Network-Level Security and Privacy Control for Smart-Home IoT Devices

Vijay Sivaraman†, Hassan Habibi Gharakheili†, Arun Vishwanath*, Roksana Boreli⋆, Olivier Mehani⋆
†University of New South Wales, *IBM Research-Australia, ⋆NICTA, Australia
Emails: {vijay, h.habibi}@unsw.edu.au, arunv@ieee.org, {roksana.boreli, olivier.mehani}@nicta.com.au

***Abstract*—The increasing uptake of smart home appliances, such as lights, smoke-alarms, power switches, baby monitors, and weighing scales, raises privacy and security concerns at unprecedented scale, allowing legitimate and illegitimate entities to snoop and intrude into the family's activities. In this paper we first illustrate these threats using real devices currently available in the market. We then argue that as more such devices emerge, the attack vectors increase, and ensuring privacy/security of the house becomes more challenging. We therefore advocate that device-level protections be augmented with network-level security solutions, that can monitor network activity to detect suspicious behavior. We further propose that software defined networking technology be used to dynamically block/quarantine devices, based on their network activity and on the context within the house such as time-of-day or occupancy-level. We believe our network-centric approach can augment device-centric security for the emerging smart-home.**

I. Introduction

The home is becoming increasingly "smart", driven by emergence of Internet-connected appliances that form part of the emerging breed of Internet-of-Things (IoT) devices. This enables consumers to remotely monitor and manage their home environment [1] lighting systems can be controlled remotely, smoke alarms can alert our mobile phone when a fire is detected, we can monitor our kids from afar, and our health/fitness data can be sent instantly from the home to the cloud for analysis. Surveys in the US [2] indicate personal or family safety, property protection, lighting/energy management, and pet monitoring as top motivations for use of such devices, with 51% of those surveyed willing to pay inexcess of $500 for a fully-equipped smart-home.

With increasing deployments of such Internet-connected devices in the house come increasing risks to both privacy and security: an eavesdropper can illegitimately snoop into family activities, even a legitimate entity (such as the device manufacturer) may be gathering data about users that they are not aware of, and of course a malicious entity may remotely take over control of the IoT devices, using it to either harm the household or to use it as a launchpad for attacking other domains. Indeed in early 2014 it was revealed that there was a large-scale attack on IoT devices including TVs and fridges [3], in which hackers were believed to have broken into more than 100,000 everyday consumer gadgets.

Funding for this project was provided by the Australian Research Council (ARC) Discovery Grant DP150100564.

In this paper we first consider some smart-home appliances available in the market today, and study their operation to reveal several security and privacy concerns. We then argue that security implementation and practise is going to be highly variable across devices, depending on factors such as device capabilities, mode of operation, and manufacturer. These motivate us to propose an approach that implements additional security measures in the network - to this end we propose the use of software defined networking (SDN) to implement dynamic security rules that can evolve based on context, such as time-of-day or occupancy of the house. We believe that our approach can augment existing security solutions implemented by device manufacturers, and additionally provide privacy capabilities that may not necessarily be supported by the manufacturer.

The rest of this paper is organised as follows: in §II we study a few IoT devices to expose their security and privacy vulnerabilities. In §III we discuss our defence mechanism that dynamically reconfigures the network to provide enhanced privacy/security based on context, and describe our prototype implementation in §IV. Relevant prior work is summarised in §V, and the paper concludes in §VI.

II. Threats for IoT in the Smart-Home

Smart home appliances of all varieties are emerging in the market, and Cisco VNI predicts that Internet-of-Things (IoT) connections will grow by 43% each year, rising from 341 million globally in 2013 to 2 billion by 2018. We procured several such devices and studied their behavior in our lab – we have previously revealed vulnerabilities of some of these devices in our earlier work [4]. We briefly elaborate on these in order to provide the context for the defense techniques that will be presented later.

The *Philips Hue Connected bulb* allows the user to wirelessly control the lighting system in the home, and consists of an Ethernet enabled bridge that accepts commands from the user app and communicates these to the bulbs using the ZigBee-Light link protocol. The data exchange between the app and the bridge is via HTTP commands and is not encrypted, so an eavesdropper can easily deduce the operations the user performs on the bulb. Further, even though the device implements access control in the form of a white-listed set of users, this list can be extracted by any attacker, who can then masquerade as a legitimate user, thereby gaining control over the bulb.

