Intrusion Detection and Prevention with Internet-integrated CoAP Sensing Applications

Jorge Granjal and Artur Pedroso
CISUC/DEI, University of Coimbra, Coimbra, Portugal

Keywords: Internet-integrated Sensor Networks, CoAP, 6LoWPAN, IoT, Intrusion Detection.

Abstract: End-to-end communications between Internet devices and Internet-integrated constrained wireless sensing platforms promise to contribute to the enabling of many of the envisioned IoT applications. In this context, communication technologies such as 6LoWPAN and CoAP are currently materializing this vision, and we may fairly observe that security in the presence of such devices, and particularly in the context of end-to-end communications with Internet-integrated WSN, will be of prime importance. Considering the constraints of sensing devices in terms of critical resources such as energy, memory and computational capability, it is clear that Internet-integrated WSN will need security against various types of attacks, particularly those originated at devices without the constraints of WSN sensors (e.g. Internet hosts). Existing encryption strategies for communications in IoT environments are unable to protect the WSN for Denial of Service (DoS) and other intrusion attacks, particularly in what regards the usage of CoAP to enable application-layer communications. Therefore, anomaly and intrusion detection will play a major role in the enabling of IoT applications in various areas. In this article, we approach a framework to cope with intrusion detection and reaction in CoAP Internet-integrated WSN, and in the context of this framework we implement and evaluate various complementary detection and prevention mechanisms. Our proposal is evaluated experimentally and ours is, as far as our knowledge goes, the first proposal with the above-mentioned goals.

1 INTRODUCTION

Most of the applications envisioned for the Internet of Things (IoT) are critical in respect to security. The IoT will be enabled by communication and security technologies based on the 6LoWPAN (IPv6 over Low power WPAN) (Montenegro et al., 2007) adaptation layer, in particular by the Constrained Application Protocol (CoAP) (Bormann et al., 2012), which was designed to support RESTful application-layer communications with heterogeneous constrained sensing and actuating platforms. Another relevant protocol is RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) (Winter, 2012), which is a routing framework adaptable to various types of IoT application areas. CoAP is beginning to enable transparent and pervasive web communications supporting IoT applications, but on the other end such Internet-integrated devices are exposed to external threats, particularly in the form of Denial of Service (DoS) attacks and attacks subverting the usage rules of CoAP. Although security mechanisms may be adopted to secure CoAP communications, such as DTLS or even IPSec, such mechanisms only protect against external attacks, and are also not able to offer protection against DoS threats. In fact, cryptography cannot detect attackers with legal keys, either internal or external, but behaving maliciously. Intrusion detection and prevention will thus play a very important role in enabling most of the envisioned IoT applications (Palattella et al., 2013), and appropriate solutions need to be designed that ate able to protect CoAP communication environments, not only from DoS-related attacks, but also from attacks design to explore vulnerabilities of the newly designed CoAP application-layer protocol. Thus, unlike in isolated WSN (wireless sensor networks) environments, 6LoWPAN networks use IP and are directly connected to the untrusted Internet, opening the door to new threats that must be prevented and dealt with. Although there are various proposals in the literature focusing on intrusion detection in WSN environments, most of such proposals are not fit for the IoT. Many proposals are designed without a central manager or controller, without considering the availability of message security and considering that sensing devices are not capable of global communica-

Granjal, J. and Pedroso, A.
Intrusion Detection and Prevention with Internet-integrated CoAP Sensing Applications.
DOI: 10.5220/0006777901640172
In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, pages 164-172
ISBN: 978-989-758-296-7

tions and identified globally (e.g. by an IP address). In the context of the IoT, the global Internet architecture is starting to encompass sensing devices, and in consequence 6LBR (6LoWPAN border routers) are assumed to be always accessible, end-to-end message security is required and devices are globally identified by IPv6 addresses (Palattella et al., 2013). Considering such goals, we consider a set of representative attacks against devices using CoAP, which we evaluate in the context of a framework for application-layer (CoAP) intrusion detection and reaction. The article is structured as follows. We begin by discussing intrusion detection prevention in the IoT in the next Section, and in Section III we discuss our framework, together with its main components and the messaging format employed for security-related management procedures. In Section IV we focus on intrusion detection and reaction mechanisms implemented and evaluated later in the article, in Section V. Finally, in Section VI we conclude our discussion.

