A Framework for Security Monitoring of Real IoT Testbeds

Vinh Hoa La[1], Edgardo Montes de Oca[1], Wissam Mallouli[1], Ana R. Cavalli[1,2]
[1]*Montimage, 39 rue Bobillot, 75013 Paris, France*
[2]*SAMOVAR, CNRS, Telecom SudParis, Paris-Saclay University, 9 rue Charles Fourier 91011 EVRY, France*
{*vinh_hoa.la, edgardo.montesdeoca, wissam.mallouli*}*@montimage.com, ana.cavalli@it-sudparis.eu*

Keywords: Intrusion Detection, Anomaly Detection, 6LoWPAN, Wireless Sensor Networks, IoT, Security Monitoring.

Abstract: Internet of Things (IoT) has been acknowledged as a novel transformation technology because of its wide range of applications in various domains, namely connected agriculture, industrial control, smart buildings and home automation. It promises innovative business models and improved user experience. However, the devices are prone to failures and malicious attacks on account of their resource-constrained characteristics. In this paper, we present a framework for security monitoring of IoT systems. It is based on MMT-IoT, which is a reactive monitoring tool to be deployed in a running IoT environment to address malicious behaviors, failures and attacks. In this paper we also present the experiments conducted on two practical IoT-6LoWPAN testbeds. The preliminary results confirmed the efficiency of the proposed solution.

1 INTRODUCTION

Computer security, also known as cyber-security or IT security, has been an emerging topic for decades. It is expected to attract even more attention due to the increasing reliance on computer systems in many different domains. Computer systems here are not limited to servers, desktops or laptops but also include *smart devices* (e.g. smart-phones, connected objects, sensor devices). The pervasiveness of these systems goes together with the growing of cyber-attacks in both volume and sophistication. According to a study made by *Symantec* [1] in 2015, nearly one million new malware threats are released every day. Two-thirds of Internet users have been victims of cyber-crime, with more than 1.5 million new victims every day.

Additionally, the incredible growth of Internet and wireless networks, based on technologies such as Bluetooth and Wi-Fi and the concept *Internet of Things* (IoT), promise to make future networks become *Internet of Every Things*. There are nowadays about 15 billion of IoT devices and they are estimated to be 50 billion connected devices in 2020, according to a report by Cisco and DHL [Macaulay et al., 2015]. As a representation, Wireless Sensor Networks (WSNs) have been attracting a lot of interest from both the research community and the public. However, the resource-constrained characteristics of physical objects in those networks presumably limit the design and development of security protocols. Whilst, sensor nodes, which usually operate in remote, unattended and even harsh environments, are prone to failures and malicious attacks.

In the last years, the research on IoT/WSNs was mainly focused on how to make the concept of IoT realistic and practical. In other words, most of the IoT research projects have been trying to qualify this technology by standardizing the communication protocols, ameliorating the performance of the IoT systems, optimizing the resource consumption, etc. Security is always considered as an important issue but difficult to achieve thoroughly because it seems contradictory with the system's performance due to the resource constraints of IoT devices.

To date, there are a number of research works on the subject of IoT security. However, they mostly concentrate on **designing** secure communication protocols, light encryption, authentication, data freshness (avoiding packet injection), etc. Recently, researchers are paying more and more attention on monitoring in general and intrusion detection in particular for IoT/WSNs. However, many existing approaches are still at the design level and not yet implemented.

In this paper, we propose a framework for security monitoring of IoT systems based on the MMT-IoT tool. The tool allows capturing and analysing the traffic generated by the IoT devices, as well as visualising the findings. The solution has been de-

[1]https://www.symantec.com/security-center/threat-report

ployed and tested on two *FED4FIRE+*[2] platforms: w-iLab.t[3] (provided by imec) and Log-a-Tec[4] (provided by the Jožef Stefan Institute - JSI). These experiments have been decisive for raising the maturity level of the MMT-IoT solution and, thus, provide a clearer view of its market potential.

The rest of this paper is organized as follows. Section 2 gives a presentation of the proposed framework. It includes an introduction of MMT-IoT, as well as the adaptations performed to make the tool applicable for the FED4FIRE+ testbeds. These testbeds are described in Section 3. The experimental results of testing MMT-IoT in the two testbeds are presented in Section 4. Finally, we conclude our study and identify necessary future work in Section 5.

2 A FRAMEWORK FOR THE MONITORING OF IOT NETWORKS

2.1 MMT-IoT a security monitoring tool for IoT networks: General architecture

Montimage has developed an extensible monitoring framework called Montimage Monitoring Tool (MMT) [Wehbi et al., 2012, Mallouli et al., 2012]. It has been conceived as a modular approach to analyse security properties of conventional networks (both wired and wireless) by means of extracting statistical information from the network protocols and feeding it to an engine to perform a temporal logic-based analysis. Montimage has adapted this technology to, respectively, the Cyber Physical Systems (CPS/IoT) and 5G networks, considering the particular requirements of these networks.

The MMT-IoT [La et al., 2016, La and Cavalli, 2016] has thus been developed to be used with the MMT-Probe software. In general, the main goal of the MMT-IoT solution is designed to avoid performing heavy operations on the IoT devices, leaving the security analysis for the traditional MMT-Probe solution. Since the latter is a Linux-based tool, it is implicitly constrained to the protocols that the Linux kernel is capable of handling. In particular, the IEEE 802.15.4 Protocol (IoT-specific Layer 2 protocol) is not natively supported by the Linux kernel. In this case, when Linux protocol stack tries to identify the layer 2 protocol, it will not understand the frame format and, in consequence, discard the packets they reach the Linux network drivers. To avoid this, an abstraction layer needs to be inserted between the IoT traffic and MMT-Probe, so the latter will be able to capture the traffic from any traditional Linux interface.

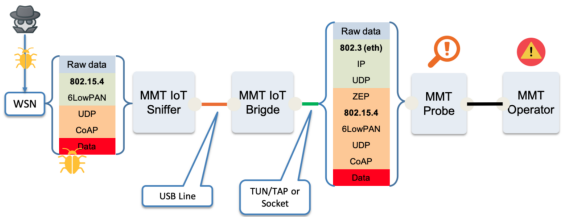

Figure 1: MMT-IoT general architecture

Figure 1 shows how the MMT-IoT technology is capable of extracting the information from the IoT protocols. In order to correctly adapt this approach (designed initially for traditional Ethernet networks) it was required to split the network extractor (sniffer) in two parts: the MMT-IoT Sniffer and the MMT-IoT Bridge. The former is the IoT endpoint that is in charge of sniffing the packets and forwarding them to a more powerful machine through a USB line. The latter recovers the transferred packets from the USB line and injects them (encapsulated using the ZEP protocol) in the loopback interface of the machine, making the packets ready to be analysed by MMT-Probe (in charge of parsing the protocol communications and analysing them) and visualised by the MMT-Operator (in charge of collecting the information provided by the probes and visualising statistics and alarms).

The described technology is the core of the proposed IoT framework, that is tested in the context of this paper to determine its performance and scalability, as well as to evaluate the effectiveness of the attack/anomaly detection on IoT networks.

2.2 Adaptation of MMT-IoT to the Zolertia Re-Mote Devices

As mentioned before, Montimage required the deployment of its software on real IoT devices in order that it can be used in the FED4FIRE+ platforms. To this end, Montimage selected the Zolertia Re-Mote devices [5] for its ease of deployment and its wide range of support for IoT operating systems.

Figure 2 presents the general architecture of the experiments performed using MMT-IoT. The number

[2] https://www.fed4fire.eu/
[3] https://doc.ilabt.imec.be
[4] http://www.log-a-tec.eu/
[5] https://zolertia.io/

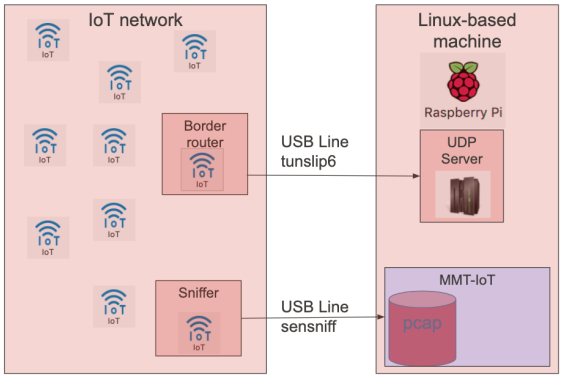

Figure 2: General architecture for the experiments

of the Zolertia devices acting as the clients may vary, depending on the objective of the experiment or the availability of the devices. In summary, the following adaptations have been performed in order to facilitate the application of MMT-IoT:

- Border Router firmware: The Border Router is the edge device placed between the IoT network and the traditional IP network. In the context of this work, the Border Router connects with a Linux-based machine that acts as the gateway collecting the sensed data sent by the IoT 6LoWPAN devices (i.e. UDP clients), and transferring it to a dedicated UDP Server via a USB line tunslip6 connection. Tunslip [6] is a tool used to bridge IP traffic between a host and another network element over a serial line. The firmware is for forming a 6LoWPAN network and routing the messages to the server.
- Client firmware: The clients and the Border Router self-organise among themselves to form a 6LoWPAN network. The normal clients are configured to report sensed data every 10 seconds [7], whilst the attacker client behaves interchangeably in one of the three modes (Normal, DoS attack and Dead modes) changing every 60 seconds.
- Sniffer: The sniffer includes a peripheral firmware capturing all network frames and streaming them to the host and a python script reading network packets captured by the peripheral, converting them to the PCAP format and piping them to the Linux-based machine via the USB line.
- MMT-IoT: A number of adaptations/modifications have been integrated to MMT so that the tool is able to work with IEEE 802.15.4/6LoWPAN traffic. This included developing new plugins (for parsing these packets), enabling the configuration of the network stack, specifying new security rules, and designing new dashboards.

[6] https://github.com/contiki-os/contiki/blob/master/tools/tunslip6.c

[7] Note that values are purely illustrative and depend on the business application

3 SET-UP OF THE EXPERIMENTS

In the context of these experiments, Montimage used two FED4FIRE+ testbeds to assess and test the prototype and get feedback to improve the solution to reach TRL6 [8]. In particular, Montimage pursued two principal objectives with these experiments:

- Analyse the performance and the scalability of the MMT-IoT solution in a real IoT scenario (represented in Figure 2),
- Perform security analysis in IoT networks by means of detecting well-known attack types.

3.1 w-iLab.t Testbed

The w-iLab.t platform allows its user to run remote experiments in a fully automated way. Because of limited interference from outside, repeatable experimentation is enabled, which allows the experimenters to compare results between different experiment runs. Montimage has created credentials on the FED4FIRE+ portal, with which all testbeds in the federation can be accessed. For this experiment it was necessary to reserve all the available nodes (embedded PC of type Intel NUC) from the "Datacenter" floor of the testbed (as shown in Figure 2). Every node in the w-iLab.t testbed has at least one Zolertia Re-Mote sensor connected over USB. This sensor node can be accessed after logging in to the embedded PC. In addition to the reservation, to access these nodes (over SSH) it was required to design an experiment using the jFED-Experimenter tool. Before doing this, we defined the two scenarios corresponding to the two main objectives of the experiments run on the w-iLab.t:

- A main objective of using the w.iLab.t platform is to perform scalability tests of the MMT-IoT software. In this sense, we needed a big number of nodes that are capable of generating test traffic that will be captured by a central sniffer. More precisely, 21 clients were flashed with normal (i.e. reporting the data every 10 seconds) or attacker (i.e. behaving interchangeably in the

[8] TRL: Technology Readiness Level

Normal mode, DoS attack mode and Dead mode) firmware so that we could increase the bitrate of sensed data on the air to determine the limit of the sniffer in terms of capturing traffic. In addition, the MMT-IoT could be tested to determine if it could detect the DDoS (Distributed Denial of Service) attack when multiple nodes act as attackers.

- Another objective is performing security tests on IoT networks. In this sense, we modified the firmware flashed on the IoT device that generates the malicious traffic so that it can be analysed and detected by the MMT-IoT. Indeed, 7 normal clients and 1 attacker were used. MMT-IoT is tested to see if it could capture all the traffic with no packet lost, extract the statistics, visualise them using graphs, as well as detect 3 types of anomalies:
  - DoS attack: Instead of reporting every 10 seconds like the normal nodes, the attacker sends sensed data 100 times faster, i.e. 10 messages/second;
  - Dead node (node failure): A normal node that should send sensed data, but that stopped reporting (e.g. due to a failure).
  - Incorrect FCS (Frame Check Sequence): The FCS field contains a number that is calculated by the source node based on the data in the frame. This number is added to the end of a frame that is sent. When the destination node receives the frame, the FCS number is recalculated and compared with the FCS number included in the frame. If they are different, the frame will be considered malformed (intentionally or not) or modified between the route from the source node to the destination node. During the period of the DoS attack mode, the attacker sends messages with incorrect FCS. In reality, an incorrect FCS can signify a malformed packet (e.g. due to a misconfiguration or an error in the implementation), a jamming attack (i.e. the attacker abuses the network by generating frames that should be ignored) or a message manipulation attack (i.e. the attacker intercepts and modifies a frame's content).

