SPIN: a User-centric Security Extension for In-home Networks

SIDN Labs Technical Report SIDN-TR-2017-002

Date
June 30, 2017

Page
1/12

Classification
Public

Authors
Cristian Hesselman, Jelte Jansen,
Marco Davids, and Ricardo de O. Schmidt

Contact
sidnlabs@sidn.nl

Contact
T +31 (0)26 352 5500
support@sidn.nl
www.sidn.nl

Offices
Meander 501
6825 MD Arnhem
The Netherlands

Mailing address
PO Box 5022
6802 EA Arnhem
The Netherlands

Abstract

We present our ongoing work on a system to curb the security risks that the Internet of Things (IoT) is widely expected to introduce in smart homes, such as exposing large numbers of vulnerable IoT devices that can be misused for massive DDoS attacks on core Internet systems. Our system for Security and Privacy for In-home Networks (SPIN) extends a user's home network with network-level functions that monitor the security of IoT devices and automatically block their traffic in case of vulnerabilities or compromises. SPIN takes a unique user-centric approach in that it (1) allows users to easily deploy the system through one or more pluggable SPIN devices, (2) protects users' privacy by keeping all processing and threat handling within the home, (3) allows users to configure the system with their security preferences, for instance in terms of traffic blocking behavior. SPIN is also unique because it embraces collaborative security by design by enabling the security community to provide malicious traffic patterns. The contribution of our work is the design of the SPIN system and a first implementation that enables users to block traffic from their IoT devices for privacy protection purposes.

Keywords: Internet of Things, security, privacy, smart homes.

1 Introduction

The Internet of Things (IoT) [1] is rapidly entering our daily lives with networking and computing capabilities more and more being extended to devices and things that we normally do not think of as computers. According to Gartner [2], the number of IoT devices will grow by 31% in 2017 and in 2020 the installed base may reach as many as 20.4 billion IoT devices worldwide.

While the IoT promises to enable many new types of services and applications, IoT devices are often poorly secured and consequently pose a threat to the security and stability of the core systems of the Internet, such as to the Domain Name System (DNS). In October 2016, for example, DNS operator Dyn was hit by a Denial of Service (DoS) attack carried out through millions of IoT

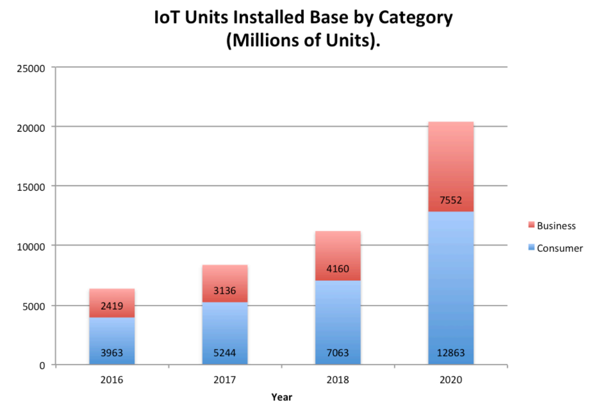


Figure 1. Gartner predictions for IoT growth (adapted from [2]).

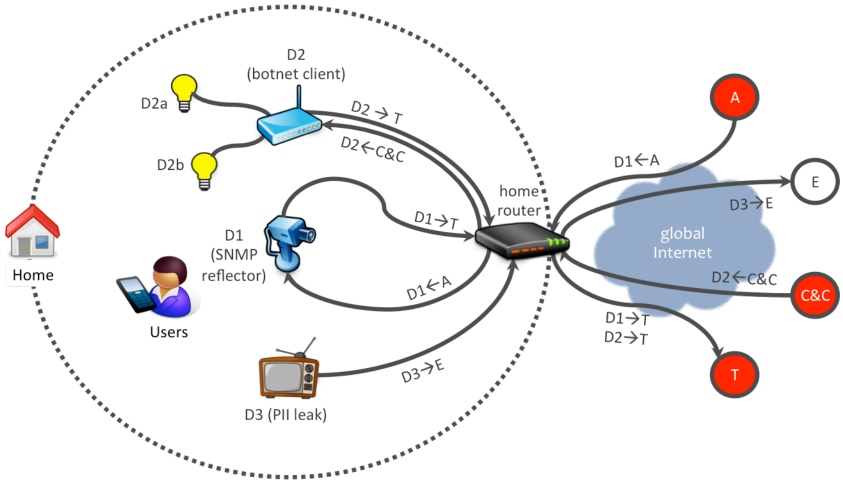


Figure 2. SPIN threat model.

devices compromised with the Mirai botnet [3] that allegedly reached an aggregate magnitude of 1.2 Tbps. Other potential targets of such attacks include operators of Top-level Domains (such as .nl, operated by SIDN), hosting providers, and application service providers.

Another consequence of poorly secured IoT devices is that they compromise the security and privacy of end-users, for instance because they allow attackers to obtain the video feed of a vulnerable baby monitoring system [4]. This jeopardizes users' trust in het Internet and their home environment, in particular because the average end-user typically finds it hard to distinguish well from poorly secured IoT devices and in many cases even lack the interest in these characteristics [7].

