Creating a Secure Internet-of-Things through Dynamic Software Defined Networks

Hieu Do, Brian Kao, Theodore Kim, Nick Nguyen, and Weiwen Ying

***Abstract*—This paper proposed a solution for users to secure the IoT devices and make it more efficient for management by using SDN. Software Defined Networks (SDN) is an approach to cloud computing that facilitates network management and enables programmatically efficient network configuration in order to improve network performance and monitoring. Also, we analyzed the security level of SDN considering the computational, physical and functional properties of IoT devices. From there, we designed and implemented a decentralized and virtualized solution with three components: the security policy, the network controller, and the network hardware. Finally, we evaluated our solution as well as considered possible alternatives.**

Keywords—Internet-of-Things, SDN, Software Defined Networking, Firewalls, Network Security, Public Key Infrastructure

I. Introduction

With the recent surge of consumer demand for "smart technologies," common household items such as light bulbs and refrigerators are now being networked and exposed to the public internet to provide extra functionality and connectivity to computers and smartphones **[1]**. These technologies, dubbed "Internet-of-Things (IoT)" for the connectivity being introduced to common items, presents an opportunity for increased consumer convenience. However, they also may serve as a vulnerability in previously secured networks. Still experimenting in a nascent field, IoT developers have yet to develop a thorough security model that can prevent intrusion and unauthorized use of their devices and its network. As the IoT ecosystem grows, so too does the number of unsecured access points to the network. The 2016-2017 Mirai DDoS attack of major networks like Twitter and PlayStation **[2]** are examples of network intrusion via an IoT device vulnerability. Since any device in an IoT system can be the weak point, it is important that all devices on the network have robust security safeguards.

Software Defined Networks (SDN) is an approach to network architecture design which aims to increase traditional networks' flexibility, load balancing capabilities, and centralization by separating its control flow and data layers with a software implemented control structure. Yang et al. **[3]** first proposed the idea of abstracting network routing decisions from network hardware to specialized, configurable controllers. The first practical implementation of network virtualization, project Ethane (later developed into the OpenFlow SDN architecture), was developed by Casado et al. **[4]** as a means of managing complex enterprise networks.

The appeal of SDN architectures is the virtualization of network functions from hardware devices, such as switches and routers, to user defined software functions, which typically are executed by some SDN enabled switch and are managed from some privileged terminal or server (referred to as a the SDN controller). A software defined network would feature routers and switches which have no native protocol for handling network traffic, but rather forward the network control data (i.e. packet / datagram / frame headers) to the controller, which would decide how the hardware should route the data packet **[5]**.

While originally developed for datacenter networks to dynamically manage load balancing and security needs for the needs of different applications (such as VoIP versus database traffic), SDN has since been expanded to a variety of other use cases. Examples of such applications are the aggregation of multiple networks within enterprise wide area networks (SD-WAN), network microsegmentation to achieve multiple levels of data confidentiality over a single connection (i.e. a secure tunnel versus a unsecure tunnel), and securing networks which IoT devices inhabit.

The latter application of Software Defined Networking, and the topic of this paper, was proposed initially by Flauzac, Gonzalez, and Nolot (2015) **[6][7]**. They suggest securing networks with M2M (machine-to-machine, the method of communication most IoT devices employ which requires no user interaction) nodes using a distributed SDN network to safeguard local network traffic to detect and block suspicious network traffic patterns. This paper expands on Flauzac et al.'s approach to include new use cases to add redundancy and remote verification to the network to add a capacity to *prevent* IoT attacks rather than simply isolate infected devices.

II. Background

Given the unique nature of IoT devices (compared to more traditional network endpoints such as computers, servers, and even smartphones) as single purpose, user independent entities, network security solutions must account for the needs of the IoT platform. For example, given the limited processing capabilities of IoT devices, the solution should achieve low computational overhead for individual devices. Moreover, a network is only as strong as its weakest device,

therefore the solution should be applicable to old, possibly unsecured devices. It should similarly be updatable as IoT devices often have a lifespan longer than that of the security standards they implement. IoT network security solutions should be atomic; an individual compromise or suspicion thereof within the network should not compromise other endpoints. Finally, as the majority of breaches occur from unauthorized accesses to these devices, the solution should completely mitigate all accesses to the network from these devices to prevent compromised devices from accessing the rest of the network.