978-1-4673-7701-0/15/$31.00 ©2015 IEEE

The *Belkin WeMo motion sensor and switch kit* connect to the Internet via WiFi and allow the user to control the power socket for any electrical home appliance such as desk lamp, coffee machine, or room heater. We were easily able to attack this device by first conducting an SSDP discovery to obtain the IP address of the WeMo devices and the ports they are listening on, and to learn the SOAP commands and their arguments supported by these devices. The attacker can then enable remote-access on the device by registering as a legitimate user by sending an appropriate SOAP-formatted POST command – this then gives the attacker remote access to the device from anywhere in the world, which is indeed scary.

The *Nest smoke-alarm* sends reports and alerts to the user's mobile app, giving them peace-of-mind that their house is safe no matter where they are. However, it comes equipped with sensors that detect motion and light – this can potentially let it detect when the user is in the same room or if he/she has turned on/off the lights. These kind of capabilities immediately raise a privacy concern for the user who may feel that they are being monitored and tracked within their home. We do note that all data exchanges with the Nest smoke-alarm are encrypted, so eavesdroppers cannot read into the communications.

The *Withings Smart Baby Monitor* comes with an IP camera that allows the user to monitor their baby at home via an App on their phone. We captured and analysed WiFi packets to/from the baby monitor, and found all the data exchange to be in plain-text; however, access to the camera does require obtaining a one-time access token from the server. We created a "man-in-the-middle" attack in which we allow the victim's app to authenticate itself to the server and obtain the session id, but then hijack the connection using ARP poisoning, allowing the attacker to replace the source IP address to his own to gain access to the camera feed.

The *Withings Smart Body Analyzer* is a weighing scale that can also measure body fat, heart rate and BMI. It can connect to the Internet either using WiFi or by Bluetooth pairing with the Withings Health Mate App. Once again we found that the user's personal information (name, weight, height, age, gender) is sent unencrypted in plain-text over the WiFi channel. We also found, by capturing Bluetooth packets, that the scales transmits its MAC address and a secret key, which are together used to generate an MD5 digest used by the server for authentication; by capturing this information, we are easily able to recreate the digest, allowing us to masquerade as the device unbeknownst to the server.

The five representative smart-home devices discussed above have poor implementation of security, and are susceptible not just to passive eavesdroppers gaining private information, but also active attackers who may capture information that allows them to masquerade as legitimate users or launch man-in-the-middle attacks. We believe these privacy/security issues are likely to be widely prevelant amongst most IoT devices in the market, and therefore necessitate solutions that can work across the entire gamut of smart-home devices, as discussed next.

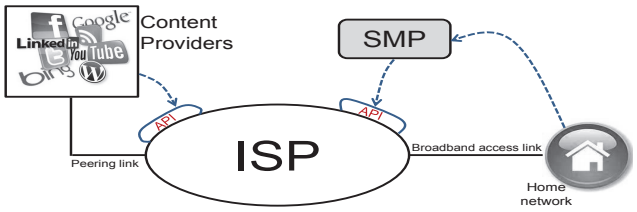


Fig. 1. High level architecture

III. Network-Level Defence

Users purchasing IoT devices for their smart-home assume that the manufacturer has embedded appropriate privacy/security safeguards in the devices. As we have shown in the previous section, this is not the case – indeed many of these devices can be compromised with very little effort on the part of the attacker. We therefore propose a radically different approach, one where security problems are detected and solved at the network-level. There are many reasons we believe our approach has merit:

- Network-level security can be implemented across the entire range of IoT devices, rather than device-level security that is specific to a particular device;
- Unlike device-level security that is embedded into devices and is hence difficult to upgrade, network-level security can be implemented in the cloud, and can be enhanced on a continuous basis;
- Network-level security can be offered by a third-party who has expertise in this specific area, rather than by the device manufacturer who may not have the drive or the skills to implement security properly;
- Network-level security adds an extra layer of protection that can augment any device-level security implemented by the manufacturer.

Indeed, network-level security is routinely used in enterprise networks today (e.g. products like HP Network Protector [5]), as an addition to protection software (like virus-checkers) installed in clients. We believe that network-level security will be even more relevant to IoT, since the heterogeneity in devices is much greater than for desktop/laptop computers that typically run a small handful of operating systems (Windows, MacOS or Linux).

