Paper

Battery Drain Denial-of-Service Attacks and Defenses in the Internet of Things

Philokypros P. Ioulianou[1], Vassilios G. Vassilakis[1], and Michael D. Logothetis[2]

[1] *Department of Computer Science, University of York, York, United Kingdom*
[2] *Department of Electrical and Computer Engineering, University of Patras, Patras, Greece*

https://doi.org/10.26636/jtit.2019.131919

Abstract—IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is a popular routing protocol used in wireless sensor networks and in the Internet of Things (IoT). RPL was standardized by the IETF in 2012 and has been designed for devices with limited resources and capabilities. Open-source RPL implementations are supported by popular IoT operating systems (OS), such as ContikiOS and TinyOS. In this work, we investigate the possibility of battery drain Denial-of-Service (DoS) attacks in the RPL implementation of ContikiOS. In particular, we use the popular Cooja simulator and implement two types of DoS attacks, particularly version number modification and "Hello" flooding. We demonstrate the impact of these attacks on the power consumption of IoT devices. Finally, we discuss potential defenses relying on distributed intrusion detection modules.

Keywords—battery drain, ContikiOS, Cooja simulator, denial-of-service, intrusion detection, IoT, RPL.

1. Introduction

The Internet of Things (IoT) has found numerous applications in different domains, such as home automation, industrial control, health monitoring, intelligent transportation, and smart grid [1], [2]. IoT devices usually have limited resources, low computational power, small batteries, as well as limited memory and storage. Nevertheless, IoT devices are able to collect data, exchange small pieces of data through the Internet or directly with other devices, and perform lightweight computations.

Many IoT networks rely on the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [3]. In 2012, the Internet Engineering Task Force (IETF) standardized RPL, which has been designed for resource-constrained devices. Open-source RPL implementations are supported by well-known IoT operating systems (OS), such as ContikiOS [4] and TinyOS [5].

Nowadays, many security approaches and cryptographic mechanisms exist for securing traditional networks. However, oftentimes these measures cannot be applied to IoT devices due to their limited capabilities mentioned above. As a result, many IoT devices are equipped with weak or no security measures [6] and become targets of cyberattacks. Such attacks have multiplied over the past years [7]. Recent examples of cyberattacks include the Mirai botnet [8] and its evolution, Chalubo botnet [9], which exploited the default/weak passwords or OS vulnerabilities in more than 100,000 IoT devices (such as IP cameras and home routers) and launched Distributed Denial of Service (DDoS) attacks affecting multiple targets. Such incidents suggest that IoT devices offer weak security and that proper defenses should be implemented to secure consumers, businesses and critical infrastructure.

Many attacks on IoT devices that have been launched recently exploit the properties of RPL and typically include DoS [10], [11] and routing attacks [12], [13]. Detection of and effective defense against such attacks is currently an open research problem [14], [15].

In this paper, we consider the RPL implementation of ContikiOS, namely ContikiRPL [16]. We focus on two popular types of DoS attacks: "Hello" flooding [11] and version number modification [17], [18], which can drain the batteries of IoT devices. We have implemented these attacks using the Cooja simulator [19], which is used to simulate the behavior of ContikiOS. We demonstrate how these attacks may impact the power consumption of IoT devices and render some devices unreachable. Following the presentation of our simulation results, we discuss potential defenses and detection approaches. In particular, we propose a modular Intrusion Detection System (IDS) that comprises a set of distributed detection modules and a border router acting as a centralized detection module.

The rest of the paper is structured as follows: In Section 2, we describe the Cisco 7-layer IoT model we have adopted. In Section 3, we briefly review the possible attacks in IoT networks. In Section 4, we demonstrate the most significant IDS solutions currently existing for IoT. In Section 5, we describe our "Hello" flooding and version number modification attack launched with the use of the Cooja simulator and present the simulation results obtained. In Section 6, we present our proposed IDS design. In Section 7, we conclude and discuss potential future research directions.

2. Cisco's 7-Layer IoT Model

In this work, we adopt Cisco's 7-layer model [20], as it is one of the most detailed IoT references. Figure 1 shows its layers. In this paper, we mostly focus on layers 1–3.

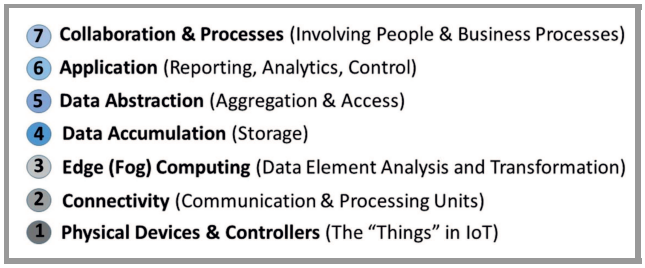

Fig. 1. Cisco's 7-layer IoT model.

Beginning from layer 1, smart or physical devices transmit or receive data. Layer 2 refers to the connectivity among the devices, located within the same network or across multiple networks. In many implementations, data may be transferred reliably between IoT devices using the existing network infrastructure. Layer 3 includes functionalities related to data analysis and transformation. Specifically, the processing of network packets occurs in that layer, so that packets are understandable to the higher layers. Layer 4 is where data is accumulated and is available for use by specific applications. Data is abstracted in layer 5. In other words, data from various sources is collected and processed to be easily accessible by applications. IoT applications read the information in layer 6. The top layer, layer 7, is where the end user's business processes live. The IoT system will become useful only when people cooperate and make use of IoT applications and their data.

