A Novel Framework for Collaborative Intrusion Detection for M2M Networks

Junaid Arshad*, Mohammad Mahmoud Abdellatif†, Muhammad Mubashir Khan‡ and Muhammad Ajmal Azad§
* School of Computing and Engineering, University of West London, United Kingdom
† Faculty of Engineering, The British University in Egypt, Egypt
‡ Dept. of Computer Science & IT, NED University of Engineering and Technology, Pakistan
§ School of Computing Science, Newcastle University, United Kingdom
Email: junaid.arshad@uwl.ac.uk, mohammad.abdellatif@bue.edu.eg, mmkhan@neduet.edu.pk, muhammad.azad@ncl.ac.uk

Abstract—**The proliferation of sensor devices has introduced exciting possibilities such as the Internet of Things (IoT). Machine to Machine (M2M) communication underpins efficient interactions within such infrastructures. The resource constraints and ad-hoc nature of these networks have significant implications for security in general and with respect to intrusion detection in particular. Consequently, contemporary solutions mandating a stable infrastructure are inadequate to fulfill these defining characteristics of M2M networks. In this paper, we present COLIDE (COLlaborative Intrusion Detection Engine) a novel framework for effective intrusion detection in the M2M networks without incurring high energy and communication cost on the participating host and edge nodes. The framework is envisioned to address challenges such as flexibility, resource constraints, and the collaborative nature of the M2M networks. The paper presents a detailed system description along with its formal and empirical evaluation using Contiki OS. Our evaluation for different communication scenarios demonstrates that the proposed approach has limited overhead in terms of energy utilization and memory consumption.**

I. Introduction

The use of sensor devices has increased dramatically over the last few years leading to their proliferation across diverse domains such as wearables, intelligent appliances, and vehicles. As these devices have the ability to be connected to a network, it introduces exciting possibilities such as the Internet of Things (IoT). IoT has received significant attention as a disruptive technology and is considered fundamental to the networks of the future. A recent study from Gartner predicted the number of sensor devices to increase to 8.4 billion in 2017 [1]. This has direct impact on industrial applications such as automotive industry, commercial security cameras, as well as consumer applications such as wearables, smart TVs, and smart meters.

A typical IoT network consists of devices with resource constraints such as limited processing power, energy resources, and communication range. In view of these constraints, 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) [2], [3],[4], allows the resource constrained sensor devices to send and receive information as IPv6 packets. Therefore, it facilitates communication between low power wireless personal area networks using IPv6 by performing header compression and fragmentations to fit the large sized IPv6 packets in the smaller link layer frames such as those defined by IEEE 802.15.4 [5]. This enables the *things* to use the IP based Internet, leveraging standards and technologies developed over the last few decades. For a typical LoWPAN, this connectivity is achieved by using an edge router which facilitates connectivity among the devices participating within a LoWPAN as well as with the Internet. The focus of our research is to investigate and address novel challenges for security of M2M networks which we believe are two-fold: firstly, these devices are typically resource constrained thereby limiting their ability to host sophisticated security system. Secondly, the ad-hoc nature of 6LoWPAN network allows devices to connect to other devices at runtime typically for short time periods therefore creating a volatile infrastructure. However, current approaches to intrusion detection for 6LoWPAN networks are generally focused at standalone intrusion detection components integrated within the host or the cluster head. These approaches are limited in that they consider a restricted view of the events within an M2M network and therefore are limited in their ability to address complex, multi-stage attacks. We believe that due to the adhoc nature of such systems, a collaborative intrusion detection approach will enable the edge routers to use collective information from various devices to have rigorous view of the characteristics of events visible to them.

Our contribution presented via this paper is COLIDE - a collaborative intrusion detection framework for M2M based IoT networks, that leverages collaboration among IoT nodes for effective intrusion detection without incurring high communication, processing and energy resources. To this end, the framework envisages collective use of the information from host and network based detection systems. We believe correlating the events from multiple devices can facilitate minimizing the false positive rate, improve the detection rate under distributed attacks and also minimize the workload for the end host. The proposed framework is envisioned to address challenges such as the flexibility, resource constraints of the nodes, and the collaborative nature of the M2M networks.

The rest of the paper is structured as follows. Section II describes the related work regarding intrusion detection in an IoT network. Section III presents our collaborative approach and Z-notations for intrusion detection in an IoT network. A detailed description of the experimentation setup is presented in section IV with thorough evaluation presented in section V. Section VI presents an overall discussion about the effectiveness of the scheme followed by conclusions of the paper in section VII.