A key cause for these IoT security breaches is that manufactures aim at getting (cheap) products to the market as early as possible and sacrifice good security and privacy solutions [5]. Because there is little incentive for them to change this strategy [6], it is very likely that there will always be a large number of vulnerable IoT devices in the wild. This becomes even more alarming when considering that the IoT will grow to billions of devices with an increasing diversity in terms of hardware, firmware and field-upgradability [1]. For DNS operators and other service providers this means potentially larger and more distributed attacks, while for end-users it means more untrusted devices that integrate into their daily lives more intimately.

These developments motivated us to design and implement the system for *Security and Privacy for In-home Networks* (SPIN), which provides network-level security functions that monitor and automatically block vulnerable IoT devices. The goal of the SPIN system is to protect (1) DNS infrastructure operators and other service providers on the Internet from DDoS attacks and (2) to protect users' security and privacy in their homes. SPIN focuses on home networks because they are typically not as well-managed as corporate ones. Our view is that SPIN is an element of a wider integrated approach to IoT security, which for instance also involves setting up a commonly applied security certification mark for IoT devices [8] [9].

Our contributions are the design and initial implementation of the SPIN system. SPIN takes a unique user-centric approach in that it (1) allows users to easily deploy the system through pluggable SPIN devices that automatically monitor and block traffic for groups of IoT devices in the home, (2) protects users' privacy by keeping all processing and threat handling on the SPIN devices in their home, (3) allows users to configure the system with their security control preferences, for instance in term of the system's traffic blocking behavior. SPIN is also unique because it enables the security community to provide traces of malicious traffic, thus extending the systems' threat detection capabilities. Together, this sets our work apart from similar systems, such as those of [4] and

[10].

In the remainder of this paper, we discuss our ongoing work on the SPIN system. In Section 2, we discuss the threat model that guides our work. Section 3 discusses our design goals and we describe the SPIN architecture in Section 4. Section 5 presents the implementation of our proof-of-concept and Section 6 outlines the differences of existing approaches to ours. Finally, in Section 7 we draw our final considerations and discuss future work.

2 Threat Model

The SPIN system focuses on detecting reflective distributed DoS attacks, compromises of IoT devices, and leaking of personally identifiable information. Figure 2 shows an example of each of these attacks and the effects they have on the home network and services on the Internet.

Distributed Reflection Denial of Service (DRDoS): In a DRDoS attack, an adversary uses a large number of different reflectors in the Internet (e.g., open resolvers) to overwhelm an attack target with traffic. The attacker sends spoofed requests to the reflectors, which reply to the target instead of to the requestor (attacker). An attacker can amplify the attack by requesting the reflectors to use responses that are much larger than the original requests.

D1 in Figure 2 is an example of a device that acts as a reflector in an SNMP-based reflective amplification attack. D1 receives SNMP requests from adversary A and sends responses to target T. In this particular example, the manufacturer configured D1 to respond to SNMP requests by default. This functionality might also be enabled for other protocols that can be abused in reflection and amplification attacks, such as NTP and the DNS protocol [11] [12].

Device compromises: A device is compromised when an adversary gains unauthorized access to it. The adversary then uses the compromised device for malicious purposes, such as obtaining user credentials, or adding the device to a botnet to take part in DDoS attacks [3] [13]. Compromising a device can be done in various ways, such as through (weak) password guessing [14], a same site scripting attack [15], a DNS rebinding attack [16], or by manipulating the device's access control list [4].

Figure 2 illustrates an example in which adversary A obtains unauthorized access to device D2, which is a Zigbee-Ethernet bridge connecting light bulbs (D2a and D2b) to the home network. A installed a botnet client on D2, which starts sending traffic to target T. This will likely be an (external) IP address that D2 does not normally connect to. D2 also receives traffic from unusual (external) IP addresses, suggesting communication with a botnet command and control (C&C) [17].

Information leaks: Devices on the home network may run firmware that shares personally identifiable information with servers on the Internet [18], perhaps without user's consent. In the example of Figure 2, device D3 is a smart TV that covertly sends information to endpoint E. E could for instance be a social media service, an ad network that profiles the user's viewing behavior, or a server of the TV manufacturer interested in the users' viewing behavior.

3 Design Goals

The goal of the SPIN system is to provide a user-centric security extension for home networks. The system's main functions are (1) to detect anomalies on the home network, (2) to automatically block suspicious inbound and outbound traffic to/from IoT devices, and (3) to inform the end user about the system's actions and anomalies found. These functions are conceptually similar to a traditional firewall, except that SPIN aims at serving end-users instead of IT professionals and that it protects heterogeneous IoT devices rather than general-purpose computers such as PCs or laptops.

Our key design goals for the SPIN system are: in-home deployment (Section 3.1), enable monitoring of any IoT device (Section 3.2), modular deployment (Section 3.3), user configurability (Section 3.4), and support for collaborative security (Section 3.5). We present the resulting SPIN architecture in Section 4.

3.1 In-Home Deployment

Our first design goal is to enable users to deploy the SPIN system using equipment in their home. This keeps the information that the SPIN system collects to carry out its task (e.g., traffic measurements and a longitudinal model of network topology) as well as the user traffic itself within the home, which is essential for a system that aims to improve users' security and privacy.

