Towards In-Network Security for Smart Homes

Martin Serror∗, Martin Henze∗, Sacha Hack§, Marko Schuba§, Klaus Wehrle∗
∗Communication and Distributed Systems, RWTH Aachen University, Germany
§FH Aachen University of Applied Sciences, Germany
{serror,henze,wehrle}@comsys.rwth-aachen.de, {s.hack,schuba}@fh-aachen.de

ABSTRACT

The proliferation of the Internet of Things (IoT) in the context of smart homes entails new security risks threatening the privacy and safety of end users. In this paper, we explore the design space of in-network security for smart home networks, which automatically complements existing security mechanisms with a rule-based approach, i.e., every IoT device provides a specification of the required communication to fulfill the desired services. In our approach, the home router as the central network component then enforces these communication rules with traffic filtering and anomaly detection to dynamically react to threats. We show that in-network security can be easily integrated into smart home networks based on existing approaches and thus provides additional protection for heterogeneous IoT devices and protocols. Furthermore, in-network security relieves users of difficult home network configurations, since it automatically adapts to the connected devices and services.

CCS CONCEPTS

• Security and privacy → Network security; Intrusion/anomaly detection and malware mitigation;

KEYWORDS

Internet of Things, Smart Homes, Network Security, Attack Mitigation, Anomaly Detection

1 INTRODUCTION

With the proliferation of the Internet of Things (IoT) [2], all sorts of Internet-connected devices, ranging from simple sensors to more complex controllers and home appliances, find their way into private homes [14]. For the convenience of their users, these IoT devices are either directly or indirectly (via a bridge) connected to the home network enabling continuous monitoring and control through the Internet. This trend, referred to as smart homes, encompasses home appliances, smartphones, heating and cooling, and alarm systems [33]. As a result, users benefit from a streamlined automation of daily tasks, enhanced monitoring and alarm capabilities, and even reduced energy consumption. These benefits have led to the unabated success of smart home deployments, e.g., Gartner predicts that 20 billion devices will be connected to the Internet by 2020, where consumer devices will represent the largest group with around 13 billion anticipated devices [20].

Besides these benefits, however, smart homes introduce serious security challenges [6, 12, 15, 24], which mainly result from the unrestricted interconnection of IoT devices among each other and with the Internet [16]. Since the local communication of IoT devices is often unencrypted [7], devices rely on well-known standard passwords [31], or enable remote access, e.g., via telnet, an attacker can exploit these security flaws to gain unauthorized (root) access to devices [29]. Once the attacker gained access, further attacks are possible, such as eavesdropping on network traffic to access sensitive information, manipulating devices, e.g., deactivating an alarm system, or even installing ransomware [43]. As a result, a single compromised device in a home network, e.g., a malware-laden smartphone [36], often suffices to gain access to other devices in this network. Considering currently deployed IoT devices, an Internet-wide analysis of reachable IoT devices shows that the vulnerability rate strongly depends on the device type, varying between 0.44 % and 40 % [31]. Recent attacks, such as Mirai [1], showed that security flaws of IoT devices have far-reaching consequences, since these devices can be turned into a powerful botnet performing Distributed Denial-of-Service (DDoS) attacks on vital online services.
We observe that these security risks are often exacerbated by the methodology of underlying home network deployments where, once a device is connected to the home network, there are no mechanisms on the network layer that restrict the communication capabilities of this device to the bare minimum required for the intended functionality. For example, if a smart home device such as a web radio gets infected by malicious software, it can establish new connections and attack other devices and servers without limitations. However, conceptually there is no need to allow communication of a web radio with other devices to realize the intended functionality, i.e., streaming audio from a distinct number of Internet hosts. From a different perspective, typical users of IoT devices are often overwhelmed with securely configuring their smart home network. The new possibilities enabled by a large variety of IoT devices and services invites users to connect devices with often lenient default security configuration to their home network, without thinking about the consequences, or simply because they do not know about the potentially involved risks.

