Protecting Home Networks From Insecure IoT Devices

Elmer Lastdrager[*], Cristian Hesselman[*†], Jelte Jansen[*] and Marco Davids[*]
[*]SIDN Labs Arnhem, NL and [†]University of Twente Enschede, NL
{firstname.lastname}@sidn.nl

***Abstract*—We present our ongoing work on SPIN, a much-needed open source measurement platform that enables researchers and other users to easily analyze the security features of devices in the "Internet of Things" (IoT), specifically in-home networks. SPIN accomplishes this by mapping and enhancing network-level measurements in the home network and by making them available through a well-defined interface. This enables all kinds of new applications for research and commercial purposes, such as privacy managers for consumers that visualize insecure IoT devices and their connections, and new algorithms that automatically block botnet traffic to protect the public Internet against IoT-powered DDoS attacks. SPIN is a flexible distributed system that runs in the home network and that keeps users in control. We have validated SPIN in our lab through prototype implementations.**

***Index Terms*—Internet of Things, Security, Privacy, Measurements, Home Networks**

I. Introduction

The "Internet of Things" (IoT) [1] is a much-hyped term that typically refers to connecting a large number of heterogeneous everyday objects to the Internet and to each other. Devices that used to be "dumb", are increasingly becoming "smarter" by adding processing power and a network connection to them. Examples are fridges, door locks, baby monitors, and light bulbs.

While the IoT can help people in their daily lives [2], it can also jeopardize their security, privacy, and safety because IoT devices are often insecure, in particular devices used in home networks [3]–[6]. Examples are baby monitors that remote adversaries can exploit to reroute its video feed [4], devices like a vacuum cleaner that dynamically build up an indoor map of a house and silently shares it with the device's manufacturer for advertising purposes [7], and devices with programming errors that render the device inoperable (e.g., a smart door lock [8]) or that result in a shutdown of all communications within a house [9].

In addition to these consumer risks, the IoT also poses a large-scale security threat to the Internet because insecure, compromised IoT devices enable massive DDoS attacks that can take down parts of the Internet. This was exemplified by the Oct 2016 DDoS attack on DNS provider Dyn [10], which was carried out by an estimated 100,000 IoT devices infected with the Mirai botnet and led to outages of popular services such as Spotify and Twitter. Consumers are unlikely to be interested in such attacks [11], since they do not know the victims that are being DDoS'ed by the devices in their network and likely will not even notice the relatively low volume of DDoS data on their high-speed Internet connection.

Protecting consumers and the Internet against insecure IoT devices requires a combination of several interventions, such as legal, regulatory, and technical [1]. From a technical perspective, IoT security would greatly benefit from measurement platforms that enable researchers and companies to easily develop, deploy, and evaluate new security methods for home networks, such as novel immersive user interfaces that notify users of security events [12] or algorithms to discover and block botnet traffic [13]. To the best of our knowledge such platforms currently do not exist.

As the authoritative DNS operator of the .nl top-level domain in the Netherlands, a critical Internet service, we feel a responsibility to help fill this gap. We expect the IoT to form a growing security risk because manufacturers often have little incentive to provide firmware updates to fix vulnerabilities [14], while consumers will connect more and more devices to their networks, usually without caring much about their security posture until a hacked device harms them (e.g., [15]). The IoT-powered DDoS attack on fellow-DNS operator Dyn was our trigger to act.

Our contribution is SPIN (Security and Privacy for In-home Networks), an open source measurement platform that creates a dynamic and easy-to-use data model of the IoT devices in a home network. We have been working on SPIN since 2017 [16]. SPIN already contains two applications that use the SPIN network model: a privacy manager that enables consumers to disable IoT devices they consider harmful through a visual interface, and a "reverse firewall" that automatically blocks traffic flows from IoT devices towards the Internet (as opposed to a regular firewall that works the other way around), for instance to prevent them from taking part in an IoT-powered DDoS attack. Our rational is that the added value of SPIN's easy-to-use network model will not only stimulate new SPIN-based security methods, but will also help getting both applications deployed on a large scale.

