Demo: NFV-based IoT Security at the ISP Level

Yehuda Afek*, Anat Bremler-Barr‡, David Hay†, Lior Shafir*, Ihab Zhaika†

*The School of Computer Science, Tel Aviv University, Tel Aviv, Israel.
†School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel.
‡Computer Science Department, Interdisciplinary Center, Herzliya, Israel.

***Abstract*—This demo focuses on demonstrating features of a new system to protect IoT devices in customer premises at the ISP level. The core of the system is deployed as a Virtual Network Function (VNF) within the ISP network, and is based on the Manufacturer Usage Description (MUD) framework, a white-list IoT protection scheme that has been proposed in recent years.**

As MUD is designed for on-premise deployment, the system makes the necessary adaptations to enable its deployment outside the customer premise. Moreover, the system includes a mechanism to distinguish between flows of different devices at the ISP level despite the fact that most home networks (and their IoT devices) are behind a NAT and all the flows from the same home come out with the same source IP address.

Our demo follows closely a proof-of-concept that we have done with a large national level ISP, showing how our system can identify the various IoT devices that are connected to the network and detecting any unauthorized communications.

I. Introduction

In this paper, we demonstrate the features of an ISP level system that protects the IoT devices in a large number of homes, presented in [1]. This scalable, managed service does not require any cooperation or installation on the client premise or on the IoT devices themselves. Furthermore, it monitors the IoT traffic and detects malicious behavior, including outgoing DDoS traffic, without being on the critical path, and it filters bad traffic by ACLs on either the PoP router or the client CPE. The CPE itself is considered an IoT device and traffic destined or that originates at the CPE is monitored as well.

Our system follows a *whitelist approach* for IoT protection: While non-IoT devices such as desktops, laptops, and smartphones connect with many services and websites, IoT devices should typically connect with only a few endpoints (e.g., corresponding with their users, specific cloud services, and some generic services). A recent initiative calls for IoT device vendors to provide a *Manufacturer Usage Description (MUD)* for their products [2], which will allow security tools to monitor the compliance of the device with its usage description. In this demo, we will assume that IoT devices have MUD. Each IoT device specifies a URL for its MUD when connecting to the network (e.g., in the DHCP discovery message as a specific option reserved for this protocol). MUDs, stored as files, consist of whitelists describing the devices' legitimate communication, specified (among other parameters) by the domain names of legitimate endpoints.

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

To protect an IoT device, given its MUD file, it is necessary to resolve the domain names in the list to get the corresponding IP addresses (which might vary in different geographies and at different times), then to monitor the device traffic ensuring it communicates only with these IP addresses and complies with the MUD file (e.g., uses the specific ports and protocols specified in the file). If a deviation is detected, offending connections should be blocked and alerts on suspicious activity are issued. Suggested implementations of the MUD standard as well as other IoT protection systems, implement these steps within the internal/local area network (LAN), i.e., for home networks on the CPE, [3]–[7], though sometimes on a separate device. Our system implements these steps at the ISP level. The system combines an off-path Virtual Network Function (VNF) with the existing ISP's on-path enforcement capabilities. The VNF holds the MUD rules (where domain names are already resolved to IP addresses) and monitors a large number of home networks by these rules. Upon a violation, the corresponding connection is blocked in one of the ISP's on-path routers or switches, e.g., using ACLs.

As home networks apply Network Address Translation (NAT), implying all devices use the same IP address and port numbers are arbitrary, perhaps the major challenge when working outside the LAN is to distinguish between connections originating from different devices. Our design overcomes this difficulty by dynamically installing *packet marking rules* on the home gateway router (often called a Customer Premise Equipment or CPE), using its standard configuration protocol TR-69 [8]. Marking is done on the DSCP header field (thus not introducing extra overhead) only on packets originating from IoT devices. Furthermore, marking rules are installed only when the device first connects to the CPE.

II. NFV-based System Description

The system goal is to ensure that all packets of an IoT device complies with the MUD file rules. This implies that, for each packet, the system needs to decide whether it conforms with a MUD file (or another form of a whitelist), and if not, to block the packet. The MUD enforcement has thus two logical components, monitoring and enforcing. In the *whitelist monitoring* (WLM), it is determined whether a packet/connection complies with a whitelist/MUD file or

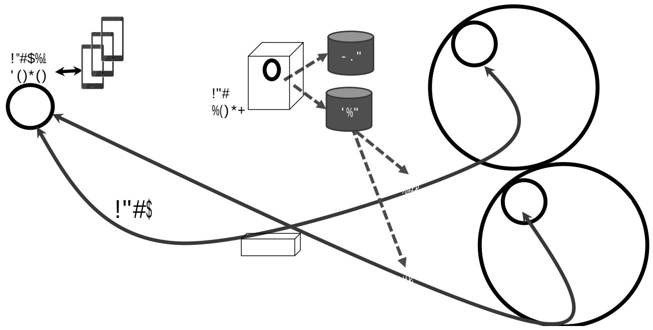

Fig. 1: Illustration of our Framework. The WLM's data plane is marked with solid lines, while the WLM's control plane is marked with dash lines. In this illustration, the WLM VNF protects two LANs that communicate with the Internet through two CPEs. Both LANs have a camera installed in them, which connects to the same service in the cloud. The WLM VNF contains MUD files for all IoT devices in the system. It verifies that both cameras communicate only with endpoints within their MUD file.