2 INTRUSION DETECTION AND PREVENTION USING CoAP

In classic approaches on intrusion detection and prevention in the Internet three complementary approaches are normally considered: signature-based, anomaly-based and specification-based systems. A signature or misuse-based IDS first defines patterns of the known attacks and checks the traffic against such known attacks. Mechanisms in this class are usually characterized by low-false alarm rates, although they need to store large data sets (signatures and also the data to be analyzed) and are limited in detecting new attacks. In anomaly-based intrusion detection, normal network behaviors are first classified, and compared with monitored operations and communications, in order to detect anomalous activities. This class of systems possess the ability to detect new attacks but can be characterized by a high false-alarm rate. Finally, specification-based systems are a variant of anomaly-based systems, and work by specifying normal network operations in detail and monitoring any breaking of that specification. Such system decrease the false detection rate but on the other hand the operation patterns must be usually created by specialists. The current trend in IDS research in the context of the IoT is to combine these before mentioned methods, in order to jointly benefit from the qualities of the various approaches. Other useful characterization of intrusion detection and prevention is in what respects the topology of the employed architecture, which may be either distributed, centralized or hybrid. In the former, the role of detection and reaction to attacks may be supported by various devices in the network, while in distributed systems one single system is responsible for such tasks. A hybrid system combines distributed intrusion detection supported by devices in the network with a central manager, usually responsible for more complex analysis and decisions operations. In this article, we consider the implementation of a hybrid intrusion detection and prevention architecture employing signature-based, as well as DoS detection. Looking at recent (less than five years) research proposals dealing with intrusion detection and prevention in 6LoWPAN and CoAP environments, we find proposals mostly focused on protecting against attacks on routing using RPL in 6LoWPAN environments. A first approach towards IDS in IoT environments is presented in (Raza et al., 2013), in the form of SVELTE, a system designed to protect WSN from attacks against routing operations, in particular spoofed or altered information, sinkhole and selective-forwarding. Attacks are detected by maintaining a dedicated routing information in the 6LBR, which is constructed from RPL information and also from information reported by the various sensors, for the purpose of detecting inconsistencies in the routing tree. SVELTE is mostly focused on RPL-based 6LoWPAN networks, and this proposal doesnt address security against attacks at the network and upper layers, neither DoS or other types of attacks. In (Le et al., 2012) the authors focus again on threats against RPL, and propose a two-layer IDS architecture designed to detect internal attacks on routing operations, based on three components: an RPL specification-based monitor, an anomaly-based used in cooperation with the specification-based to monitor the node performance and a statistical-based component to reveal the attacker source. Although this work performs a good job in discussing the applicability of WSN IDS systems to IoT 6LoWPAN environments, it is mostly focused on internal attacks against RPL. Also, the described system model is also not materialized in the form of concrete detection and reaction mechanisms. In (Lee et al., 2014) the authors propose an intrusion detection method based on evaluating over time the energy consumption of sensing devices. The authors classify sensing devices with irregular energy consumptions as malicious attackers, by considering energy consumption models built for communications in a 6LoWPAN network, in both the mesh-under and route-over operation modes of IEEE 802.15.4. From simulation, the authors state that this strategy may allow to detect misbehaving nodes and that such nodes may thus be excluded from operations in the network. One limitation of this approach is that IoT applica-