3. Attacks in IoT

The majority of IoT devices have weak or no security at all, making it easy for an attacker to exploit them. As a result, critical information may be stolen from devices or they may be used to cause harm in other networks. Below, we briefly review the most significant attack types against IoT devices. The 7-layer model by Cisco is assumed to describe attacks.

At the layer of physical devices, replacing firmware of a smart device with its malicious counterpart could permit the attacker to read data in transit or data stored in the device. Another method of hardware exploitation is the non-network side-channel attack. In that attack, electromagnetic signals of the device are monitored by the attacker to expose the status of the device. DoS attack constitutes another threat for smart devices. Resource exhaustion and battery draining are some examples of DoS attacks [21]. In these attacks the attacker may prevent a device from sleeping by periodically transmitting "Hello" messages or may drain the limited power resources by submitting heavy computational tasks. Apart from DoS attacks, the adversary could attack the network by cloning a node. In this way, packets received by the node could be redirected or modified.

At the connectivity level, eavesdropping is a popular attack method in which the goal of the attacker is to export confidential information including usernames and passwords. Therefore, the attacker may learn about the network infrastructure, enter and modify device data or steal important information. Also, at this level, devices are vulnerable to Man-in-the-Middle (MitM), routing and replay attacks [22] in which attackers try to spoof and drop packets or even modify routing information. In addition, connectivity level DoS attacks may exert a negative impact on the performance of the IoT network. Some examples of DoS attacks include packet flooding and signal jamming, whose goal is to corrupt the device's communication signal. Last but not least, IoT devices may be exploited and transformed into bots to carry out DoS attacks against selected targets. Chalubo and Mirai botnets are the most recent examples of this threat [8], [9].

At the edge computing level, servers could be exploited by injecting malicious input and stealing important data. Similarly, attackers may try to leak information from a device or server to learn which services are used in the vulnerable IoT network. Database warnings or errors, for example, provide valuable information to attackers.

4. Intrusion Detection Systems for IoT

Over time, IDSs have been considered by researchers as security measures for keeping IoT networks secured. However, traditional network detection algorithms have different requirements than those based on IoT. Thus, adapting traditional methods in IoT environments is a challenging task. Certain IoT characteristics, such as the limited processing power of intelligent devices, different network structures and a variety of IoT device protocols, introduce new challenges which an IoT-based IDS must take into account [23]. Below, we present the latest IDS solutions for IoT.

The first developed IDS that aims at protecting smart devices irrespective of specific IoT protocols or applications is Kalis [24]. Kalis is a network-based, hybrid signature/anomaly-based, hybrid centralized/distributed, online IDS. The detection strategy selected depends on the specific features of the protected network. Furthermore, Kalis obtains knowledge from network-installed modules and tries to prevent intrusion by taking into account the current topology of the network and by conducting traffic analysis. Moreover, it can be extended to support new protocol standards and may improve detection performance by allowing knowledge sharing between the nodes. It is implemented on routers using the OpenWRT firmware [25]. Evaluation is performed using 6 TelosB devices programmed in TinyOS [5]. Experimental results show that Kalis achieves 100% accuracy in detecting most of the attacks. Thus, it offers better detection performance than Snort [26] and other traditional IDS solutions.

Svelte IDS is another interesting work in the field [27]. This is an anomaly- and signature-based IDS, developed to prevent RPL-based routing attacks affecting IoT devices [3]. Some of the attacks considered include selective forwarding, sink-hole attack and spoofed or altered information. As far as the approach to node placement is concerned, Svelte has a centralized module, called 6LoWPAN border router (6BR), which carries out heavy calculations, and a number of resource-restricted modules monitoring network devices. The 6BR consists of three components. The first one is the 6LoWPAN mapper which gathers information from sensors to regenerate the network. The second component is the detection system which uses the obtained information to detect potential intrusions. The third component is a mini firewall that prevents the entry of malicious traffic into the network. In IoT devices, the first and third components are integrated.

Although quite a few IoT-based IDSs have been developed recently, current solutions have certain constraints. Kalis, for example, requires deployment of detection modules specific for the attack type. This could create a complex network resulting in poor detection performance. Additionally, it utilizes Wi-Fi for communication. This means that interference between smart sensors and Kalis nodes could occur if nodes are in close proximity. Svelte has also some limitations as it is a host-based IDS, meaning that the sensor's software must be modified. This, however, would be very challenging for larger networks, which is a typical case in many IoT application domains. Another major issue is that Svelte has a high false detection rate. This was proved by Matsunaga *et al*. [28], who proposed a scheme to reduce false detection rate. However, further experiments are needed to ensure that the solution is robust and scalable.

In conclusion, a technologically enhanced solution is needed to protect IoT networks against several possible attacks. We considered the aforementioned limitations during the design of our proposed IDS.

5. Implementing Battery Drain DoS Attacks in Cooja

Before designing an effective IDS, the initial step is to implement and study the impact of several attacks on each device and on the entire network. After that, by launching attacks using various configuration parameters and intensities, different detection methods can be implemented, tested, and enhanced.

We use the Cooja simulator [19] for testing and experimentation, which is becoming increasingly popular among IoT researchers. It is also particularly suitable for experiments in the real world, since the developed applications can be directly uploaded to real hardware. Cooja can be used to simulate the behavior of ContikiOS – a popular open source IoT operating system [29].

In this work, two IoT-specific DoS attacks have been implemented in Cooja, namely version number modification and "Hello" flooding. These attacks exploit the RPL protocol's features and affect the power consumption of IoT devices. Cooja provides an implementation of the RPL protocol, called ContikiRPL [16].