II. Related work

Intrusion detection within IoT systems has received significant attention over the last few years. For instance, in [6], Raza et al. presented an intrusion detection system for IoT taking into consideration the unique network elements of IoT i.e. network protocols developed for the constrained devices including RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [7] and 6LowPAN [3],[4]. They proposed a hybrid intrusion detection architecture which included lightweight elements within the sensor nodes along with a central node at the 6LOWPAN Border Router (6BR) node. The work presented is different from COLIDE framework in that COLIDE only focuses on the generic IoT system which can include devices of any types (constrained or unconstrained) whereas [6] have specifically designed for the constrained devices. Furthermore, our proposed system is also able to function within an untrustworthy and flexible environment where different devices can form ad-hoc networks without previous handshakes to deliver a certain service in a coordinated manner.

In [8], Sheikhan and Bostani presented an intrusion detection similar to the [6] such as; both focus on using the network traffic of the devices for intrusion detection purposes, consider the resource-constrained 6LowPAN devices based system, and finally, both approaches are focused on the sinkhole and selective-forwarding attacks. However, the proposed COLIDE framework is capable of working with diverse devices and a range of issues including different types of attacks, inherent flexibility of the IoT networks, and the lack of trust among the participant devices. In [9], Kasinathan et al. developed an architecture to protect against Denial of Service (DoS) attacks within 6LowPAN networks. In [10] Jun and Chi correlated the large number of alerts and network traffic as a part of intrusion detection. Le et al. [11] investigated the security aspects of Routing Protocol for Low-power and lossy network (RPL). This work is similar to the COLIDE framework in that it also uses a network of monitoring nodes, which are used to sniff and monitor communications within their neighborhood. Obaid et al. [12] presented one of the early efforts to develop an intrusion detection system for IP-based wireless sensor networks. The authors presented RIDES which is a hybrid IDS combining both anomaly and misuse based intrusion detection approaches. Although our approach has similarities with [12] in that it also uses both signature and anomaly based detection systems, however [12] proposed to use both detection engines at node and edge router level. We believe this has significant performance overheads especially at node level due to the limited resources available. On the contrary, we propose using signature based intrusion detection system at the node level and the anomaly based IDS at the edge router thereby significantly reducing the performance overhead.

In [13], Abduvaliyev et al. presented an effort to develop an energy efficient intrusion detection system for wireless sensor networks. The proposed system is a combination of anomaly and misuse based intrusion detection aimed at protecting the cluster heads which the authors believe is the first target for any attack on WSN. In [14] Wenchao et al. used K-Nearest Neighbor (KNN) machine learning algorithm to identify malicious nodes in the wireless sensor networks. Saeed et al. [15] proposed intrusion detection and prevention system using random neural networks. However, modelling the behaviour patterns of device usage and processing it via multi-stage neural networks could have a high-energy consumption. The confidentiality and integrity of data exchanged between IoT devices and the core system can be protected through the use of cryptography mechanism [16], [17]. However, the energy resources required for encrypting the data make these approaches infeasible to be deployed in a real system implementation.

In [18], Zhou et al. proposed a decentralized multi-dimensional alert correlation system for the collaborative intrusion detection. The system consist of two algorithms implemented in a fully distributed CIDS, first algorithm clusters alerts locally at each device, before reporting significant alert patterns to a global correlation stage. The authors in [19] proposed a self-adapting, knowledge-driven IDS for IoT network running different communication protocols. Sedjelmaci et al. proposed an efficient and lightweight intrusion detection mechanism for securing the vehicular network in [20]. The approaches utilizes the rules-based intrusion detection to identify different type of attacks. In [21], Alessandro et al. proposed an IDS architecture for the IoT that uses the Raspberry Pi equipped with Snort intrusion detection system. The authors in [22] proposed intrusion detection system for the visual sensor networks based on traffic pattern matching and then a hierarchical self-organizing map (HSOM) is employed to learn traffic patterns and detect intrusions.

In summary, COLIDE makes contribution to the existing knowledge within intrusion detection for IoT networks with respect to working with diverse devices and a range of issues including different types of attacks, limited node-level resources and the inherent flexibility of the IoT networks.

III. COLIDE: A collaborative intrusion detection framework for M2M

COLIDE is a collaborative intrusion detection system for IoT infrastructures which takes into account resource constraints, flexibility and diversity of devices and emphasizes cooperative nature of such systems. A graphical representation of the COLIDE framework is presented in Fig 1 which presents its different components and interactions between them.

As presented in Fig 1, intrusion detection is performed at two levels, the node level and the edge router level. These are described below in more detail.

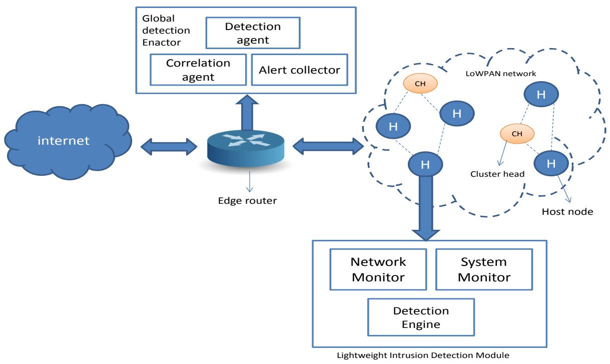

Fig. 1 Building block of collaborative intrusion detection system for IoT.

A. Node level detection

The node level detection module is envisaged to be a lightweight module present at all end-hosts (sensor nodes). This module is envisioned to take advantage of the unique visibility offered by a node level system to improve the overall intrusion detection. In this regard, we propose using signature based intrusion detection within nodes due to its efficiency with respect to consumption of computational resources as compared with anomaly based detection. Furthermore, the choice of implementation for network vs host based monitoring at node level is rendered application specific as it will influence the types of attack that can be protected against. For instance, a Denial of Service (DoS) attack targeted to flood specific nodes within a LoWPAN can be detected by monitoring network traffic whereas a backdoor channel attack targeted at gaining unauthorized access to a node can be detected by monitoring system events. The detection engine for the node level module therefore processes the information generated by the monitor(s) using existing signatures to detect any attack attempts.

In order to aid formal representation, let us define an event within a given host H as E_{hi}. As the proposed system is flexible in terms of implementation of the host based component, this event can represent a system event such as a system call or a network event such as a network packet. In both scenarios, E_{hi} will be composed of a number of parameters which will be important to decide if an event is malicious or non-malicious. For instance, for a network packet, these parameters can include protocol, inter-arrival time, packet size etc. Therefore, E_{hi} can be represented as:

$$E_{hi} : \{pr_1, pr_2, pr_3, \cdots pr_n\}$$

where pr_n is a specific parameter for an event. Within the context of the above scenario, the Detection Engine DE is expected to categorize E_{hi} as malicious or non-malicious. The intrusion detection policy P_h is used to contribute towards this decision. Therefore, if SE_{hi} represents the state of an event E_{hi}, the following can be represented as the intrusion detection function.

$$SE_{hi} : DE(E_{hi}, P_h)$$

We define the following data models to be used throughout our formal description.

EH: a set of events for the host H
SE: state of an event; it can be malicious or non-malicious
SEH: state of an event for host H;
PH: Detection policy for host H
HOSTS: a set of hosts within a LoWPAN

```
[Node Level Intrusion Detection]
$\Delta Host_{ID}$
$EH_{ID}? : \mathbb{N}$
$EH_i? : EH$
$PH_i? : PH$
$SEH_i? : SE$
----------------
$EH_{ID} > 0$
$EH \neq <>$
$PH \neq <>$
$SEH \neq <>$
    if $EH_i \in EH$ then
        $SEH_i = DE(EH_i, PH_i)$
        if $SEH_i = Malicious$ then
            Intrusion Response$(EH_i, SEH_i)$
```

B. Edge router detection

An edge router is an important component within IoT systems as it enables connectivity between the LoWPAN(s) and the Internet. We envisage to leverage the enhanced computational capabilities of edge routers to achieve rigorous intrusion detection for IoT systems. In particular, the edge router detection module is envisaged to monitor traffic for LoWPAN(s) attached to it thereby monitoring traffic for all the devices within a LoWPAN. Among others, this enables detection of attacks affecting multiple devices within a LoWPAN due to the level of visibility offered by the edge router.

Within the proposed system, the edge router detection module has three components: Alert Collector, Correlation Agent, and Detection Agent. As the edge router is expected to monitor all the devices within a LoWPAN, a method is required to identify threats/alerts for individual sensors. Alert Collector is to achieve this function by communicating with individual IoT devices to gather alerts from the node level monitoring components. A typical intrusion is usually not an isolated event that can be achieved within a single transaction or network event but it is usually a series of steps each of which may target a specific vulnerability with the aim to achieve the overall successful intrusion. Correlation Agent component is made to facilitate countermeasures for such attacks by correlating malicious events at network and system levels as monitored by the node level monitors. This enables improved visibility into the events within IoT devices and facilitates the overall intrusion detection process. Historically, anomaly based intrusion detection approaches have demonstrated better efficiency especially with respect to detection of complex, multistage, and zero day attacks however at the cost of increased resource consumption. Due to the increased capability of edge router devices, we propose implementing anomaly based intrusion detection at the edge router. The Detection Engine at the edge router is envisioned to achieve this by making use of the alerts collected and correlated by the Alert Collection and alert Correlation components. The Global Detection Enactor (GDE) has three subcomponents i.e. Detection Agent, Correlation Agent (CA), and Alert Collector (AC). As the GDE is responsible for a LoWPAN, it is expected to monitor hosts within that LoWPAN. Within the context of multi-stage attack detection, we envisage DA to have a profound role in assessing the correlation between independent events and envisage a scheme inspired by [23]. The details of these components are envisioned to be included in the extended version of this paper.

