Extending MUD to Smartphones

Ina Berenice Fink, Martin Serror, Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany
{fink, serror, wehrle}@comsys.rwth-aachen.de

***Abstract*—The tremendous success of the IoT is overshadowed by severe security risks introduced by IoT devices and smartphone apps to control them. Therefore, academia and industry increasingly acknowledge the use of in-network security approaches, such as IETF Manufacturer Usage Description (MUD), to restrict undesired communication. However, actual communication patterns of smart homes are not sufficiently covered by such policy-based approaches. In this paper, we propose to enforce MUD on authenticated smartphones to efficiently filter malicious traffic close to its origin and hinder further spreading. Such enforcement allows us to successfully mitigate the threat of malicious apps and IoT devices in smart home networks.**

I. Introduction

The tremendous success of the Internet of Things (IoT) has led to a vast amount of deployed smart devices, particularly in the consumer domain and smart homes [18]. In this context, IoT devices offer all sorts of convenient services to their users, such as the automation of daily tasks or alarm and monitoring systems. These services rely on local and global connections of devices and cloud services via home networks and the Internet. However, connecting IoT devices to one's home network also bears severe security risks, since such devices are known for their frequent security flaws [1], [6]. Indeed, research has shown that many deployed IoT devices suffer from missing or weak encryption of communication [2], the use of well-known standard passwords [17], and complicated or a lack of update mechanisms [25]. Among others, these vulnerabilities facilitate the massive infiltration of malware on IoT devices, e. g., to form large botnets, such as the infamous Mirai botnet [14]. To summarize, a single compromised device in a home, e. g., a legacy device with known vulnerabilities, often suffices for attackers to access the local network and infect other IoT devices. Therefore, improving the security of smart homes can, unfortunately, not solely rely on new device generations with sophisticated security mechanisms, as this neglects the interconnection of such devices and the risks that emerge when an attacker gained access to the local network.

Hence, we aim to explore *in-network security* further, offering additional protection to the security mechanisms implemented on the devices. In particular, we consider policy-based approaches, restricting the network traffic of each IoT device to the required connections for completing its functions [22], [19]. Such approaches build on the observation that many IoT devices have a limited purpose with clearly defined communication patterns, e. g., a sensor that periodically reports the measured temperature. Thus, this idea follows the principles of *least privilege* and *defense in depth*, to minimize the attack surface and limit the damage if a device gets infected or taken over by an attacker. Most prominently, the Internet Engineering Task Force (IETF) recently proposed Manufacturer Usage Description (MUD) [11], a standard allowing IoT devices to signal the network their required connections in the form of rules. The standard defines a format for these rules and how a central entity, e. g., the home router, can securely retrieve the rules of a connected device from a trusted source.

However, considering that a user typically interacts with IoT devices using a smartphone [6] questions the efficiency and effectiveness of simple central enforcement of MUD rules at the router. Indeed, similar to [16], we observe that different communication scenarios are possible for the communication between smartphone and IoT devices. Since MUD omits multi-purpose devices, like smartphones or tablet computers, it currently only covers *indirect communication*, where the traffic flows through a cloud service. Moreover, the standard does not define *how* and *where* the corresponding traffic enforcement should occur. Therefore, we argue that to improve the security of smart homes substantially, in-network approaches, such as MUD, need to account for the different IoT communication scenarios, and especially for the pivotal role of smartphones. Additionally, malicious network traffic should be filtered close to its origin before spreading further in the network.

In this paper, we thus propose to mitigate the risks emanating from smartphones and the IoT devices that smartphones interact with by extending MUD to restrict access to the local network more effectively. More precisely, we propose implementing a central *Local MUD Manager (LMM)*, which we complement with *Mobile MUD Enforcement Engines (MMEEs)* running on different smartphones associated with the home network. The MMEEs allow enforcing MUD rules close to the corresponding devices, which reduces the amount of unwanted network traffic and even enables MUD enforcement for IoT devices directly connected to a smartphone, without restricting their intended functionality. In the following, we provide an example attack scenario to illustrate the deficiencies of MUD, which we then summarize in Sec. III.