An alternative is to bind the SPIN functions to a third party located outside the home network [4] or to every IoT device in the home. The former however requires sharing of information on the network's topology with third parties (e.g., a cloud service), while the latter would likely be hard to achieve because of legacy devices and the limited resources typical of IoT devices (cf. the IETF's work on security protocols for IoT devices [19]).

We run all of the SPIN functions (threat detection, automatic traffic blocking, user notification) as a service on a device that users can easily plug into their home network, such as a mini home router or a bridge to a non-IP radio network. SPIN devices need to be relatively resource-rich to run the SPIN service and need to be able to monitor the network's traffic, for instance by putting it in the data forwarding path (e.g., on the home router of Figure 2) or by connecting it to the network in promiscuous mode.

3.2 Monitor any IoT Device

Our second design goal is that the SPIN system should enable users to monitor any IoT device, thus allowing them to easily deploy the system without having to go through device-specific procedures, such as loading threat detection modules for specific types of sensors into the SPIN system.

To accomplish this, we design the SPIN service to operate at the network-level, which means that it analyzes network traffic (e.g., IP headers, packet lengths, and DNS payloads) and analyzes the generic properties of IoT devices, such as security configurations (e.g., default passwords) and if they are susceptible to misuse in reflection attacks. The SPIN service thus does not rely on application-level information in the payloads or IP packets (other than DNS payloads) or on device-specific security properties, such as the proprietary pan-tilt-zoom API of specific IP cameras.

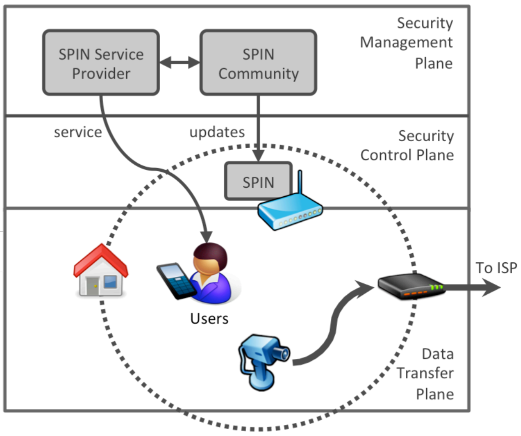


Figure 3. SPIN business roles.

The advantage of network-level security is that it is generic and works for a wider range of IoT devices than device-specific approaches. This is an important feature in the IoT, since devices are much more heterogeneous than in the traditional Internet of PCs and laptops. Our network-level approach is similar to that of Sivarama et. al. [4], except that they deploy their system in the cloud rather than using devices in the home network.

Operating at the network-level also has the advantage that the SPIN service will be able to transparently work with encrypted traffic because it does not depend on packet payloads. This will further increase users' trust in the system as it will not attempt to decrypt traffic. It also makes the SPIN system more future proof, because more and more IoT devices will encrypt their traffic, for instance as a result of new encryption protocols for IoT devices [19] and the mass uptake of Let's Encrypt certificates [20].

A future challenge for SPIN is the interoperability with systems that analyze device-specific and proprietary security characteristics, such as the pant-tilt-zoom functions of IP cameras. Such systems augment our network-level approach, but are outside the scope of our current work.

3.3 Modular Deployment

Our third design goal is to enable users to deploy the SPIN system in a modular and incremental way by using multiple SPIN devices, each protecting a subset of the IoT devices in the home.

The advantage of this approach is that it reduces the impact of reflection (DoS) attacks and device compromises because SPIN devices can block outbound malicious traffic closer to an affected IoT device. For example, if Zigbee router (D2 in Figure 2) and the home router are both SPIN devices, then the Zigbee router would be able to isolate malicious traffic coming from D2a or D2b to the Zigbee part of the network. Users can also create different types of IoT device groups, such as based on network technology (e.g., Zigbee or Wi-Fi), or on location (e.g., a group for each floor of the house).

A modular SPIN setup also allows the SPIN service to run on devices that connect to non-IP parts of the network. For example, the SPIN service on the Zigbee router has detailed knowledge about the properties of Zigbee devices (D2a and D2b) and their respective traffic flows. The SPIN service on the home router will not have this information because Zigbee is a link-level technology, which means that the Zigbee router multiplexes all traffic from the Zigbee devices onto one IP connection. Multiple SPIN devices thus together provide a more detailed view of the user's home network, which for instance allows for improved threat detection and enables an IoT service provider (see Section 3.4) to better help users if they concede in sharing information about their network.

A challenge is the development of a SPIN protocol that enables multiple SPIN devices in the home to share their partial views of the network. We envision that in these setups a limited number of devices will run a full SPIN service, while others will act as SPIN agents that only monitor and block traffic and rely on the SPIN services for data analysis and decision making.

3.4 User Configurability

Our fourth design goal is that the SPIN system should provide central facility that enables users to configure the system with their security and privacy control preferences across SPIN devices, specifically pertaining to the following four areas:

Level of automation: Users should be able to configure to what level they want the SPIN system to automatically block IoT devices. We expect that most users will want to the system to operate automatically and only receive indications of the blocks that the system put in place. Some (expert) users may however also want to manually control this behavior.