In order to detect and resolve security/privacy issues for IoT, we propose an external entity, called the "Security Management Provider" or SMP, that develops, customizes, and delivers to the user extra safeguards at the network level for the IoT devices in their household. A simple example might involve the SMP adding the appropriate access control rules that protect a specific IoT device, while a more complex example might involve dynamic policies that change access control depending on the context (e.g. the family members being present or absent from the house). Sophisticated security offerings like these that require a combination of data-analytics and network-control are lacking today, and can be fulfilled by the architecture we propose next.

Fig. 1 shows that the SMP interacts on the one side with the ISP network (or home router equipment) via dynamic APIs, and on the other side with home users via easy-to-use GUIs. The job of the SMP is to exercise (limited) configuration control over the ISP network and/or home-router on behalf of the consumer, without being directly on the data path. From an ISP's point-of-view, the "user-facing" network-level APIs developed in this paper (right) complement the "content-provider" facing ones we proposed earlier [6] (left).

SMP role/benefits: The SMP provides customization interfaces (portals/apps) to users, translating these into network-level operations invoked via APIs. We intentionally decouple the SMP from the infrastructure operator/vendor so that multiple entities can compete for this role – an ISP or home router vendor may of course develop the SMP capabilities in-house, bundling it with their offerings to increase retention and revenue; a content provider (e.g. Google, Netflix) or cloud service operator (e.g. Amazon, Apple) may also have an interest in this role so it can improve delivery of its own services; or a new entrant may take up this role with a view towards greater visibility and analytics of home network usage. We believe that by teasing out the role of the SMP, our architecture exposes a wealth of business models that have the potential to spur competition and overcome the current stagnation in residential Internet offerings.

ISP/Home-router-vendor role/benefits: Today's home-routers (much like commercial routers) are vertically integrated, with diverse feature sets and management-interfaces bundled onto the device at production time. Our architecture encourages such vendors to forego user-interface development, and instead focus on supporting APIs that allow an external entity (the SMP) to configure network behavior (our prototype leverages open-source platforms such as OpenWRT and OVS). This reduces the development burden on vendors, allowing them to focus on their competitive advantage, while the cloud-based control model can give them better feedback on feature-usage on their devices. A similar argument applies to the ISP, who today provides little more than Internet connectivity, which is recognized as a low-margin business with little revenue growth. Our architecture gives ISPs a way to monetize on mass-market residential Internet service customization, without taking up the burden of customer management, by exposing network-level capabilities via suitable APIs to an external entity (the SMP). Lastly, the network configurations underlying the APIs can be automated using SDN technology (as described in the next section), and the ISP can therefore support them at low cost.

Consumer role/benefits: The consumer's need for securing their smart-home is more likely to be met by an SMP specialized in the task, than by a generalist ISP or router vendor selling a bundled product. User preferences can be learnt, stored in the cloud, and restored even if the subscriber changes ISP or the home-router. Features and look-and-feel can be personalized from the cloud, and configuration options updated as technologies and use-cases evolve. In the next section we illustrate our prototype that used SDN technology to offering IoT security-as-a-service to residential consumers.

