Transparent Microsegmentation in Smart Home IoT Networks

Amr Osman
TU Dresden

Armin Wasicek
Avast Inc., USA

Stefan Köpsell
TU Dresden

Thorsten Strufe
TU Dresden

Abstract

Driven by the Internet-of-Things (IoT) and 5G, the growing size and complexity of smart home networks leads to an increased attack surface. Smart home IoT devices are typically online 24/7, have out-of-date firmware, are not regularly patched against the latest security vulnerabilities, and often collect sensitive data and send it to the cloud. In this work we propose microsegmentation as a mean to reduce the attack surface of smart home networks with the assistance of the edge cloud. We implement two network functions that cooperate to enforce fine-grained network security policies in smart homes. One function builds an inventory of all devices and their vulnerabilities. The second utilizes that information to dynamically allocate IoT devices to microsegments, and isolates them from one another using inter- and intra- segment network-level security policies. We evaluated our approach using three different IoT network security metrics and IoT topologies. In the best case, microsegmentation reduces the attack surface exposed to a Mirai-infected IoT webcam by as much as 65.85% at the cost of preventing 2.16% of the otherwise-valid network flows between devices.

1 Introduction

Realizing the vision of a smart home is a key driving force for the adoption of Internet-of-Things (IoT) devices in consumer's homes. Temperature control, lights, intruder alarm, and irrigation are examples for common tasks that are already taken over by IoT devices. Household robotics and Augmented and Virtual Reality (AR/VR) are on the horizon to help with chores and entertain. The common denominator for this new wave of devices is that they are networked systems.

An IoT system typically consists of two parts: the device part placed in the smart home network and another part that lives in the cloud. For collaborating, both need to exchange information in a device-to-cloud model. In addition, some IoT devices might work under a device-to-device model. For instance, a home assistant could forward a user's voice command either by contacting the receiving IoT device directly, or by sending the command to the device's cloud for when it is forwarded to the device. Either way, home networks become much more sophisticated through the increased number of devices as well as their communication activities.

This increased complexity raises security and privacy concerns. Some IoT devices might introduce vulnerabilities, because they do use outdated software versions, or their developers didn't use secure coding practices. Some vulnerabilities come from the fact that the devices use insecure protocols like UPnP. Other IoT devices might violate their user's privacy by scanning the network, or querying other devices without authorization. A proven way to mitigate these issues is to apply network segmentation. Network segmentation has been successfully applied to enterprise networks. However, it has not yet been applied to home networks for two main reasons: until now, home networks never had the scale, and because the administrative burden is too high for the average user.

The new mobile standard 5G enables a whole new set of networking technologies, network virtualization being one of them. The combination of Software-defined Networking (SDN) and Network Function Virtualization (NFV) transforms the network edge into a virtual datacenter. Home networks are one of the first targets of these ongoing virtualization efforts. The capability of defining virtual networks on top of the physical infrastructure enables a whole new wave of innovation in networks.

This paper discusses microsegmentation, a technique to segment home networks with a particular focus on IoT devices. A network segment, or microsegment, is a partition in the network that hosts a set of devices. The lifecycle of microsegments is handled in an automatic fashion such that this process requires minimal interaction (and knowledge) from the user. Devices are placed in microsegments based on their category. Microsegmentation is enabled through Software-defined Networking (SDN), a new paradigm to be used in home networks. In particular, the contributions are

- Lifcycle management of microsegments including their creation, update, and deletion via an SDN controller
- Isolation and filtering of microsegments via OpenFlow rules and actions
- Classification of IoT devices on the home network and using that information to compute placement

The rest of the paper is organized as follows: Section 2 defines microsegmentation and what it means for smart homes. Next, Section 3 explains how microsegmentation can be architected in virtual home networks. In Section 4 the prototype implementation is evaluated. Section 5 refers some related work. The paper concludes in Section 6.

2 Smart home microsegmentation

Microsegmentation is a novel network segmentation paradigm that facilitates granular security policies within network segments as well as between segments. It is different to network segmentation techniques like subnets, Virtual Local Area Networks (VLANs), firewalls and other perimeter defense mechanisms. Those mechanisms are unable to selectively isolate lateral traffic within a perimeter, are static in nature, and are coupled with the underlying infrastructure. Microsegmentation overcomes these challenges by dynamically identifying each device on the network and assessing its security, thus, building a complete device inventory and subsequently placing these devices into functional security groups that are decoupled from the underlying infrastructure. This section highlights the core assumptions and defines the scope of our effort to bring microsegmentation to smart home networks.