tions may not always present a homogeneous energy consumption profile over time, and other is that in this work the focus is not on attacks at the network or application layers. In (Rghioui et al., 2014) the authors also focus on discussing DoS attacks against RPL operations. Despite their discussion on various type of attacks against the routing layer, as well as various recommendations on aspects such as the placement of IDS system, as well as on the types of systems to employ and the parameters to consider in various scenarios, the article does not propose nor evaluate any particular mechanisms. In (Kasinathan et al., 2013) the authors propose a DoS architecture for 6LoWPAN, designed to integrate with the network framework developed within the EU FP7 project ebbits. This framework is focused on critical network environments, and the proposed architecture has been designed and evaluated for industrial environments. A preliminary implementation is also discussed and evaluated using a penetration testing system. The proposed system is focused on jamming attacks, but is dependent on information modules available on the eebits architecture, and security-related notifications are dependent on the usage of a dedicated communications medium for security-related data. The authors in (Surendar and Umamakeswari, 2016) propose an improvement to SVELTE when dealing on attacks against routing, particularly focused on sinkhole attacks. The proposal is evaluated through simulation while, again, this proposal focuses on attacks against a particular class of attacks against routing in 6LoWPAN environments. In (Shreenivas et al., 2017) SVELTE is again improved by adding a new parameter, in the form of a link reliability metric which helps in preventing the 6LBR and neighbouring nodes to actively engage with malicious intruders. The implementation and evaluation are based on Contiki and COOJA is used for simulations, and the authors claim that their proposal improves the true positive rate of SVELTE. In (Rghioui et al., 2015) the authors address the design of a system based on detecting misbehaving nodes in a 6LoWPAN networks, with the assumption that neighbour nodes in such a network behave similarly (in terms of communications) during the lifetime of the application. The proposal is evaluated against its performance (true positives and false positives) and found to perform better, the same applying to energy. We note that this work is focused again on attacks against routing operations in a 6LoWPAN network, in this case considering that all nodes in a given DODAG are supposed to operate similarly. From the previous analysis, we may observe that, on the one hand, intrusion detection and prevention in 6LoWPAN environments is very recent. On the other hand, most of the existing proposal focus on attacks against routing using the RPL framework. If it is true that some proposals deal with attacks (DoS attacks) disrupting 6LoWPAN operations, we have found no proposals considering the conjugation of DoS attacks with attacks perpetrated at the application-layer.

3 A FRAMEWORK FOR INTRUSION DETECTION AND PREVENTION WITH CoAP

We now proceed by presenting the proposed framework for intrusion detection and prevention in CoAP Internet-integrated sensing environments. With this goal in mind, we start by identifying the system and security requirements, and proceed to discuss the operation of the various modules that constitute our framework. We also address security-related management and later in the article the intrusion detection techniques implemented.

3.1 System and Security Requirements

As previously discussed, IoT applications are being envisioned and implemented in areas as diverse as smart cities, surveillance and smart energy, among others, that will require fundamental security assurances from the infrastructure, and this certainly applies also to the capability to detect and react timely to attacks against the availability of such devices and of the IoT application. As we have previously analyzed, there is currently a lack of systems and mechanisms designed to enable intrusion detection in CoAP networks, and therefore this motivates us towards the proposal of a framework with the following goals:

- Cross-layer attack detection: being able to detect attacks at the network (6LoWPAN), transport (RPL) and application (CoAP) layers.
- Detect attacks originated at external (namely, Internet hosts) and internal devices (e.g. other sensing devices, either in the same or in a separate WSN domain).
- Intrusion prevention and filtering: be able to react (timely) to attacks by blocking attackers.
- Extensibility: support intrusion and prevention mechanisms at the various layers, according to the requirements of the IoT application at hand.
- Configurability: support reconfigurable detection and reaction policies, during the lifetime of the application.

Taking into consideration the previously identified atributes, we proceed by describing our proposed framework for intrusion detection and prevention in IoT CoAP environments.

3.2 Intrusion Detection and Prevention Framework

For an intrusion detection and prevention system to be feasible in most Internet-integrated WSN environments, it must cope with the resource constraints of sensing and actuating platforms, while on the other hand being able to adapt and benefit to devices with less constraints, as is usually the case with 6LoWPAN Border Routers (6LBR). As previously discussed, for this reason, we adopt a hybrid approach to the support of intrusion and detection functionalities. Figure 1 illustrates the system model for the proposed system.

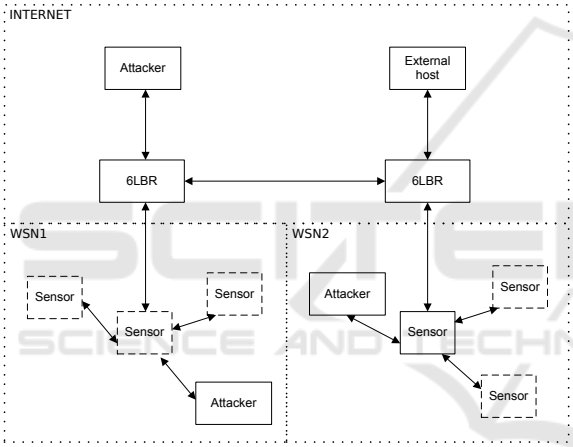

Figure 1: System architecture for intrusion detection and prevention in CoAP Internet-integrated networks.