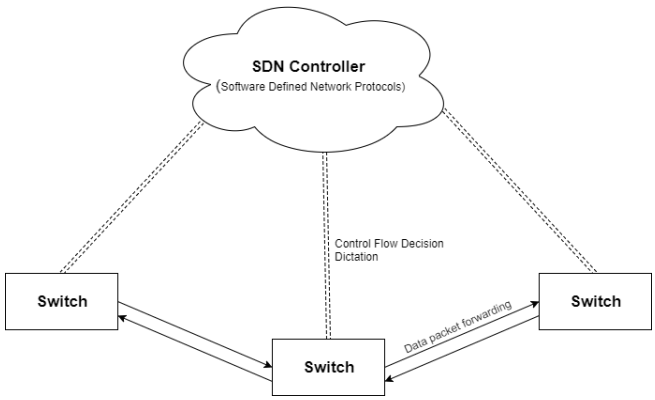

FIGURE 1
A TYPICAL SDN ARCHITECTURE

Ideally, the solution would be network independent, where each device mitigates its own security and not require special accommodations from its network. However, with estimates projecting that the number of IoT devices eventually outnumbering the number of humans utilizing them **[1]**, it is becoming unreasonable to assume that every device can be made perfectly secure. Therefore, research into improving the security of networks which utilize distributed sensing devices just as specialized networks are used in diverse use cases are being explored.

Traditionally, network security is ensured through a variety of preventative network policy mechanisms such as firewalls, MAC address filtration and network authentication. However, Yu et al. **[8]** argues that these network security measures are insufficient for IoT device networks, particularly the ineffectiveness of firewalls. Firewalls are static blacklists of network requests to block data packets based upon some condition (usually packet header content or connection state). They are not applicable to IoT devices as detecting irregularities in their network traffic will require observing the state of dependent devices and environmental factors (i.e., the smart thermometer may request information from the lightbulb only when the dryer is on). Hence, the complexities of the devices' interactions may lead to hole in the firewall policy and vulnerabilities for attackers to exploit.

Sanchez, Lopez, and Skarmeta **[9]** proposes a solution which would implement a lighter-weight variant of PANA (Protocol for Carrying Authentication for Network Access) to authenticate computationally deficient devices such as IoT sensors on a network and achieve secure access control. While their protocol would prevent unauthorized devices access to a network, it would not prevent malicious actors controlling a compromised device from gaining full autonomy within the network.

Hence, the adaptation of SDN architecture to IoT networks aims to create a more adaptable and aware model of IoT device network behavior. Flauzac et al.'s **[6]** initial solution involves creating a SDN enabled network on top of an existing, traditional network to monitor relationships between devices (i.e. what devices has communicated with one another before, what are the circumstances of their relationship, does device A only speak to device B if device