In the remainder of this article, we first provide an overview of the general concept of SPIN (Section II). We then discuss our design goals (Section III), the SPIN architecture (Section IV), and our implementation (Section V). Section VI discusses related work. Finally, we draw conclusions and discuss future work in VII.

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

II. SPIN Concept

Figure 1 shows an overview of the SPIN concept for a simple home network consisting of two light bulbs (*L1* and *L2*), a thermostat (*TS*), and a smart window (*W*). The light bulbs connect to the home network through a network bridge (*N1*), for instance because they use Zigbee as their link-level protocol. The thermostat and the smart window also connect to network bridges (*N2* and *N3*, respectively) and all three bridges connect to each other and to the Internet through the home router (*N4*).

SPIN's task is to protect users and the Internet from vulnerable IoT devices. In Figure 1, devices *TS* and *L1* have been compromised by adversaries *A* and *B*, respectively. *A* obtains the current temperature in the house from the thermostat and sends it to server *S* under *A*'s control, thus getting an indication if there is somebody at home or not. *B* has infected the light bulb with a botnet client [10], which sends DDoS traffic to target *T* along with a large number of other infected IoT devices in other houses. *B* might also carry out the DDoS attack through a large number of reflectors on the Internet (e.g., open resolvers) and amplify the attack by requesting the reflectors to use responses that are much larger than the original requests [17], [18]. *A* and *B* may have compromised *TS* and *L1* in various ways, such as through (weak) password guessing [19], a same site scripting attack [20], a DNS rebinding attack [21] or by manipulating the device's access control list [22]. They might also misuse *L1* and *TS* for other purposes, such as obtaining user credentials.

In the scenario of Figure 1, SPIN's privacy manager notifies the user that *TS* is connecting to server *S*, which differs from *TS*' normal behavior. The user subsequently decides to block the outgoing flow to *S* and SPIN instructs node *N2* to drop packets from *TS* to *S* ("block(*TS*, *S*)" in Figure 1). Similarly, SPIN's reverse firewall detects that *L1* is sending traffic with a botnet signature and automatically blocks the flow by instructing *N1* to drop packets from *L1* to *T*. In both cases, the user may contact a specialized service provider (not shown in Figure 1) to help getting *L1* and *TS* cleaned, for instance by installing a new firmware version.

Figure 1 also illustrates that the capabilities of the nodes in a home network may differ. For example, *N1*, *N2*, and *N4* are SPIN-enabled, which means that the SPIN system can measure their traffic and block traffic flows passing through them. However, *N3* is not SPIN-enabled, for instance because it is built into the house and cannot be upgraded and put under SPIN control. As a result, the SPIN system measures the traffic to and from smart window *W* and block its traffic flows at *N4*.

III. Design Goals

We set four design goals for SPIN, which are: (A) provide a measurement tool for IoT security applications in home networks, (B) allow for full in-home system deployment, (C) control security and privacy at the network-level, and (D) keep the user in control. We discuss the rationale behind these goals and indicate how the SPIN system architecture (see Section V) addresses them.

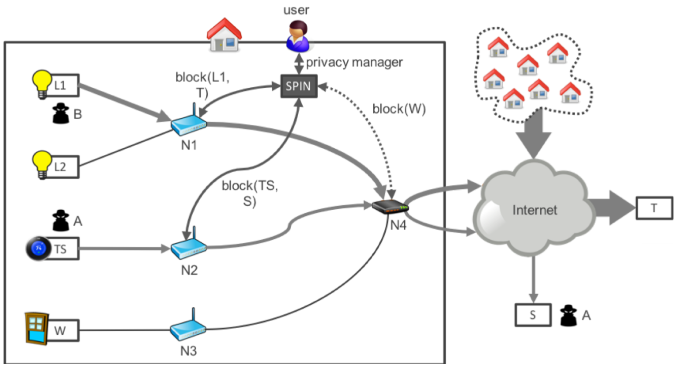


Fig. 1. SPIN concept.