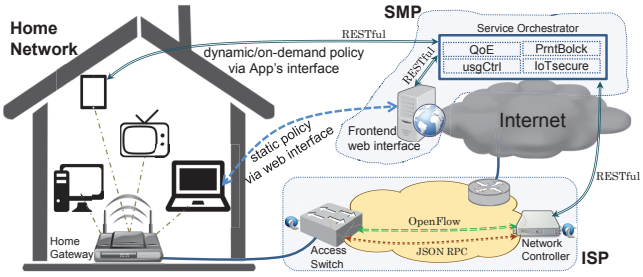


Fig. 2. Overview of prototype design

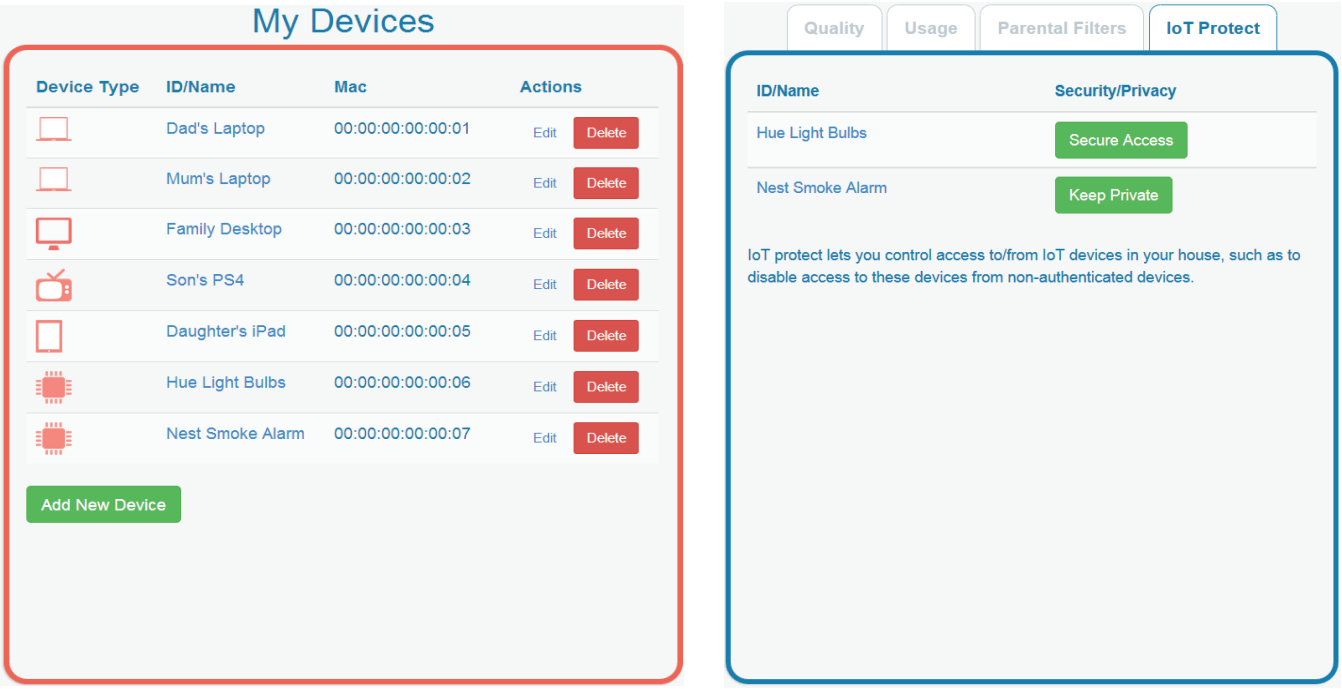
IV. Prototype and Evaluation

We have implemented a prototype of our system that uses our proposed three-party architecture and APIs to provide the above four customization capabilities to subscribers. Our system includes the access switch (OVS) enhancements and controller (FloodLight) modules for the ISP network, and the security orchestrator (Ruby on Rails) and web-GUI (Javascript/ HTML) operated by the SMP. To emphasize their distinction, our ISP controller operates in our University data-center, while the SMP runs in the Amazon cloud. Our implementation has been tested in two settings: (a) an SDN-enabled campus network (emulating an ISP network) spanning over 3000 WiFi access points, and (b) a small number of houses where it works over-the-top (OTT) of legacy (non-SDN) ISP network.

Our implemented design is depicted in Fig. 2, and http://api.sdnho.me/ shows our user-interface live. We assume that the ISP's access switches are SDN-enabled, and further assume that the ISP has visibility of the subscriber's household devices. This starting point is chosen for convenience since: (a) existing SDN controllers have better support for Layer-2 protocols, (b) MAC addresses are static unlike IP addresses that are usually dynamic, and (c) there is a trend towards ISPs providing managed home gateways, either by giving the subscriber a physical home gateway or a virtual instance in the cloud (e.g. vCPE). This assumption will be relaxed later and the solution will be shown to work over-the-top of legacy ISPs with today's NAT-enabled home gateways.

ISP Access switch: Our access switch runs `Open vSwitch 1.9.0` (OVS), and exposes standard OpenFlow APIs. Each home is associated with a physical port on this switch, and for each home we create an instance of a virtual bridge within OVS that maintains the flow-rules and queues for that household.

ISP Network controller: We used the Floodlight (v0.9) OpenFlow controller for operating the ISP network, and developed Java modules to implement the RESTful APIs exposed to the SMP, as shown in Fig. 2). Successful API calls result in appropriate flow table rules being added/removed at the respective OVS bridge serving this subscriber. We added a new module to FloodLight to implement the API for access-control, that provides a wrapper to the FloodLight firewall module so



(a) Home network devices

(b) IoT Device Config

Fig. 3. Web interface showing (a) devices, (b) bandwidth, (c) filters, and (d) usage.

that access control policies (based on remote IP) can be pushed by the external SMP entity for a specific household device.