5.1. RPL Overview

RPL organizes nodes along a destination-oriented, directed, acyclic graph (DODAG) [30]. The root node initiates the creation of graphs by regularly generating DODAG information object (DIO) messages, which are advertised via link-local multicasts. DIO messages include such information as identity of the root, the metrics used for routing and the depth of the originating router (called "rank").

The "Hello" flooding attack in RPL occurs when a large number of DODAG information solicitation (DIS) messages are transmitted by the malicious node to other nodes. This causes the recipient nodes to reply by sending DIO messages. Consequently, network floods with packets and the node's batteries are drained. Similarly, in the version number modification attack [18], the malicious node changes the DODAG version number before forwarding the received DIO messages to the next hop. Nodes receiving a malicious DIO message with the modified version number reset their trickle timer, store the new version number in their memory and advertise it to their neighbors via DIO messages. Note that the root uses the version number to control the so-called "global repairs" of the RPL network and to ensure that the latest routes are available to nodes in DODAG. Global repair is the repair mechanism that is initiated by the root to rebuild the network. During this process, it increases the version number of RPL DODAG and the whole DODAG is reconstructed. This method ensures a loop free and optimized tree based on the objective function used. Still, this makes IoT nodes perform useless computations and waste their energy. Thus, modifying the version number will cause unnecessary global rebuilds of the DODAG, create loops in the topology, as well as exhaust the nodes.

5.2. Simulation Scenarios and Results

Below are two scenarios, simulated in Cooja, which show the effects of the DoS attacks mentioned above. Applications of the UDP client-server model are used on top of each node. Seven Tmote Sky nodes [31] were simulated running ContikiOS. The network, depicted in Fig. 2, consists of one server (root node with ID 1) and six client nodes with IDs from 2 to 7.

In the first scenario, no compromised nodes exist. Each node is configured to send messages to the server at specific intervals. These messages contain various information about the sending node, such as its battery indicator and temperature. In the second scenario, node 7 is malicious/compromised and performs DoS attacks. Specifically, node 7 has been configured to transmit a big number of DIS messages to its neighbors. In addition, it changes the DODAG version number, so that global repairs are initiated.

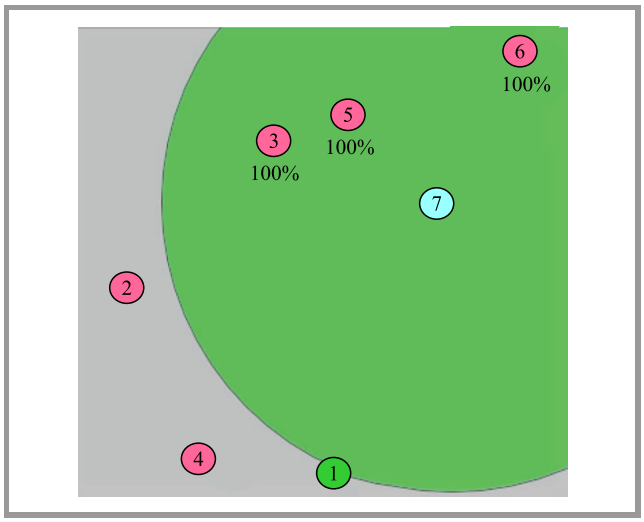

Fig. 2. Network topology for Cooja simulations.

The simulation time in our experiments in each scenario is 10 min. The configuration used for each node is shown in Table 1. The server is the receiver of all messages exchanged in the network. As a result, it is always powered on. The Radio Duty Cycle (RDC) driver is responsible for saving as much as possible power for the device. ContikiOS implements several RDC drivers, but the server uses NullRDC, which does not save power. The Medium Access Control (MAC) driver is responsible for reliably transferring packets via the radio medium. If any collisions occur, it re-transmits the packets until they are delivered. All nodes in our scenarios use the Carrier Sense Multiple Access (CSMA) driver at the MAC layer to guarantee packet delivery. In contrast with benign and malicious nodes, the server does not transmit DIS messages.

Benign nodes send data to the server. They are configured to send a DIS message every 60 s until they successfully join the network. The malicious node broadcasts 80 DIS message every second, thus launching the "Hello" flooding attack. Benign and malicious nodes in this scenario are configured to use NullRDC as RDC driver and CSMA as MAC driver. This setup will keep the devices always on.

In the first scenario, the network topology is formed as shown in Fig. 3. The numbers displayed on each link indicate the expected number of transmission (ETX) that a node

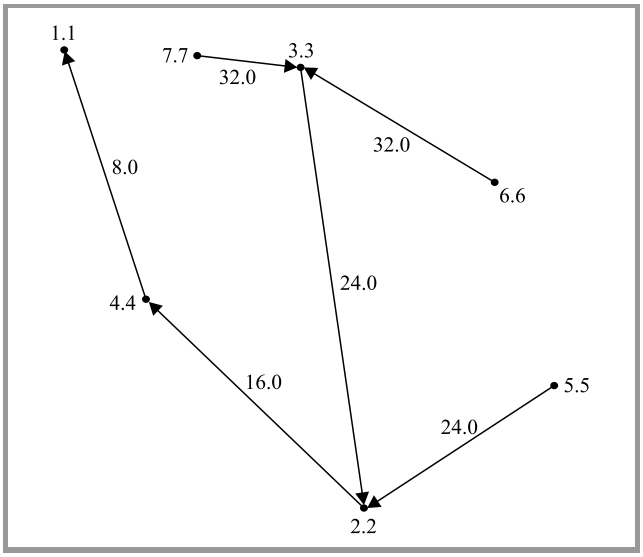

Fig. 3. Scenario 1 (normal operation): network topology.

needs to make to the destination in order to successfully deliver a message. For instance, the ETX value of node 4 next to the server (node 1) is 8. In Fig. 3 we also note that messages of node 7 must be transmitted via nodes 3, 2 and 4 to get to the server. Note that node 7 is not malicious and runs the same code as all other nodes in this scenario. In Fig. 4 the power consumption of each node is shown. Measurements were collected using the PowerTracker tool in Cooja. As expected, all nodes are almost always on (99.87% of the time, on average) and have very low values of radio TX and radio RX. This is normal for small-sized networks.