Devices to monitor: Users should be able to configure which IoT devices in their home they want to put under SPIN security control and which devices they want to secure through other means. For example, many users have high-end general-purpose computers at home such as laptops, PCs, and tablets that are protected through virus scanners and do not need to be monitored by the SPIN system.

Use of network measurements: Keeping in line with the SPIN system's local in-home deployment model (see Section 3.1), users should be able to define how SPIN should store network and device measurements, for example in terms of retention time.

Device-specific security services: The SPIN system forms a first line of defense because it operates at the network-level (see Section 3.2) and needs to interface with other external services to help users resolve device-specific threats.

We foresee a business model around the SPIN system for this purpose, in which a SPIN Service Provider (see Figure 3) provides IoT security services to users, for instance to help them updating the firmware of a specific IoT device in response to a notification from the SPIN system. This may require users to share information about their network with their SPIN Service Provider on a case-by-case basis, which they explicitly need to concede to because the SPIN system keeps such information in the home (see Section 3.1). SPIN Service Providers should be trusted entities, such as a user's Internet access provider, the manufacturer of an IoT device, or a new type of business.

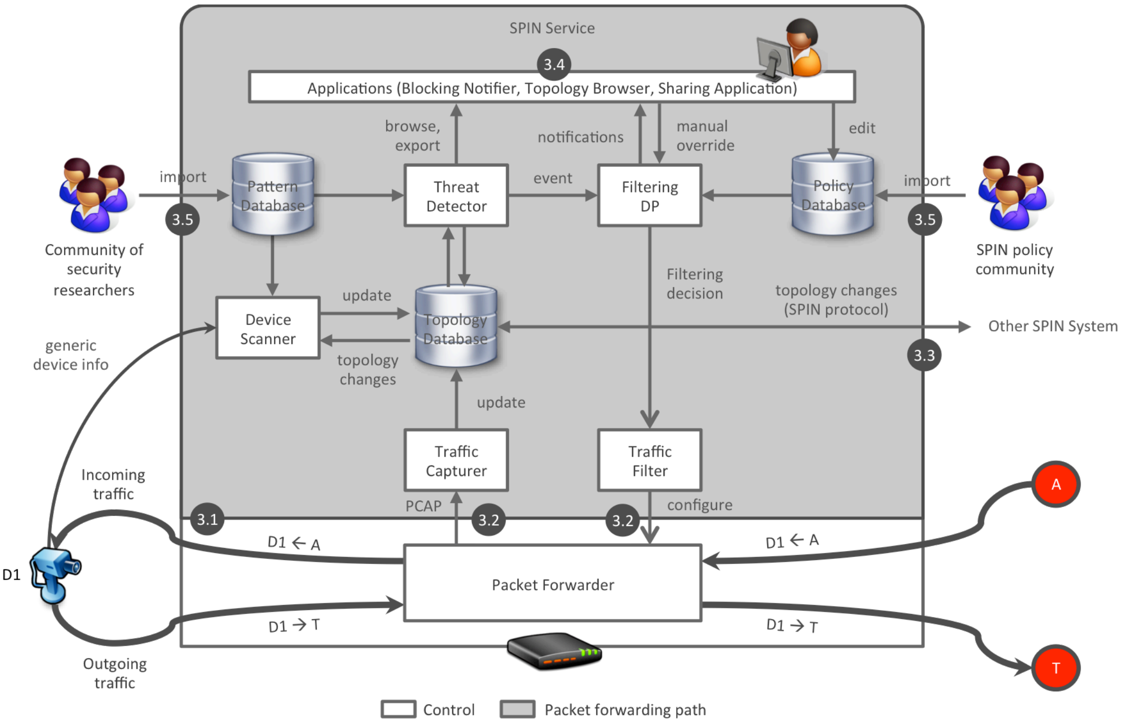


Figure 4. Architecture of a SPIN device.

The SPIN Community maintains and updates the SPIN open source software and provides traces of malicious traffic (see Section 3.5). The SPIN Service Provider works with them, for instance to donate software or to leverage the community's expertise to provide services.

3.5 Support Collaborative Security

Our fifth and final design goal for SPIN is to support collaborative security [1], which we believe is the only viable way to attain a reasonable level of security in the IoT.

Collaborative security in the context of SPIN means that the system trusts the SPIN Community (e.g., device manufacturers, CERTs, and ethical hackers) to provide descriptions of malign traffic patterns to be watched for within the home network (Section 3.2). This requires some form of governance regarding which descriptions to accept into the system, for instance through a global and multidisciplinary "IoT Security Experts Group". It also requires an open interface that supports standardized languages to describe traffic patterns, for instance based on the rule syntax of Snort [21] or OpenBSD packet filtering [22].

4 SPIN Architecture

Figure 4 shows the SPIN architecture, which revolves around the SPIN security service (shaded area). The service runs on in-home network elements such as routers and bridges and interfaces with their packet forwarding engines (transparent area). The bottom part of Figure 4 shows the example of device D1 (camera) of Figure 2, which is engaged in a reflection attack. The numbers (3.x) in Figure 4 correspond to our design goals of Section 3.x.