It is also important to identify the trust model assumed in the definition and usage of our framework. We consider that devices assuming the role of intrusion detection, prevention and filtering of communications are trusted, and this applies to both constrained sensing devices and gateways (6LBR entities). Thus, we focus on security against external attackers, from the point of view of the communications and also of the devices participating in normal operations. We also assume that more intensive processing is performed by the 6LBR, thus allowing us to benefit from the resouces available in this platform, while the various sensors in the WSN domains also cooperate in the task of detecting and reacting to attacks against the security of the network. As illustrated in Figure 1, attackers may be internal for a given WSN domain but also external (e.g. an Internet host). Thus, the cooperation of the various devices in the role of attack detection and blocking is fundamental in order to timely stop attacks, either at the end device being attacked or, if necessary, by blocking communications between the Internet and WSN domains (at the 6LBR). Thus, the hybrid approach allows for the rapid detection of attacks and the timely distributed reaction to such attacks, also with the help of border routers. For example, if certain conditions arise, the 6LBR may block communications between WSN domains, or originated at the Internet, and also to instruct the various sensing devices in the WSN to start blocking communications from a particular origin device. We proceed by identifying the modules that materialize our intrusion detection and reaction system, in the context of the sensing device in Figure 2 and of the 6LBR in Figure 3.

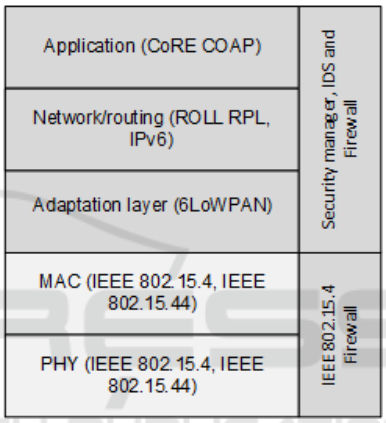

Figure 2: IDS and firewall in the stack (sensing device).

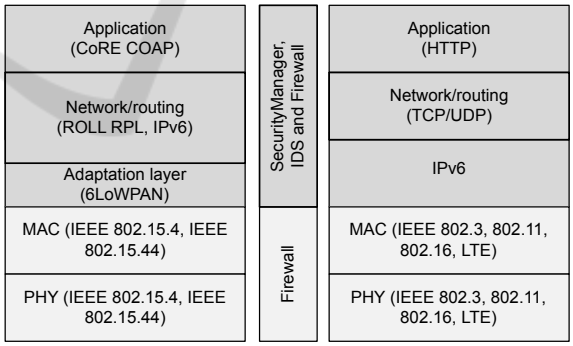

Figure 3: IDS and firewall in the stack (6LBR).

We note that encryption can be enabled either at the physical and data-link layers in the WSN, through IEEE 802.15.4 encryption, or by using DTLS or IPSec. As illustrated in Figures 2 and 3, we enable detection of attacks by analyzing traffic at various layers of the communications stack, in particular from the 6LoWPAN adaptation layer up. The framework also supports a firewall to filter out undesired communications at the various stages (layers) of process-

ing in the context of the 6LoWPAN/CoAP networking stack. Thus, a given message may be blocked, if conditions arise, at the network, routing or application (CoAP) layers.

3.3 Security Management

As previously discussed, the security module in sensing devices and in the 6LBR assumes the role of generating, receiving and processing security management messages. Such messages allow us to transmit critical information regarding attacks that are detected and on how the various devices may act in order to coordinately stop such attacks. As we focus on intrusion detection and prevention on CoAP IoT networks, security management messages are transported in the payload of CoAP confirmable messages, as such being inherently protected from packet lost (Bormann et al., 2012). In Figure 4 we illustrates the format for security management messages exchanges between devices in the context of our framework and we also assume that communications in this context may be protected via encryption either at the neywork layer using IPSec or at the transport layer using DTLS. In this figure, at the top we illustrate the format for security messages originated at a sensing device, and at the bottom security messages originated at the 6LBR.

Figure 4: Format of security notification messages.

In the ip_index we store a position in a vector storing structures storing an IP address, the number of distinct requests received in the sensor from that source IP address and a flag indicating weather messages from that address are already blocked or not. The index_request indicates the type of request received by the constrained device (and that has subsequently motivated the generation of this security notification message). Finally, in the num_requests field we transport the number of requests received so far by the sensor, of type index_request, from the IP address stored in the position ip_index of the vector. As for messages originated at the 6LBR, ip_to_block is used again as an index to a position in a vector maintained in the destination sensing device, containing the information about the blocked origin device. We may also note that the vector maintaining information about each entity know in the network, together with the counter of requests received from the that entity and blocking information, must be maintained in the various devices of the system in a synchronized fashion. This information, together with the identification of the type of request, allows for the exchange and updating of the information required for the security management modules to detect and act upon received 6LoWPAN and CoAP communications.