II. Example Attack Scenario

This section presents a typical example attack in smart homes [21] that is currently not covered by MUD and similar approaches. Here, the user installs an app on her smartphone, unknowingly that this app embeds malware. Research has shown that even apps that are officially listed in Google Play or Apple's App Store may infiltrate malware [4], [5], [24]. Since MUD only considers IoT device traffic, the smartphone's

© IEEE, 2020. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in *IEEE LCN 2020*.

communication per se is not restricted. Hence, the malware on the smartphone liberally scans the local network for IoT devices, e. g., using Simple Service Discovery Protocol (SSDP), and forwards information about potential targets to a malicious server on the Internet. The server then returns instructions to manipulate the local router's port forwarding table, which the malware executes using Universal Plug and Play (UPnP). Thus, the attacker does not even require direct access to the router. Finally, the manipulated port forwarding allows the malicious server to attack other local IoT devices from the Internet. In many cases, access to IoT devices cannot be restricted to specific endpoints without forfeiting functionality, e. g., local or remote access from a smartphone with dynamically changing IP addresses, thus rendering MUD impotent against such scenarios.

We highlight the importance of the mentioned security risks by identifying a real-world target for the described attack based on the security analysis in [13]. The examined TP-Link camera can be controlled via apps or web interfaces over a broad range of ports from the local network using TCP or UDP and the Internet via UDP. Further, the camera uses weak default log-in credentials, sent in plaintext in the header of audio and video streams [13]. Thus, a malicious app within the local network can easily access the camera, posing a severe privacy threat, as described in the previous scenario.

This example illustrates the severeness of IoT security risks, where a single device often suffices to compromise an entire smart home network, even with MUD enabled. Based on these observations, we continue with a general analysis of the open issues and challenges for in-network security in smart homes.

III. Problem Statement

Policy-based in-network security for the IoT leverages the fact that many IoT devices have a limited purpose, facilitating the specification of their communication behavior as Access Control Lists. One approach to this is MUD [11], allowing the definition of profiles for IoT devices that specify their allowed network connections, e.g., based on IP addresses, protocols, and ports. In turn, all other communication can be blocked, avoiding undesired behavior in the first place. Therefore, MUD is increasingly considered by related work to improve smart home security [9], [10], [20], [15].

However, MUD only provides a rough reference architecture regarding the actual enforcement of the network policies. In particular, taking into account the typical interactions of users with their IoT devices, we note that complex communication scenarios, including smartphones, are not covered by MUD. The main reason for this limitation is that the MUD policies apply to entire devices, and it is not possible to differentiate between different services running on the device. Furthermore, filtering traffic at a central networking device excludes direct communication between an IoT device and a smartphone. Concretely, we identify the following shortcomings of MUD:

(i) *Limited applicability:* policies only apply to single-purpose IoT devices, although they communicate with other (general-purpose) devices, e. g., smartphones.

(ii) *Limited expressiveness:* policies apply to an entire device. They cannot be differentiated according to services and apps, e. g., it is impossible to restrict access to a dedicated control app while blocking all other traffic.

(iii) *Limited enforcement:* policies are centrally enforced at networking devices. Thus, subnets might remain untouched, whereas enforcement on the respective device decreases the attack radius. Furthermore, distributed enforcement increases robustness against single failures.

Therefore, to successfully leverage in-network security for smart homes, existing policy-based approaches, such as MUD, need to address these shortcomings. In the following, we give a short overview of related work before diving into our proposal to counter the explored deficiencies in Sec. V.

IV. Related Work

Goutam et al. [7] point out that IoT devices usually connect to cloud endpoints or control devices, e. g., smart hubs, but not to each other. Meanwhile, attacks often originate from other IoT devices. Thus, the authors employ user-based device categorization to implement the complete isolation of IoT devices. However, they do not address the security risks emerging from control devices. Habibi et al. [8] propose a more sophisticated approach by implementing whitelisting where policies are automatically learned and enforced by a proxy server deployed on a router. Besides allowed endpoints, their policies include statistical parameters, e. g., sent packets per minute. However, attacks targeting the learning process can lead to faulty device policies. Moreover, this approach exclusively focuses on external connections, neglecting the risks of compromised devices attacking the network from the inside. Similarly, Hamza et al. [9] combine Software-Defined Networking (SDN) and MUD to narrowly define and restrict IoT devices' behavior and forward non-complying traffic to an Intrusion Detection System. Like [8], this approach suffers from lacking consideration of threats in the same network.