In the second scenario, nodes use the same RDC and MAC driver configuration as before. Node 7 has, however, been modified to transmit 80 DIS messages and to increase the DODAG version number before transmitting the received

Table 1
Types and configuration of nodes

Node type	Description	Radio Duty Cycle (RDC)	Medium Access Control (MAC)	RPL conf. (DIS interval)
Server	Receives messages without doing any processing or sending acknowledgments	NullRDC	CSMA	—
Benign node	Creates a mesh network by using RPL protocol and sends data to server	ContikiMAC or NullRDC	CSMA	60 s
Malicious node	Uses RPL protocol to broadcast DIS control messages to neighbors (flooding attack) and modify version number	ContikiMAC or NullRDC	CSMA	Every second

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	99.91%	0.01%	0.06%
Sky 2	99.79%	0.05%	0.11%
Sky 3	99.92%	0.03%	0.10%
Sky 4	99.86%	0.06%	0.06%
Sky 5	99.91%	0.02%	0.11%
Sky 6	99.79%	0.02%	0.07%
Sky 7	99.88%	0.02%	0.07%
AVERAGE	99.87%	0.03%	0.08%

Fig. 4. Scenario 1 (normal operation): power consumption measurements.

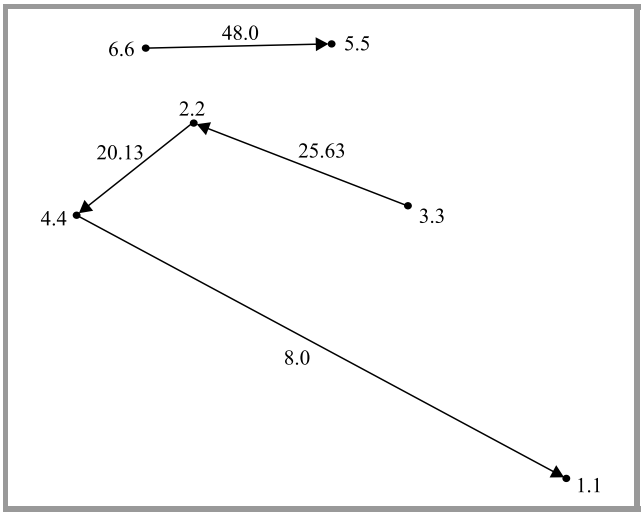

Fig. 5. Scenario 2 (attack): network topology.

DIO messages to the server. Changing the version number leads to global repair and to the formation of two different DODAGs. Every few minutes, global repair is triggered. As a result, the routes change quickly. The topology of the network is therefore not stable and some nodes may be disconnected from the server or from other nodes. There is one such situation Fig. 5, where at that particular moment nodes 5 and 6 do not have a route to the server. The impact of the attack is demonstrated in Fig. 6, which shows the measurements of power consumption. In adjacent nodes 3, 5, and 6, the attack caused high radio RX, and high radio TX in node 7. As a result, both malicious/compromised and neighboring nodes are depleted of energy.

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	99.88%	0.04%	0.12%
Sky 2	99.89%	0.11%	0.36%
Sky 3	99.80%	0.12%	2.12%
Sky 4	99.92%	0.14%	0.14%
Sky 5	99.80%	0.15%	2.09%
Sky 6	99.83%	0.09%	2.04%
Sky 7	99.91%	1.82%	0.32%
AVERAGE	99.86%	0.35%	1.03%

Fig. 6. Scenario 2 (attack): power consumption measurements.

In the previous scenarios, nodes used the same RDC and MAC drivers. However, using a different RDC driver may produce different results. In ContikiOS, ContikiMAC is another option for RDC driver. For this reason, the two scenarios were repeated using the ContikiMAC RDC driver in benign and malicious nodes, while keeping the same MAC driver. Starting with the normal scenario, the nodes' power consumption is shown in Fig. 7. As expected, ContikiMAC enables sleep mode and this is clearly shown by the very low percentage of radio on for all nodes except for the server.

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	99.82%	0.11%	0.18%
Sky 2	1.93%	0.87%	0.05%
Sky 3	1.94%	0.87%	0.06%
Sky 4	1.16%	0.24%	0.06%
Sky 5	1.25%	0.35%	0.07%
Sky 6	1.20%	0.30%	0.04%
Sky 7	1.29%	0.39%	0.06%
AVERAGE	15.34%	0.45%	0.07%

Fig. 7. Scenario 1 using ContikiMAC: power consumption measurements.

In addition, radio TX is 0.45%, on average, which means that nodes sleep most of the time and send very few packets within the network. Average radio RX is even lower than radio TX, because it is the server that is the destination of the majority of packets. The corresponding network topology is shown in Fig. 8. It may be noticed that all nodes communicate with the server by using their next hop.

