Towards a Multilayer Strategy Against Attacks on IoT Environments *

Davino Mauro Junior, Walber Rodrigues, Kiev Gama, José A. Suruagy, Paulo Andre da S. Gonçalves
Universidade Federal de Pernambuco - Centro de Informática (CIn/UFPE)
Recife, Brazil
{dmtsj, wmr, kiev, suruagy, pasg}@cin.ufpe.br

***Abstract*—The Internet of Things market has seen a large growth in numbers in the recent past. With it, security is becoming a usual concern among consumers. By taking as a starting point an already existing categorization of typical IoT attacks grouped by TCP/IP network layers, we did a non-exhaustive search of solutions addressing each attack. We found that solutions are typically focused on a single layer or even a specific attack only. Furthermore, these solutions lack flexibility to incorporate new attacks. To avoid the non-practical approach of having multiple non-extensible tools, this paper presents an ongoing work that focuses on a multilayer approach to IoT threats. The proposed system leverages an autonomic architecture to analyze network traffic in a distributed manner, detecting suspicious behavior with preconfigured rules being applied by a Complex Event Processing (CEP) engine.**

***Index Terms*—Security; Internet of Things; MAPE-K;**

I. Introduction

The Internet of Things (IoT) is growing rapidly, with over 24 billion devices expected to be installed by 2020 [1]. This exponential growth reflects on various ecosystems; from industrial to connected houses. Security solutions focused on these ecosystems are crucial as these devices often have access to sensitive information.

The Smart Home ecosystem represents a big part of the IoT market share [2], thus requiring special attention. Devices which on a recent past were considered rare, e.g., smart cameras and smart locks, are now actively present in consumer's houses, leading to new challenges both in security and privacy domains. In terms of security, there are multiple scenarios that causes concern among users who wish to purchase these devices. For example, Fernandes et al. showed how a compromised smart lock from the IoT Smart Home platform Smart Things could allow an attacker to enter the house without dweller consent [3].

Privacy is also a serious concern among users. For instance, Nest smart cameras, Google's solution for IoT indoor and outdoor surveillance, were recently targeted by hackers. After compromising a Nest baby monitor, an attacker did broadcasts of voice messages threatening parents, who experienced a dreadful situation, feeling invaded and unconfortable [4]. This is just one, among many examples, but cases like this demonstrate the importance of addressing the security concerns involving these smart devices.

Different approaches and tools addressing these attacks were presented in the recent past. Due to the large number of IoT attacks, it is important to categorize them to have a better understanding to appropriately provide solutions to detect and neutralize such threats. Nawir et al. [5] and Ashraf et al. [6] proposed a categorization for the different types of IoT attacks. The former uses a fine-grained classification based on TCP/IP network layers, while the latter provides a coarse-grained perspective of three layers (Cloud, Network and M2M layers). Since we are focused on home IoT, we focused on the fine-grained one.

In section II-B, we did a non-exhaustive search of IoT threat defense tools. We matched them with their respective attacks grouped under different network layers, as presented by Nawir et al. [5]. This categorization brought up two main limitations of these solutions. First, these solutions often act on a single network layer, forcing the user to have multiple tools in place as to prevent the attacks. Second, most of these solutions do not provide any means to extend the tool in question for new attacks, lacking flexibility.

Thus, there is an open challenge concerning the integration and flexibility of the approaches to many IoT threats into a tool that could concentrate these defense mechanisms. It would allow an easier adoption as well as the ability to easily incorporate solutions to new threat patterns that may appear. As an attempt to address these limitations, we present our ongoing work on the *IoT-Flows* platform, which consists on a new security system for IoT environments focusing on two limitations of the state of the art: **extensibility** and **multilayer defense**. Our approach followed the MAPE-K reference architecture [7], typically used in autonomous systems.

II. Background and Related Work

In this section, we revisit a study on the taxonomy of IoT attacks presented by Nawir et al. [5], using it as baseline for a non-exhaustive search on IoT solutions for each attack.

A. Categorization of IoT Attacks

Nawir et al. propose a security attack categorization for each network layer [5]. Table I shows the attacks distributed on the different layers. In the **Physical** layer, the study categorizes jamming and tampering attacks. *Jamming* consists in constantly, deceptively or randomly compromising the communication channel with meaningful data, whereas *Tampering* is physically attacking the device [8]. On the **Data Link** layer, attacks are classified as collision, resource exhaustion and unfairness. *Collision* consists on sending packets at the same time of legitimate data packets, harming specific packets instead of the whole channel [8]. Resource *Exhaustion* are attacks that force the device to consume its resources deliberately, for example,

This material is based upon work supported by RNP (Brazil) and NSF (USA) under Grant Nos. 1740897 and 1740916.

sending multiple requests to devices that uses batteries until the battery dies. Some MAC protocols give priority to devices that are very low on battery, the *Unfairness* attack consists in creating a low battery device to have priority when sending packets denying real traffic [8].

The majority of attacks are reported on the **Network** layer, i.e., spoofed, altered or replayed routing information, selective forwarding, sinkhole, sybil, wormholes, HELLO flood and acknowledgement spoofing. The attacks based on *Spoofed, Altered or Replayed routing information* are based on unprotected ad-hoc networks, where the routing can be compromised. *Selective forwarding* is related to a denial of service on a specific node with packets being dropped. *Sinkhole*, as described by [9], consists on a network node pretending to have the shortest path to other node ir order to drop packets, modify data or interfere in clustering algorithms. *Sybil* attacks are based on a node obtaining multiple fake identities and misleading other nodes on the network. *Wormholes* attacks are based on different devices on distinct connected networks, where the data from one network is sent to another in order to create real, but misleading information. *HELLO* messages are used by some protocols to establish connection or neighborhood relation and are used with a high power transmitter to create fake proximity relations [8]. *Acknowledgement spoofing* uses ack messages to pretend that a disabled node is alive or the connection between two nodes is strong than it apparently is.

On the **Transport** layer, there are flooding and de-synchronization attacks. *Flooding* consists on exploring the natural vulnerabilities of TCP and UDP protocols as the UDP have no flow control and the TCP protocol is vulnerable to SYN (new connection request) flood. De-synchronization is the interference of an attacker in order to interrupt an active connection between two nodes, using fake packets containing error or specific control flags [10]. On the **Application** layer, the vulnerabilities reported are related to system reliability and cloning attack [5]. *Reliability* issues are often related to execution problems like buffer overflow. *Cloning* is the capability of attackers steal information from devices or steal the device credentials.

B. Categorization of Solutions

Building on the taxonomy of the IoT attacks previously described, we did a non-exhaustive search with the main goal of distributing the state-of-the-art solutions on the different network layers presented. Table I shows the results.

Starting on the **Physical** layer, Namvar et al. presented a novel anti *Jamming* strategy which promotes an IoT controller to protect the IoT devices against malicious radio jammers [11]. For *Tampering* attacks, a team at the NEC Corporation developed a lightweight-architecture for tampering detection on IoT devices, using real-time inspection with no impact on the device normal operations [12]. On the **Data Link** layer, *Collision* attacks are similar to jamming, thus inheriting the same solution. Ruckebusch et al. presented a new architecture designed with IoT in mind that mitigates *Exhaustion* attacks, an after-effect of the previous attacks [13]. Regarding *Unfairness* attacks, Djedjig et al. presented a trust-based defense model to detect malicious behaviour, calculating trust levels for participating nodes [14].

On the **Network** layer, solutions often tackle more than one type of attack. SVELTE is one such case, applying real-time intrusion detection to detect *Spoofing* and *Sinkhole* attacks [15]. Huijuan et al. describes a system that uses watermarked packets to identify whether a network node is doing a *Selective Forwarding* attack, using a trust value to identify how many marked packets are dropped related to normal packet loss, skipping nodes with low trust [16]. Due to the fact of *Sybil* attacks are based on creating fake nodes in networks, Sohail et al. developed a system based on device mobility that is capable of differentiating if a node is fake or not by reading the RSSI (Received Signal Strength Indication) pattern [17]. For Pongle et al., the *Wormhole attack* is very location related, so the developed system periodically broadcasts the nearby RSSI. Then, this information is used by other devices to infer if a node is nearby or not, classifying it as compromised [18]. Singh et al. also used the RSSI but to mitigate *HELLO* attacks, considering that devices have a homogeneous signal strength, any other value too different is considered strange. If the value is close to the standard, the node will be asked to solve a puzzle that increases increases exponentially in difficulty per HELLO message. If the node fails to answer in an assigned time, the node is labeled as strange [19].

Flooding attack is commonly used on the **Transport** layer for distributed denial-of-service (DDoS) in IoT environments. Dao et al. presented how attack behaviour learning can be used to detect flooding attacks with smart filters distributed on the network [20]. *Desynchronization* attacks can be mitigated using authentication protocols like the one proposed by Fan et al. using a RFID protection scheme [21]. In the **Application** layer, common defenses include access policies to control information flows between applications and IoTdevices. One such solution was presented by Demetriou et al. with HanGuard, applying SDNs to enforce policies on Smart Home networks [22]. Another approach involves increasing data security on IoT applications. Fernandes et al. developed FloFence, a framework that enables developers to secure function executions involving sensitive data in Android-OS's created processes [23].

Although these solutions give powerful tools to secure smart devices, they are not designed to incorporate future attacks into their security models, therefore lacking flexibility when a new attack appears and not making difficult to improve the tools in an easy way. Also, the solutions do not provide a multilayer defense against the attacks, forcing the user/consumer to have multiple tools in place to improve security of their devices. In the following, we propose a system that focus on solving both of these problems.

III. The IoT-Flows Platform

A. Approach

The importance of autonomic systems for IoT security was already pointed out in a review that gathers different solutions emphasizing on autonomic computing to mitigate IoT threats [6], although with a limitation of typically addressing just one or sometimes two layers from TCP/IP. Besides the architectural approach, the concepts of self-healing and self-protection are the main aspects taken from the *self-** principle.

Table I: Categorization of State of the art solutions under the IoT Attacks taxonomy defined by Nawir et al. [5]

Layer	Attacks	Methods/Strategies	State-of-the-art solution
Physical	Jamming	Creates radio interference and exhaustion on IoT devices	Namvar et al. [11]
	Tampering	Creates compromised nodes	NEC Corporation [12]
Data Link	Collision	Simultaneous transmission of two nodes on the same frequency	Namvar et al. [11]
	Exhaustion	By repetetively colliding the nodes	Ruckebusch et al. [13]
	Unfairness	Using above link layer attacks	Nabil et al. [14]
Network	Spoofed, altered or replayed routing information	Creates routing loops, extend or shortening sources routes, attracting or repelling network from select nodes.	Raza et al. [15]
	Selective forwarding	Choose what information to gather before transmitting.	Deng et al. [16]
	Sinkhole	Compromised node tries to attract network traffic by fake advertising its fake routing update	Raza et al. [15]
	Sybil	Single node duplicates its node to be in multiple locations.	Abbas et al. [17]
	Wormholes	Selectively tunneling or retransmit information to the IoT devices.	Pongle et al. [18]
	HELLO flood	Uses HELLO packets as weapon to launch the attack on IoT system	Singh et al. [19]
	Acknowledgement spoofing	Spoof the link layer acknowledgments for overhead packets.	Raza et al. [15]
Transport	Flooding	Repeat the request of a new connection until the IoT system reaches maximum level.	Dao et al. [20]
	De-synchronization	Disruption of an existing connection.	Fan et al. [21]
Application	Clock skewing, Selective message forwarding, Data aggregation distortion	The adversaries usually masquerade like normal behavior in IoT system. Attackers also can still choose a message that he/she intend in the IoT system and launched their own malicious activities.	Demetriou et al. [22] Fernandes et al. [23]

We propose the *IoT-Flows* platform, which is a system that employs this autonomic approach in a distributed architecture with multiple components, each one possessing a specific and unique responsibility. Following an old design principle– Separation of Concerns–one could see each component as a module addressing a different concern, i.e., a problem [24]. A component could, for example, deal with the problem of monitoring the network in a distributed manner and filter the traffic data, while another component would analyze this data and look for signs of suspicious behaviour. This separation of concerns alleviates the complexity of a security enforcement system monitoring different networks with heterogeneous devices. The architecture of the system is based on the MAPE-K, a traditional architecture blueprint originally designed for self-adaptive systems. For brevity reasons, we will not discuss the architecture on this paper, please refer to the original white paper for a detailed discussion [7].

IoT-Flows focuses on securing communication in-between IoT devices. It acts on the different network layers, providing a **multilayer defense** for IoT environments. The system is able to monitor the traffic on the different WiFi networks that the smart devices are connected to. It also provides **extensibility**, allowing the user of the system to incorporate new attacks into the defense model, in the form of Complex Event Processing (CEP) rules. Currently we have developed patterns against attacks in three TCP/IP layers: **Network**, **Transport**, and **Application** layers. For instance, while monitoring the network, the system is able to detect that an IoT device is being targeted for *Acknowledgement Spoofing* with a fake device trying to masquerade the official device. At the same time, on the transport layer, the attacker would be flooding the IoT device with multiple requests and could also act on the application layer, trying to masquerade normal behaviour requests, like turning the device on or off. The system is able to detect any of these behaviours while monitoring the network traffic and applying pre-configured rules that analyze the packets being sniffed. Once a suspicious behavior is detected, the system can alert the user or block all requests directed to the IoT device in question, therefore stopping the attack.

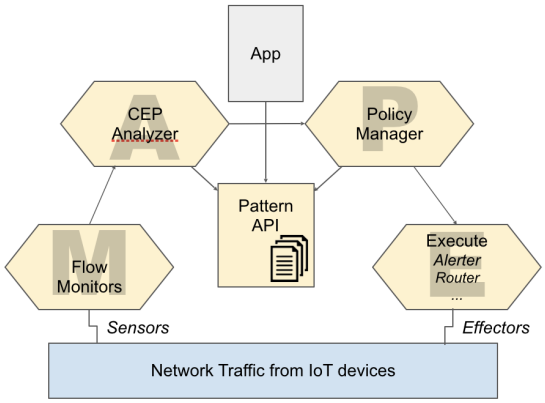

Figure 1: Architecture of the IoT-Flows platform.

IoT-Flows allows the user to download new security patches that provides detection of new attacks while also providing manual configuration, if needed. We have tested the approach of having an extensible mechanism based on Complex Event Processing rules that allows to easily include the identification of new attacks. Some drawbacks are the need to understand the rule language of the CEP Engine (Esper[1]) and understanding the metadata of the packet structure in order to write the rules.

B. Architecture

Based on the architectural parts of the MAPE-K blueprint [7], we define the following components (shown in Figure 1): (i) Flow Monitor(s), (ii) Pattern API, (iii) CEP Analyzer, (iv) Policy manager, (v) Execute components (Alerter/Router/etc). We briefly describe each component below according to their matching role on the MAPE-K architecture, however, MAPE-K's Knowledge component is currently not implemented thus we do not describe it. Implementation details were omitted due to space limitations.

Network Monitor. The network monitor relates to the *Monitor* component on the MAPE-Karchitecture. Its main

[1] www.espertech.com/esper

responsibility is to aggreggate and filter traffic data, i.e., the resource that is monitored, generating "events" to be analyzed by the next component.

Pattern API. This is a core component of the system where the rules are stored and relates to the Analysis and Plan parts of the MAPE-K. A *Pattern* maps a rule to a certain action. For example, one could create a rule to block flows from a certain IoT device to an unwanted address. The system would detect this behavior trying to match the pre-defined rules to the network data being analyzed by the *Monitors*. Then, once matched, the system would then perform a pre-configured action, e.g., alert the user or block the network request. A support Web App allows platform users to write these rules and deploy them into *IoT-Flows*.

CEP Analyzer. This component is based on the Analysis part of the MAPE-K architecture. Its main responsibility is to receive the aggregated traffic data from the monitors and apply Complex Event Processing (CEP) to match these data against pre-defined patterns.

Policy Manager. This component acts as a bridge between the *CEP Analyzer* and the *Execute* components, i.e., the Alerter, Router, etc. It maps a pattern to a pre-defined action. For example, once the system detects a suspicious behavior that needs a certain action, say a DDoS attack that needs to trigger an email alert for certain users and also to block requests, this component would both

Execute Components. These components are logically one in responsibility. The *Alerter* component is responsible for generating an email or SMS alert to the user after a suspicious activity is detected or the component is configured to do so under certain circumstances, e.g., if a smartphone tries to connect to an IoT device. The *Router* component is responsible for applying enforcement policies to the devices, e.g., it could block a request to an unwanted endpoint originating from an IoT device on the network.

IV. Conclusions and Future Work

Securing IoT devices against both old and new network attacks is crucial. After revisiting the different types of IoT attacks categorized in literature, we did a non-exhaustive search for state-of-the-art solutions targeting those attacks and grouped them under the respective target network layers. We found that solutions for the different types of IoT attacks exist, but they lack flexibility and are often bound to a single network layer. Trying to address these limitations, we presented our ongoing work on the *IoT-Flows* platform. It consists of a system acting as a line of defense that is able to integrate defense mechanisms for IoT attacks expanding across different network layers. The proposed system is also extensible, giving flexibility to incorporate patterns that can identify new attacks and act upon their detection. Our approach is based on the MAPE-K reference architecture for autonomous systems.

In the future, apart from completing and evaluating the proposed system, we envision a crosslayer solution for IoT environments using input from different layers in a simultaneous manner as to prevent complex attacks. In addition, we plan to implement MAPE-K's Knowledge component by employing machine learning techniques to analyze traffic history.

References

[1] Business Insider, "There will be 24 billion iot devices installed on earth by 2020," 2016. [Online]. Available: https://www.businessinsider.com/there-will-be-34-billion-iot-devices-installed-on-earth-by-2020-2016-5

[2] Forbes, "2017 roundup of internet of things forecasts," 2005. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#7b71aae11480

[3] Michigan News, "Hacking into homes: 'smart home' security flaws found in popular system," 2016. [Online]. Available: https://news.umich.edu/hacking-into-homes-smart-home-security-flaws-found-in-popular-system/

[4] Washington Post, "'I'm in your baby's room': A hacker took over a baby monitor and broadcast threats, parents say," 2018. [Online]. Available: https://www.washingtonpost.com/technology/2018/12/20/nest-cam-baby-monitor-hacked-kidnap-threat-came-device-parents-say/

[5] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of things (iot): Taxonomy of security attacks," in *Electronic Design (ICED), 2016 3rd International Conference on*. IEEE, 2016, pp. 321–326.

[6] Q. M. Ashraf and M. H. Habaebi, "Autonomic schemes for threat mitigation in internet of things," *Journal of Network and Computer Applications*, vol. 49, pp. 112–127, 2015.

[7] IBM, "An architectural blueprint for autonomic computing,," 2005. [Online]. Available: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

[8] A. Tayebi, S. Berber, and A. Swain, "Wireless sensor network attacks: An overview and critical analysis," in *Sensing Technology (ICST), 2013 Seventh International Conference on*. IEEE, 2013, pp. 97–102.

[9] H. Shafiei, A. Khonsari, H. Derakhshi, and P. Mousavi, "Detection and mitigation of sinkhole attacks in wireless sensor networks," *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 644–653, 2014.

[10] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *IEEE Pervasive Computing*, no. 1, pp. 74–81, 2008.

[11] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, "Jamming in the internet of things: A game-theoretic perspective," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[12] NEC Corporation, "Lightweight-architecture tamper detection technology to protect iot devices," 2018. [Online]. Available: https://www.nec.com/en/global/rd/technologies/falsification_find/index.html

[13] P. Ruckebusch, E. De Poorter, C. Fortuna, and I. Moerman, "Gitar: Generic extension for internet-of-things architectures enabling dynamic updates of network and application modules," *Ad Hoc Networks*, vol. 36, pp. 127–151, 2016.

[14] D. Nabil, D. Tandjaoui, I. Romdhani, and F. Medjek, "Trust-based defence model against mac unfairness attacks for iot," 07 2017.

[15] S. Raza, L. Wallgren, and T. Voigt, "Svelte: Real-time intrusion detection in the internet of things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[16] H. Deng, X. Sun, B. Wang, and Y. Cao, "Selective forwarding attack detection using watermark in wsns," in *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, vol. 3. IEEE, 2009, pp. 109–113.

[17] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat, "Lightweight sybil attack detection in manets," *IEEE systems journal*, vol. 7, no. 2, pp. 236–248, 2013.

[18] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in internet of things," *International Journal of Computer Applications*, vol. 121, no. 9, 2015.

[19] V. P. Singh, S. Jain, and J. Singhai, "Hello flood attack and its countermeasures in wireless sensor networks," *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 3, p. 23, 2010.

[20] N. Dao, T. V. Phan, U. Sa'ad, J. Kim, T. Bauschert, and S. Cho, "Securing heterogeneous iot with intelligent ddos attack behavior learning," *CoRR*, vol. abs/1711.06041, 2017. [Online]. Available: http://arxiv.org/abs/1711.06041

[21] K. Fan, W. Jiang, H. Li, and Y. Yang, "Lightweight rfid protocol for medical privacy protection in iot," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, April 2018.

[22] S. Demetriou, N. Zhang, Y. Lee, X. Wang, C. A. Gunter, X. Zhou, and M. Grace, "Hanguard: Sdn-driven protection of smart home wifi devices from malicious mobile apps," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2017, pp. 122–133.

[23] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "Flowfence: Practical data protection for emerging iot application frameworks." in *USENIX Security Symposium*, 2016, pp. 531–548.

[24] R. J. Mitchell, *Managing complexity in software engineering*. IET, 1990, no. 17.