4 IMPLEMENTATION STRATEGY

We proceed by discussing the implementation of the proposed framework in the Contiki operating system (Dunkels et al., 2004), starting with the mechanisms that allow us to internally integrate security with the processing of 6LoWPAN and CoAP communications.

4.1 Counters and Thresholds for Attack Detection

As previously noted, the proposed framework is extensible, and extensibility here is in fact two dimensional. On the one hand, filtering and analysis mechanisms can be integrated in the framework, as illustrated in Figure 2, at the network, routing and application layers. On the other hand, such mechanisms may be designed to distinguish between different types of requests at such layers. The combination of both types of analysis and filtering lays the ground for the support of various intrusion detection strategies. We start by noting that it is fundamental to be able to detect and prevent resource exhaustion attacks at CoAP networks, particularly because the network congestion control in CoAP is not controlled by the server, in fact being implemented via transmission parameters in CoAP messages sent by client (Le et al., 2012). With such aspect in mind, we consider the detection of the number of requests that a constrained device received in a specific period (in a minute), above which the security and the stability of the WSN environment (and thus of the IoT application) may be at risk, and also attacks employing other types of messages, as well as those subverting the usage rules of the CoAP protocol. Overall, we maintain, in each sensing device, separate counters for the following types of messages received by the system during a previous time window:

- Number of valid CoAP requests to resources (sensors and actuators) available and published by the device;
- Number of invalid CoAP requests, thus malformed requests or requests to resources (sensors and actuators) that are not available in the device;
- Number of messages not intended for CoAP processing, such as ICMP, TCP and others;

The previous counters, as well as thresholds that may be activated for each type of attack, allow us to detect and react to attacks using the security manager and firewall modules, respectively.

4.2 Intrusion Detection and Prevention Policy

We now describe the intrusion detection and reaction policy considered for the purpose of evaluating the impact of the proposed mechanisms, that we discuss later in the article. In both the sensing devices and the 6LBR the rules in the policy are implemented as a set of pre-configured rules in memory, which describe the conditions triggering particular actions, with the help of the previously discussed counters. We also note that the framework is generic, in the sense that it allows for the activation of different detection and blocking policies, in line with the particular requirements of the IoT application at hand. Next we describe the security policy considered in the context of our experimental evaluation of the proposed mechanisms, that we have implemented in the constrained sensing devices. As can be seen, the security policy is defined as a set of rules that identify the conditions upon which intrusion detection and prevention takes place.

```
# For requests to resource CoAP1 received from a device with source
# IP address orig1, communications are immediately blocked
If IP=orig1 and res=CoAP1 then block

# Notify the 6LBR if more than 5 requests are received for the CoAP
# resource CoAP2, irrespective of the source IP address
If IP=* and res=CoAP2 and NReqMin>=5 then notify

# Enable security against attackers sending  undesired and malformed
# CoAP requests
If IP=* and res=malformed and NReqMin>=1 then notify
If IP=* and res=malformed and NReqMin>=3 then block and notify

# Establish a threshold to control the acceptable number of messages
# that can be accepted and processed in the device
If IP=* and res=* and NReqMin>=5 then notify
```

We start by considering requests that are already blocked, as well as requests to a particular resource that, above a particular threshold, trigger a notification to be sent to the 6LBR. We also enable security against undesired or malformed CoAP requests, by notifying and blocking further communications entering the device. Finally, we enable DoS protection via a limit of requests per minute above which we notify the 6LBR. Next we describe the same security policy, now as enforced by the 6LBR.

```
# Block external (e.g. from the Internet) requests to resource CoAP2
# on sensor1, if a notification is received from that device and
# when the number requests per minute is above 5
If NotifSource=sensor1 and IP=external and res=CoAP2 and NReqMin>=5
then blockdst=sensor1

# Notify all sensors in the WSN about a device making requests to
# resource CoAP2, when at least two sensing devices have sent alerts
If NotifSource=* and NNotif>=2 and IP=internal and res=CoAP2
then notifyblock=all

# Block malformed messages received from an internal sensing device
# and notify other internal devices also block such communications
If NotifSource=* and NNotif>=3 and res=malformed and IP=*
then notifyblock=all and blockdst=all

# Notify internal devices about all communications exceeding 7
# messages per minute, irrespective of their origin, type and CoAP
# resource requested
If NotifSource=* and NNotif>=2 and res=* and IP=* and NReqMin>=7
then notifyblock=all
```