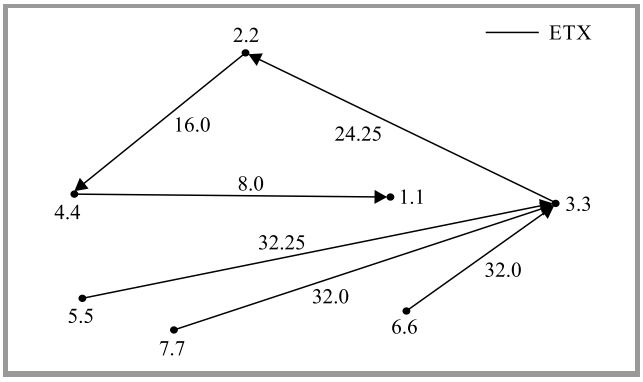

Fig. 8. Scenario 1 using ContikiMAC: network topology.

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	100.00%	0.07%	0.24%
Sky 2	3.78%	2.08%	0.19%
Sky 3	59.37%	2.20%	35.70%
Sky 4	4.28%	2.67%	0.03%
Sky 5	59.60%	1.99%	36.06%
Sky 6	60.17%	2.42%	36.08%
Sky 7	63.98%	47.33%	0.94%
AVERAGE	50.17%	8.39%	15.61%

Fig. 9. Scenario 2 using ContikiMAC: power consumption measurements.

Using the same node configuration, the second scenario with a malicious node was repeated. In this case, the power consumption of nodes is affected by the malicious node, as shown in Fig. 9. Although nodes should be sleeping most of the time, they are ON for 50% of the time. This is true for the server as well. The difference compared with the nor-

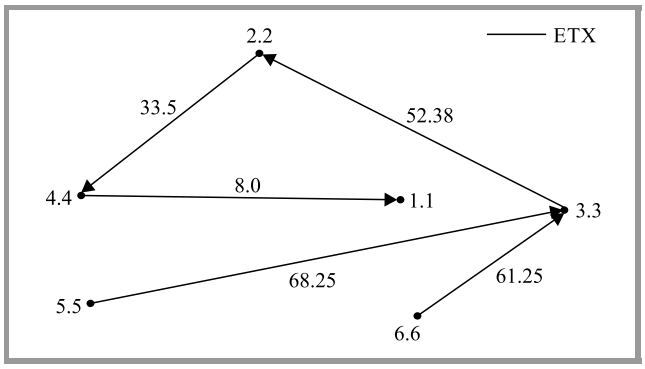

Fig. 10. Scenario 2 using ContikiMAC: network topology.

mal scenario is about 35%, which is significant. The reason for this behavior is caused by the fact that the malicious node 7 broadcasts DIS messages requiring a DIO reply from its neighbors. This is also the reason why nodes 3, 5 and 6 have the highest radio RX percentage in comparison with other nodes. These nodes are closer to node 7 and are more affected than other nodes. Furthermore, node 7 transmits all the time and has, therefore, the highest percentage of radio TX. Looking at the network topology in Fig. 10, one may see that some links have unusually high ETX values. The reason for that are the global repairs initiated, which assign different ETX values to links. Nodes located near the malicious node have a worse ETX value in their links in comparison with other nodes. Moreover, the malicious node is not shown in the presented network topology because it never joins the DODAG and, therefore, no information is sent to the server.

6. Proposed IDS

6.1. Architecture and Components

In addition to the typical sensor nodes, two new types of devices are considered: i) IDS routers for the handling of both the detection module and the firewall, and ii) sensor-like devices, called IDS detectors, for the monitoring and transmission of suspicious traffic to the router. In a typical scenario of a small IoT network, one IDS router acts as the border router (BR) of the network, while several IDS detectors are deployed near the nodes. This scenario is demonstrated in Fig. 11. This means that devices that need to communicate with an external server send all requests via the IDS router. The router analyses all passing traffic, and determines whether or not the sending node is malicious.

IDS detectors monitor packets to help detect malicious nodes. Malicious devices may try to interrupt normal network operation internally without having to communicate with the router or external networks. In such cases, detectors log packets and if the behavior of a node corresponds to a known attack, relevant information is sent to the IDS router.

In the scenario shown in Fig. 11, we have 5 Tmote Sky sensors and an IDS consisting of one router and 2 detectors.

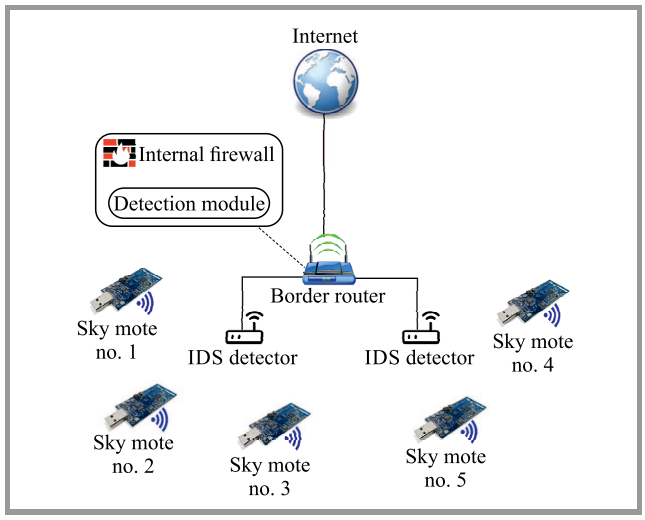

Fig. 11. High-level IDS architecture

The router is Internet connected and has two components: a detection module and a firewall. These components contribute to the internal and external protection of the network. The detection module executes specific algorithms to determine if malicious nodes exist or not in the network, while the firewall generates and enforces rules for stopping malicious traffic. The detectors are wired to the router in order to prevent interference or eavesdropping via a wireless channel. If the communication of the router and the detectors needs to be wireless, a proper secure wireless communication scheme will be used (e.g. [32]). IDS detectors capture any traffic exchanged among nearby sensors. Afterwards, a decision whether traffic should be forwarded to the router or not is taken based on a lightweight algorithm. We assume that detectors are resource-constrained. Algorithms that need heavy calculations or large memory are therefore not appropriate.