The SPIN service consists of the following components: a Filtering Decision Point (Section 4.1), a Threat Detector (Section 4.2), a Topology Database (Section 4.3), a Device Scanner (Section 4.4), a Traffic Capturer (Section 4.5), a Traffic Filter (Section 4.6), a Pattern Database (Section 4.7), a Policy Database (Section 4.8), and User Applications (Section 4.9).

4.1 Filtering Decision Point

The Filtering Decision Point dynamically decides when to block an IoT device's incoming or outgoing traffic. For example, the SPIN service on the home router of

Figure 2 would decide to block SNMP requests from the adversary (A) to the networked camera (D1) as well as outbound responses from D1 to T, thus partially isolating D1 because it can no longer participate in the reflection attack on T. Similarly, the SPIN service would respond to D2's compromise by blocking traffic to D2 from suspicious IP addresses and would block outgoing IP packets that the TV (D3) sends to endpoint E on the Internet to prevent the device from leaking personally identifiable information.

The Filtering Decision Point responds to events from the Threat Detector (Section 4.2) and enforces its decisions through the Traffic Filter (Section 4.6). The Filtering Decision Point might also support other filtering decisions, such as rate limiting.

4.2 Threat Detector

The Threat Detector continuously analyzes traffic patterns on the home network and checks them against the network's "normal" traffic patterns and against known malicious traffic patterns. It for instance notifies the Filtering Decision Point if it notices that a connected light bulb suddenly starts sending unusual volumes of DNS traffic to a specific destination or when it starts port-scanning. The Filtering Decision Point will subsequently decide on whether or not to block the traffic of that specific light bulb.

The Threat Detector consists of a collection of data analysis algorithms. They retrieve and update a model of the network topology from the Network Topology Database (Section 4.3), and receive descriptions of malicious traffic patterns from the Traffic Pattern Database (Section 4.7).

4.3 Topology Database

The Topology Database is a key component in the SPIN architecture and stores a longitudinal description of the home network's topology in the form of a sequence of time stamped graphs $G_0 \ldots G_N$, with G_t describing the topology of the network at time t. Each graph G_t consists of: nodes (e.g., IoT devices and devices running the SPIN service) and the external services that they communicate with and an edge for each pair of nodes that exchange traffic. Nodes have attributes that describe network-level properties that are directly measurable (e.g., any enabled reflector ports), and higher-level assertions based on these measurements (e.g., the probability that the device has been compromised or if it has a weak password). The attributes of the edges describe the distribution of traffic between nodes per graph as well as the traffic distribution across graphs.

The Topology Database shares its graphs with other SPIN devices in the same home and also receives graphs from them, thus allowing SPIN devices to collaboratively build up a complete overview of the home network. Interactions between different instances of the Topology Database take place via the SPIN peer-to-peer protocol.

The Threat Detector (Section 4.2), the Device Scanner (Section 4.4) and the Traffic Capturer (Section 4.5) update the graphs in the Topology Database based on network measurements, device measurements, and data analysis algorithms, respectively.

4.4 Device Scanner

The Device Scanner actively probes the devices on the network to collect their directly measureable network-level properties, such as whether they have open ports, whether they are susceptible to reflection attacks, and whether they can be logged onto with an easy-to-guess password. The Device Scanner also acts as a virtual adversary in that it replays attack traffic from the Pattern Database (Section 4.7) and checks if the device's output produces a pattern that indicates that the device is vulnerable or has been compromised.

The Device Scanner responds to changes in the graphs that the Topology Database (Section 4.3) stores, for instance when a new device appears on the network.

4.5 Traffic Capturer

The Traffic Capturer monitors all traffic passing through the SPIN device and updates the edges of each graph G_t in the Topology Database (Section 4.3). It obtains its traffic measurements from the SPIN device's forwarding engine.

The Traffic Capturer ignores the traffic of specific IoT devices that the user does not put under SPIN security control, for instance if the device is a laptop that the user has secured through other means (e.g., a virus scanner).

4.6 Traffic Filter

The Traffic Filter blocks IP addresses for inbound and outbound traffic of IoT devices, based on filtering decisions from the Filtering Decision Point (Section 4.1). As a result, the Traffic Filter partially isolates IoT devices by instructing the packet forwarder to drop packets for certain IP addresses.

The Traffic Filter can also limit traffic rates, but this specific functionality is planned as future work.

4.7 Pattern Database

The Pattern Database contains descriptions of flows of network traffic (flow specs) that lead to or result from specific device compromises on the home network. For example, traffic patterns within the SPIN context can be similar to those used to identify anomalies within SCADA (Supervisory Control and Data Acquisition) networks. Like in SCADA networks, we expect that IoT devices in home networks will have strong traffic patterns. This allows for whitelisting of traffic flows [23], where flows that do not follow known patterns are marked as suspicious. Barbosa et. al. [24] also explore patterns in the periodicity of SCADA traffic, in which devices are expected to communicate within well-determined intervals.

The Pattern Database gets flow specs from the security community (see Section 3.3) in through an open and standardized interface (Figure 3).

4.8 Policy Database

The Policy Database contains descriptions of configurable filtering policies, which state when to block the incoming or outgoing traffic of an IoT device. They can for instance be expressed using an extension of Snort rules [21] or OpenBSD's syntax for packet filtering rules [22].

The condition of a filtering policy is an event that the Threat Detector (Section 4.2) discovers, such as open ports on devices, easy-to-guess passwords, an IP address that an IoT device normally does not connect to, or a match on a malicious network pattern. The policy action defines what filtering actions to take when the event occurs.

The SPIN system uses a combination of pre-defined policies, user-defined polices, and policies received through collaborative security actions (Section 3.3).

4.9 User Applications

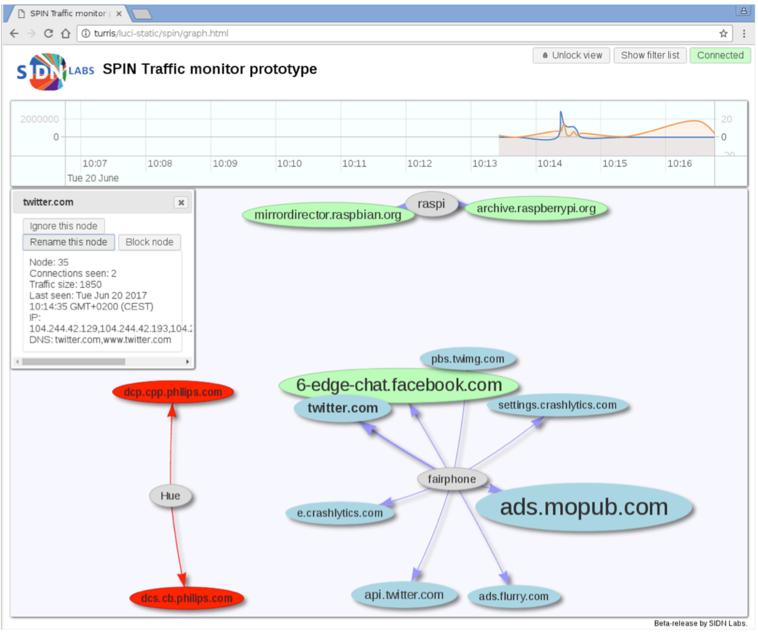
We distinguish three SPIN applications, which enable users to interact with the SPIN system:

Blocking notifier: informs users of the blocking decisions that the Filtering Decision Point made, for instance when it configured the Traffic Filter to block incoming traffic from adversary A in Figure 2, thus preventing device D1 from further participating in the reflection attack on target T.

Topology browser: enables users to easily review and update the network topology graphs that the Topology Database (Section 4.3) stores. For example, the user of Figure 2 would utilize the topology browser to manually block device D3 (smart TV), preventing it to connect to endpoint E. The browser also enables users to define which IoT devices the SPIN system should protect and how the Topology Database should store the measurements of the Traffic Capturer and the Device Scanner based on the user's privacy preferences.

Sharing application: enables users to share network graphs with a SPIN Service Provider (Figure 3). This will typically happen when the SPIN system partially blocks an IoT device, for instance as a result of the device's outgoing traffic matching a malicious pattern. Although part of our SPIN design, the sharing application and the SPIN Service Provider are out of the scope of our current implementation.

Specific user interface designs for the SPIN applications are outside the scope of our work as well, but prior art exists in this area [25].



Node: 35
Connections seen: 2
Traffic size: 1850
Last seen: Tue Jun 20 2017
10:14:35 GMT+0200 (CEST)
IP:
104.244.42.129,104.244.42.193,104.2
DNS: twitter.com,www.twitter.com

Figure 5. SPIN prototype (Topology Browser).

5 Implementation

We have developed a working prototype of the SPIN service, which focuses on blocking traffic to and from IoT devices for privacy protection purposes (cf. device D3 in Figure 2). Our implementation includes a first version of the Traffic Capturer (Section 4.5), an in-memory version of the Topology Database (Section 4.3), and the Topology Browser (Section 4.9). The source code is available in the form of an open source package for OpenWRT devices [26], which we bundled with our Valibox firmware for DNSSEC validation [27].

The prototype consists of three modules: (1) a kernel module to capture or block traffic (the Traffic Capturer and Traffic Filter combined), (2) a user-space module to control the kernel module, aggregate the data and distribute it via an MQTT message broker to any clients or front-ends (the Topology Database), and (3) a Javascript dashboard that can be used from a browser in the local network (the Topology Browser).

The experimental setup that we used to validate our prototype consists of three IoT devices (a Philips Hue lamp, an Android mobile phone, and a Raspberry Pi running Raspbian) and a GLiNET AR150 mini-router that acts as the SPIN device and that runs the SPIN service.

Figure 5 shows a screenshot of the SPIN dashboard (Topology Browser), which displays the network graphs that our in-memory version of the Topology Database stores. The nodes in Figure 5 represent IoT devices or services on the Internet and the arrows indicate a "sends traffic to" relationship. The grey nodes are devices within the local network, while outside destinations are represented by blue nodes (traffic within the last 10 minutes), green nodes (most recent traffic), or red nodes (blocked traffic). The nodes are identified by an IP address, a MAC address, a domain name, or a user-given name, depending on available information. If one IoT device has multiple domain names or IP addresses, then the SPIN dashboard shows them as one node, and the user can review them by selecting the node.

The SPIN dashboard enables the user to manually block certain devices or remote addresses by denying all traffic to and from their respective nodes. The dashboard also allows users to rename nodes so they have more memorable names, so that they are easily identifiable with user-friendly, easy-to-remember names (e.g., "fridge", "thermostat", or "living room TV"). The user can furthermore configure the SPIN device to completely ignore certain (IoT) devices, thus removing them from SPIN security control. Users for instance use this feature to ignore devices whose security they manage themselves, such as laptops or PCs. Such high-end devices typically also generate a lot of traffic, which fills up the dashboard and reduces its readability. By default, a SPIN device only monitors other devices and does not monitor itself, but the user can remove that restriction and observe the traffic of the SPIN device as well.

Since SPIN is a network-level system (Section 4.5), the Traffic Capturer only processes and reports traffic metadata (IP addresses, IP protocol, port and traffic size) and does not inspect packet payloads. The only exception is DNS query responses, which SPIN inspects and reports separately, so that the dashboard can show domain names instead of IP addresses for remote hosts.

Our software targets OpenWRT-based devices, but can also be built and run on other Linux-based systems. A screencast of the SPIN dashboard for a SPIN device running in an actual home network is available in [28].

6 Related Work

Leverett et al. [9] discuss the role of standardization and certification within the IoT ecosystem and how the European Union regulatory framework should look like in a decade from now, having the underlying principle of maximizing social welfare by reducing risk. Besides standards and traditional testing of devices, the authors describe the need for monitoring systems to avoid security breaches and vulnerabilities being exploited with malicious intent. In addition, Bugeja et al. [29] survey security and privacy challenges in home networks at the levels of service, communication and device. They also present potential mitigation approaches for these challenges. Again, the importance of traffic and device monitoring to ensure security and privacy in home networks is discussed.

There exist multiple initiatives from both academia and industry that address the problem of protecting users' security and privacy in home networks. Simpson et al. [10] propose a framework to handle security within a home network that, by observing all traffic to and from the home network, can autonomously intervene once a threat is identified. Like our SPIN system, their proposal is modular and each module can tackle a different sort of threat. Unlike SPIN, however, their framework does not put the user in control of what goes on in the home network. In addition, while SPIN has a distributed character, the framework in [10] solely operates on the home gateway.

Sivaraman et al. [4] combine network monitoring, Software Defined Networking (SDN), and dynamic security rules to ensure security and privacy within home networks. Their proposal is different from ours in that they centralize IoT security control with a third party outside the home, which interacts directly with devices in the home network to enforce security policies. Our SPIN system, on the other hand, fully runs locally on devices in the home.

On the industry side, Turris Omnia [30] is a project by CZ.NIC that aims at helping users protecting their home network using a special-purpose router. The router reports potential threats in the traffic flows between the Internet and the home network to a centralized control point, which decides on what steps to take and that also informs other routers within the Turris network. This is unlike the SPIN system, which handles threats locally at the home network. Another difference with Turris is that the SPIN system provides users with a live graphical view of the network topology and traffic and allows them to block traffic, which is not possible with Turris.

The Dowse project [25] by Dyne.org implements a transparent proxy that focuses on privacy protection. This proxy allows users to visualize connections that happen within the home network. Their motivation is the growth of unconscious abuse, i.e., connections and information exchange without user consent. The work of Dowse is complementary to the SPIN system because they focus on experimenting with user interaction, whereas we focus on the underlying network-level system.

Finally, Dojo [31] by Dojo-Labs is a device that claims to protect the home network from malware, viruses and cyber attacks, while keeping privacy intact. Dojo is however a closed proprietary product, which makes it impossible for us to outline its differences to SPIN.

7 Conclusions and Future Work

We proposed the system for Security and Privacy in In-home Networks (SPIN), which reduces the security risks that the billions of poorly secured IoT devices will pose to core Internet systems, service providers, and end-users. SPIN extends a user's home network with network-level functions that monitor and dynamically block the traffic to and from vulnerable or compromised IoT devices. The system follows a unique user-centric approach: (1) it enables users to easily deploy the system through one or more SPIN devices, (2) it keeps the user's data and threat handling within the home network, and (3) it enables users to configure their security control preferences. We discussed the SPIN system's design and a first validation of the SPIN concept in the form of an application that enables users to block for privacy protection purposes.

Our future work is to further develop the technologies to refine the SPIN architecture, for instance for the longitudinal storage, representation, and visualization of SPIN network graphs, threat detection algorithms, description languages for traffic flows and blocking policies, and SPIN protocol that enables multiple SPIN devices to share their view of the home network. We will further extend the SPIN architecture, for example to include the protection of SPIN devices themselves and to enable easy interaction with the security community. We will also evaluate the SPIN system in several ways, for instance in terms of its threat detection accuracy and usability by non-experts. Finally, we plan a small-scale pilot with the next version of our prototype to collect feedback from real users and to validate the SPIN system in an operational setting.

References

[1] K. Rose, S. Eldridge, and L. Chapin, "The Internet of Things: an Overview", ISOC, Oct. 2015, https://www.internetsociety.org/doc/iot-overview

[2] "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016", press release, Feb. 2017, http://www.gartner.com/newsroom/id/3598917

[3] "Dyn Analysis Summary Of Friday October 21 Attack", blogpost, http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[4] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, Olivier Mehani, "Network-level security and privacy control for smart-home IoT devices", IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, United Arab Emirates, Oct 2015

[5] B. Dickson, "How insecurity is damaging the IoT industry", Oct 2016, https://bdtechtalks.com/2016/10/23/how-insecurity-is-damaging-the-iot-industry/

[6] B. Schneier, "Click Here to Kill Everyone", Jan 2017, http://nymag.com/selectall/2017/01/the-internet-of-things-dangerous-future-bruce-schneier.html

[7] "IoT? I don't care, SC Magazine US, Feb. 2 2017", https://www.scmagazine.com/iot-i-dont-care/article/634990/

[8] "ICSA Labs IoT Certification Program announcement", May 2016, https://www.icsalabs.com/press-release/icsa-labs-rolls-out-internet-things-security-testing-program

[9] E. Leverett, R. Clayton, and R. Anderson, "Standardisation and Certification of the 'Internet of Things'", 16th Annual Workshop on the Economics of Information Security (WEIS17), La Jolla, USA, Jun 2017, https://www.cl.cam.ac.uk/~rja14/Papers/weis2017.pdf

[10] Anna Kornfeld Simpson, Shwetak N. Patel, Franziska Roesner, Tadayoshi Kohno, "Securing Vulnerable Home IoT Devices with an In-Hub Security Manager", 1st International Workshop on Pervasive Smart Living Spaces (PerLS 2017), co-located with IEEE PerCom

2017, Mar 2017, https://www.cs.washington.edu/tr/2017/01/UW-CSE-17-01-01.pdf

[11] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and its Potential for DDoS attacks – a Comprehensive Measurement Study, ACM Internet Measurement Conference 2014 (IMC 2014), Nov 2014, Vancouver, BC, Canada, https://wwwhome.ewi.utwente.nl/~rijswijkrm/pub/imc101-vanrijswijk.pdf

[12] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, Christian Rossow, "IoTPOT: Analysing the Rise of IoT Compromises", 9th USENIX Workshop on Offensive Technologies (WOOT'15), Washington, DC, http://christian-rossow.de/publications/iotpot-woot2015.pdf

[13] "Fridge Caught Sending Spam Emails in Botnet Attack", CNET, Jan 2014, https://www.cnet.com/news/fridge-caught-sending-spam-emails-in-botnet-attack/

[14] M. Garrett, "I bought some awful light bulbs so you don't have to", 2016, https://mjg59.dreamwidth.org/40397.html

[15] 'What is same site scripting and what are some exploit scenarios?", http://tinyurl.com/hkfnehd

[16] "DNS rebinding", WikiPedia, https://en.wikipedia.org/wiki/DNS_rebinding

[17] A. Sood, S. Zeadally, and R. Bansal, "Cybercrime at a Scale: A Practical Study of Deployments of HTTP based Botnet Command and Control Panels", IEEE Communications Magazine, Jul 2017

[18] "Samsung's Smart TVs don't just spy, they transmit your speech in unencrypted plaintext", ExtremeTech website, Feb 2015, https://tinyurl.com/samsungspeech

[19] S. Keoh, S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A Standardization Perspective," IEEE Internet of Things Journal, June 2014.

[20] M. Aertsen, M. Korczyński, G. Moura, S. Tajalizadehkhoob, and J. van den Berg, "No domain left behind: is Let's Encrypt democratizing encryption?", CM, IRTF & ISOC Applied Networking Research Workshop 2017 (ANRW 2107), Prague, Czech Republic, Jul 2017, https://www.sidnlabs.nl/downloads/papers-reports/No%20domain%20left%20behind%20is%20Let%20s%20Encrypt%20democratizing%20encryption.pdf

[21] Snort rules, http://snort.datanerds.net/writing_snort_rules.htm

[22] OpenBSD Packet Filtering Rules, https://www.openbsd.org/faq/pf/filter.html#syntax

[23] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras, "Flow whitelisting in SCADA networks", International Journal of Critical Infrastructure Protection, Vol 6, Issues 3–4, Dec 2013, pp 150-158

[24] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras, "Exploiting traffic periodicity in industrial control networks", International Journal of Critical Infrastructure Protection, Vol 13, Jun 2016, pp 52-62

[25] DOWSE Homepage, http://dowse.equipment/

[26] https://github.com/SIDN/spin

[27] Valibox Homepage, https://valibox.sidnlabs.nl

[28] SPIN screencast, https://www.youtube.com/watch?v=jynMCQ1fyvM

[29] J. Bugeja, A. Jacobsson, and P. Davidsson, "On Privacy and Security Challenges in Smart Connected Homes", European Intelligence and Security Informatics Conference, Aug 2016, http://dspace.mah.se/bitstream/handle/2043/21507/2857a172.pdf

[30] Turris Omina, https://project.turris.cz/en/

[31] Dojo-Labs Homepage, http://www.dojo-labs.com/

Change Log

Version	Changes
Jun 28, 2017	Initial publication
Jun 30, 2017	Expanded comparison with Turris Omnia in Section 6.