SMP Security Orchestrator: We implemented a security orchestrator in Ruby-on-Rails that holds the state and the logic needed by the SMP to manage security for the subscriber. It interacts on one side with the ISP via the aforementioned APIs, and on the other side with the front-end portal and user apps (described next) via RESTful APIs, as shown in Fig. 2. It uses a MySQL database with tables for subscribers, devices, policies, user preferences and statistics. It acts upon REST commands from the user portal/apps (described next) by retrieving the appropriate state information corresponding to the subscriber's command, and calling the appropriate sequence of ISP APIs to achieve that functionality.

Web-based portal: provides the front-end for users to customize their services, and is implemented in Javascript and HTML. Snapshots are shown in Fig. 3, and we encourage the reader to see it live at http://api.sdnho.me/. Upon signing in, the user sees their household devices listed in the left panel, while the right panel shows services including a tab for security settings. Fig. 3(a) shows 7 devices for this user, comprising laptops, desktop, iPad, TV, and IoT devices. The *IoTProtect* tab, shown in Fig. 3(b), allows the user to delegate security/privacy of any of their IoT devices to the SMP; the SMP holds the knowledge base on appropriate methods to protect that specific device, and can insert appropriate access control rules via the network API, potentially using context information from the home.

Evaluation: We demonstrate the utility of having the SMP providing IoT protection as a value-add service using two specific devices: the *Philips Hue light-bulb* and the *Nest smoke-alarm*. The light-bulb in our lab connects to the Internet via a WiFi bridge, to which existing Android/iOS apps send desired commands to adjust bulb settings. Even though the bridge maintains a white-list of authenticated clients, this list is sent over-the-air in plain text when queried. We have written a Python script that uses the captured white-list information to construct attack packets that can be played from the Internet to masquerade as a legitimate device and take control of the bulb (the attack is documented in [4]). A user today would most likely be unaware of this attack, let alone know how to block it. In our architecture, the user delegates protection of this device to the SMP. The SMP invokes the network API to insert appropriate access control rules that allow only known clients (belonging to residents of the house) to access the bulb. To support roaming, we wrote a mobile app, installed on the user's phone, that sends heartbeat messages to the SMP with its public IP address, that is then dynamically programmed into the home/edge-router's ACL. This method secures access to the bulb at the network-level, and can be applied to a range of IoT devices with minimal burden on the user.

We also applied our method to enhance privacy of the Nest smoke-alarm installed in our lab. This device connects via the WiFi network to cloud-based servers for providing real-time emergency alerts to the user app. Since the device contains motion and light sensors, there is a legitimate concern that it can track users inside their house and report these to Nest. We captured traffic from our smoke-alarm over several days, and found that (encrypted) traffic was exchanged with authentication (`frontdoor.nest.com`),

alarm notification (`transport-04.rts08.iad01.production.nest.com`), and logging (`log-rts08-iad01.devices.nest.com`) servers. We then built a capability by which the user can request the SMP to protect their privacy when using this device. This prompts the SMP to make network API calls to block the device from accessing the logging server (to which the device sends 250KB of data daily); importantly, this does not disable its core-functionality, i.e. the user still receives notifications on their app when the device detects smoke. This principle can be extended to other devices – for example, we have developed functionality that blocks Dropcam from uploading video to the cloud when the user is at home, something that would otherwise have to be done manually each time by the privacy-conscious user.

To summarize, we have built and demonstrated the value of a specialized provider who offers IoT security/privacy as a service, and dynamically manages the firewall rules for the user (either at the ISP access switch or in the user's home gateway) that monitor and control network operations for each IoT device. We have evaluated this with a small handful of IoT devices, but the general principle can be applied more broadly to any IoT device, providing better security/privacy assurances than provided by device manufacturers today.

V. Prior Work

Research into security and privacy of IoT is still in its infancy, and much of the prior work has focused on understanding and identifying potential threats and adapting existing security techniques to the IoT environment – see a recent survey article [7]. A majority of the work advocates embedding security architectures within the IoT device, including securing the communication protocols. For example, [8] proposes optimizing the DLTS communication protocol for securing IoT data exchange, [9] avises implementation of IEEE 802.15.4 compliant link layer security procedures, and [10] presents a lightweight encryption/decryption method for ID authentication among sensor nodes. Concepts from Artificial Immune System (AIS) have been imported to detect attacks on IoT, and an IoT intrusion detection system with dynamic defense was developed in [11], [12]. VIRTUS [13], a middleware solution for management of applications in IoT environments adopts open standards such as XMPP and OSGi. Access-control mechanisms based on an optimised implementation of elliptic-curve digital signatures (ECDSA) and token-based access to CoAP resources have been developed in [14].