Each scenario was executed (i.e. triggering of the attack, start of the sniffing process along with the security analysis) for at least 5 minutes, in order to generate enough traffic that could be later analysed.

3.2 Log-a-Tec Testbed

Montimage prepared a set of equipment forming a "small" IoT 6LoWPAN network. It includes four Zolertia RE-Motes, one Raspberry Pi and the accessories (cables, USB hub, power adapters, etc.) to constitute:

- A border router mote that acts as the gateway collecting the sensed data from other motes and forwards the reports via the USB line to the server deployed in the Raspberry Pi.
- A sniffer mote that runs in a passive manner: listening for air traffic, capturing and piping it via the USB line connected to the Raspberry Pi.
- A normal client mote that reports the data every 10 seconds.
- An attacker mote that behaves interchangeably in the three modes described before (Normal, DoS attack and Dead mode).
- A Raspberry Pi that feeds the motes in terms of batteries, hosting the server dealing with the sensed data and receiving the sniffed traffic which is then analysed by MMT-IoT.

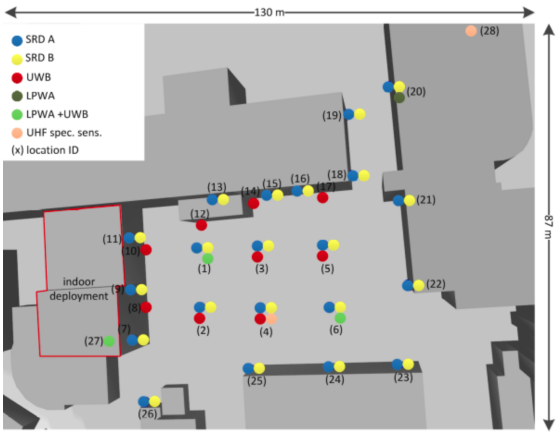

Figure 3: The distribution of the Log-a-Tec Testbed

The devices were sent to JSI to be placed near the Log-a-Tec Testbed whose distribution is depicted in Figure 3 (on the ground and in the office building). The devices of JSI at the yellow dots are configured to form a 6LoWPAN [G. Montenegro and Culler, 2007, Vasseur et al., 2011] network on the same channel (channel 15) as Montimage's Zolertia Re-Motes. The devices at the position 4 and 5 were offline at the moment of the experiments. Montimage's devices and the Border Router of JSI's network were set up at different places:

- Outdoor: at the position 6,
- Outdoor: in a park, on a table between the position 3 and 5,
- Indoor: in the office at the position 9.

4 EXPERIMENTAL RESULTS

In each experiment, the traffic captured is analysed by the MMT-Probe to extract the statistics, to verify the security violations defined in MMT-Security rules, and to visualise the obtained results on a MMT-Operator dashboard. The current version of the dashboard permits displaying the traffic bitrate, the most active nodes, the most active links, the topology, the security alerts and their distribution. In the context of this work, anomalies were detected based on specific symptoms, such as the following:

- DoS attack: MMT-IoT raises an alert about a possible DoS attack if it sees a node reporting data more than 4 times in one second. The values "4 times" and "one second" are configurable and depend on the behaviour of the real use case.
- DDoS attack: MMT-IoT raises an alert about a possible DDoS attack if it sees two nodes reporting at least twice in a second. Similarly to the item above, "two nodes" and "twice a second" are configurable parameters depending on the real use case.
- Dead node (node failure): MMT-IoT raises an alert about a possible dead node if it sees a node that should send sensed data but does not. For instance, should send a report every 10 seconds but stops for at least 20 seconds.
- Incorrect FCS: MMT-IoT calculates the FCS and compares it with the FCS included in the packet of the sender. A difference triggers an alert of an incorrect FCS that might be related to a misconfiguration, an implementation error, a malformed packet, a jamming attack, or a manipulation attack in which the payload was modified intentionally.