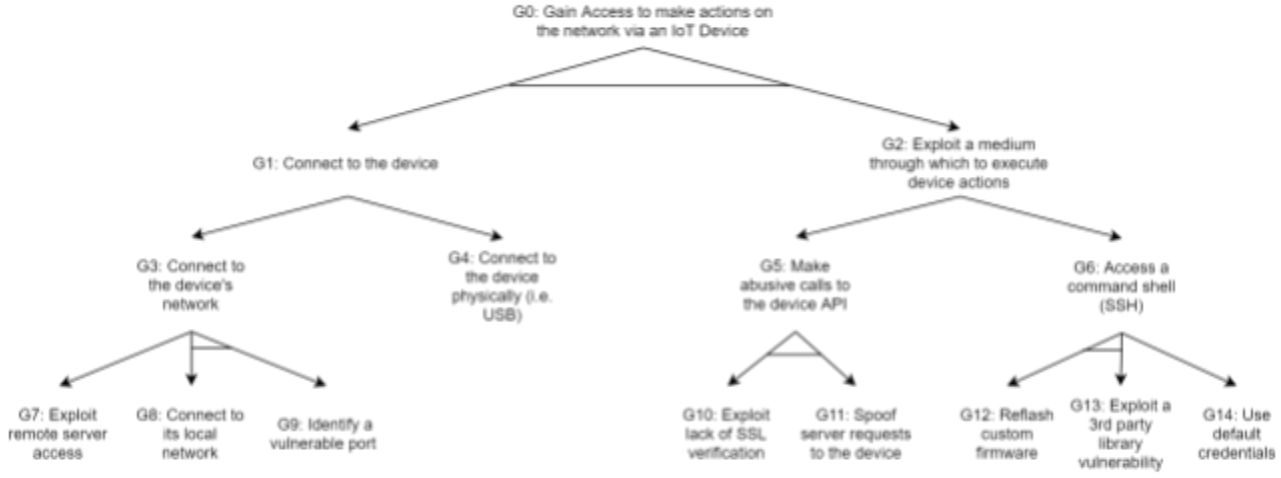

FIGURE 2
IoT NETWORK THREAT MODEL

A initiates the exchange). In a later implementation of their solution, Flauzac et al. **[7]** suggests a decentralized approach to SDN controller distribution to ensure that if one controller is compromised, that the remainder of the network remains secure.

III. THREAT MODEL

As previously discussed, the potential threats facing networks with IoT devices are unique to the computational, physical and functional properties of IoT devices. Attackers may attempt to infiltrate IoT networks through a variety of vulnerabilities that may be present in the diverse IoT devices present in any given network. Figure 2 outlines the potential methods of attack that could be taken to gain unauthorized access to an IoT enabled network.

Table 1 outlines how each threat is addressed by the solution proposed by Flauzac et al. **[6][7]**

TABLE 1
IoT Network Threat and Potential Solution

Threat	Solution
G7, G8, G9: Remote access to the network and its IoT devices	The SDN enabled network is observant of communications between remote sources and IoT devices on a network. If the remote source does not have a trusted relationship with the device (i.e. the device had never opened a request with that remote IP) the controller will block the request or place the device in an untrusted state
G6, G12, G13, G14: Exploitation of device software vulnerabilities	While software vulnerability exploitation cannot be prevented by a network controller, certain application layer packets types are blocked by default, and must be manually enabled by the network administrator (i.e. SSH packets to and from each device are blocked by default)
G4: Physical Device exploitation	The solution assumes that each device has already been compromised.
G5: Abuse calls to the device API	Ideally, the SDN controller has a means of evaluating packets between remote servers and IoT end devices for legitimacy (i.e. has this request from the device elicited this response from the server before? If not, place the device in an untrusted state)
G10, G11: Exploiting devices' unconditional trust of remote servers	The solution does not prevent attackers from impersonating remote servers and making illegal / malicious requests to the individual IoT devices. The alterations proposed by this paper attempt to rectify this exploit path by having the controller establish a legitimate connection with the server and compare its response with the response of the unverified remote server

IV. DESIGN

The solution should provide a decentralized, virtualized (i.e. the solution should not have to be implemented by each device) implementation of a network security policy which protects against each of the attack cased presented in the previous section. Moreover, it should not interrupt or interfere with the normal operations of the network, as it would be without the solution.

There are three main components to the solution which need to be designed in parallel: the security policy, the network controller, and the network hardware.

The policy must be defined thoroughly enough to exclude any and all abusive use cases of the system. While, in many cases, a generic security policy may suffice, a custom configured policy to the needs and properties of each individual network would be ideal. The SDN controller, the hardware responsible for receiving control plane data from the rest of the network and dictating a secure course of action, must be safe from exploitation itself, as an exploit in the "brain" of the system would compromise the system as a whole. Finally, the hardware used in the network, including the IoT sensors themselves, must be visible within the SDN architecture so that the network policy applies to the network as a whole.

V. IMPLEMENTATION

The following implementation is an adaptation of the solution presented by Flauzac et al. **[6][7].** There are multiple platforms on which to develop an SDN controlled network, the most widely used being OpenFlow communication protocol and architectures. Using the OpenFlow architecture, end systems are connected to an OpenFlow (SDN) enabled switch, which is connected through the data plane upwards in the network topology and to the SDN controller via the control plane.

To ensure redundancy in the SDN network architecture, Flauzac et al. **[7]** proposes that multiple SDN controllers be connected in parallel. The parallel controllers are able to see (at least partially) the rest of the network. When switches query a controller for a forwarding action based upon the information available to it via the control plane, it compares responses from multiple controllers and performs the actions dictated by the majority of controllers. Each controller is treated with equal privilege, hence creating a decentralized redundant network of controllers.