A. Measurement API and SPIN Applications

We aim for SPIN to provide an easy-to-use measurement API that provides a high-level and longitudinal model of a home network and its IoT devices to application developers, allowing them to abstract away from the particularities of device and network measurements. Our motivation is to enable researchers to easily develop and evaluate new in-home applications, such as privacy managers (see Section II) and algorithms to detect newly emerging botnets. Our goal is to stimulate new research into IoT security and privacy applications this way and perhaps the development of commercial SPIN-based products and services as well.

Our other aim is to provide applications that use the SPIN tool. We already built two applications: (1) a privacy manager that visualizes which IoT devices on the network exhibit potentially suspicious behavior and that enables consumers to manually block them, and (2) a "reverse firewall" that proactively mitigates IoT-powered DDoS attacks by automatically blocking suspicious outgoing network traffic (cf. device *L1* in Figure 1). We envision that these applications will also use external sources (e.g., feeds from IoT honeypots [18]) that describe traffic patterns in a standardized language, for instance based on the rule syntax of Snort [23] or OpenBSD packet filtering [24].

B. In-home Deployment

Our second objective is that SPIN can be fully deployed on network equipment within the home without any components running in the cloud (cf. [22]). In this way, SPIN and SPIN applications keep device and network measurements as well as information inferred from them within the home. Users thus stay in control of their data, which is essential for a system that aims to improve users' security and privacy.

The implication is that SPIN needs to be able to operate in widely varying home networks, for instance in terms of size and computing and networking capabilities. We therefore designed SPIN as a distributed system with relativity lightweight measurement and flow blocking functions running on relatively low-end network equipment in the packet forwarding path (e.g., routers and bridges), while its more advanced control functions run "off path" on more high-end, always-on devices such as a network-attached storage. Our distributed design still allows for centralized implementations, such as solutions that run all of SPIN's functions on the home router.

An advantage of our split-level approach is that SPIN can block DDoS traffic close to the source device through its on-path functions, which also reduces the impact of a DDoS on the local network. For example, in the home network of Figure 1, SPIN can block the DDoS flow from device *L1* at *N1* rather than at the home router (*N4*), which reduces the impact of the flow on the local network.

A consequence of our approach is that interoperability between SPIN's control functions and its on-path measurement functions becomes more important to allow devices of different vendors to interoperate.

C. Network-level Control

We also aim for SPIN to provide network-level security and privacy control. This means that SPIN (1) analyzes network traffic (e.g., IP headers, packet lengths, and DNS payloads) and the generic security properties of IoT devices (e.g., if they use default passwords or if they are susceptible to reflection attacks [17]) and (2) blocks flows at intermediate network equipment, such as bridges and routers. SPIN therefore does not rely on IP packet payloads (except for DNS payloads) and is unaware of device-specific functions.

The advantage of a network-level security approach is that it is generic and works for a wide range of IoT devices, which is important because IoT devices are much more heterogeneous than personal computers and laptops. Another advantage is that consumers can continue to use the IoT devices of their choice and are not locked into using specific types of devices, such as those supported by their access provider. A network-level system furthermore allows consumers to easily deploy a SPIN-based system because they do need to go through device-specific procedures, such as loading threat detection modules for specific types of devices and firmware versions. Also, network-level operation makes SPIN encryption-agnostic, which is important as we expect more and more IoT devices to use encrypted connections.

A consequence of being device and application agnostic is that SPIN is a first line of defense against insecure IoT devices that requires supplementary services to help users, for instance to clean a device by upgrading its firmware. SPIN also needs a clear notification service for consumers because IP-level blocking might render devices inoperable and is therefore a temporary solution.

D. User Control

Our fourth design goal is for SPIN-based systems to keep the consumer in control. To accomplish this, we provide a central preference manager as part of the SPIN system that enables users to configure their security and privacy control preferences, specifically pertaining to the following three areas:

- Level of automation: consumers should be able to configure to what level they want SPIN applications like our reverse firewall to automatically block IoT devices. We expect that most consumers will opt for automatic blocking with notifications, but some (expert) users may want to manually control this behavior.
- Devices to monitor: users should be able to indicate which IoT devices 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: in line with SPIN's in-home deployment model, users should be able to express how SPIN should store network and device measurements, for example in terms of retention time.