not; In the *whitelist/MUD enforcement* (WLE), based on the output of the WLM component, the packet is either dropped or permitted. While WLE must be *on the traffic path* observing every packet originated or destined for IoT devices; the WLM may be performed on a copy of the traffic, thus being off the critical path. In our implementation WLM, is implemented as a VNF that receives only a single packet of each connection, sent from an IoT device, to decide whether the entire connection should be permitted or blocked. See Illustration in Figure 1.

Our implementation consists of a control-plane (inside the WLM VNF as well as control communication with other control-planes, such as on the CPEs and other components, see Figure 1) and a data-plane (in the CPE, the WLE, and the WLM VNF). The control-plane main tasks are to maintain the data-plane's whitelists (e.g., by resolving domain names in MUD files to IP addresses) and to remotely configure CPEs to mark packets of IoT devices. Packet marking, ACL enforcement, and checking whether a connection from an IoT complies with its whitelist, are all done in the data-plane.

.

III. Demo Details

In this demo, we will show some features of our system. It follows a proof-of-concept deployment we have conducted in a large national-level ISP network.

Specifically, we will have the following three components in our demo: **(i) A CPE** (deployed on a Raspberry Pi). The CPE will run OpenWRT [9], a commonly-used open-source operating system of CPEs. We have implemented a TR-69 client with a TR-181 data model for OpenWRT, that supports the required operations enabling it to remotely extract the MUD file URL (when an IoT device connects to the CPE) and to mark packets of this device using a specific DSCP value in their IP header (see [1] for further details). The CPE will connect with IoT devices (through WiFi) and with an *upstream PoP router*. Moreover, it will connect to a commercial Auto-Configuration Server (ACS) through the standard TR-69 management protocol, used today to manage CPEs in the ISP level; **(ii) Upstream PoP router** (deployed on another Raspberry Pi). The role of the upstream router is to duplicate the traffic and send a copy to our VNF. Furthermore, it will block traffic using ACLs, when it receives a notification from our VNF (see WLE in Fig. 1); **(iii) Virtual Network Function**. This is the core component of our system, where whitelist/MUD monitoring is done (see WLM in Fig. 1). As mentioned before, our VNF consists of a control plane and a data plane. The data-plane is implemented using Open vSwitch (OVS) version 2.8.1 [10] with OpenFlow 1.3. The control plane is implemented as applications (in Python) over Ryu—a common open-source OpenFlow controller [11]. Our implementation leverages both the caching capabilities of OVS and its supports of DPDK. We have also deployed a commercial ACS in the cloud, and our VNF communicates with it (e.g., to extract MUD file URLs or to instruct CPEs to mark packets) using the ACS's northbound interface.

We will demonstrate in details the steps taken by our system when a new device joins a home network until we start monitoring it. Then, we will show how our system reacts to a whitelist violation, alerting its users and blocking the traffic. Whitelist violation will be simulated by supplying an incomplete MUD file to the system.

Acknowledgments: This work was partly supported by the Federmann Cyber Security Resarch Center in conjunction with the Israel National Cyber Directorate

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 *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS'2020)*, 2020, accepted for publication.

[2] E. Lear, R. Droms, and D. Romascanu, "RFC 8520: Manufacturer Usage Description Specification," Internet Engineering Task Force, March 2019. [Online]. Available: https://datatracker.ietf.org/doc/rfc8520/

[3] McAfee, " McAfee: Built-in Protection for Your Connected Devices," 2019, https://securehomeplatform.mcafee.com/.

[4] F. Roberts, "Trend micro partners with asus to beef up iot security in homes," *Internet of Business*, Jan 2017. [Online]. Available: https://internetofbusiness.com/trend-micro-asus-iot-security/

[5] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, pp. 76–79, 02 2017.

[6] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *IEEE ICDCS*, 2019, pp. 756–767.

[7] M. Özçelik, N. Chalabianloo, and G. Gür, "Software-Defined Edge Defense Against IoT-Based DDoS," in *IEEE CIT*, 2017, pp. 308–313.

[8] The Broadband Forum, "TR-069: CPE WAN Management Protocol," 2018, Issue 1 Amendment 6. URL https://www.broadband-forum.org/download/TR-069_Amendment-6.pdf.

[9] OpenWrt, "The OpenWrt project," 2020. [Online]. Available: www.openwrt.org

[10] B. Pfaff, J. Pettit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, "The design and implementation of open vswitch," in *NSDI*, 2015, pp. 117–130. [Online]. Available: http://www.openvswitch.org/

[11] Ryu SDN Framework Community, "Ryu SDN Controller," 2017. [Online]. Available: https://osrg.github.io/ryu