2.1 Requirements

In smart home networks, microsegmentation performs identification and isolation of IoT devices that connect to the cloud via the residential gateway. The task of identification encompasses classifying single devices on the home network in pre-defined device classes. Then the isolation task aims to shield communication between sets of devices, either partially, selectively, or totally.

In particular, our approach achieves granular security policies at the network-level through fulfilling these requirements:

1. **Isolation**: controlling communication between devices within each microsegment, between microsegments, and external endpoints in the cloud or Internet.
2. **Scalability**: sustaining a large number of microsegments, IoT devices and home networks.
3. **Edge readiness:** virtual network functions in the edge cloud must seamlessly augment the home network.
4. **Automatic segment allocation**: newly connected devices should be automatically recognized, identified and appropriately put into a microsegment.
5. **Adaptability**: dynamically changing the current set of microsegments configuration at runtime as new devices are added to the smart home.

2.2 Communication setting

We consider a single smart home network that is connected to the Mobile Edge Cloud (MEC). This is aligned with the latest developments on the virtualization of residential gateways (RG) in 5G networks [13]. Inside the home network a mixture of various IoT and non-IoT devices that communicate using IP is present. These devices can be either wirelessly connected or attached to the RG via a wired connection. Also, they communicate either directly peer to peer, or through a broker such as MQTT [9], or other higher-layer IoT protocols based on TCP/IP.

2.3 Threat model

Traditionally, the RG served as a security perimeter. As devices increasingly connect to their backends and exchange information through the cloud, this perimeter is dissolving. Single devices are perforating the perimeter by enabling port forwarding (e.g., via UPnP), holding open connections to their cloud endpoints, or have vulnerabilities in their companion mobile apps. Specifically, IoT devices often engage in machine-to-machine type of communication, are always on, numerous, and hardly supervised and patched against the latest security updates by their owners. All these factors increase the attack surface of the home network.

We consider attackers that might leverage one of the many entry points in the home network and then perform lateral movements to infect other devices, steal information, abuse resources, or cause other damage. After getting past the RG that is mostly the only firewall in a home network, the attacker can connect to any other resource in the home network without restrictions.

3 Software-Defined Secure Isolation

Figure 1. introduces a microsegmentation architecture that satisfies the requirements above. Two main network domains come to play: first, the Smart Home network (SH), and second, the Edge Cloud network (EC) hosting control functions. In support of the SH additional Virtual Network Functions (VNFs), or computation-heavy tasks can be offloaded to the Virtual Private Edge Cloud (VPEC). The VPEC is a virtual cloud network in the edge that is bridged with the SH over a Layer 2 tunnel. Example VNFs include an IoT analytics Micro-Service (MS), an IoT broker (e.g. MQTT), virtual IoT

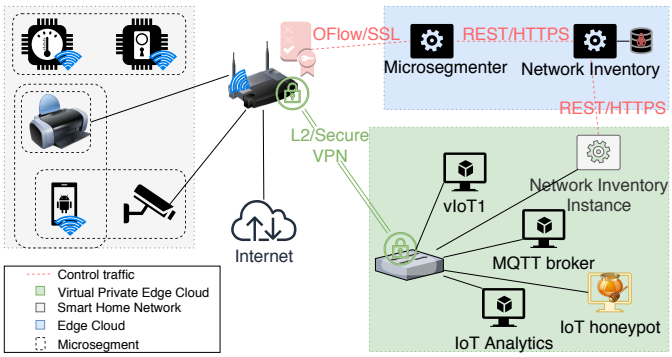

Figure 1: Software-Defined Secure Isolation

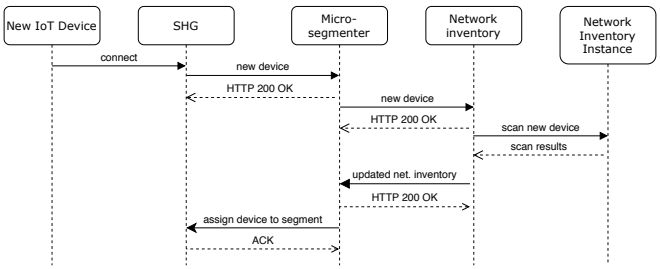

Figure 2: Isolation lifecycle

devices or even an IoT honeypot to absorb and analyze attacks on the SH.