Collaboration between the router and the detectors helps monitor traffic from both internal and external interfaces. Some malicious devices, for example, may attempt to communicate with a remote & control server to download malicious files or instructions [33]. Other devices that are compromised may exchange traffic locally. The presented design takes all types of communications into account in order to block malicious nodes. The router captures Wi-Fi and IEEE 802.15.4 traffic. It is also capable of detecting attacks from ZigBee and IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) devices [3].

The proposed IDS stores malicious patterns in the router's detection module in the same way as other signature-based solutions. The router connects the internal network to the Internet and is assumed to have sufficient computing power to execute algorithms required to detect various attack types.

6.2. Mitigating Attacks

As mentioned earlier, the goal of the proposed IDS is to detect and prevent a large number of different types of

attacks, for instance DoS attacks that may occur in IoT networks to exhaust sensor node batteries. Moreover, routing attacks usually exploit RPL which is a routing protocol that many smart IoT network sensors currently use. Selective forwarding, sinkhole attack and clone ID are some of other well-known routing attacks [23], [34], [35].

The previous attacks can be detected using various techniques. Measuring the packet dropping rate, packet sending rate, the received signal strength (RSS), packet interval, and monitoring the number of node IDs in the network are some of the mitigation methods [36]. Specifically, the packet sending rate could be a good metric for detecting malicious nodes, as smart devices do not exchange many packets. A device that behaves abnormally and sends too many packets could be considered a malicious one.

Another useful metric is the packet interval. Each device is configured to communicate with other in specific time intervals. Malicious devices could exploit this feature and send more requests in small intervals. Our IDS can detect this behavior by taking into account the time intervals of all nodes in the network and by calculating the average time interval which will be considered normal behavior. Thus, any node exceeding the normal packet interval threshold will be considered as malicious. A threshold-based detection is a lightweight mechanism that allows IDS detectors to perform fast calculations and detect compromised nodes. According to [6], these well-known attacks may have a significant impact on the availability, as well as on the integrity of IoT systems.

Regarding the scalability of the proposed IDS, it is expected to have good efficiency even in large networks. To ensure that, only the suspicious traffic will be forwarded from detectors to the router. This means that detectors will perform some specific computations (e.g. packet sending rate and packet interval) and will forward the node's traffic to the router for further investigation (e.g. signature matching) only if the metric of interest is above the threshold value. Apart from that, the router will have an overall picture of the network and will block suspicious nodes.

6.3. Detection Module and Firewall

As mentioned before, the detection module at the router plays an important role in the proposed IDS. This component determines if a node is malicious or not. Decision will be taken based on the information collected for each individual device. For example, a device sending a large number of packets with a high rate or a node with RSS value above the threshold value may be regarded as malicious. As a consequence, that device may be removed from the network, its IP address will be blacklisted, a proper firewall rule will be generated and the network administrator will be notified. However, complex attacks, such as selective forwarding, are not easy to detect and may need more time to identify. For this reason, the detection module will store signatures of known IoT malware. Packets matching a stored malicious signature will be blocked and the source and destination nodes will be blacklisted.

The firewall at the router is added as an extra layer of protection. The IP addresses of malicious nodes will be blocked if they match any stored firewall rules. The detection module will proceed to banning a node from the network only if it has information of its malicious behavior. In this case, a new firewall rule will be created including the IP of the node, and traffic between the node and the Internet will be stopped.

As regards the strategy of placing IDS modules, two methods are usually distinguished: network-based and host-based [36]. In the network-based method, the agent is placed near the base station, so that it can monitor the traffic sent by the devices, creating an additional communication overhead. In the host-based method, the agent is embedded in all nodes, consuming a significant portion of the node's resources and energy. In this work, a hybrid approach that combines both methods, has been followed. A centralized node (i.e. the router) keeps signatures, analyzes traffic and detects sensor or Internet attacks. Some decentralized nodes (i.e. detectors) execute lightweight tasks, such as monitoring and sending suspicious packets to the router. The advantage of this placement strategy is that traffic can be captured and attacks can be detected from all network segments. Furthermore, deploying detectors near sensors helps detect attack attempts faster and more efficiently, instead of waiting for the malicious packets to pass via the router.

7. Conclusion and Future Work

In this paper, we studied the impact of battery drain DoS attacks on IoT devices. Cooja simulator is the platform chosen for implementing IoT-based attacks. It supports application development for ContikiOS. The demonstrated scenarios include attack scenarios where a compromised sensor performs DoS attacks based on "Hello" flooding and version number modifications. As demonstrated, the attack may negatively affect the energy consumption of IoT devices.

We also proposed a new IDS for securing IoT networks and devices. The proposed IDS follows the hybrid placement approach for effective detection of intrusions originating both from external and internal networks. Some of the advantages of the proposed IDS are: i) no firmware modification of IoT devices is needed, ii) the detectors are wired to avoid jamming and other wireless attacks, iii) support for generic IDS modules, and iv) support for heterogeneous devices (e.g. ZigBee and 6LoWPAN).