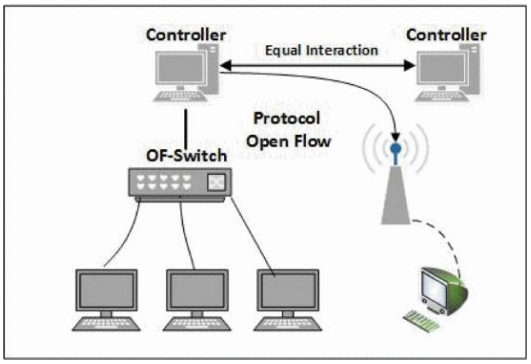

FIGURE 3
A Decentralized SDN Network for IoT application

The SDN enabled network operates on a system of tiered "trust relationships" between endpoints of the network. As proposed by Flauzac et al. **[6].** Ideally, only two trusting endpoints are allowed communication with one and other. Two endpoints with a neutral relationship (neither trusted nor untrusted) can establish a trusting relationship with one and other by completing one round of communication (a request, a valid response, such as a HTTP status of 200, and another request) to demonstrate that the two devices are meant to be communicating with one and other. Untrusted devices communicating between each other will be blocked, whereas an untrusted device communicating with a previously trusted device will only be allowed previously recorded packet types (i.e. an untrusted device that previously only sent HTTP packets to its trusted partner, will only be allowed to send HTTP packets).

A device's trust can be lost after anomalous behavior is detected (a new remote server is accessed, an anomalous response from a server, etc.). On the other hand, trust can be regranted either after a certain timeout or after human intervention. See Table 2 for a complete description of network trust actions.

Devices will also be assigned an authentication level which dictates the privileges of the device on the network **[7]**, such as the ability to send potentially "dangerous" packet types such as SSH. All IoT devices are initially assigned the lowest privilege level, and can be elevated by the network administrator either at setup or upon request (i.e. repeatedly sending SSH packets would prompt the administrator for elevation).

TABLE 2
TRUST RELATIONSHIP AND NETWORK ACTION

Source Reputation	Destination Reputation	Description of Network Action
Trusted	**Trusted**	The communication is allowed unconditionally
Trusted	**Untrusted**	Only previously recorded packet types (no new packet types) can be sent between the two devices until full trust is restored. This is to limit compromised devices from abusing using their network foothold
Untrusted	**Trusted**	Same as Trusted / Untrusted
Neutral	**Trusted**	Would never occur as both devices will become trusted at the same time or one will become untrusted individually. Once trusted, no device will return to the neutral state
Untrusted	**Untrusted**	Unconditionally block the communication between the devices
Neutral	**Neutral**	Allow communication. Elevate trust after a full round of communication (request, response, request). Record all traffic for later debugging

Finally, the network would provide redundancy for possible "man-in-the-middle" attacks on the network devices. This is an important component of the system's ability to not only detect and recover from infiltration, but prevent it as well. The controller should compare unencrypted requests to servers (HTTP) from devices that may not have previously implemented that functionality, with responses from SSL secured requests from the SDN controller. In this way, the controller can attempt to detect illegitimate servers and their API responses.

VI. EVALUATION

SDN enables the network to be automated and centrally managed, and these functions can be implemented through remote configuration and management. From a centralized point, managers can create automated protocols that allow management of data flows. Also, managers can set up a policy for new devices that will be connected to IoT soon, which basically allows users to predict and respond to IoT access devices. And the inherent scalability of SDN allows for the rapid addition of new IoT devices. Programming the response protocol for new network devices means that the network can scale (or shrink) as needed, and the dynamic response system greatly reduces the risks of IoT.

Virtualization of SDN components enables network devices and traffic to be dynamically reconfigured, including configuring bandwidth automatically, and unconfiguring bandwidth. Therefore, as IoT traffic grows, SDN can reasonably allocate bandwidth. The global IoT environment means the influx of large amounts of data and devices, but the analysis of this information will lead to smarter, more automated predictions. And this helps devices know more about each other, which will reduce the traffic issues.

SDN can improve network visibility. If switches can route the low-priority to the line with utilization below a certain level as the preferred alternate path, then the transmission will definitely get faster.

Risks and possible solution:

One major risk is the human component of the system: how do human administrators have an informed role in the management of the network? In enterprise use cases, network administration can be achieved by a dedicated IT professional. However, given the increasing popularity of IoT technology amongst consumers, a more user-friendly administration method must be presented for their use; a phrase like "Device A is requesting an elevation in privilege status," is meaningless to the average user. While not currently offered on the market, usable SDN technology is in development for consumer use, including nascient forms such as the Norton Core router, according to a study conducted by Alshnta, Abdollah and Al-Haiqi **[12]**.