Three main components enable microsegmentation:

1. **SDN-enabled wireless Smart Home Gateway (SHG)**: either in the form of new hardware, or realized as a soft switch running on a traditional RG using opensource firmware (e.g. Open vSwitch (OVS) on OpenWRT)

2. **Microsegmenter VNF**: an SDN application that can re-program the SHG via a protocol such as OpenFlow.

3. **Network inventory VNF**: in contrast to static annotations [8], this component fingerprints and identifies devices on the SH using traffic analysis [11] for example. It also scans the IoT devices for vulnerabilities and conducts exploitability assessments [16].

In our testbed, the SHG was implemented using OpenWRT and OVS on a RG. All local wired and wireless switch ports were attached to OVS. The Microsegmenter was implemented as an application using Ryu SDN framework. The Network Inventory was implemented wrapping the code from the Avast Wifi Inspector that contains an algorithm for device classification. Both the Microsegmenter and the Network Inventory run as Microservices (MS) inside docker containers. Finally, the SHG bridges to the VPEC on Layer 2 through a secure VPN such as L2TP/OpenVPN. All remaining VNFs and virtual IoT devices in the VPEC are implemented as MSes inside docker containers.

3.1 Microsegmentation

The OVS switch runs in default-deny or "secure" mode and the wireless interface runs in client-isolation mode. Hence, every network flow must be explicitly allowed through an OpenFlow (OF) rule on the SHG. OF rules on devices are enforced through their MAC addresses on Layer 2, without the need for port numbers. That facilitates accounting for both wireless and wired devices.

Given a device d, and an IP gateway gw, communication from or to external networks such as the Internet is permitted by inserting OF rules in the format: $MAC_d \leftrightarrow MAC_{gw}$: *NORMAL* where *NORMAL* is the output action as in the OF protocol specification. Pairs of devices within a microsegment communicate using similar OF rules between their source and destination MACs. If two devices communicate over the wireless interface, we explicitly forward the packets back to the wireless interface as appropriate. This additional rule is required because the *NORMAL* OF action doesn't forward an incoming packet to its input port to prevent forwarding loops. Similarly, packets are sent back to the tunnel interface when necessary to facilitate microsegmentation across the SH and the VPEC.

The *cookie* field of OF rules is used to group the OF rules in different segments. This facilitates deletion and modification of segments at run-time. Finally, our implementation of an SDN-based DHCP relay and an ARP proxy handles L2 broadcast such as DHCP and ARP. As a result, broadcast packets are only forwarded to devices within the same microsegment.

3.2 Isolation Lifecycle

The microsegmentation isolation mechanisms are fully automated and can be re-configured without human interaction at run-time, unlike traditional network segmentation approaches. Figure 2 shows the interaction between the components depicted in our proposed architecture to automatically assign a new IoT device to a corresponding microsegment. A similar process happens periodically or when the SHG is restarted. LLDP is employed to detect topology changes and hosts entering or leaving the SH.

4 Evaluation

We've implemented a Microsegmenter to create, update, and delete segments in a network and document its properties in 4.1. The network inventory service is based on a commercial device fingerprinter and vulnerability scanner, the Avast Wi-Fi Inspector. As new devices come on the market, this service is retrained and updated independently from the microslicer that assumes it receives current and correct information from the network inventory. Sections 4.2, 4.4, and 4.3 evaluate the concept of microsegmentation using existing IoT datasets.

4.1 Scalability of microsegmentation

We evaluate the theoretical scalability limits of our approach to microsegmentation empirically. As the number of smart homes and IoT devices connected to the edge clouds is on the rise, our approach should support a large number of network segments, smart home networks and devices.

Number of smart homes: The Microslicer identifies a SHG by a 64-bit field and supports 2^{64} smart home networks.

Number of segments in a smart home: The cookie field of the OF rule format, a 64-bit field, denotes a segment, therefore, each smarthome includes 2^{64} segments.

Number of devices in a segment: Since each MAC address has 48 bits, $2^{48}-2$ devices are supported (after accounting for broadcast addresses).

Number of Openflow rules required: The OF switch starts in default-deny or "secure" mode; each allowed network flow must be explicitly allowed by an OF rule.