As a future work, we plan to simulate attacks on larger networks of different topologies. In addition, the "Hello" flooding attack will be further studied using different temporal and spatial distributions of RPL messages. Apart from that, we aim to develop and test the proposed IDS in Cooja. IDS performance will be evaluated by simulating attack scenarios and obtaining useful metrics, such as detection rate and false positives rate. The IDS will be tested for DoS, routing and other types of attacks. Finally,

to test their performance in a real world IoT environment, IDS modules will be imported into ContikiOS.

References

[1] V. G. Vassilakis, I. D. Moscholios, J. S. Vardakas, and M. D. Logothetis, "On the digital certificate management in advanced metering infrastructure networks", in *Proc. IEICE Inform. and Commun. Technol. Forum ICTF*, Poznań, Poland, 2017.

[2] B. A. Alohali and V. G. Vassilakis, "Secure and energy-efficient multicast routing in smart grids", in *Proc. 10th IEEE Int. Conf. on Intell. Sensors, Sensor Netw. and Inform. Process. ISSNIP*, Singapore, 2015 (doi: 10.1109/ISSNIP.2015.7106929).

[3] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, IETF, March 2012.

[4] Contiki: The Open Source OS for the Internet of Things [Online]. Available: http://www.contiki-os.org/ (accessed: 2019.01.14).

[5] TinyOS: An OS for Embedded, Wireless Devices [Online]. Available: https://github.com/tinyos/tinyos-main (accessed: 2019.01.14).

[6] Gemalto. The State of Internet of Things Security [Online]. Available: http://www2.gemalto.com/iot/index.html (accessed: 2019.01.14).

[7] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks", in *Proc. 3rd Int. Conf. on Elec. Design ICED*, Phuket, Thailand, 2016, pp. 321–326 (doi: 10.1109/ICED.2016.7804660).

[8] Symantec Security Response, Mirai: What you need to know about the botnet behind recent major DDoS attacks, Oct. 2016 [Online]. Available: https://www.symantec.com/connect/blogs/mirai-what-you-need-know-about-botnet-behind-recent-major-ddos-attacks

[9] T. Easton, "Chalubo botnet wants to DDoS from your server or IoT device", Oct. 2018 [Online]. Available: https://news.sophos.com/en-us/2018/10/22/chalubo-botnet-wants-to-ddos-from-your-server-or-iot-device

[10] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks", in *5th IEEE Int. Conf. on Cyber Secur. and Cloud Comput. CSCloud and 4th IEEE Int. Conf. on Edge Comput. and Scalable Cloud EdgeCom*, Shanghai, China, 2018, pp. 12–17 (doi: 10.1109/CSCloud/EdgeCom.2018.00012).

[11] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6LoWPAN based Internet of Things", in *Proc. 9th IEEE Int. Conf. on Wirel. and Mobile Comput., Netw. and Commun. WiMob*, Lyon, France, 2013, pp. 600–607 (doi: 10.1109/WiMOB.2013.6673419).

[12] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of Things", *Int. J. of Distrib. Sensor Netw.*, vol. 9, no. 8, pp. 1–11, 2013 (doi: 10.1155/2013/794326).

[13] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in RPL-based Internet of Things", *Int. J. of Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016 (doi: 10.6633/IJNS.201605.18(3).07).

[14] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey", *IEEE Commun. Surveys & Tutor.*, vol. 19, no. 4, pp. 2502–2525, 2017 (doi: 10.1109/COMST.2017.2751617).

[15] P. P. Ioulianou, V. G. Vassilakis, I. D. Moscholios, and M. D. Logothetis, "A signature-based intrusion detection system for the Internet of Things", in *Proc. IEICE Inform. and Commun. Technol. Forum ICTF*, Graz, Austria, 2018.

[16] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL", in *Proc. 9th ACM/IEEE Int. Conf. on Inform. Process. in Sensor Netw.*, Stockholm, Sweden, 2010, pp. 406–407 (doi: 10.1145/1791212.1791277).

[17] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-version number and rank authentication in RPL", in *Proc. IEEE 8th Int. Conf. on Mob. Ad-hoc and Sensor Syst. MASS 2011*, Valencia, Spain, 2011, pp. 709–714 (doi: 10.1109/MASS.2011.76).

[18] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A study of RPL DODAG version attacks", in *Proc. IFIP Int. Conf. on Autonomous Infrastruc., Manag. and Secur.*, Brno, Czech Republic, 2014, pp. 92–104 (doi: 10.1007/978-3-662-43862-6_12).

[19] F. Osterlind *et al.*, "Cross-level sensor network simulation with Cooja", in *Proc. 31st IEEE Int. Conf. on Local Comp. Netw.*, Tampa, FL, USA, 2006, pp. 641–648 (doi: 10.1109/LCN.2006.322172).

[20] "The Internet of Things Reference Model", Cisco, 2014 [Online]. Available: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf

[21] Y. Yang *et al.*, "A survey on security and privacy issues in Internet-of-Things", *IEEE Internet of Things J.*, vol. 4. no. 5, pp. 1250–1258, 2017 (doi: 10.1109/JIOT.2017.2694844).

[22] F. Ayotunde Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey", *J. of Network and Comp. Appl.*, vol. 88, pp. 10–28, 2017 (doi: 10.1016/j.jnca.2017.04.002).

[23] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things", *J. of Network and Comp. Appl.*, vol. 84, pp. 25–37, 2017 (doi: 10.1016/j.jnca.2017.02.009).

[24] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino. "Kalis – a system for knowledge-driven adaptable intrusion detection for the Internet of Things", in *Proc. IEEE 37th Int. Conf. on Distrib. Comput. Syst. ICDCS 2017*, Atlanta, GA, USA, 2017, pp. 656–666 (doi: 10.1109/ICDCS.2017.104).

[25] OpenWRT: a Linux OS for Embedded Devices [Online]. Available: https://openwrt.org (accessed: 2019.01.14).

[26] M. Roesch *et al.*, "Snort: Lightweight intrusion detection for networks", in *Proc. of the 13th USENIX Conf. on System Admin. LISA'99*, Seattle, WA, USA, 1999, vol. 99, pp. 229–238.

[27] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: real-time intrusion detection in the Internet of Things", *Ad Hoc Netw.*, vol. 11. no. 8, pp. 2661–2674, 2013 (doi: 10.1016/j.adhoc.2013.04.014).

[28] T. Matsunaga, K. Toyoda, and I. Sasase, "Low false alarm rate RPL network monitoring system by considering timing inconstancy between the rank measurements", in *Proc. 11th Int. Symp. on Wirel. Commun. Syst. ISWCS 2014*, Barcelona, Spain, 2014, pp. 427–431 (doi: 10.1109/ISWCS.2014.6933391).

[29] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki – a lightweight and flexible operating system for tiny networked sensors", in *Proc. 29th IEEE Int. Conf. on Local Comp. Netw.*, Tampa, FL, USA, 2004, pp. 455–462 (doi: 10.1109/LCN.2004.38).

[30] E. Baccelli, M. Philipp, and M. Goyal, "The P2P-RPL routing protocol for IPv6 sensor networks: Testbed experiments", in *Proc. 19th Int. Conf. on Software, Telecommun. and Comp. Netw. SoftCOM 2011*, Split, Croatia, 2011, pp. 656–666 [Online]. Available: https://hal.archives-ouvertes.fr/hal-00651603/document

[31] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research", in *Proc. 4th Int. Symp. on Inform. Process. in Sensor Netw.*, Boise, ID, USA, 2005, pp. 364–369 (doi: 10.1109/IPSN.2005.1440950).

[32] B. A. Alohali, V. G. Vassilakis, I. D. Moscholios, and M. D. Logothetis, "A secure scheme for group communication of wireless IoT devices", in *Proc. 11th IEEE/IET Int. Symp. on Commun. Syst., Netw., and Digit. Sig. Process. CSNDSP 2018*, Budapest, Hungary, 2018 (doi: 10.1109/CSNDSP.2018.8471871).

[33] S. Khattak, N. R. Ramay, K. R. Khan, A. A Syed, and S. Ali Khayam, "A taxonomy of botnet behavior, detection, and defense", *IEEE Commun. Surveys & Tutor.*, vol. 16, no. 2, pp. 898–924, 2014 (doi: 10.1109/SURV.2013.091213.00134).

[34] P. Pongle and G. Chavan, "A survey: attacks on RPL and 6LoWPAN in IoT", in *Proc. Int. Conf. on Pervasive Comput. ICPC 2015*, Pune, India, 2015 (doi: 10.1109/PERVASIVE.2015.7087034).

[35] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO suppression attack against routing in the Internet of Things", *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2524–2527, 2017 (doi: 10.1109/LCOMM.2017.2738629).

[36] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-service attacks on 6LoWPAN-RPL networks: threats and an intrusion detection system proposition", *J. of Adv. Comp. Sci. & Technol.*, vol. 3, no. 2, pp. 143–153, 2014 (doi: 10.14419/jacst.v3i2.3321).

Philokypros P. Ioulianou received his B.Sc. degree in Computer Science from the University of Cyprus in 2016, and M.Sc. in Advanced Computer Science with specialization in Computer Security from the University of Manchester in 2017. He is currently a Ph.D. student at the University of York, UK. His research interests are in the area of computer and network security, IoT (Internet of Things) and wireless sensor security.

https://orcid.org/0000-0001-7436-4470
E-mail: pi533@york.ac.uk
Department of Computer Science
University of York
York, United Kingdom

Vassilios G. Vassilakis received his Ph.D. degree in Electrical and Computer Engineering from the University of Patras, Greece in 2011. He is currently a lecturer in Cyber Security at the University of York, UK. He's been involved in EU, UK, and industry funded R&D projects related to the design and analysis of future mobile networks and Internet technologies. His main research interests are in the areas of network security, Internet of Things, next-generation wireless and mobile networks, and software-defined networks. He is published over 90 journal/conference papers. He is served as an Associate Editor in IEICE Transactions on Communications, IET Networks, and Elsevier Optical Switching & Networking.

https://orcid.org/0000-0003-4902-8226
E-mail: vv573@york.ac.uk
Department of Computer Science
University of York
York, United Kingdom

Michael D. Logothetis received his Dipl. Eng. degree and Doctorate in Electrical Engineering, both from the University of Patras, Patras, Greece, in 1981 and 1990 respectively. From 1991 to 1992 he was Research Associate in NTT's Telecommunication Networks Laboratories, Tokyo, Japan. In 2009 elected (Full) Professor in the ECE Department of the University of Patras. His research interests include teletraffic theory, simulation and performance optimization of telecommunications networks. He has published over 200 conference/journal papers. He has become a Guest Editor in: Mediterranean Journal of Electronics and Communications, Mediterranean Journal of Computers and Networks, IET Circuits, Devices & Systems, IET Networks and Ubiquitous Computing and Communication Journal. He is a member of the IARIA (Fellow), IEEE (Senior), IEICE (Senior), FITCE and the Technical Chamber of Greece (TEE).

https://orcid.org/0000-0001-6315-5382
E-mail: mlogo@upatras.gr
Department of Electrical & Computer Engineering
University of Patras
Patras, Greece