As with most application, virtualization like that achieved with SDN increases the surface area for attack. Rather than a data path with vulnerabilities, there is now an increased likelihood for compromise of the control plane, with illegitimate routing information being sent along with data (IP spoofing, DNS spoofing, etc.). While these "spoofing" vulnerabilities may be able to solve these issues, the computational and network resources to achieve this would be unreasonable. Hence, the solution is dependent on the continued advance of SDN technologies and may not be viable at the current state of the technology.

VII. RELATED SOLUTIONS

Using an SDN enabled network provides a means of implementing a diverse set of security policies and threat analysis mechanisms. One such mechanism presented by Hodo et al. **[11]** leverages machine learning algorithms (specifically a neural network) to detect threats within a network of IoT devices. The artificial neural network is trained both previous to deployment and during operation using packets sent between IoT device and remote connections. The resulting outputs then classifies each received packet as either normal or a threat. The network controller then moves to isolate threatening packets and the endpoints involved in that communication.

Another SDN solution proposed by Yu et al **[8]** attempts to validate device behavior by observing the global state of the network. This solution builds a global firewall where each traffic condition (i.e. when to block a given packet or connection) depends on the current network states of the other devices connected on the network. The goal of Yu et al.'s solution is to take into account the interactions between devices to identify anomalous behaviors within the network. For example, the thermometer will talk to the washing machine, but never at the same time as its talking to the user's smartphone.

VIII. CONCLUSION

In this paper, we have gone over how IoT systems can be protected by using SDN. SDN is not an exact system but more like a concept. It breaks the shackles of traditional network architecture, and make the control plane and data plane of network equipment separate. SDN is fully applicable to the efficient transmission of information in the IoT network layer in the existing technical capabilities and ideas. While the solution presented here is purposefully vague in its security policy (ideally, each user would custom define their desired policy), the mechanism does handle all possible abuse cases. Though SDN is not very mature today, it will become much more popular after further test and practices.

REFERENCES

[1] Cerullo, Gianfranco, et al. "IoT and Sensor Networks Security." Security and Resilience in Intelligent Data-Centric Systems and Communication Networks. 2018. 77-101

[2] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., ... & Kumar, D. (2017, August). Understanding the mirai botnet. In *USENIX Security Symposium* (pp. 1092-1110)

[3] Yang, L., Dantu, R., Anderson, T., & Gopal, R. (2004). *Forwarding and control element separation (ForCES) framework* (No. RFC 3746)

[4] Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., & Shenker, S. (2007, August). Ethane: Taking control of the enterprise. In *ACM SIGCOMM Computer Communication Review* (Vol. 37, No. 4, pp. 1-12). ACM.

[5] What's Software-Defined Networking (SDN)? (n.d.). Retrieved December 6, 2018, from https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/

[6] Olivier, F., Carlos, G., & Florent, N. (2015). New security architecture for IoT network. *Procedia Computer Science*, *52*, 1028-1033.

[7] O. Flauac, C. González, A. Hachani and F. Nolot (2015). SDN Based Architecture for IoT and Improvement of the Security. *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, 688-693.

[8] Yu, T., Sekar, V., Seshan, S., Agarwal, Y., & Xu, C. (2015, November). Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks* (p. 5). ACM.

[9] Sanchez, P., Lopez, R., & Skarmeta, A. (2013). PANATIKI: A Network Access Control Implementation Based on PANA for IoT Devices. *Sensors*, *13*(11), 14888–14917. MDPI AG

[10] B. Anggorojati, P. N. Mahalle, N. R. Prasad and R. Prasad (2012). Capability-based access control delegation model on the federated IoT network. *The 15th International Symposium on Wireless Personal Multimedia Communications*. 604-608.

[11] Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016, May). Threat analysis of IoT networks using artificial neural network intrusion detection system. In *Networks, Computers and Communications (ISNCC), 2016 International Symposium on*(pp. 1-6). IEEE.

[12] Alshnta, A., Abdollah M., & Al-Haiqi, A. (2018, May). SDN in the home: a survey pf home network solutions using Software Defined Networking. In, *Cogent Engineering* 5(1). 1-40.