Let n be the number of devices in a segment including the gateway. Then to allow bi-directional communication, two rules per device-pair are required, in total $n(n-1)$. Internet communication requires two OF rules: one per direction for each device to reach the gateway, in total $2(n-1)$. In addition, we use 8 additional rules for our internal logic *irrespective of the number of segments*. Given s segments, the total number M of required OF rules is:

$$M = s[n(n+1)-2]+8 \quad (1)$$

Although the polynomial space overhead is not desirable, it is practically preferable to keep the size of each segment small to leverage the isolation benefits of microsegmentation. Nonetheless, we leave this optimization for our future work.

4.2 Effectiveness of microsegmentation

A vulnerability score for smart home networks based on the composition of IoT devices, their CVEs, and their behavior in the network is described in [14]. An attack graph is constructed in a way that even if IoT devices might not interact on the network, there might still be an attack path between devices. Segementation isolates devices, so microsegmentation corresponds to removing edges on the graph.

To show the effectiveness of microsegmentation, we compute network vulnerability for different configurations: (i) baseline, without any microsegmentation, (ii) microsegmentation based on functional groups from [15], (iii) full isolation of each device. For that analysis we use the IoT vulnerability metrics from two recent papers, *exploitability score* [14] and *network exposure score* [1]. Both papers' authors base their numbers on exploitability and impact scores from CVEs and share their data. Figure 3a shows an example based on the data from [14]. An IoT network ('Amazon Echo','HP Inkjet', 'Osram Lightify Pro', 'Belkin WeMo', 'Roku Media Player', 'Philips Hue', 'Ring', 'Google Home') has a baseline exploitability score of 0.96. When microsegmenting 'HP Inkjet', 'Ring', and 'Google Home', the score drops to 0.88 and 0.83. A full isolation scenario, which would also inhibit

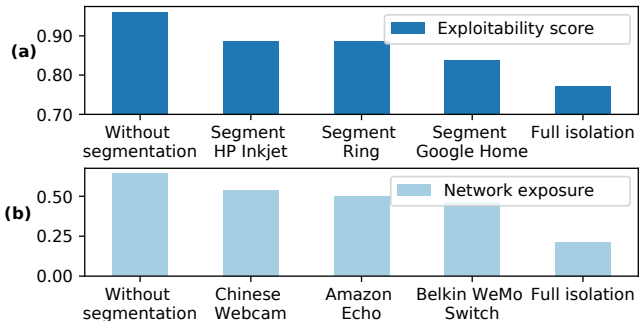

Figure 3: Effectiveness of microsegmentation in an IoT network: reduction of (a) exploitability score [14] and (b) network exposure score [1] as we segment away and isolate one device at a time from left to right. We move from no microsegmentation at all to full-isolation where every device is confined to its own microsegment.

certain desired communication, scores 0.77. A drop in the score represents a reduction of the attack surface and can be used to compare the effectiveness of different segmentation strategies relative to each other. Segmenting 'Ring' has no effect, because there are no known vulnerabilities in the attack graph that can be removed for lateral movement. Thus, segmenting 'HP Inkjet' and 'Google Home' is more effective than segmenting 'Ring'.

Figure 3b illustrates that the same holds true with data and and metrics from [1] using a subset of their topology that consists of 15 IoT devices: 'Amazon Echo', 'Amazon Fire TV,' 'August Doorbell Cam', 'Belkin WeMo Motion Sensor', 'Belkin WeMo Switch', 'Bose SoundTouch 10', 'Chamberlain myQ Garage Opener,' 'Chinese Webcam', 'Google Home', 'LIFX Virtual Bulb', 'Nest Guard', 'Philips HUE Hub', 'Ring Doorbell', 'Samsung SmartTV', and 'TP-Link WiFi Plug'. Here, we transitioned from a baseline of 0.64 to 0.21 which is an effective reduction of the attack surface by 43% when full isolation is employed.

The exposability score of the network is the inverse of the minimum normalized device, mobile app, cloud and network security scores of all devices:

$$Exposability\,score = 1 - \underset{1}{\overset{k}{MIN}}(min(d_i, m_i, c_i, n_i))$$

where k is the number of devices, d is the device score, m is the mobile app score, c is the cloud score, and n is the network score. All: d, m, c, n are normalized in the range $[0-1]$.