Unlike these prior works, our work does not embed security into the IoT device, but instead moves it to the network, managed under external control using SDN principles. Other works have also proposed SDN-based mechanims for security control, predominantly in the enterprise network context: Jingling [15] out-sources enterprise network features to external providers, HP offers SDN apps for security in enterprise networks [5], VeloCloud [16] offers cloud-based WAN management for branch offices, and LinkSys has recently introduced a cloud-managed smart WiFi router [17]. These parallel efforts corroborate that network-level security for IoT using SDN is likely to gain traction in years to come, and our work facilitates adaption of enterprise/WAN models to the home environment.

VI. Conclusions and Future Work

The increasing uptake of consumer IoT devices poses security and privacy concerns at an unprecedented level. Unlike many of the prior works that have investigated security solutions embedded into the device, we have proposed a solution that identifies and blocks these threats at the network-level. We have advocated a three-party architeture in which a specialist provider offers security-as-a-service, prototyped it using open-source SDN platforms, and evaluated its efficacy in protecting multiple smart-home devices. We believe our work is a first step towards IoT security that applies to a broad range of IoT devices; our future work will apply our solution to several other smart-home devices for which we have identified security/privacy vulnerabilities.

References

[1] C. Wan and D. Low, "Capturing Next Generation Smart Home Users with Digital Home," Huawei, White Paper, Jun. 2013.

[2] iControl. (2014) 2014 State of the Smart Home. http://www.icontrol.com/docs/pdf/2014_State_of_the_Smart_Home_-_Final.pdf.

[3] Business-Insider. (2014) Refrigerator Hacked. http://www.businessinsider.com.au/hackers-use-a-refridgerator-to-attack-businesses-2014-1?op=1.

[4] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli, "An Experimental Study of Security and Privacy Risks with Emerging Household Appliances," in *Proc. First International Workshop on Security and Privacy in Machine-to-Machine Communications (M2MSec)*, Oct 2014.

[5] Hewlett-Packard. (2014) HP Network Protector SDN Application Series. http://h17007.www1.hp.com/docs/sdn/4AA5-1462ENW.pdf.

[6] V. Sivaraman, T. Moors, H. H. Gharakheili, D. Ong, J. Matthews, and C. Russell, "Virtualizing the Access Network via Open APIs," in *Proc. ACM CoNEXT*, Dec 2013.

[7] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: A review," in *Proc. of Computer Science and Electronics Engineering (ICCSEE)*, March 2012.

[8] S. Keoh, S. Kumar, and H. Tschofenig, "Securing the internet of things: A standardization perspective," *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 265–275, June 2014.

[9] D. Altolini, V. Lakkundi, N. Bui, C. Tapparello, and M. Rossi, "Low power link layer security for iot: Implementation and performance analysis," in *Proc. of Wireless Communications and Mobile Computing Conference (IWCMC)*, July 2013.

[10] Q. Wen, X. Dong, and R. Zhang, "Application of dynamic variable cipher security certificate in internet of things," in *Proc. of Cloud Computing and Intelligent Systems (CCIS)*, Oct 2012.

[11] C. Liu, J. Yang, Y. Zhang, R. Chen, and J. Zeng, "Research on immunity-based intrusion detection technology for the internet of things," in *Proc. of Natural Computation (ICNC)*, July 2011.

[12] C. Liu, Y. Zhang, and H. Zhang, "A novel approach to iot security based on immunology," in *Proc. of Computational Intelligence and Security (CIS)*, Dec 2013.

[13] D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, and M. Spirito, "The virtus middleware: An xmpp based architecture for secure iot communications," in *Proc. of Computer Communications and Networks (ICCCN)*, July 2012.

[14] A. Skarmeta, J. Hernandez-Ramos, and M. Moreno, "A decentralized approach for security and privacy challenges in the internet of things," in *Proc. of Internet of Things (WF-IoT)*, March 2014.

[15] G. Gibb, H. Zeng, and N. McKeown, "Outsourcing network functionality," in *Proc. ACM SIGCOMM HotSDN workshop*, Aug. 2012.

[16] VeloCloud. Cloud-Delivered WAN. http://www.velocloud.com.

[17] LinkSys. Smart WiFi Router. http://www.linksys.com/en-us/smartwifi.