In this paper, we address these security risks that are imminent to the current deployment model of smart homes and IoT devices in home networks. To this end, we argue that security in smart homes must be addressed automatically and within the local network in addition to the security mechanisms already deployed directly at the device or service side. Notably, our proposed approach of in-network security offers orthogonal protection to the security mechanisms implemented on the devices by monitoring the local network, detecting suspicious traffic, and preventing attacks from and to IoT devices connected to the home network.

In contrast to the state-of-the-art in this area [4, 8, 18, 34, 35, 37], we believe that the communication of IoT devices must be restricted to the bare minimum required for the intended functionality to proactively prevent security breaches. To this end, we propose to equip each IoT device with a specification of the required communication to fulfill its services. Consequently, all other traffic is blocked and a device’s communication is monitored for non-specification-compliant behavior. Our approach thus differs from existing network security approaches such as intrusion detection [28], as it builds upon explicit permissions to prevent unauthorized network traffic instead of generously allowing all communication first and then trying to detect anomalous behavior.

In particular, our contributions lie in a thorough description of the security threats in current smart homes as well as a clear motivation for in-network security in such deployments (Sec. 2). Based on our analysis, we propose three main components for realizing in-network security:

1. Specification of network compliant behavior for individual IoT devices in smart home networks (Sec. 3);
2. IoT traffic filtering to enforce compliance with the specified communication behavior (Sec. 4); and
3. Anomaly detection to be able to dynamically counter attacks within the smart home network (Sec. 5).

In the following, we explore the design space for each of these components by focusing on existing as well as promising future approaches to realize individual parts of in-network security, before we point out future research challenges (Sec. 6).

2 PROBLEM ANALYSIS

The main challenges for securing smart home networks arise from the heterogeneity of IoT devices regarding their capabilities, modes of operation, and applications [19]. With an increasing number of IoT devices, users are overburdened with configuring and maintaining each device in a secure manner [17]. Typically, it is often unclear what kind of network permissions a specific device should have. As a result, users tend to blindly trust the IoT devices they connect to their home network. In the following, we illustrate the potential security threats in smart homes and derive the necessity for an in-network solution to mitigate these threats.

2.1 Security Threats in Smart Homes

Smart homes introduce serious security threats ranging from exposing sensitive information to unauthorized third parties [15] over hacked devices that are used to launch DDoS attacks against innocent bystanders [1, 3] to physical harm, e.g., by manipulating a smart lock to break into a home [11]. In response, the research community invested enormous efforts to identify, describe, and categorize these security threats [6, 8, 21, 24, 25]. We provide an overview of the most important security threats in smart homes, especially with respect to network communication, alongside an example scenario in the following.

In our (fictitious) example, a user buys an IP camera to monitor the drive way in front of her home. To quickly configure the camera, she uses her smartphone to connect to the camera via Bluetooth. Once the connection is established, the configuration menu of the camera allows the user to select her home network from a list of nearby WiFi networks and subsequently to enter the corresponding password. The IP camera is now connected to her home network and has full access to the local network and to the Internet, just like any other connected IoT device. To allow the user to access her camera remotely, the camera automatically and unknowingly to the user opens a port in the firewall of the user’s router, e.g., using Universal Plug and Play (UPnP).

It is, however, unknown to the user that the camera is globally reachable and that the current firmware image includes several security flaws, e.g., well-known standard passwords [31] or the potential for remote code execution which
Towards In-Network Security for Smart Homes

2.2 The Need for In-Network Security

To augment security in smart homes, we thus require an in-network approach that automatically adapts to the heterogeneity of smart home networks by restricting the communication capabilities of IoT devices without limiting their desired functionality. To this end, we propose a new approach to network security that restricts both internal communication (i.e., with other devices in the same home network) and external communication (i.e., with Internet- and cloud-based services) of individual IoT devices to the extent necessary for delivering their intended functionality. That is, depending on the purpose of the IoT device in the smart home network, we allow certain incoming and outgoing connections of the device that are necessary for its intended functionality, while blocking all other connections.