4.3 Case study: Mirai

To test our approach in a real adversarial setting, we emulated a network topology that is vulnerable to Mirai [2] based on the dataset provided in [15]. The topology includes more than 28 IoT and non-IoT devices that belong to six different functional groups. The ratio of infected devices can be measured by scanning the network similary to Mirai [2]. Three configurations are tested: (i) baseline, without any microseg-

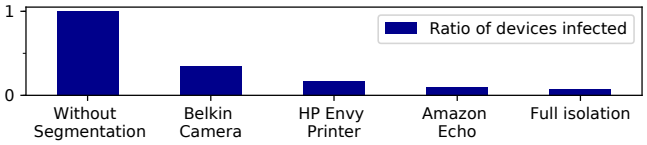

Figure 4: Effectiveness of microsegmentation against Mirai. The y-axis shows the fraction of devices infected, and the x-axis shows the potential attacker starting positions. Note that the starting position does not influence the results in a no-microsegmentation scenario and a full-isolation one.

From	To
HP Envy Printer	Laptop
Samsung Smart Cam	Belkin Motion Sensor
Samsung Smart Cam	Samsung Galaxy Tab
Belkin Motion Sensor	Samsung Smart Cam
Insteon Camera	Samsung Galaxy Tab
Samsung Galaxy Tab	Samsung Smart Cam

Table 1: Network flows blocked due to microsegmentation

mentation, (ii) microsegmentation based on functional groups, (iii) full isolation of each device.

Figure 4 shows the fraction of the network infected in each case. For configuration (ii), we assume that the attacker has control of either 'Belkin Camera', 'HP Envy Printer', or 'Amazon Echo' and starts the scanning the local network to discover other vulnerable IoT devices.

This practical evaluation confirms our theoretical results. Furthermore, we observe that using the Belkin Camera for the lateral movement gives the attacker the highest advantage and network visibility. The reason is that the cameras functional segment contain the biggest number of IoT devices. Compared to the baseline, microsegmentation based on functional groups allowed us to reduce the attack surface by 65.85% had the attacker used the Belkin Camera for lateral movement.

4.4 Impact on functionality

To assure that microsegmentation is not detrimental to the functionality of the smart home, we evaluate the level of disruption to the user using same topology from Section 4.3 and the traffic traces of one day [1](06.10.2016). We replayed the traffic in two configurations: (i) baseline without any microsegmentation, and (ii) microsegmentation based on functional groups. Then, we measured the percentage of network flows that were admitted in (ii) compared to the baseline. The resulting observation is that microsegmentation based on functional groups only deviates from the baseline by 2.16%. This deviation comes from flows that would cross the functional microsegments. Table 1 displays the blocked flows.

The proposed microsegmentation can allow explicit flows such as the ones between the Samsung Galaxy Tab and the Samsung Smart Cam. There were also some suspicious flows in the traffic like a one-way flow from printer to laptop, or a flow between Samsung Smart Cam and Belkin Motion Sensor. In conclusion, this simple microsegmentation strategy is capable of identifying and blocking flows that could be malicious or breaching privacy by crossing functional boundaries.

[1] https://iotanalytics.unsw.edu.au/iottraces

5 Related work

Virtualizing home networks has been the subject of a recent Broadband Forum publication [13]. In this setting, traditional home routers are split between an on-premise, physical device and a set of virtual services on the edge of the ISP network. Feamster et al [5] investigated the effects of outsourcing the security of home networks to such managed services.

Network slicing or segmenting are long-term research topics. Yiakoumis et al propose slicing home networks for the purpose of reducing costs and improving QoS [17]. Going forward, the terms "microslicing" [3], "microsegmentation" [10] or "micronets" [6] were used to describe similar network segregation efforts using SDN. Most recently, this kind of network segmentation has been described in the upcoming NIST publication on home IoT device security [4]. Our work fits in this general research area, but adds automation to how network segments are managed and devices are assigned.

Manufacturer Usage Descriptions (MUD) [8] assume that IoT devices have communication patterns that are known a-priori to the manufacturer. Although IoT device communication can certainly be specified by the manufacturer, the network behavior of PCs and mobile devices depends rather on their users. Microsegmentation could be used to enforce MUD files, but it doesn't necessarily require a communication specification. It is complementary to MUD, because microsegments also protect devices that do not have a MUD file. Contrarily to MUD that provides suggestions, our approach aims to implement directives within the scope of a home network.

The SoK in [1] surveys IoT device security. The paper in [7] surveys the composition of vulnerable, internet-facing home IoT devices and [14] investigates how to measure the security of individual IoT devices. Identifying or fingerprinting IoT devices encompasses automatically identifying the type and class of devices connected to an IoT network [12].