From the policy considered for the 6LBR we may observe that the gateway is able to control and block the forwarding of 6LoWPAN and CoAP communications according to security warnings received from devices in the WSN domain, while also sending to devices in the WSN domain notifications instructing such devices on how to act on communications received from the (suspect) IP origin address.

5 EXPERIMENTAL EVALUATION

The proposed intrusion detection and prevention framework has been implemented and experimentally evaluated, with the goal of determining the effectiveness (in regard of its capability of detecting and reacting to attacks in a timely fashion) of the proposed mechanisms, as well as its impact in the critical resources available in constrained sensing platform, in particular memory, computational power and energy. We start our discussion by describing our experimental evaluation setup.

5.1 Experimental Evaluation Setup and Goals

As previously discussed, we have implemented and experimentally evaluated the proposed framework and security mechanisms. In Figure 5 we illustrate the experimental evaluation scenario. As illustrated, CoAP requests, as well as other types of messages targeting a constrained sensing device, may be originated either at another WSN device or at an external (Internet) host.

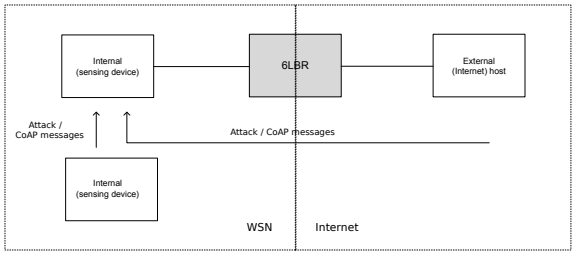

Figure 5: Experimental evaluation setup.

We have modified the source code of the Contiki operating system (Dunkels et al., 2004), with the goal

of implementing and experimentally evaluating the proposed intrusion detection and prevention mechanisms, using a MTM-CM5000MSP TelosB. As for the support of the 6LBR, we employ a Raspberry Pi model B device running Linux (Raspbian) to support forwarding of communications, as well as security management and filtering. In this setup we send different types of request messages to a CoAP sensing device and at different rates, with the goal of evaluating the effectiveness of the implemented mechanisms in dealing with DoS, as well as attacks against CoAP. Our main goal is to experimentally evaluate the impact of intrusion detection and prevention on the resources of constrained devices and on the operations of IoT applications employing 6LoWPAN and CoAP, in particular via the following strategy:

- Evaluate the impact of security on the memory of the sensing device. Memory is a scarce resource on such devices, and as such this is important in order to ascertain on the effectiveness of our proposal.
- Evaluate the energy consumption on the constrained sensing device in the presence of attacks, in comparison with normal operations, as this directly influences the achievable lifetime of IoT applications.

With the previous goals in mind, we proceed by discussing the three considered experimental evaluation scenarios.

5.2 Evaluation Scenarios

We have considered the following complementary experimental evaluation scenarios, with the following goals:

- Scenario E1 - Filtered CoAP external requests: in this scenario the sensing device is already blocking requests to resource coap1, originated at a known external (in the Internet) device.
- Scenario E2 - Notify and block at the 6LBR (external attacker): in this scenario we consider a known device in the Internet sending CoAP requests to resource coap2 on the constrained sensing device. According to the previously discussed security policy, the sensing device is configured to notify the 6LBR when a given IP address transmits 5 or more requests per minute to a given CoAP resource. Upon receiving such notifications, 6LBR blocks further forwarding of communications.
- Scenario E3 - Notify and block (internal attacker): in this scenario, we consider a known internal attacker (in the same WSN as the attacked device), sending requests to the resource coap2. The attacked sensing device is configured to notify the 6LBR when receiving 5 or more requests per minute directed to that resource, and upon receiving such notifications the 6LBR is configured to notify the devices in the WSN to block further requests received from the attacker.

