J., Norenzayan, A.: Most People are Not WEIRD. Nature **466**(7302), 29–29 (2010)
10. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: DDoS in the IoT: Mirai and Other Botnets. Computer **50**(7), 80–84 (2017)
11. Kolomeets, M., Chechulin, A., Kotenko, I., Saenko, I.: Access Control Visualization Using Triangular Matrices. In: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 348–355 (2019). DOI 10.1109/EMPDP.2019.8671578
12. Lear, E., Droms, R., Romascanu, D.: Manufacturer Usage Description Specification. RFC 8520 (2019). DOI 10.17487/RFC8520. URL `https://rfc-editor.org/rfc/rfc8520.txt`
13. Lueth, K.L.: State of the IoT 2020: 12 billion IoT Connections, Surpassing non-IoT for the First Time. [Online]. Available on: `https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time`
14. Maxion, R.A., Reeder, R.W.: Improving user-interface dependability through mitigation of human error. International Journal of human-computer studies **63**(1-2), 25–50 (2005)
15. Oliveira, D., Rosenthal, M., Morin, N., Yeh, K.C., Cappos, J., Zhuang, Y.: It's the Psychology Stupid: How Heuristics Explain Software Vulnerabilities and How Priming can Illuminate Developer's Blind Spots. In: Proceedings of the 30th Annual Computer Security Applications Conference, pp. 296–305 (2014)
16. O'Neill, M., et al.: Insecurity by Design: Today's IoT Device Security Problem. Engineering **2**(1), 48–49 (2016)
17. Rajivan, P., Moriano, P., Kelley, T., Camp, L.J.: Factors in an End User Security Expertise Instrument. Information & Computer Security (2017)
18. Reeder, R.W., Bauer, L., Cranor, L.F., Reiter, M.K., Bacon, K., How, K., Strong, H.: Expandable Grids for Visualizing and Authoring Computer Security Policies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1473–1482 (2008)
19. Scott, J., Ophoff, J.: Investigating the Knowledge-Behaviour Gap in Mitigating Personal Information Compromise. In: HAISA, pp. 236–245 (2018)
20. Smetters, D.K., Good, N.: How Users Use Access Control. In: Proceedings of the 5th Symposium on Usable Privacy and Security, pp. 1–12 (2009)

21. Tanabe, R., Tamai, T., Fujita, A., Isawa, R., Yoshioka, K., Matsumoto, T., Gañán, C., van Eeten, M.: Disposable Botnets: Examining the Anatomy of IoT botnet infrastructure. In: Proceedings of the 15th International Conference on Availability, Reliability and Security, pp. 1–10 (2020)
22. Vaniea, K., Ni, Q., Cranor, L., Bertino, E.: Access Control Policy Analysis and Visualization Tools for Security Professionals. In: SOUPS Workshop (USM), pp. 7–15 (2008)
23. Xu, T., Naing, H.M., Lu, L., Zhou, Y.: How Do System Administrators Resolve Access-denied Issues in the Real World? In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 348–361 (2017)
24. Yaqoob, I., Ahmed, E., ur Rehman, M.H., Ahmed, A.I.A., Al-garadi, M.A., Imran, M., Guizani, M.: The Rise of Ransomware and Emerging Security Challenges in the Internet of Things. Computer Networks **129**, 444–458 (2017)