6 Conclusion

Using a novel edge-cloud system architecture, this work implemented microsegmentation to protect smart home IoT networks from internal attacks involving lateral movements. Our work identifies and transparently quarantines malicious devices from accessing the LAN and WAN. Non-malicious devices are automatically classified based on functionality and are accordingly assigned to confined network microsegments. We demonstrated the effectiveness and transparency of our approach by improving key IoT network exploitability metrics on multiple smart home topologies that were microsegmented.

Discussion Topics

Desired Feedback With this paper we are exploring if the community agrees that SDN-based microsegmentation is a feasible security mechanism for smart homes, and overcomes the limitations of other segmentation approaches such as the VLANs and VxLANs.

Controversial Points We expect controversy over the thought that microsegmentation could degrade the network's utility and the reliable operation of its automatic management. Furthermore, the proposed edge-cloud system architecture is debatable. Evading microsegmentation by ARP spoofing and masquerading behind other devices is also a potential topic, although, past literature has shut down this argument.

Discussion Points We seek discussion over the trade-offs and mechanics of more advanced microsegmentation strategies. Also, over the security implications of overlapping microsegments, i.e., devices being part of more than one microsegment at the same time.

Open Issues A more dynamic approach to place devices in microsegments will be our next step. More sophisticated traffic heuristics could help. To enhance space complexity, the OF rules used to implement microsegmentation need to be reduced. Finally, modeling and verifying possible microsegmentation strategies before executing them in the network is also highly relevant.

Depreciating Circumstances Microsegmentation falls apart when it "breaks the Internet" and stalls the vital functions of the smart home. Also, we cannot implement our approach without the recent advances in programmable networks such as SDN.

References

[1] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. SoK: Security evaluation of home-based IoT deployments. In *IEEE S&P*, 2019.

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, and Elie Bursztein et. al. Understanding the mirai botnet. In *USENIX Security 17*, 2017.

[3] Mathieu Boussard, Nicolas Le Sauze, Serge Papillon, Pierre Peloso, and Remi Varloot. Secure application-oriented network microslicing. In *IEEE NetSoft*. IEEE, 2019.

[4] Donna Dodson, Whitney Polk, Murugiah P. Souppaya, and William Barker et. al. Securing small business and home internet of things (iot) devices: Mitigating network-based attacks using manufacturer usage description (mud). 2019.

[5] Nick Feamster. Outsourcing home network security. In *ACM SIGCOMM HomeNets 10*. ACM Press, 2010.

[6] Steve Goeringer. Cablelabs micronets: A new approach to securing home networks. Whitepaper, Cablelabs, 2018.

[7] Deepak Kumar, Kelly Shen, Benton Case, Deepali Garg, Galina Alperovich, Dmitry Kuznetsov, Rajarshi Gupta, and Zakir Durumeric. All things considered: An analysis of iot devices on home networks. In *USENIX Security 19*, pages 1169–1185, Santa Clara, CA, 2019.

[8] Eliot Lear, Ralph Droms, and Dan Romascanu. Manufacturer Usage Description Specification. RFC 8520, 2019.

[9] Roger Light. Mosquitto: server and client implementation of the mqtt protocol. *Journal of Open Source Software*, 2(13):265, 2017.

[10] Olli Mämmelä, Jouni Hiltunen, Jani Suomalainen, Kimmo Ahola, Petteri Mannersalo, and Janne Vehkaperä. Towards micro-segmentation in 5g network security. 2016.

[11] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. Profiliot: A machine learning approach for iot device identification based on network traffic analysis. In *ACM SAC*, 2017.

[12] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. *IEEE ICDCS*, 2017.

[13] David Minodier and Gregory Dalle. Network enhanced residential gateway. Technical Report TR-317, Broadband Forum, 2016.

[14] Josh Payne, Karan Budhraja, and Ashish Kundu. How secure is your IoT network? In *IEEE ICIOT*, jul 2019.

[15] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Trans. Mobile Comput.*, 18(8), 2019.

[16] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen. Identifying vulnerabilities of consumer internet of things (iot) devices: A scalable approach. In *IEEE ISI*, pages 179–181, 2017.

[17] Yiannis Yiakoumis, Kok-Kiong Yap, Sachin Katti, Guru Parulkar, and Nick McKeown. Slicing home networks. In *ACM SIGCOMM HomeNets 11*, 2011.