Other than the three previous scenarios, we also consider the existence of a CoAP device fully exposed to internal and external communications. In this scenario, CoAP requests to resource coap1 are received from an unknown external device, and the sensor tries to honor (receive and process) all such requests, thus being fully exposed to DoS and attacks against CoAP. This scenario may thus provide us with a baseline for comparing with the aforementioned evaluation scenarios with security.

5.3 Impact on Memory

In Figure 6 we illustrate the impact of the proposed intrusion detection and reaction mechanisms, when implemented in Contiki, in the memory available in the sensing device. The implementation of CoAP with security (IDS) supports the previously discussed intrusion detection and prevention mechanisms. The impact of our IDS implementation on Contiki is of 10.6% in the case of RAM (6830 bytes are used with IDS, against 6173 without security), and of 5.6% in the case of ROM (for the program code, requiring 44926 bytes with security against 42528 bytes without security).

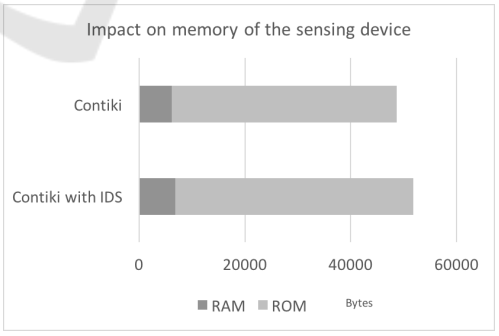

Figure 6: Impact of the proposed framework on the memory available on the sensing device.

Regarding the previously discussed measurements, we may safely consider that intrusion detection and prevention (together with the firewall and security management modules) demands an acceptable overhead on the memory available on the sensing device, thus not compromising the employment

of the previously discussed security functionalities on devices with the characteristics of the TelosB.

5.4 Impact on Energy

Energy is a critical resource on IoT environments, considering that many constrained sensing platforms still depend on batteries. Therefore, the communications and security protocols must be efficient in terms of the required energy, in order not to compromise the lifetime of the IoT application at hand. With the purpose of measuring the energy consumption of intrusion detection and prevention, we have employed Powertrace (Dunkels et al., 2011) with Energest to profile power consumption in Contiki. This tool is characterized by around 94% of reported accuracy in its energy measurements. Using Energest, we are able to measure the CPU and radio cycles spent by the sensing device, when receiving communications and processing security, measured at each 20 seconds, for a period of at least 80 seconds. In each experience, we start making requests to a sensing device in the beginning of the second period of 20 seconds and refuse the measures taken by Energest in the first period. Thus, we collect the next three 20 second periods measured by Energest, and calculate the average cycles per second used by the sensing device in a 60 second period. Energest reports the CPU, LPM, Tx and RX measurements, from which are may analytically obtain the power usage. We consider 3V as standard voltage for our calculations, and that a node is in low power mode (LPM) when the radio is off and the MCU (micro controller unit) is idle. We calculate the CPU time when the MCU is on. In Figure 7 we illustrate the power required (in mW) to process different number of CoAP requests, considering the previously discussed configurations (with and without security).

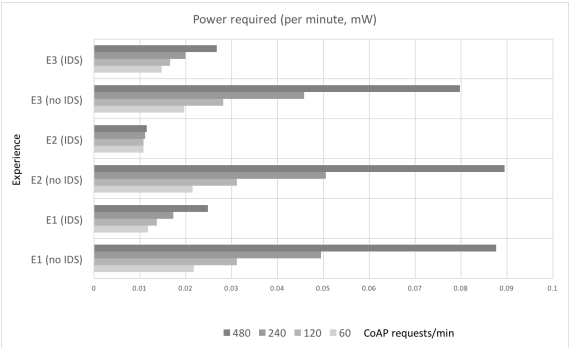

Figure 7: Power required to process CoAP requests (with and without IDS).

As can be observed in the previous Figure, the impact of the implement security mechanisms provides evident energy savings, if compared with the correspondent experimental evaluation scenarios where the sensor is exposed to attacks and thus without intrusion detection and reaction mechanisms. For example, in the scenario E3 without security the device is attacked by an internal attacker and up to 0,079 mW are required to process 480 CoAP requests in a minute, while security in this case (scenario E3 with IDS) lowers this requirement to 0,026mW, or aproximatelly 32% of the original value.

5.5 Lifetime of Sensing Applications

Another useful evaluation is on the impact of security on the lifetime of IoT applications. Thus, considering our previous measurements on the energy required in the various evaluation scenarios, and the availability of two new AA-type batteries in the sensing device, we are able to analytically derive the expected lifetime of IoT applications, that we illustrate in Figure 8.