IV. SPIN ARCHITECTURE

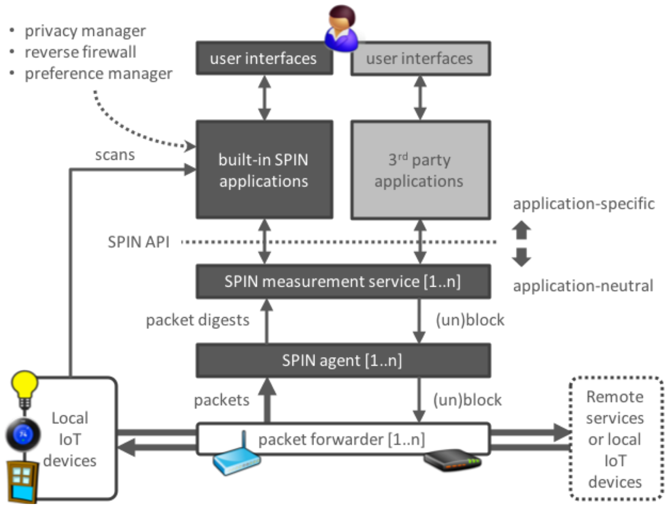


Fig. 2. Architecture of a SPIN-based system.

Figure 2 provides an overview of the architecture of a SPIN-based system, which consists of a SPIN agent, the SPIN measurement service, and (built-in) SPIN applications.

A. Spin Agents

A SPIN agent is a light-weight component that interfaces with the packet forwarding functions of a node in the home network, such as a bridge or a router. An agent publishes digests of the packets that flow through the node, for instance in terms of source and destination IP and port, number of packets and length in bytes. A SPIN agent also accepts requests to block certain IP flows, such as the flow coming from a device that is involved in a DDoS attack (cf. device *L1* in Figure 1).

A SPIN agent itself does not perform any advanced data analysis tasks, such as anomaly detection in network traffic. It is usually a low-level component that integrates with the node's operating system, but it may also interact with a node's forwarding path through other means, such as a mirror port.

A home network may contain multiple agents. For example, nodes *N1* and *N2* run a SPIN agent so that SPIN applications can instruct them to block the flows coming from *L1* and *TS*. SPIN agents require a standardized protocol to enable applications and agents running on devices of different vendors to interoperate, which might be a new topic of standardization in the IETF.

B. Measurement Service

The measurement service stores a longitudinal description of the home network in the form of a sequence of time stamped graphs $G_0 \ldots G_N$, with G_t describing the network at time *t*. Each graph G_t consists of vertices that represent the devices in the network (e.g., IoT devices and network devices) and the external services that they communicate with, and contains an edge for each pair of vertices that exchange traffic. Vertices 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). The attributes of the edges for instance describe the distribution of traffic between vertices as well as the traffic distribution across graphs.

The measurement service constructs its graphs based on the flow digests it receives from one or more SPIN agents and the higher-level assertions it gets from SPIN applications (see Section V), such as a detection module for a certain botnet.

A home network may contain multiple measurement services, each covering part of the network. This for instance covers scenarios in which a user purchases new IoT equipment with a built-in measurement service, while its existing network also contains one. In situations like these, the different measurement services need to synchronize to create a more complete view of the network, or they all need to share their graphs with one measurement service that will act as the master for the entire network. Measurement services need a standardized protocol to allow different instances to interoperate.

C. SPIN Applications

The SPIN measurement service and its API enable researchers and companies to develop and evaluate all kinds of applications, ranging from new privacy management functions to novel botnet detection algorithms. SPIN will be shipped with at least three built-in applications as an example, which are:

- A privacy manager, which uses the measurement service to visualize which IoT devices connect to each other or to services on the Internet and enables users to manually block certain traffic flows.
- A reverse firewall, which uses the measurement service to automatically detect and block IoT devices, for instance when they might be used for a large IoT-powered DDoS attacks.
- A preference manager, which keeps track of the user's preferences (e.g., automatic or manual blocking and retention time of measurements) and makes them available to the measurement service and other applications.

V. Prototype

We validated our approach by implementing our ideas. The implementation is open source and can be found on Github [25]. Currently, our implementation contains most basic components: SPIN agents, the privacy manager, and the reverse firewall. In the current prototype, the preference manager has not been implemented yet. The experimental setup that we used to validate our prototype consists of several IoT devices: a Philips Hue lamp, mobile phones (Android and iOS), a The LoraWAN-gateway, an IP camera, a smart TV, a smart speaker, and a Raspberry Pi running Raspbian. Furthermore, a GLiNET AR150 mini-router was used to act as the SPIN agent for our lab tests. However, our implementation is available to run on generic OpenWRT devices and it can be deployed on Debian, a Raspberry Pi or as a virtual machine image.

A. SPIN Agent

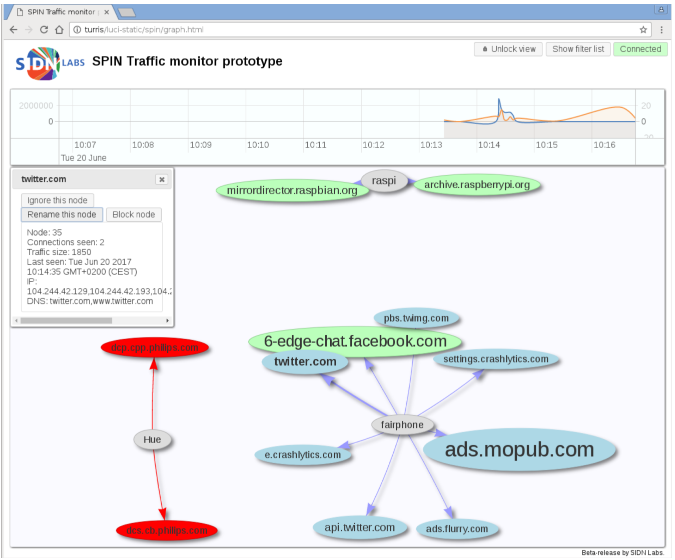
We implemented the SPIN agent (see Figure 2) as a user space daemon in combination with iptables chains and netfilter conntrack and logs. The deamon receives traffic from the netfilter logs and turns them into digests. Then, it publishes the digests through a simple MQTT message broker, which transfers all messages from the agents to the applications (e.g., the privacy manager).

B. Privacy Manager

We also developed a basic implementation of SPIN's built-in privacy manager, which visualizes traffic flows and allows the user to inspect and block specific flows. The privacy manager is a Javascript application that can be used from a browser in the local network [26].

Figure 3 shows a screenshot of the privacy manager, which displays the in-memory network graph of our lab setup. Each center node represents an IoT device in our lab setup (in grey) and the nodes around it are the services on the Internet they connect to. The arrows indicate a "sends traffic to" relationship. 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 privacy manager shows them as one node, and the user can review them by selecting the node.

The privacy manager enables the user to manually block certain devices or remote addresses by denying all traffic to and from their respective nodes. If the user choses to disable a particular traffic flow, the privacy manager will publish a command to the MQTT message broker, requesting any connected SPIN agent to block the specified traffic and will mark the blocked device (red).



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

Fig. 3. Prototype of our SPIN-based privacy manager.

C. Reverse firewall

We built a simple proof-of-concept of SPIN's built-in reverse firewall, which uses a straight forward anomaly detection algorithm: it flags and blocks a device if it connects to many different combinations of IP addresses and ports in a short period of time, or if it's traffic pattern deviates significantly from the traffic as observed before. Blocked devices cannot send or receive any data and a user needs to manually unblock it using the privacy manager.

VI. Related Work

Many researchers have signaled the risks of connecting insecure devices to the Internet in relation to the Internet of Things. For example, Leverett et al. [27] discuss the role of standardization and certification within the IoT ecosystem. They describe the need for systems to be monitored for security breaches and vulnerabilities, and discuss the role of laws and regulators. Bugeja et al. [28] discuss the challenges to security and privacy in the IoT, and propose device monitoring as one of the mitigations to insecure devices. When IoT devices are not securing their communication, their traffic may be wiretapped [29]. Apthorpe et al. [29] discusses mitigation strategies for the scenario of eavesdropping IoT devices.

Researchers that want to perform extensive measurements on IoT security have no tools available that suit their needs. There are several limited options. The Turris Omnia [30] contains a tool called majordomo that stores aggregated traffic statistics for each device. Alternatives are general-purpose tools such as netflow or iptraf.

Several academic proposals similar to SPIN have been published. Simpson et al. [31] propose an autonomous system that intervenes when a threat is identified. However, it operates on the home gateway only, and does not allow users to control what is going on in their network. Sivaraman et al. [22] propose a system that uses a combination of network monitoring, software defined networking, and dynamic security rules. However, they use a central third party outside the home for security control, whereas SPIN runs locally in the home network and supports a more distributed approach. The system of Heimdall [32] relies on data that is obtained from external sources (such as VirusTotal), it is not open source, and does not have SPIN's fine-grained network control. Huang et al. [33] has a comparable aim with an implementation but uses ARP-spoofing to reroute traffic within the local network to the device running their software, thereby impacting performance and reliability.

Several SPIN-like solutions have emerged from the industry as well. Turris Omnia [30] allows users to protect their home using a special-purpose router. It reports potential threats to a centralized control point that analyzes it and may decide to inform other Turris routers. Unlike SPIN, Turris users cannot block traffic flows, unless they manually configure firewall rules. The Dowse project [12] implements a transparent proxy focusing on the user's privacy. Dowse focuses more on the user interaction, whereas SPIN is more a network-level platform.

Recently, an abundance of proprietary and closed-source products has been released or announced. For example, Dojo [34], Norton Core by Symantec [35], Sense [36] by F-secure, Cujo [37], Bitdefender Box [38] and Akita [39] and the Fingbox [40]. Additionally, McAfee produced the McAfee Secure Home Platform, which can be used by router manufacturers to provide network protection [41]. They are different from SPIN in that they are closed-source, whereas SPIN is completely open. The impact of that is that SPIN supports measurements. Furthermore, SPIN does not suffer from a vendor lock-in.

VII. Conclusions and Future Work

The IoT will likely enable a wide range of new applications and services, but its large number of insecure devices also poses a threat to the privacy of users and the stability of the Internet. To address this problem, researchers and companies need flexible measurement platforms for home networks that enable them to easily develop and evaluate new IoT security applications, such as privacy management applications for consumers and new anomaly detection algorithms.

We propose SPIN to as a distributed system that can be flexibly deployed in a wide range of home networks and provides applications with an easy-to-use measurement-based data model of the network's IoT devices and their security features. SPIN comes with a privacy manager that uses the data model to enable consumers to control their privacy in the presence of insecure IoT devices and a reverse firewall to automatically block devices, for instance when they might be involved in a DDoS attack. We discussed our design goals for the SPIN system, the system's architecture, and a first evaluation through an implementation of two built-in SPIN applications and the underlying measurement and mapping functions.

As future work, we plan to work on implementing a full-functioning measurement service that contains a longitudinal model of the network. Furthermore, we will implement and evaluate existing anomaly detection algorithms, which will fuel the reverse firewall functionality. We plan to standardize the communication protocols. Furthermore, we are currently planning a pilot study evaluate the functionality of SPIN in various deployment scenarios in the real world using a large-scale pilot study. Finally, we are seeking for third parties that want to (commercially) use SPIN in their devices, for example Internet Service Providers.

REFERENCES

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

[2] G. Demiris and B. K. Hensel, “Technologies for an aging society: A systematic review of “smart home” applications,” *IMIA Yearbook of Medical Informatics 2008*, vol. 47, no. 1, pp. 33–40, 2008.

[3] M. Starr. (2014, Jan) Fridge Caught Sending Spam Emails in Botnet Attack. CNET. [Online]. Available: https://www.cnet.com/news/fridge-caught-sending-spam-emails-in-botnet-attack/

[4] D. Goodin. (2015, Feb) 9 baby monitors wide open to hacks that expose users’ most private moments. Arstechnica. [Online]. Available: https://arstechnica.com/information-technology/2015/09/9-baby-monitors-wide-open-to-hacks-that-expose-users-most-private-moments/

[5] S. Rosenblatt. (2013, Apr) Top Wi-Fi routers easy to hack, says study. CNET. [Online]. Available: https://www.cnet.com/news/top-wi-fi-routers-easy-to-hack-says-study/

[6] M. Zorz. (2016, Sep) IoT village uncovers 47 security vulnerabilities across 23 devices. Help Net Security. [Online]. Available: https://www.helpnetsecurity.com/2016/09/16/iot-village-def-con/

[7] A. Hern. (2017, Jul) Roomba maker may share maps of users’ homes with Google, Amazon or Apple. The Guardian. [Online]. Available: https://www.theguardian.com/technology/2017/jul/25/roomba-maker-could-share-maps-users-homes-google-amazon-apple-irobot-robot-vacuum

[8] I. Thomson. (2017, Aug) Firmware update blunder bricks hundreds of home ‘smart’ locks. Register. [Online]. Available: https://www.theregister.co.uk/2017/08/11/lockstate_bricks_smart_locks_with_dumb_firmware_upgrade

[9] K. Hill. (2015, Mar) This guy’s light bulb performed a DoS attack on his entire smart house. Splinter. [Online]. Available: https://splinternews.com/this-guys-light-bulb-performed-a-dos-attack-on-his-enti-1793846000

[10] S. Hilton. (2016, Oct) Dyn Analysis Summary Of Friday October 21 Attack. Dyn. [Online]. Available: https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[11] L. Kessem. (2017, Feb) IoT? I don’t care. SC Magazine US. [Online]. Available: https://www.scmagazine.com/iot-i-dont-care/article/634990/

[12] DOWSE. (2019, Nov) Dowse. [Online]. Available: https://www.dowse.eu/

[13] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, “A survey of botnet technology and defenses,” *2009 Cybersecurity Applications Technology Conference for Homeland Security*, Mar 2009. [Online]. Available: http://dx.doi.org/10.1109/CATCH.2009.40

[14] Cyber Security Raad, “Towards a secure connected digital society (in Dutch),” Cyber Security Raad, Tech. Rep. 3, 2017. [Online]. Available: https://www.cybersecurityraad.nl/binaries/CSR%20Advies%20IoT%20digitale%20versie%20DEF%20NED_tcm56-298518.pdf

[15] C. De Mar. (2015, Aug) Baby monitor hacker sends a frightening message to Indianapolis family. Fox 59. [Online]. Available: http://fox59.com/2015/08/27/baby-monitor-hacker-sends-a-frightening-message-to-indianapolis-family/

[16] C. Hesselman. (2017, Jul) SPIN: A user-centric security extension for in-home networks. [Online]. Available: https://www.sidnlabs.nl/en/news-and-blogs/spin-a-user-centric-security-extension-for-in-home-networks

[17] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, “DNSSEC and its potential for DDoS attacks,” *Proceedings of the 2014 Conference on Internet Measurement Conference - IMC ’14*, Nov 2014. [Online]. Available: http://dx.doi.org/10.1145/2663716.2663731

[18] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, “IoTPOT: Analysing the Rise of IoT Compromises,” in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, Aug. 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/pa

[19] M. Garrett. (2016, Feb) I bought some awful light bulbs so you don’t have to. [Online]. Available: https://mjg59.dreamwidth.org/40397.html

[20] A. Barth, C. Jackson, and J. C. Mitchell, “Robust defenses for cross-site request forgery,” *Proceedings of the 15th ACM conference on Computer and communications security - CCS ’08*, 2008. [Online]. Available: http://dx.doi.org/10.1145/1455770.1455782

[21] C. Jackson, A. Barth, A. Bortz, W. Shao, and D. Boneh, “Protecting browsers from dns rebinding attacks,” *Proceedings of the 14th ACM conference on Computer and communications security - CCS ’07*, 2007. [Online]. Available: http://dx.doi.org/10.1145/1315245.1315298

[22] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, “Network-level security and privacy control for smart-home IoT devices,” *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2015. [Online]. Available: http://dx.doi.org/10.1109/WiMOB.2015.7347956

[23] Cisco. (2019, Nov) Snort - network intrusion detection and prevention system. [Online]. Available: https://www.snort.org/

[24] OpenBSD. (2019, Nov) OpenBSD packet filtering rules. [Online]. Available: https://www.openbsd.org/faq/pf/filter.html#syntax

[25] SIDN Github. (2019, Nov) SPIN source code. [Online]. Available: https://github.com/SIDN/SPIN

[26] SIDN. SPIN screencast. Youtube. [Online]. Available: https://youtu.be/jynMCQ1fyvM

[27] E. Leverett, R. Clayton, and R. Anderson, “Standardisation and certification of the ‘internet of things’,” in *16th Annual Workshop on the Economics of Information Security (WEIS17)*, La Jolla, USA, Jun 2017. [Online]. Available: https://doi.org/10.17863/CAM.35286

[28] J. Bugeja, A. Jacobsson, and P. Davidsson, “On privacy and security challenges in smart connected homes,” *2016 European Intelligence and Security Informatics Conference (EISIC)*, Aug 2016. [Online]. Available: http://dx.doi.org/10.1109/EISIC.2016.044

[29] N. Apthorpe, D. Reisman, and N. Feamster, “Closing the blinds: Four strategies for protecting smart home privacy from network observers,” in *Workshop on Technology and Consumer Protection (ConPro ’17)*, San Jose, CA, USA, 2017.

[30] CZ.NIC. (2018, Jan) Turris omina. [Online]. Available: https://project.turris.cz/en/

[31] A. K. Simpson, F. Roesner, and T. Kohno, “Securing vulnerable home IoT devices with an in-hub security manager,” *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar 2017. [Online]. Available: http://dx.doi.org/10.1109/PERCOMW.2017.7917622

[32] J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino, “Heimdall: Mitigating the internet of insecure things,” *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 968–978, Aug 2017. [Online]. Available: http://dx.doi.org/10.1109/JIOT.2017.2704093

[33] D. Huang, N. Apthorpe, G. Acar, F. Li, and N. Feamster. (2019, Sep) IoT Inspector: Crowdsourcing labeled network traffic from smart home devices at scale. [Online]. Available: https://arxiv.org/abs/1909.09848

[34] Bullguard. (2018, Jan) Dojo. [Online]. Available: https://dojo.bullguard.com/

[35] Symantec. (2018, Jan) Norton core router. [Online]. Available: https://us.norton.com/core

[36] F-Secure. (2018, Jan) F-secure sense. [Online]. Available: https://www.f-secure.com/nl_NL/web/home_nl/sense

[37] CujoAI. (2019, Nov) Cujo. [Online]. Available: https://cujo.com/

[38] Bitdefender. (2018, Jan) Bitdefender box. [Online]. Available: https://www.bitdefender.com/box/

[39] AKITA. (2018, Jan) AKITA instant privacy for smart homes. Kickstarter. [Online]. Available: https://www.kickstarter.com/projects/429056796/akita-instant-privacy-for-smart-homes

[40] Fing. (2019, Nov) Fing. [Online]. Available: https://www.fing.io/

[41] I. Paul. (2018) D-Link’s McAfee-powered AC2600 router aims for network-wide protection. PCWorld. [Online]. Available: https://www.pcworld.com/article/3246288/d-link-ac2600-router-mcafee.html