In contrast to exclusively focusing on IoT devices, Demetriou et al. [3] emphasize the risks emerging from malware-laden smartphones by proposing HanGuard, which monitors the traffic of individual apps directly on the smartphone and validates it against policies which are shared with the router. HanGuard then forwards its decisions about how to handle the traffic to the router for enforcement. Additionally, the router applies general filtering to fully exclude unauthorized devices. However, all policies are defined by the user, depending on the user's expertise and judgment. Thus, they are not narrowly tailored to the device's advertised function as it is the case when using device-specific policies like MUD, offering a larger attack surface. Furthermore, the enforcement is carried out centrally at the router, leaving direct communication between smartphone or smart hub and IoT device unrestricted by design. Last, filtering is only based on IP addresses and ports. Thus, at least with connectionless protocols such as UDP, the router cannot reliably distinguish the traffic emerging from multiple apps on the same smartphone.

Consequently, concurrent connections of malicious and benign apps lead either to security risks or function impairment.

Another common approach to increase smart home security is to apply Machine Learning (ML) for filtering suspicious traffic. A general problem of ML-based approaches is the probability of false positives [10], [12] limiting the legit functionality of IoT devices. In contrast, policies allow undeniable decisions about whether to accept or drop traffic. Furthermore, current work in the area of ML offers approaches for intrusion detection but is not suited for proactively avoiding intrusion in the first place. We hence recognize the opportunities of ML and existing work to enable automatized identification of IoT devices and detection of malicious behavior. However, similar to [10], we consider ML rather as a sensible addition to policy-based approaches instead of a sole solution for offering all-encompassing protection against IoT device intrusion.

To summarize, the related work in this field provides valuable approaches to improve the security of IoT devices. However, an all-encompassing solution is missing until now. Thus, we present and discuss a solution for extending MUD to smartphones in the remainder of this paper.

V. Extending MUD to Smartphones

As exposed in Sec. III, retrofittable solutions to improve IoT security are urgently needed. While policy-based approaches appear promising, existing solutions cover the problem space only insufficiently when taken individually because of restricted applicability, expressiveness, and enforcement. Hence, in this paper, we propose to block undesired connections already at the corresponding control devices in addition to central policy enforcement at networking devices. Our architecture (cf. Fig. 1) consists of two main components:

① A central *Local MUD Manager (LMM)*, which is based on the *MUD manager* introduced in [11], enforces MUD policies to restrict the communication of IoT devices and their control devices within the network. While [11] states the responsibilities of the MUD manager and provides details about the handling of MUD files, we contribute a concrete design of the MUD manager and the enforcement of network element configurations: Given the benefits of SDN for smart homes [23], we envision to deploy the LMM in our local network as a controller application on top of an SDN controller. According to the SDN paradigm, the SDN controller possesses an overview of the whole network and its activities. Furthermore, it can install dynamic flow rules in the flow table of SDN switches in the same network, thus steering the network traffic. We assume that the SDN controller and thus the LMM has an IP address well known to all devices in the network, e. g., by running on the same device as the local DHCP server. We further assume that IoT devices advertise themselves and transmit MUD URLs as defined in [11] by, e. g., using DHCP. Consequently, the controller knows about all IoT devices deployed in its local network, including their MAC and IP addresses, and has their MUD files available, from which it can extract flow rules to enforce.

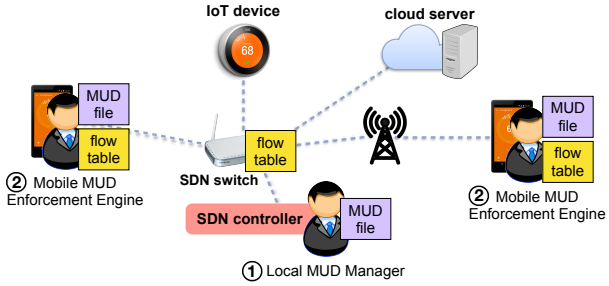

Fig. 1. General architecture: ① a central LMM filters the traffic of all local IoT devices to enable only policy-compliant communication with authenticated smartphones and the Internet. ② MMEEs embedded into smartphones' OS authenticate the smartphone to the LMM and filter app traffic.

② The LMM is complemented by distributed *Mobile MUD Enforcement Engines (MMEEs)* running directly on smartphones, which are often used to control IoT devices [16] via dedicated apps. The MUD files introduced in [11] directly address IoT devices using Access Control Lists (ACLs) to define their connections. However, the proposed MUD files essentially cover only the communication between IoT devices and cloud services. We propose to extend MUD files to consider smartphone apps additionally. Thereby, we define and restrict connections of IoT devices as well as of individual apps attempting to interact with them. To enforce rules derived from the corresponding ACLs on the smartphone, the MMEE applies app-based traffic filtering using its own flow table.