4.1 w-iLab.t Testbed

Regarding the scalability and security tests with 27 clients, we discovered that 32 Kbps was the maximum data bit rate that can be handled by the sniffer (shown in Figure 5). Above this limit, the sniffer started dropping some frames. In theory, by default the Zolertia Re-Mote is capable of working at 50 Kbps data rates. In practice, the limit in the context of our experiments was found to be lower. This can be explained by the physical factors that can affect the transmission, namely the antenna, the power, the noise, etc., and by the fact that such IoT devices are not always stable.

The DoS attack can be alerted as shown in Figure 4 but can also be intuitively observed by just looking at Figure 5. The throughput greatly increased when the DoS attack took place. Of course the variation

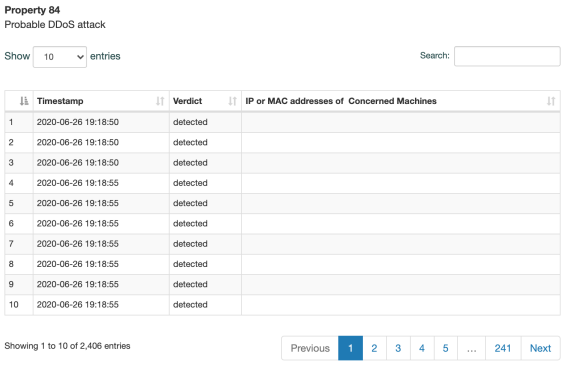

Figure 4: DDoS alerts in the test with 27 clients

between normal operation and operation under attack depends on the use case that could imply low or high number of communications, and low or high variability in the traffic.

Regarding the security tests with 7 clients and 1 attacker, MMT-IoT raised the alerts correctly when the attacker changed from the Normal mode to the DoS attack and Dead mode, as shown in Figure 6 and 7 respectively. The messages containing incorrect FCS were also detected, as demonstrated in Figure 8. The attacker could be identified not only based on the security alerts but also by looking at the traffic volume generated. For example, as displayed in Figure 9, the attacker is the most active node.

4.2 Log-a-Tec Testbed

As mentioned in the previous sections, our IoT devices were placed at different positions near the Log-a-Tec testbed. However, regardless of the position of JSI's Border Router and Montimage's devices, the sniffer captured both traffic generated by Montimage's Zolertia Re-Motes and by JSI's 6LoWPAN network. This was determined by the fact that there were only 4 Zolertia Re-motes but the sniffer captured the traffic of at least 10 nodes. In addition, after capturing the traffic and having it analysed by MMT-IoT, we observed that there was, in general, more traffic captured due to two reasons:

- The sniffer "S" (together with Montimage's equipment) is placed in a central position (at position 6 or between 3 and 5 in Figure 3),
- The sniffer is placed close to the Border Router "BR" of JSI's network.

As a result, the least traffic was observed when "S" was indoor at the position 9 (behind metallized glass) and "BR" was at 6 (Figures 10). There were only 10 nodes found generating the traffic at a rather low data rate (less than 1 Kbps). Meanwhile, most traffic was witnessed when both "S" and "BR" were on the table

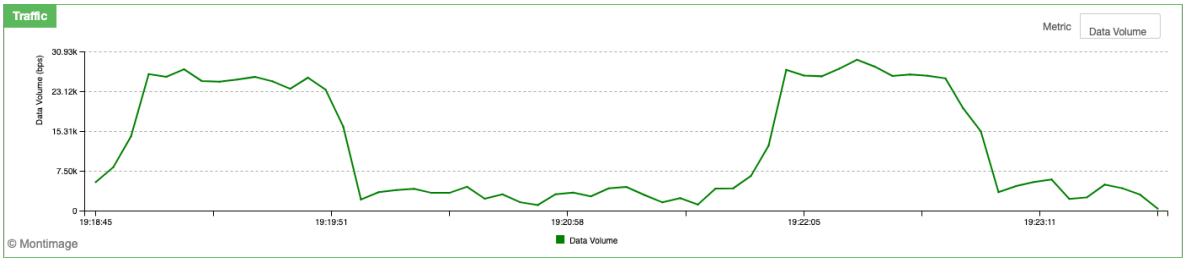

Figure 5: Traffic bitrate observed in the test with 27 clients

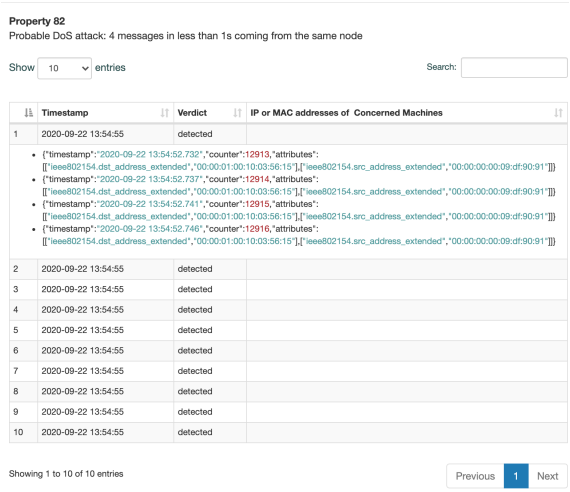

Figure 6: DoS alerts in the test with 7 clients and 1 attacker

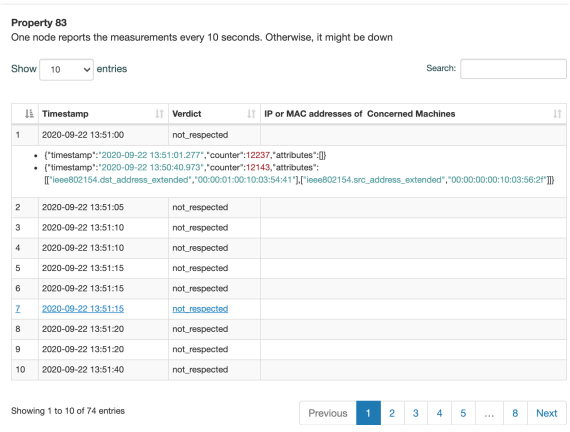

Figure 7: Dead node alerts in the test with 7 clients and 1 attacker

between 3 and 5 (Figures 11). 22 nodes were seen generating the traffic with the bitrate of up to 3Kbps.

Regarding the security aspect, MMT detected the misbehavior of the attacker node in all the performed experiments(as shown in Figure 12). The attacker was identified when it triggered the DoS attack or when it acted as a dead node and did not report any data at all. The messages containing the incorrect FCS were also alerted.

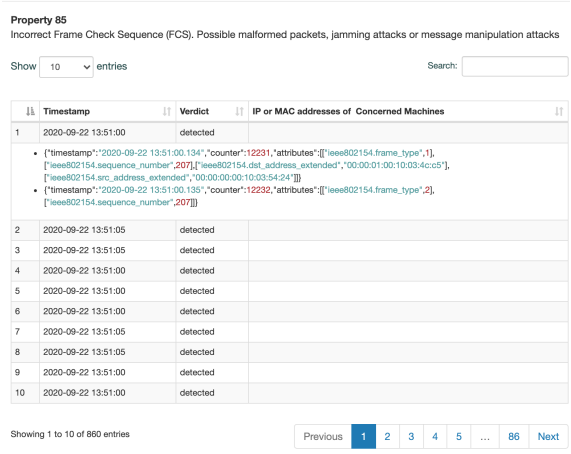

Figure 8: Incorrect FCS alerts in the test with 7 clients and 1 attacker

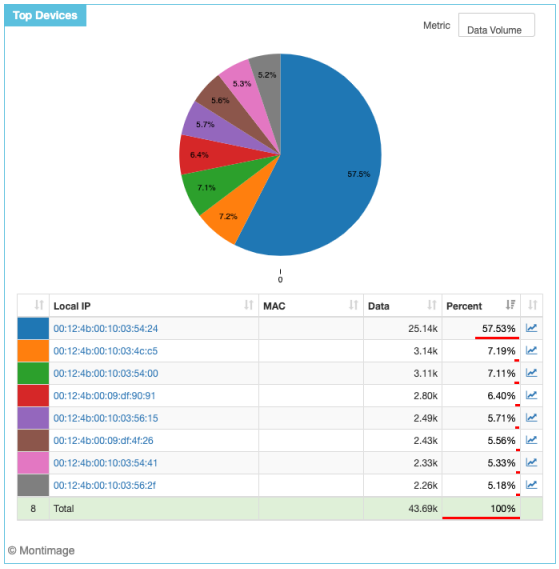

Figure 9: Most active nodes in the test with 7 clients and 1 attacker

Most surprisingly, although Montimage prepared only one Zolertia Remote acting as the DoS attacker, the DDoS attack was still detected. This means that there was another node of IJS's testbed sending data at a very high rate. For example, as demonstrated in

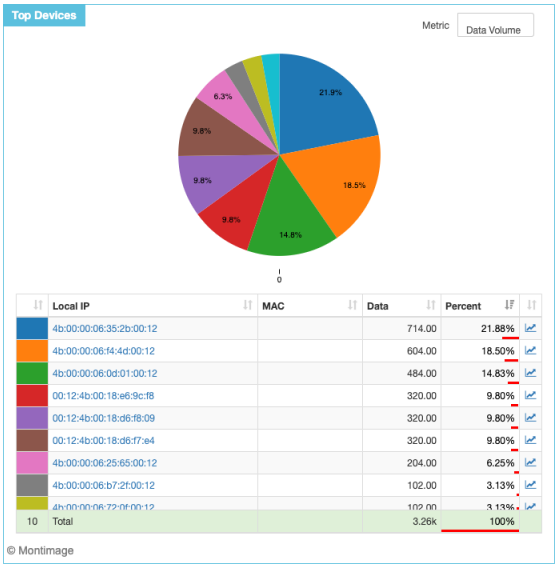

Figure 10: Active nodes observed when "S" was at 9 and "BR" was at 6

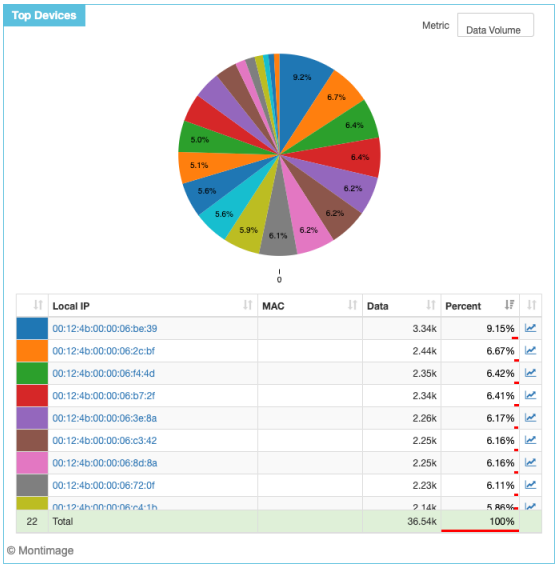

Figure 11: Most active nodes observed when both "S" and "BR" were between 3 and 5

Figure 13, there were two nodes reporting data twice in several milliseconds which were alerted as a probable DDoS attack.

This finding is explained by the fact that the devices in Log-a-Tec communicate using the 6TiSCH standard[9] where devices re-transmit the packets if the acknowledgment for the sent packets is not received. In this case, it sends 8 packets in a row within a short interval of time, since the time slot of the 6TiSCH is only 10ms. It could happen that a device leaves the network without informing the other devices so that if one tries to reach it, no ACK will be received. This behavior was captured because it violated the security policy defined for the experiment, but the detection rule can be changed to avoid generating DDoS detection alarms for these cases.

[9]https://datatracker.ietf.org/wg/6tisch

5 CONCLUSIONS

The network traffic analysis results presented in this paper confirm that the proposed framework based on the MMT-IoT tool behaves as expected in a real IoT environment. This allowed us to perform an initial validation of the technology and use the methods developed to plan a more extensive validation and stress testing in the future. With this we aim to further increase the TRL of the MMT-IoT solution, adding value to the whole MMT suite and expanding the domains of applicability of the Montimage Monitoring Tool framework. In comparison with other similar IDS (Intrution Detection System) tools proposed for IoT/ 6LoWPAN-based WSNs (e.g., SVELTE [Raza et al., 2013]), MMT-IoT attempts to passively monitor the network based on the sniffer without createing additional traffic that might be costly in the IoT systems.

Nevertheless, MMT-IoT's scalability, as mentioned above, depends on the capacity of the sniffer. Therefore, a more powerful/ dedicated node should be used for performing the sniffing. In addition, by inspecting the source code of the sniffer, the processing time should be proportional to the size of the packet, due mainly to the copying of memory and writing to the USB line. It is natural to think that bigger packets will reduce the number of captured packets, suggesting an inverse relationship between these variables. Considering this, it is not clear which will be the optimal point that maximises the throughput, considering that the size of the packet and the number of packets captured could have a potentially inverse relationship. Also, this leads to a possible evasion: the attacker may saturate the sniffer itself by injecting, for instance, "big packets". This needs to be confirmed by more experimentation and will probably lead to the development of new evasion avoidance techniques.

ACKNOWLEDGEMENTS

This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 732638, project FED4FIRE+.

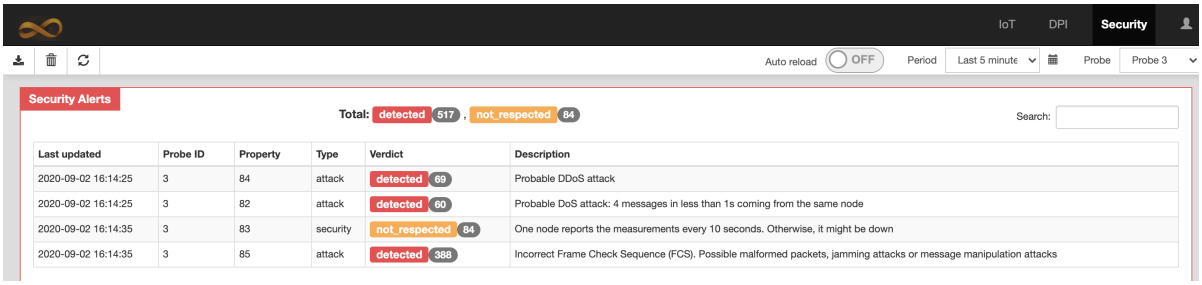

Figure 12: Anomalies detection in the test with Log-a-Tec testbed

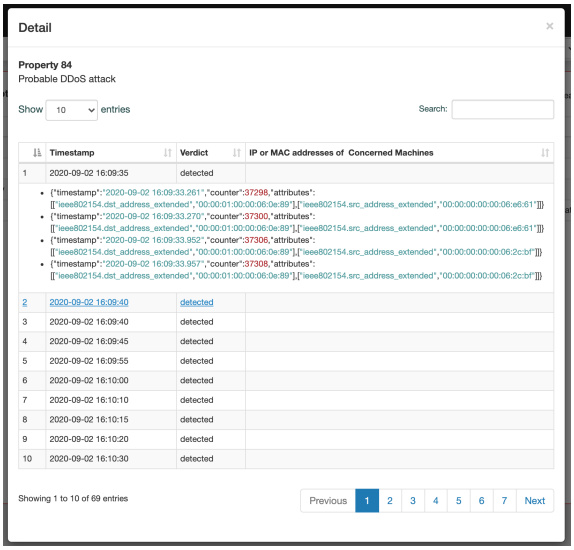

Figure 13: DDoS attack detection in the test with Log-a-Tec testbed

REFERENCES

G. Montenegro, N. Kushalnagar, J. H. and Culler, D. (2007). Transmission of ipv6 packets over ieee 802.15. 4 networks. IETF Tech. Rep. RFC 4944.

La, V. H. and Cavalli, A. R. (2016). A misbehavior node detection algorithm for 6LoWPAN Wireless Sensor Networks. In *Proceedings of 36th IEEE International Conference on Distributed Computing Systems (ICDCS 2016), Second IEEE International Workshop on Security Testing and Monitoring (STAM 2016)*.

La, V. H., Fuentes, R., and Cavalli, A. R. (2016). A Novel Monitoring Solution for 6LoWPAN-based Wireless Sensor Networks. In *Proceedings of 22nd Asia-Pacific Conference on Communications (APCC 2016)*.

Macaulay, J., Buckalew, L., and Chung, G. (2015). Internet of Things in logistics. Cisco and DHL report.

Mallouli, W., Wehbi, B., and de Oca, E. M. (2012). Online Network Traffic Security Inspection Using MMT Tool. In *Systems Testing and Validation Workshop 2012*, pages 23–31.

Raza, S., Wallgren, L., and Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks*, 11(8):2661–2674.

Vasseur, J. P., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., and Chauvenet, C. (2011). RPL: The IP routing protocol designed for low power and lossy networks. *In Internet Protocol for Smart Objects (IPSO) Alliance*, (April):20.

Wehbi, B., Montes de Oca, E., and Bourdelles, M. (2012). Events-Based Security Monitoring Using MMT Tool. In *IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST), 2012*, pages 860–863.