Introducing this new network security approach serves several purposes. First of all, the unauthorized access to IoT devices that are connected to the Internet is prevented by blocking such connection attempts. Moreover, in the case that an attacker still gained access to an IoT device, e.g., by exploiting a security flaw, the unauthorized access to other devices within the network should be impeded. Existing network security mechanisms, e.g., a firewall, typically operate on the home router protecting the network from unauthorized access from outside. In turn, for connections within the home network such security mechanisms do not take effect, which allows infected devices to easily attack other devices within the network. Therefore, our proposed communication approach only permits those connections within the network that are necessary such that the IoT devices can fulfill their tasks. Hence, we significantly reduce the attack vector of rogue devices within the home network. Finally, unauthorized connections from the home network to other networks are also blocked, e.g., to prevent infected devices to perform a DDoS attack or expose sensitive information that should not leave the home network.

To turn our vision of a new communication approach for in-network security of smart homes into reality, we require three main components as illustrated in Fig. 2: (1) a specification of network compliant behavior for each IoT device to derive what kind of communication has to be performed to deliver the desired functionality, (2) a centralized network component, e.g., the home router, that filters IoT traffic according to the specified intended communication behavior to prevent attacks from and to IoT devices, and (3) an anomaly detection system that enables us to dynamically react to attacks within the smart home network by evolving the specified intended communication behavior.

Our approach is based on the assumption that all communication traffic is routed through a central network router, which is the typical setting in today’s home networks. This,
### Table 1: Example of a service description for a web radio.

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
<th>Source</th>
<th>Destination</th>
<th>Traffic Type</th>
<th>Max. Packet Size</th>
<th>Max. Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>audio stream</td>
<td>requests &amp; ACKs</td>
<td><em>.</em>.<em>.</em>:* example.org:80</td>
<td>example.org:80</td>
<td>periodic</td>
<td>200 Bytes</td>
<td>8 kbit/s</td>
</tr>
<tr>
<td>AAC data</td>
<td></td>
<td><em>.</em>.<em>.</em>:*</td>
<td>example.org:80</td>
<td>periodic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>station info</td>
<td>requests &amp; ACKs</td>
<td><em>.</em>.<em>.</em>:* example.org:80</td>
<td>example.org:80</td>
<td>bursty</td>
<td>200 Bytes</td>
<td></td>
</tr>
<tr>
<td>firmware update</td>
<td>XML data</td>
<td>203.0.113.10:21</td>
<td>203.0.113.10:21</td>
<td>periodic</td>
<td>200 Bytes</td>
<td>8 kbit/s</td>
</tr>
<tr>
<td>remote config</td>
<td>https</td>
<td>cloud.example.org:*</td>
<td><em>.</em>.<em>.</em>:*</td>
<td>bursty</td>
<td>600 Bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>https</td>
<td><em>.</em>.<em>.</em>:*443</td>
<td>cloud.example.org:*</td>
<td>bursty</td>
<td>600 Bytes</td>
<td></td>
</tr>
</tbody>
</table>

To leverage knowledge of network compliant behavior as a foundation to realize in-network security for smart homes, we propose to specify this behavior in the form of communication rules. Communication rules specify the IP addresses (or hostnames) as well as port numbers and communication direction that need to be permitted to realize the intended functionality of a specific IoT device, similar to, e.g., rules for firewalls. Furthermore and in contrast to such simple firewall rules, communication rules also define other essential characteristics of network communication such as packet sizes, packet interarrival times, number of parallel connections, and consumed bandwidth. Each device can have multiple communication rules to account for different intended functionality of a single device, e.g., a web radio (cf. Table 1) typically receives audio streams at a bandwidth of 128 kbit/s but occasionally also needs to download new station lists or firmware updates. Moreover, the user may configure her web radio via a cloud service that interacts with the device via https.

In its easiest form, such communication rules specifying network compliant behavior are directly provided by the manufacturer of the respective device. To account for legacy devices already deployed in smart homes today or manufacturers that refrain from publishing communication rules for their devices, users can rely on a certification agency or the open-source community to provide communication rules, or even provide them by themselves [42]. To this end, recent research shows that IoT devices can be automatically identified by passively observing network traffic [27]. Such fingerprinting techniques can be used to retrieve the respective communication rules from a trusted third party, e.g., a certification agency or an open-source repository, given that the manufacturer does not provide a description of network compliant behavior in the first place.

In its easiest form, such communication rules specifying network compliant behavior are directly provided by the manufacturer of the respective device. To account for legacy devices already deployed in smart homes today or manufacturers that refrain from publishing communication rules for their devices, users can rely on a certification agency or the open-source community to provide communication rules, or even provide them by themselves [42]. To this end, recent research shows that IoT devices can be automatically identified by passively observing network traffic [27]. Such fingerprinting techniques can be used to retrieve the respective communication rules from a trusted third party, e.g., a certification agency or an open-source repository, given that the manufacturer does not provide a description of network compliant behavior in the first place.

Besides identifying individual IoT devices, we can also classify devices into groups based on their communication behavior. As a result, we can provide certain default rules to a whole group of devices with similar communication behavior. For example, when a new IoT device, for which no communication rules are available (not even from a trusted...
third party), connects to the home network, this device can be automatically matched to an existing communication class, e.g., permit only local communication or permit only communication to cloud services. This enables more specified default rules for unknown devices, which can be gradually adapted over time when more traffic data becomes available.

Especially if the communication behavior of a new device does not closely match one of the default classes, it might be necessary to automatically derive the communication rules of this device. To this end, we propose to observe the communication of the new device during a learning phase and create communication rules based on the initial behavior of this device. Here, it is important to cover the different aspects of the device’s functionality (see above) to reduce the overhead for adapting communication rules later on.

Once we have obtained the communication rules for an IoT device through one of the above channels, these rules are centrally installed at the home router that bridges the home network with the Internet. In the case that such a description is not available yet, e.g., because we first need to classify the IoT device based on its communication behavior, we apply a default set of rules. As such, the device is, e.g., not allowed to open a connection to another device or server unless the user explicitly grants permission at the home router.

For a practical realization of such a system, we expect it to be challenging to implement the real-time traffic filtering on commercial off-the-shelf router hardware. Nevertheless, as shown in the following section, there are existing approaches that target efficient traffic filtering. For future work, we thus aim at adapting these approaches to fit the requirements and limitations of the smart home use-case.

4 IOT TRAFFIC FILTERING

Based on the communication rules that specify network compliant behavior for each IoT device, we propose proactive in-network security that aims at minimizing the risk of attacks within the smart home network. This not only includes attacks from outside but also attacks from inside, e.g., by infected IoT devices connected to the smart home network. In the following, we describe how we envision the realization of in-network security based on a Software-Defined Networking (SDN) and flexible packet matching approach that efficiently enforces the communication rules in a heterogeneous environment with evolving attack vectors.

4.1 SDN-Based Attack Prevention

Software-Defined Networking (SDN) realizes central network management, which separates the data plane from the control plane [23]. This facilitates the change of network protocols and policies in existing networks, since only the software of the central controller needs to be adapted. Especially large networks, e.g., in data centers and companies, benefit from SDN due to the reduced complexity and overhead for deployment and maintenance. However, the advantages of SDN have been also considered for automating the security of home networks [10] and thus to relieve users of complex security management tasks.

In alignment with previous research efforts [34, 37] we therefore propose to follow an SDN approach in smart home networks to enforce network compliant behavior as defined by our communication rules (cf. Sec. 3). This approach does not require additional hardware, since the SDN controller can be centrally located at the (programmable) home router. As a first step for IoT traffic filtering, the communication rules for each IoT device need to be converted to flow-table entries for the forwarding tables, specifying for each data flow how it is matched and which actions have to be taken. If, for example, the communication rules for a smart lighting system specify only local communication within the home network, then all packets originating from the lighting system with a destination IP that does not lie within the home network address range are dropped. Besides dropping packets, other actions according to the communication rules are possible, e.g., limiting the traffic rate of an IoT device.

The flexibility of SDN allows to dynamically adapt the communication rules, e.g., when new devices are connected with the home network or the functionality of existing devices is extended. Even further, the enforcement of such rules can be realized on the IoT devices themselves and on the smartphones that interact with these devices [8]. Thus, we can filter malicious network traffic either directly at its source or at least close to its origin. However, the main challenge to realize this SDN-based approach is to correctly filter network packets according to the communication rules of individual devices, which we further elaborate in the following.

4.2 Flexible Packet Matching

The de facto standard protocol for configuring SDN deployments is OpenFlow [38]. However, its inflexible packet matching based on hard-coded header fields makes it unsuitable for quickly evolving protocols [5]. For smart home networks with a broad range of (unknown) IoT devices and protocols, the traditional OpenFlow deployment model needs to be evolved to enable dynamic packet matching. In the following, we shortly present two distinct approaches that enable flexible packet matching in SDN and thus are well-suited for enforcing security rules in smart home networks.

P4. To enable flexible packet matching, Bosshart et al. [5] propose Programming Protocol-independent Packet Processors (P4), a high-level language that allows to specify the behavior of a network switch. More specifically, P4 can be used...
to describe, independently of the target hardware, a parse-
match-action pipeline, i.e., a description of packet matching
sequences, packet processing, and actions, which can all be
modified during run time. Therefore, the P4 compiler first
translates the P4 program into multiple match+action tables,
which are then analyzed against dependencies with the help
of table dependency graphs to enable sequential and parallel
execution. Afterwards, the compiler adapts this represent-
tation for the target soft- and hardware. P4 thus allows to
specify match+action rules independently of the hardware
and, more importantly, to flexibly adapt to evolving proto-
cols and security threats. For in-network security in smart
homes, each IoT device can thus specify its communication
rules for network compliant behavior in a small P4 program
including the required headers, parsers, tables, and actions.
Subsequently, this program is used to filter the traffic of the
IoT device to permit only network compliant behavior.

**eBPF.** Berkeley Packet Filters (BPF) were originally speci-
**fied for efficient packet filtering in the Linux kernel already
in 1993 [26]. Until today, BPF evolved to a powerful tool for
**network monitoring, engineering, and security; especially
since its major revision known as extended BPF (eBPF) [13]
has been included into the Linux kernel in 2014. More specif-
ically, eBPF allows to specify small filter programs, which
are executed in a virtual machine in the operating system
kernel, close to the actual hardware. Since running user code
in the kernel poses certain risks, eBPF includes a verifier that
performs several security and stability checks before
loading an eBPF program [32]. Due to its efficiency and pow-
**erfulness, Jouet et al. [22] propose to use eBPF for flexible
**packet matching in OpenFlow. Similarly to P4, eBPF allows
to match packets independently of any protocol implemen-
tation and is therefore well-suited for the heterogeneous IoT
**landscape. Furthermore, Jouet et al. show that eBPF reduces
the number of flow entries and achieves efficient packet fil-
tering at line-rate. In contrast to P4, changes in the eBPF
**program specification do not entail a recompilation of the
target switch software [39]. For our approach of in-network
security for smart homes, eBPF thus allows to easily imple-
ment and modify communication rules of network compliant
behavior during operation. Such updates become necessary
when new IoT devices are connected with the home network,
or in the case of anomalous behavior, e.g., when a device
gets infected and intends to attack other devices. We detail
our discussion on the need to detect and react to anomalies
in the communication behavior in the next section.

## 5 ANOMALY DETECTION

So far, we described how our proposed approach for in-
network security realizes communication rules for network
compliant behavior (cf. Sec. 3) and enforces these rules in the
smart home network based on the principle of SDN (cf. Sec. 4).
However, taking into account the heterogeneity and the vast
growth of the number of IoT devices, it is possible that new
attack vectors evolve that were not considered in the existing
set of communication rules. Furthermore, the provided com-
munication rules might be incomplete and contain errors or
still allow for exploiting unforeseen attacks. The proposed
proactive security mechanisms of our approach therefore
must be complemented with a reactive approach that dynam-
ically adapts the current smart home configuration. To this
end, we propose an efficient network monitoring system as
well as anomaly detection based on machine learning.

### 5.1 Monitoring IoT Devices

A prerequisite for network-level anomaly detection is an effi-
cient monitoring of the network traffic. In the context
of smart home networks we expect, on the one hand, an
increasing number of IoT devices and consequently increas-
ing network traffic. On the other hand, typical networking
hardware for domestic use, e.g., routers and switches, has
limited computational resources. As a result, the monitoring
approach can only focus on essential packet characteristics,
such as packet headers, lengths, and periodicity, instead of
performing deep packet inspection, i.e., analyzing the pay-
load of individual network packets [18]. Still, such essential
packet characteristics can be used to identify traffic patterns
of IoT devices and thus to compare these patterns against
known malicious behavior. To even further reduce the costs
of monitoring in smart home networks, Sivanathan et al. [35]
propose to monitor the network traffic at flow-level granular-
ity, where not all packets of a data flow are analyzed in detail
to keep the processing overhead low. Consequently, we pro-
pose to first analyze the initial packets of a new unknown
data flow to quickly assess whether this flow is suspicious or
not and to adapt the monitoring accordingly. Furthermore,
we use the collected data post factum [40], e.g., to isolate
a culprit device with the help of the recorded data once an
attack has been detected.

### 5.2 Machine Learning Techniques

Besides detecting anomalies in the smart home network
based on known traffic patterns, a more dynamic approach
consists in using machine learning to classify data flows [30].
An advantage of using machine learning for anomaly detec-
tion is the possibility to detect unknown attacks, which were
not considered in the system yet. A disadvantage is, however,
the risk of misclassification leading to a too restrictive net-
work configuration with reduced functionality. Therefore,
a machine learning approach should only complement the
aforementioned rule-based approach, e.g., by adapting exist-
ing communication rules or reacting on imminent attacks.
As proposed in [4], a learning module can be trained for each data flow during run time based on the monitored traffic (cf. Sec. 5.1). Additionally, known attack patterns, e.g., TCP SYN flooding attacks, can be fed into the learning module as training data [9]. A classification module then classifies data flows based on the input of the learning module, e.g., using decision trees, neural networks, random forests, or support vector machines [9]. In case an anomaly has been detected, the corresponding communication rules are adapted to prevent further risks to the smart home network.

6 CONCLUSION AND FUTURE RESEARCH CHALLENGES

In this paper, we propose automatic in-network security for smart homes to tackle the security challenges introduced by heterogeneous IoT devices and protocols, and to relieve users of complicated security configuration of their home networks. Therefore, we are convinced that each IoT device should describe the communication required to realize its functionality, such that these communication rules can be enforced in the smart home network to realize network compliant behavior. Based on existing SDN-based approaches and efficient packet matching engines, we explore the design space for realizing such a system in the context of smart homes. Furthermore, network compliance behavior-driven in-network security can be complemented with machine-learning techniques to dynamically react to imminent attacks and to adapt communication rules during operation.

For future work, we aim at a prototypical realization of in-network security for smart homes to empirically measure its performance and to show the security benefits of this approach. We are particularly interested in achieving a balanced definition of communication rules, which does not limit the functionality of the IoT device and at the same time achieves a strong level of security for the smart home network. In this context, the automatic classification of IoT devices into different groups based on their initial traffic is of special interest to support legacy devices, where no communication rules are available. Furthermore, we expect that integrating machine learning techniques into the rule-based system will significantly improve the detection rate of malicious network traffic, compared to systems that exclusively apply machine learning without any domain knowledge. Finally, we consider it promising to extend the concepts of in-network security to related but yet different environments, such as the Industrial Internet of Things (IIoT).

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