Existing policy-based approaches insufficiently address smartphones due to dynamic IP addresses. To counter this limitation, we leverage the MMEE as a medium for smartphones to disclose themselves as valid control devices, i. e., the smartphone ensures to permit only communication of apps when complying with the present policies. To ensure conformance between MUD files of apps and IoT devices, the LMM transmits a list of the locally available IoT devices (defined by their URI and IP address) and the apps with which the devices are allowed to communicate (defined by their package name), to the MMEE. The MMEE aligns the received information with its own and maps the IP addresses of the IoT devices with the URIs defined in the apps' MUD files. Then, smartphone apps are only allowed to communicate with devices in the same local network after the respective smartphone has established a secure connection to the LMM and authenticated itself with valid certificates. Besides, communication between apps and IoT devices is only allowed if mutually defined in their MUD files. By forwarding traffic to the Internet as usual, non-IoT functionality remains untouched.

Overall, our architecture comprises various interaction scenarios: the MMEE filters direct communication between smartphone apps and IoT devices directly on the smartphone while both MMEE and LMM enforce communication rules involving networking devices and cloud services.

VI. Discussion

In Sec. III, we debate the shortcomings of existing policy-based security approaches for the IoT. We point out that existing approaches, including MUD, sufficiently cover only

one communication scenario. In particular, while traffic from malicious servers on the Internet is blocked at the local router, traffic to and from smartphone apps remains untouched, no matter if benign or malicious. We address the described shortcomings with the design presented in Sec. V as follows: we solve the problem (i), i. e., limited applicability of policies, by proposing the MMEE to authenticate and authorize individual smartphones for local communication while excluding smartphones without MMEE. Thus, even smartphones with dynamic IP addresses can be explicitly allowed or excluded from communication with IoT devices by defining corresponding Access Control Entries (ACEs) within their MUD files.

Next, we address (ii), i. e., the limited expressiveness of policies, by extending the concept of MUD to single apps running on smartphones with the MMEE acting as policy enforcer. In this way, we can apply policies to individual apps instead of inevitably allowing traffic of all installed apps if allowing communication with a specific smartphone.

Last, we tackle issue (iii), i. e., the limited enforcement, by proposing a distributed approach that enforces MUD policies close to the origin of potentially malicious traffic. Hence, we can restrict traffic even if smartphones are directly connected to IoT devices, e. g., via Bluetooth, while existing approaches only provide traffic filtering at central networking devices.

By combining the proposed solutions, our approach addresses the vulnerabilities in smart homes left open by existing solutions, such as MUD, by fine-granularly filtering traffic originating from malicious apps regardless of the communication scenario. At the same time, it enables traffic of benign control apps to support their advertised functions.

To illustrate a concrete example of how our approach can improve security, we describe an example attack scenario in Sec. II, assuming that the user installs a malicious app on her smartphone. While the original MUD standard theoretically allows the blocking of unauthorized external servers, their definition might not be possible in every case, enabling the described attack. In contrast, our solution already prevents the discovery of the device via service discovery protocols by blocking all local traffic of malicious apps directly on the smartphone because of the apps' missing MUD files. Additionally, even if the discovery of IoT devices is somehow enabled, the MMEE blocks the manipulation of the local router's port forwarding table. In case that the smartphone cannot be authenticated due to the lack of an MMEE, the LMM blocks the smartphone's local traffic entirely. Last, even if manipulation of the port forwarding table somehow succeeds, the LMM may still block the attack if a narrow definition of external communication partners is possible.

Besides sophisticated attacks as the one addressed above, our approach can also prevent attacks involving the direct access of malicious smartphone apps to IoT devices.

To conclude, extending MUD to smartphones helps significantly decrease attack surfaces originating from smartphones and malicious IoT devices. Nevertheless, we still intent our solution to be deployed in combination with traditional security measures, such as encryption and authentication.

VII. Conclusion

To mitigate the IoT security threats arising from malware-laden smartphones and malicious IoT devices, we propose extending the centralized LMM with distributed MMEEs running directly on the authenticated smartphones. It enables fine-granular filtering of network traffic where only authorized smartphone apps are allowed to communicate with IoT devices according to predefined MUD rules. For future work, we are particularly interested in evaluating our approach in heterogeneous IoT deployments with MMEEs running on different devices and further to propose our extensions within the IETF.