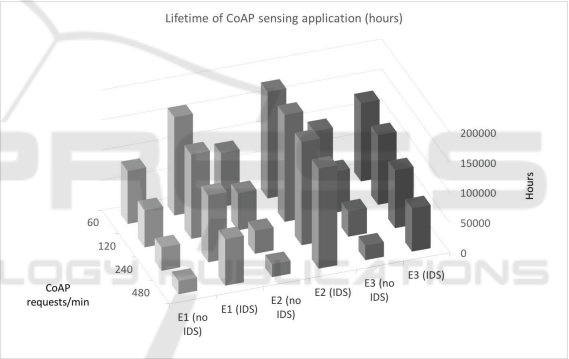

Figure 8: Lifetime (analytical) of CoAP applications (with and without IDS).

It is important to note that the previously illustrated results consider only the energy required for processing communications and security, thus not accounting the impact of other operations related with the IoT application at hand. Nevertheless, it allows us to ascertain on weather intrusion detection and prevention may compromise the goals of the application, in what respects the lifetime of sensing devices running on batteries. The effectiveness of the proposed security mechanisms is again visible in this evaluation. In the worst scenario (E3 for 480 CoAP requests per minute), security still provides approximatively 71700 hours of lifetime, in contrast with only approximately 24060 without security. Also, if we consider the baseline measurements (E1 without security), for 480 requests per minute the achievable lifetime (21900 hours) is less than one third of the counterpart with security (approximately 72240 hours).

6 CONCLUSIONS AND FUTURE WORK

In this article we propose an architecture for distributed intrusion detection and reaction in Internet-integrated CoAP sensing environments, and evaluate its effectiveness in detecting and reacting to attacks, as well as the impact of the proposed security mechanisms on the memory and energy of constrained wireless sensing devices. As we have observed, the proposed framework is flexible and extensible, so that other attacks (which can also be detected at the network, routing and application layers) can be supported in the future. As future work, we plan to extend the detection and filtering capabilities of the framework to detect new types of application attacks against the CoAP protocol.

REFERENCES

Bormann, C., Castellani, A. P., and Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62–67.

Dunkels, A., Eriksson, J., Finne, N., and Tsiftes, N. (2011). Powertrace: Network-level power profiling for low-power wireless networks.

Dunkels, A., Gronvall, B., and Voigt, T. (2004). Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE.

Kasinathan, P., Pastrone, C., Spirito, M. A., and Vinkovits, M. (2013). Denial-of-service detection in 6lowpan based internet of things. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, pages 600–607. IEEE.

Le, A., Loo, J., Lasebae, A., Aiash, M., and Luo, Y. (2012). 6lowpan: a study on qos security threats and countermeasures using intrusion detection system approach. *International Journal of Communication Systems*, 25(9):1189–1212.

Lee, T.-H., Wen, C.-H., Chang, L.-H., Chiang, H.-S., and Hsieh, M.-C. (2014). A lightweight intrusion detection scheme based on energy consumption analysis in 6lowpan. In *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, pages 1205–1213. Springer.

Montenegro, G., Kushalnagar, N., Hui, J., and Culler, D. (2007). Transmission of ipv6 packets over ieee 802.15. 4 networks. Technical report.

Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L. A., Boggia, G., and Dohler, M. (2013). Standardized protocol stack for the internet of (important) things. *IEEE communications surveys & tutorials*, 15(3):1389–1406.

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.

Rghioui, A., Khannous, A., and Bouhorma, M. (2014). Denial-of-service attacks on 6lowpan-rpl networks: Threats and an intrusion detection system proposition. *Journal of Advanced Computer Science & Technology*, 3(2):143.

Rghioui, A., Khannous, A., and Bouhorma, M. (2015). Monitoring behavior-based intrusion detection system for 6lowpan networks. *International Journal of Innovation and Applied Studies*, 11(4):894.

Shreenivas, D., Raza, S., and Voigt, T. (2017). Intrusion detection in the rpl-connected 6lowpan networks. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 31–38. ACM.

Surendar, M. and Umamakeswari, A. (2016). Indres: An intrusion detection and response system for internet of things with 6lowpan. In *Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on*, pages 1903–1908. IEEE.

Winter, T. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks.