Flow Based Security for IoT Devices using an SDN Gateway

Peter Bull, Ron Austin, Evgenii Popov, Mak Sharma, & Richard Watson
Centre for Cloud Computing
School of Computing & Digital Technology
Birmingham City University
Email: Peter.Bull@BCU.ac.uk

***Abstract*—With near exponential growth predicted in the number of Internet of Things (IoT) based devices within networked systems there is need of a means of providing their flexible and secure integration. Software Defined Networking (SDN) is a concept that allows for the centralised control and configuration of network devices, and also provides opportunities for the dynamic control of network traffic. This paper proposes the use of an SDN gateway as a distributed means of monitoring the traffic originating from and directed to IoT based devices. This gateway can then both detect anomalous behaviour and perform an appropriate response (blocking, forwarding, or applying Quality of Service). Initial results demonstrate that, while the addition of the attack detection functionality has an impact on the number of flow installations possible per second, it can successfully detect and block TCP and ICMP flood based attacks.**

I. Introduction

The proliferation of Internet of Things (IoT) based devices brings with it additional considerations with regards to securing the network. For the number of IoT devices in use to reach the number (e.g. 24 billion by 2020 [1]) and ubiquity within society predicted, there will need to be a flexible method of integrating these devices within overall network architectures. This integration will potentially be dynamic, as users and devices leave and join the networks within their range (e.g. a person with body area network sensors that connect to local networks). This brings challenges with regards to ensuring the secure configuration and integration of devices to the network. Experimental exploitation of current smart devices (e.g. Phillips Hue light bulbs, Nest thermostat) [2] has demonstrated the need for a better approach to handling IoT security. Current approaches of firewall zoning and IDS/IPS are too constrained by a traditional network architecture and are potentially computationally heavy when considering the increase in network devices.

Software Defined Networking proposes a fundamental redesign of how networks are architected, with a view to increasing flexibility and aiding device management. This is achieved through the separation of control and data planes, and the introduction of a centralised (though potentially distributed) controller. The potential use of an SDN based switch as a gateway for IoT devices provides opportunities for packet inspection, and traffic pattern analysis at the local (i.e. distributed) or global (i.e. centralised) level. This paper proposes a new adaptive flow based security mechanism for IoT devices using an SDN gateway. This adaptive mechanism will perform dynamic analysis of traffic patterns from IoT devices to determine when devices are acting in a malicious manner, or are being the target of external exploitation.

The remainder of the paper is organised as follows: Section II explores the background of IoT device security, focusing on current mechanisms and future requirements. Section III discusses the area of Software Defined Networking and its use in IoT Gateways. This includes their use in determining anomalous behaviour through traffic analysis. Section IV introduces the proposed mechanism and discusses its place within the wider network architecture. Section V presents initial proof of concept implementation work and experimentation results. VI concludes and discusses areas for potential future directions.

II. Background

This section provides an overview of current approaches to IoT infrastructure security, focusing on the unique characteristics of IoT devices that mean traditional approaches to network security may not be possible. The section concludes with consideration of security within the IoT based network itself.

A. IoT Device Security

IoT based devices are fundamentally based around the principles of wireless networks [3], which brings with it a wide variety of security issues. Firstly, nodes authentication is critical to ensure data privacy and prevention of illegal access to IoT devices. Implementation of authentication requires application signature-based identification and access confirmation mechanism or centralised server that authorises requested access.

Due to wireless data transmission, network traffic, transferred between nodes must be encrypted to protect the confidentiality of information. Securing this data requires optimal cryptography algorithms and adequate key-management systems. In this case, key-management includes secret key generation, storage, distribution and updating, where according to [4], distribution of public and secret keys to legitimate users is a key issue.

The majority of proposed security solutions use cryptographic algorithms, that normally require high amount of resources. Considering that most IoT devices are associated with low energy and computing resources capabilities, such solutions cannot be implemented to IoT devices with an application of traditional cryptographic mechanisms [5]. Implementing encryption algorithms that are optimised for low power devices is a well established area of work, with examples including; [6] experimental evaluation of different types of Attribute-Based Encryption algorithms to analyse applicability and efficiency, and [7] public key encryption algorithm optimisation for the use in resource constrained wireless sensor networks.

B. IoT Network Security

Data privacy is one of the most critical qualities in an IoT infrastructure due to potentially high amounts of sensitive data transferred (giving current context/status of people, device, or objects, for example). Trust management is a significant part of achieving IoT node privacy. Trust management can be insured by strong authentication mechanisms implemented on the node and by separation and globally managed network access control [8].

According to [9], access control in such a dynamic environment as an IoT infrastructure requires a centralised controller that will manage traffic flows and exclude unauthorised access to IoT nodes through the network infrastructure. Integration of network access control allows the simplification of authentication mechanisms on endpoint nodes to save computation time and therefore power.

Another critical issue is the securing of network level routing in wireless sensor network. Attacks toward routing infrastructure can cause the unavailability of network components, and data loss [10]. While limitations of energy and computing resources make traditional routing protocols inefficient for an IoT infrastructure, routing protocols, adapted for Wireless Sensor Networks are required.

As [4] discuss, Distributed Denial of Service (DDoS) attacks are currently the most common network attacks, and especially in the IoT. Centralised data collectors and IoT gateways becomes a single point of failure for a group of sensors of other IoT devices. That makes infrastructure critically vulnerable for DDoS attacks. Similarly, separate devices not aggregated to groups are also vulnerable to DDoS attacks due to limited resources.

C. Summary

Computation and power constraints common within IoT devices mean that many traditional approaches to network security are not practical. While work in the separate areas discussed within this section goes part, or most of the way to addressing these limitations, there is scope to further explore the potential advantages that a centralised gateway (as is required for many IoT devices to access a standard IP network).

III. SDN Based IoT Gateways

This section briefly introduces the are of Software Defined Networking (SDN) and discusses its applicability to both acting as a gateway for IoT devices and acting as a security mechanism in itself.

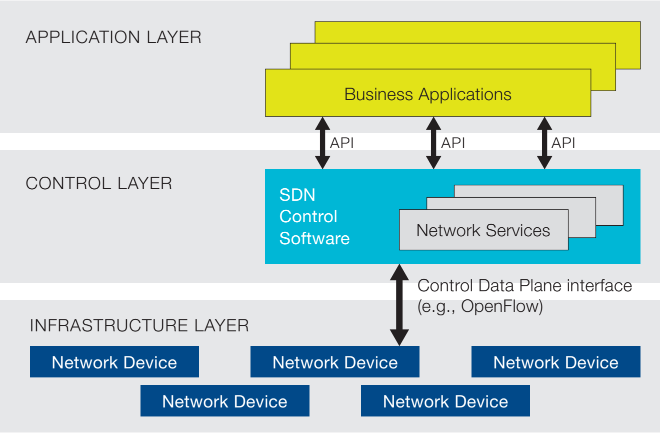

Fig. 1. Software-Defined Network Architecture [12]

A. SDN Background

When configuring traditional network devices (e.g. switches, or routers) the low level details of configuration and features etc. (referred to as the control plane) are dependent on the proprietary operating system of that device. This means that networks can be difficult to reconfigure in a dynamic manner, and complex to manage. Software Defined Networking (SDN), as discussed in [11], aims to address this through the decoupling of control and data planes to allow for the software based configuration of devices, as illustrated in Figure 1. The high level architectural overview of SDN, provided by [12], goes on to detail how it consists of three main principles:

- Decoupling of controller and data planes
- Logically centralised control
- Exposure of abstract network resources and state to external applications

These principles work together to facilitate the control and configuration of the network through a software based element, thus allowing all the associated advantages (such as dynamic control). A logically centralised controller provides a means of monitoring the overall network performance and dynamically adjusting configuration (be it re-routing traffic, or applying a new bandwidth rate limit to a greedy flow, etc.).

B. SDN-Based Security Research Projects

SDN itself does bring with it new challenges in terms of security, with the potential vulnerabilities inherent in centralised control, that require further research. There are, however, a number of new areas in which the dynamic and centralised functionality provided by SDN can enable new network security mechanisms. Ongoing projects in the area of SDN based security architectures demonstrate how the global network view and centralised control provided by SDN can be applied to enhance the efficiency of IoT security mechanisms and mitigate part of the most critical threats.

A centralised network controller allows the management and control of every traffic flow from the entire network, and can be used for development of flexible access control systems. [13] suggest a SDN-based network control approach where traffic flows are permitted or blocked based on security levels of the source and destination. Following a similar theme, [14] propose a network architecture with required validation of the address for every source of data. In this, source address validation is performed by the network controller in reactive mode for every packet that is sent to the controller. This places a potentially high computational load on the controller and may result in scalability issues, however, this could potentially be addressed through distribution of control.

One of the most common and significant security threats deeply researched is that of Distributed Denial of Service (DDoS) attacks. A number of projects are currently seeking to use SDN based security systems as means of mitigating such attacks. For example:

- [15] suggest a network architecture with application identification for every new traffic flow. DDoS defence is performed by limitation of flow amount for the same application per data source.
- [16] propose DDoS attack detection mechanism based on the network traffic statistic analysis. Controller analyse amount of transferred traffic and frequency of specific event which are associated with DoS attack.
- [17] developed a data traffic analysis method based on self-organising maps that detects malicious traffic flows.
- [18] developed flow count detection method based for anomaly detection based on analysis of previous network behaviour and comparison with current state. This methods allows the detection of abnormal traffic flow, that are associated with DDoS attacks.

Currently significant effort is made in the area of wireless network security enhancement by implementation of SDN architecture:

- The OpenRoads project [19] (available under open source licence) decouples the network mobility services from the underlying physical infrastructure. doing so it allows the management and control of the network dynamically by several sources.
- Odin [20] is an enterprise SDN based system, that uses an abstraction of light virtual access points to virtualise network infrastructure. It allows to build layered network with simple configuration and network management.
- OpenAPI [21] develops a wireless network architecture aimed at service quality and security management in ISP networks, achieved by virtualisation of last-mile access infrastructure.

C. SDN-Based Security Applications

In addition to the ongoing SDN-based security research projects, there are a small number of commercially developed security applications that are designed to integrate with SDN controllers.

1) Radware DefenseFlow: Radware [22] is a member of the Opendaylight project and have created DefenseFlow, an SDN-based DDoS mitigation hardware/software package, which allows all appliances within the network to become part of the DDoS mitigation process. DefenseFlow performs two main tasks: monitoring behaviour of protected traffic and forwarding attacked traffic to mitigation & scrubbing centres.First, Protected objects within the network are specified, these may be servers, devices or subnetworks. Once the protected objects are defined, DefenseFlow determines the appropriate places within the network to dynamically deploy traffic counters relating to the protected object.

The counters are monitored over a period of time and are used to determine network operation baselines. When a counter experiences a behavioural anomaly, an attack is declared against that particular protocol and mitigation procedures commence. A QoS policy is configured, limiting the rates of the traffic under attack and the offending traffic flows can be dropped or diverted to Out-of-Path filtering and scrubbing tools. Any legitimate packets are filtered out and placed back into the network. SDN mechanisms, such as flow tables and counters, allow granular control of traffic flows allowing only suspicious traffic flows to be diverted to the scrubbing centre. This represents an improvement over similar statistical based systems which, after an attack is declared, redirect all traffic towards the scrubbing centre.

2) Brocade: Brocade offer a similar solution to DefenseFlow, called Real-Time SDN and NFV Analytics for DDoS Mitigation. Its main difference is the use of SFlow for capturing flow counters instead of Netflow. Whereas Netflow monitors all traffic flows and places a substantial load on the CPU, sFlow uses packet sampling to create accurate flow statistics. According to [23] sFlow statistics are leveraged to identify long lived large flows (those flows with a duration of tens of seconds), and using approximately 10% of the available bandwidth. Whilst other flow types can be used to execute a DDoS attack, they do not pose the kind of immediate threat as a long lived large flow. Upon flood attack detection, mitigation actions take place and OpenFlow rules are pushed from the controller to the switches.

D. Summary

This section has considered a number of projects relating to the use of SDN as a means of securing the network from attack. The DefenseFlow and Brocade models are nominally very similar in operation, and both currently require propriety hardware or software to fully implement, which limits the practicality of their widespread deployment as means of securing IoT gateways.

Research projects, such as [18] provide solutions that are potentially computationally simple enough to run within small-scale gateway devices. The lack of consideration for wider traffic flow characteristics and the need to distribute control,

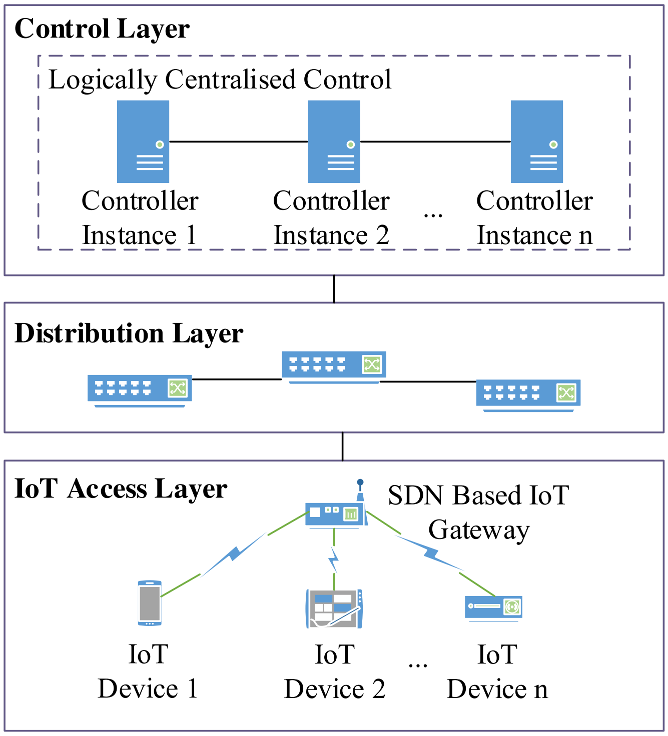

Fig. 2. IoT Software-Defined Network Architecture

however, means that they are not readily applicable to IoT based systems.

IV. Proposed Flow Based Security Mechanism

The author's previous work [24] demonstrated the potential for using simple flow analysis to determine whether flows are periodic, and if so, to pre-emptively install their corresponding rule into the flow table prior to a packets arrival at a switch. It can be seen that a similar approach can be taken in analysing the traffic coming from and going to IoT devices through an SDN gateway to determine where traffic patterns either fall outside of expected (potentially historically derived, or manually specified) patterns or match known patterns used to exploit these devices.

The use of traffic analysis as a means of protecting the network is not unique in itself, however, the use of an SDN gateway and controller allows for a holistic view of the network and removes the need for additional dedicated hardware. It is proposed here to use the IoT gateway as both an SDN switch and integrated controller to allow for analysis of traffic patterns and their corresponding flow rules closer to the edge of the network. This functionality would focus on the security provisioning, with wider network functions (such as routing, QoS, etc.) facilitated by centralised (though still potentially distributed) control. Figure 2 provides an illustration of the proposed system architecture.

Moving functionality back to the network edge may seem contrary to the initial concept of SDN devices with centralised control, however it is necessary here in order to meet the performance requirements, and also to inherently move the point of security closer to the IoT devices themselves. This is particularly important for an SDN network where OpenFlow messages are transferred to the controller, and therefore DoS based attacks can propagate their effects through the network.

Placing devices at the network edge lowers their potential load, and therefore means that smaller-scale devices may potentially be used. Project such as [25] demonstrate that even low powered devices, such as the Raspberry Pi used here, can act as an SDN switch. This largely removes cost as a barrier, where the performance of the device can be proven to be sufficient for the system in development.

V. Proof of Concept Implementation and Initial Experimentation

Initial Implementation focused on providing a proof-of-concept for the use of an SDN controller to perform the necessary attack detection and mitigation. This was developed using version 1.3 of the OpenFlow protocol and the Pox controller.

The implemented controller consists of three main components; the main switch functionality (forwarding, etc.), a statistics manager, and a set of mitigation actions. For the purposes of this proof of concept, the Pox L2 learning switch module was used as a base around which additional functionality was written.

The statistics manager is necessary for gathering data with regards to the current and historic data rates and characteristics of flows. For this implementation, this was limited to data throughput rate and interval of transmission. Future expansions could, however, utilise a more context based approach to traffic analysis (e.g. considering time/day, and correlating data transmission from other devices).

Upon detection of an anomalous flow, the proposed mechanism executes an appropriate mitigation action (as shown in Figure 3). The three possible actions are; Block, Forward, or Apply QoS.

- Blocking a flow effectively blacklists that device from transmitting or receiving data across the network. If this were a device that was under attack from an external source, then it would be prudent to block access to the device for the source of attack (based on source IP address) and to update the central controller to attempt to block this traffic closer to the source (be it internal or external). If the device itself has been compromised then traffic can be blocked and the device reported to a management system to be investigated (to check for physical compromise, or software update required, etc.).
- Forwarding a flow to a quarantined part of the network could allow for deeper inspection of the traffic before reaching a decision about how to treat a device.
- Where a decision is not clear, or blocking a single source not possible, applying Quality of Service to limit the impact of any attack (e.g. by limiting the bandwidth rate at which data can flow to/from a device) can provide a means of limiting the effect of an attack.

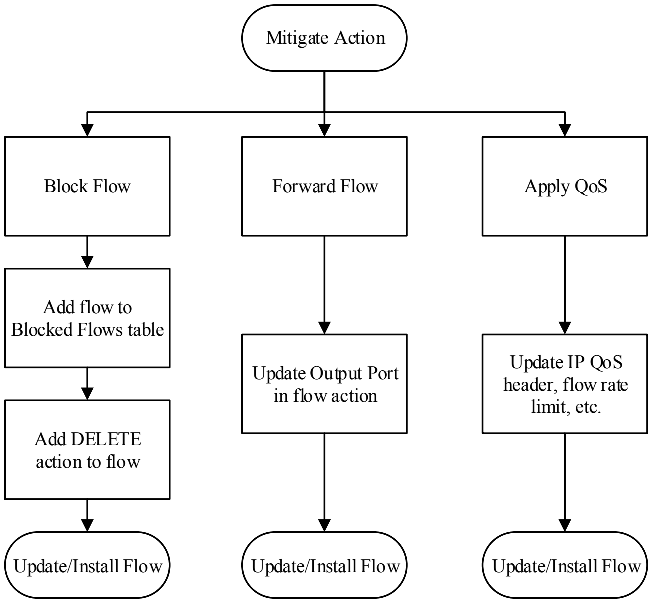
Mitigate Action
Block Flow
Forward Flow
Apply QoS
Add flow to Blocked Flows table
Add DELETE action to flow
Update Output Port in flow action
Update IP QoS header, flow rate limit, etc.
Update/Install Flow
Update/Install Flow
Update/Install Flow

Fig. 3. Mitigate Action

The choice between mitigation actions is, at this stage, a design-time decision. This could, however, be built-up in a more dynamic manner as new threats are discovered and assessed based on the responses to similar attacks.

A. Experimental Topology

For the purposes of experimentation a simple topology was developed, consisting of an IoT device source, a sync destination, and an attacking node, using an SDN based gateway with associated controller as the intermediary device (illustrated in Figure 4). Mininet was used as a means of emulating nodes and SDN devices, while Pox was used as the controller, running the custom developed controller including the proposed security mechanism.

Experimentation focused on the use of TCP flood and ICMP based attacks, using the Low Orbit Ion Cannon tool [26], confirming the ability of the SDN based security mechanism to block and mitigate these attacks. To show the affect of this attack on genuine traffic from the device a TCP stream of 1.5Mb/s was generated from the IoT device to a sync destination. The link itself was configured with a maximum bandwidth of 1.5Mbps to simulate resource constrained links. Note that, while in reality data rates from IoT based devices may be lower, the intention here is to show the effect of the attack on the genuine traffic.

B. Initial Results

Scenario 1 investigated the use of A TCP Flood attack from the attacker to the IoT Device. The genuine traffic was sent using TCP at approximately 1.5Mbps (the capacity of the link). The Low Orbit Ion Cannon tool was used to send 2Mbits per second of traffic, thus saturating the link. Figure 5 shows the results of this attack. At 5 seconds, the attack was started and at approximately 10 seconds, the attack is mitigated by blocking this flow. The genuine traffic flow then recovers and continues its optimal transfer rate.

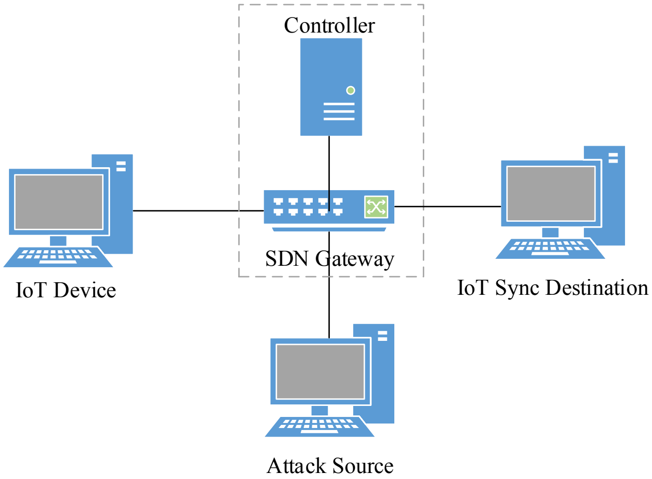

Fig. 4. Experiment Topology

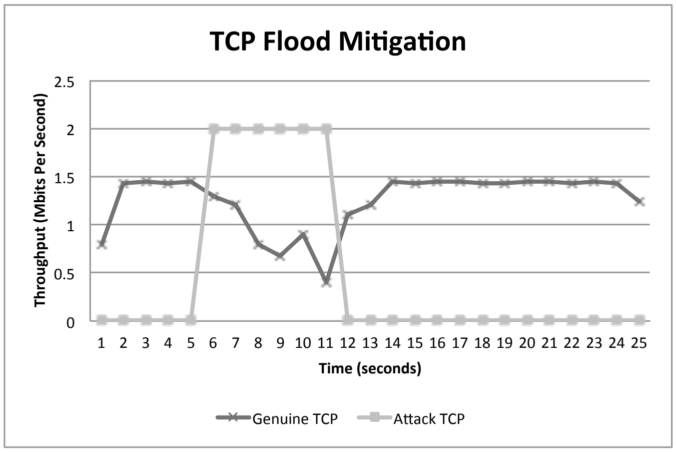

Fig. 5. TCP Flood Experimentation Results

Scenario 2 follows a similar pattern, whereby a genuine TCP flow transmits at approximately 1.5Mbps and at 5 seconds an ICMP based attack is started on the IoT Device (Figure 6). This can be seen to have a detrimental effect on the genuine traffic stream, until at approximately 9 seconds the attack is successfully blocked, and the TCP stream begins to recover.

Figure 7 shows the effect on controller performance (measured in terms of flow installations per second) of the additional functionality implemented for this mechanism. The L2 learning switch gave an average of 5.2 flow installations per second, while the proof of concept implementation resulted in an average of 2.6 flow installations per second. This is a clear difference and a potential consideration when implementing such mechanisms. The distributed nature of the approach described within this paper, however, means that the number of devices connected to each gateway/controller should be

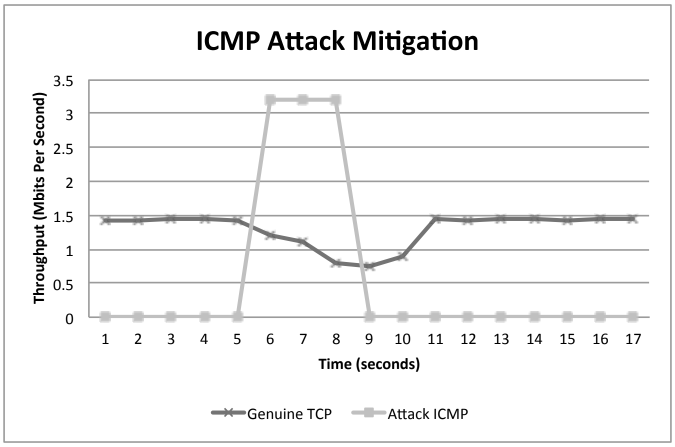

Fig. 6. ICMP Flood Experimentation Results

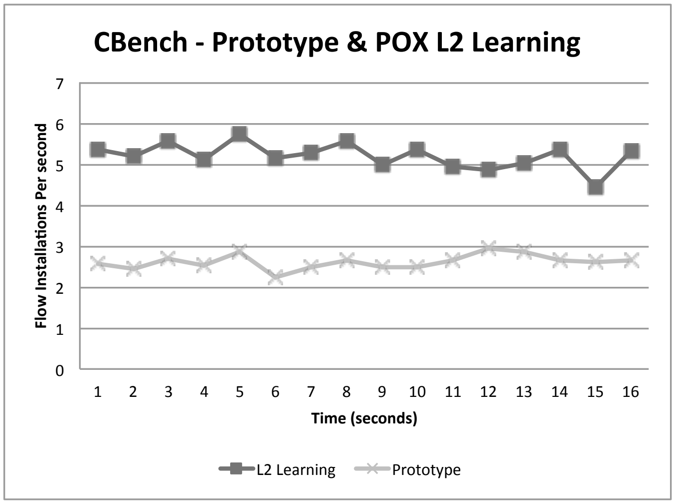

Fig. 7. CBench Controller Flow Installation Performance

relatively low, so the lower number of flow installations per second should not be prohibitive.

VI. Conclusion & Future Work

This paper has provided an overview of the need for a flexible and dynamic method of IoT security, and has discussed the potential for an SDN based gateway to address this issue. A flexible flow based security mechanism for IoT based devices has been proposed and developed using the Pox controller. While the initial test results only cover the blocking of a small number of attacks, it has both successfully validated the approach and provided a platform upon which future expansions can be developed.

Future work will focus on two main directions. Firstly, the expansion of the current implementation, and hardware based test-bed development. Secondly the dynamic traffic analysis tools component shall be abstract as a means of providing this as a service to other network applications running on top of SDN controllers.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli, “An Experimental Study of Security and Privacy Risks with Emerging Household Appliances (Position Paper),” pp. 79–84, 2014.

[3] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, “Security of the Internet of Things: perspectives and challenges,” *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[5] A. Y. Ding, J. Crowcroft, S. Tarkoma, and H. Flinck, “Software defined networking for security enhancement in wireless mobile networks,” *Computer Networks*, vol. 66, pp. 94–101, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2014.03.009

[6] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, “Performance evaluation of attribute-based encryption: Toward data privacy in the iot,” in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 725–730.

[7] S. Zhu, S. Setia, and S. Jajodia, “Leap+: Efficient security mechanisms for large-scale distributed sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 4, pp. 500–528, 2006.

[8] R. H. Weber, “Internet of things–new security and privacy challenges,” *Computer Law & Security Review*, vol. 26, no. 1, pp. 23–30, 2010.

[9] J. Kurose and K. Ross, *Computer Networking - A Top Down Approach*, 4th ed. Addison Wesley, 2007.

[10] R. Roman, P. Najera, and J. Lopez, “Securing the internet of things,” *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[11] Y. Jarraya, T. Madi, and M. Debbabi, “A Survey and a Layered Taxonomy of Software-Defined Networking,” *IEEE Communication Surveys & Tutorials*, vol. 16, no. 4, pp. 1955–1980, 2014.

[12] Open Networking Foundation, “SDN Architecture Overview version 1.1,” Tech. Rep., 2014.

[13] X. Liu, H. Xue, X. Feng, and Y. Dai, “Design of the multi-level security network switch system which restricts covert channel,” in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. IEEE, 2011, pp. 233–237.

[14] G. Yao, J. Bi, and P. Xiao, “Source address validation solution with openflow/nox architecture,” in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*. IEEE, 2011, pp. 7–12.

[15] J. Suh, H.-g. Choi, W. Yoon, T. You, T. Kwon, and Y. Choi, “Implementation of a content-oriented networking architecture (cona): A focus on ddos countermeasure,” in *Proceedings of European NetFPGA developers workshop*, 2010.

[16] C. YuHunag, T. MinChi, C. YaoTing, C. YuChieh, and C. YanRen, “A novel design for future on-demand service and security,” in *2010 IEEE 12th International Conference on Communication Technology*, 2010, pp. 385–388.

[17] R. Braga, E. Mota, and A. Passito, “Lightweight ddos flooding attack detection using nox/openflow,” in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*. IEEE, 2010, pp. 408–415.

[18] Y. Zhang, “An adaptive flow counting method for anomaly detection in SDN,” *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies - CoNEXT ’13*, pp. 25–30, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2535372.2535411

[19] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, “Blueprint for introducing innovation into wireless mobile networks,” in *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*. ACM, 2010, pp. 25–32.

[20] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, “Towards programmable enterprise wlans with odin,” in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 115–120.

[21] V. Sivaraman, T. Moors, H. Habibi Gharakheili, D. Ong, J. Matthews, and C. Russell, “Virtualizing the access network via open apis,” in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 31–42.

[22] Radware Ltd., “DefenseFlow 2.0 - Flow Based Detection,” Tech. Rep., 2015.

[23] R. Krishnan, A. Ghanwani, Dell, S. Kini, Ericsson, D. Mcdysan, Verizon, D. Lopez, and Telefonica, “Large Flow Use Cases for I2RS PBR and QoS,” 2014.

[24] P. Bull, R. Austin, and M. Sharma, “Pre-emptive Flow Installation for Internet of Things Devices within Software Defined Networks,” in *3rd International Conference on Future Internet of Things and Cloud*, 2015, pp. 124–130.

[25] H. Kim, J. Kim, and Y.-B. Ko, “Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking,” *16th International Conference on Advanced Communication Technology*, pp. 758–761, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6779064

[26] Praetox, “Low Orbit Ion Cannon,” 2014. [Online]. Available: https://sourceforge.net/projects/loic/