Acknowledgments

This research was supported by the research training group "Human Centered System Security" sponsored by the German federal state of North Rhine-Westphalia.

References

[1] J. Bugeja, A. Jacobsson, and P. Davidsson, "On Privacy and Security Challenges in Smart Connected Homes," in *IEEE EISIC*, 2016.
[2] M. Capellupo, J. Liranzo *et al.*, "Security and Attack Vector Analysis of IoT Devices," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*. Springer Int'l Pub., 2017.
[3] S. Demetriou *et al.*, "HanGuard: SDN-driven Protection of Smart Home WiFi Devices from Malicious Mobile Apps," in *ACM WiSec*, 2017.
[4] M. Egele, C. Kruegel *et al.*, "PiOS: Detecting Privacy Leaks in iOS Applications," in *NDSS*, Feb. 2011.
[5] L. García and R. J. Rodríguez, "A Peek Under the Hood of iOS Malware," in *IEEE ARES*, Aug. 2016.
[6] D. Geneiatakis, I. Kounelis *et al.*, "Security and Privacy Issues for an IoT based Smart Home," in *IEEE MIPRO*, 2017.
[7] S. Goutam, W. Enck, and B. Reaves, "Hestia: Simple Least Privilege Network Policies for Smart Homes," in *ACM WiSec*, 2019.
[8] J. Habibi, D. Midi *et al.*, "Heimdall: Mitigating the Internet of Insecure Things," *IEEE Internet of Things Journal*, vol. 4, no. 4, Aug 2017.
[9] A. Hamza *et al.*, "Combining MUD Policies with SDN for IoT Intrusion Detection," in *ACM IoT S&P*, Aug. 2018.
[10] ——, "Detecting Volumetric Attacks on loT Devices via SDN-Based Monitoring of MUD Activity," in *ACM SOSR*, Apr. 2019.
[11] E. Lear *et al.*, "Manufacturer Usage Description Spec." RFC 8520, Mar. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8520.txt
[12] F. Li, A. Shinde *et al.*, "System Statistics Learning-Based IoT Security: Feasibility and Suitability," *IEEE IoT J*, vol. 6, no. 4, 2019.
[13] F. Loi, A. Sivanathan *et al.*, "Systematically Evaluating Security and Privacy for Consumer IoT Devices," in *ACM IoT S&P*, Nov. 2017.
[14] A. Marzano, D. Alexander *et al.*, "The Evolution of Bashlite and Mirai IoT Botnets," in *IEEE ISCC*, Jun. 2018.
[15] S. N. Matheu *et al.*, "Extending MUD Profiles Through an Automated IoT Security Testing Methodology," *IEEE Access*, vol. 7, 2019.
[16] S. Notra *et al.*, "An Experimental Study of Security and Privacy Risks with Emerging Household Appliances," in *IEEE CNS*, Oct. 2014.
[17] M. Patton *et al.*, "Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT)," in *IEEE JISIC*, Sep. 2014.
[18] S. S. I. Samuel, "A Review of Connectivity Challenges in IoT-Smart Home," in *IEEE ICBDSC*, Mar. 2016.
[19] M. Serror, M. Henze, S. Hack, M. Schuba, and K. Wehrle, "Towards In-Network Security for Smart Homes," in *ARES*. ACM, Aug. 2018.
[20] S. Singh *et al.*, "Clearer than Mud: Extending Manufacturer Usage Description for Securing IoT Systems," in *ICIoT*. Springer, 2019.
[21] V. Sivaraman, D. Chan *et al.*, "Smart-Phones Attacking Smart-Homes," in *ACM WiSec*, 2016.
[22] V. Sivaraman *et al.*, "Network-Level Security and Privacy Control for Smart-Home IoT Devices," in *IEEE WiMob*, Oct. 2015.
[23] N. Soetens, J. Famaey *et al.*, "Sdn-based management of heterogeneous home networks," in *CNSM*, Nov. 2015.
[24] H. Wang, H. Li *et al.*, "Why are Android Apps Removed From Google Play? A Large-Scale Empirical Study," in *IEEE/ACM MSR*, Jun. 2018.
[25] T. Yu *et al.*, "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the IoT," in *ACM HotNets*, Nov. 2015.