IV. Experimental Setup

The implementation for the COLIDE framework was achieved in Contiki OS, the operating system for IoT used widely in research and industry [24]. The evaluation was performed using Contiki v2.7 and its built-in emulator Cooja [25].

A. *IoT environment*

The proposed IoT system is presented in Fig 2. It consists of a Border Router (BR) that acts as the DODAG root for the 6LoWPAN network and connects it to the Internet through a SLIP interface to a computer. This computing unit has higher processing power than the IoT devices.

The first tier of nodes that are a part of the 6LoWPAN are referred to as routers/IDS nodes and can forward messages to the root as well as sending periodic informaThe first tier of nodes that are a part of the 6LoWPAN are referred to as routers/IDS nodes and can forward messages to the root as well as sending periodic information regarding the behavior of other nodes under them. Information such as source and destination IP/Port of the messages being sent. The IDS nodes are always one hop away from the root.

tion regarding the behavior of other nodes under them. Information such as source and destination IP/Port of the messages being sent. The IDS nodes are always one hop away from the root.

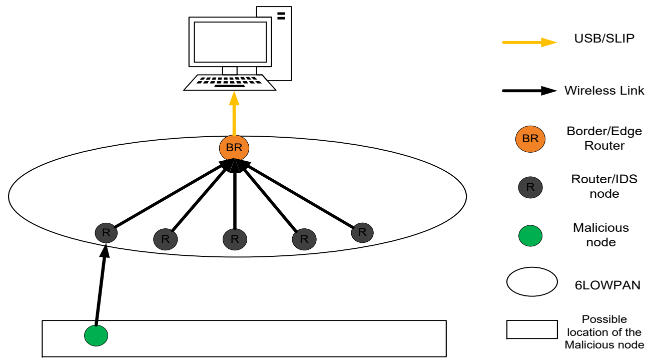

Fig. 2: Experimental Setup

The malicious nodes are located in the second tier and can only join the network through one of the router nodes in the first tier. Its location can vary but for the sake of simplicity and to retain focus, we demonstrate experimentation with only one malicious node that will always be two hops away from the root. The experimentation within this setup have been performed with Cooja using Tmote Sky motes [26]. Tmote Sky uses CC2420 IEEE802.15.4 transceiver and has 48kb of flash and 10kb of RAM. The simulated network was created with one BR, 5 router nodes, and a malicious node. In order to simulate the mobile behaviour of the IoT devices, the location of the malicious node was changed between simulation runs. However, due to the performance of RPL, the information it sends is always forwarded through its preferred parent, normally the router node closest to it. We have conducted experimentation in order to measure a baseline to use to compare the performance of the proposed IDS system. This is done by simulating the network shown in the above mentioned figure without the presence of the malicious node. Router nodes in the baseline setup do not have any special additional code, they act as RPL routers and exchange only RPL control messages among themselves and the root. Another issue that we have taken into consideration is the Radio Duty Cycle (RDC). In WSN, the rate of data transmission is usually low when compared to other networks. And so, it is not logical to keep the radio on all the time when there are no active transmissions in order to save the power of the nodes. This gave birth to many RDC protocols that control the rate which nodes can turn on or off their radios in between transmissions. In Contiki OS, the prominent RDC protocol is referred to as ContikiMAC. It takes into consideration the sleep patterns of different nodes in the network when transmitting or listening. Contiki also features an RDC protocol that keeps the radio on all the time whether there is active communication or not. It is called NullRDC. In this paper, we have simulated the network both with duty cycling using ContikiMAC and without duty cycling using NullRDC.

Mainly we are focusing on two parameters, the power consumption of the intrusion detection system and the extra memory foot print caused by adding the IDS features to the router nodes. These two metrics are discussed in more details in the following subsections.

B. *Power Measurements*

As the nodes in an IoT network are usually resource constrained, any additional feature to be added to them will have to take into consideration the extra power consumptions it adds to the nodes. Power measurements were made using the powertrace tool included in Contiki OS [27]. This tool shows the time each mote spends in one of four states. Mainly: transmitting (T_x), receiving (R_x), low power mode (LPM), and processing (CPU). Using these values, the energy (E) of a node can be calculated using the following formula

$$E(mWs) = T_x*19.5+R_x*21.8+LPM*0.0545+CPU*1.8 \tag{1}$$

These values are taken from the Tmote Sky data sheet and they are shown in Table I.

TABLE I: Base measurement units for Tmote-Sky nodes.

Typical Operating Conditions	MIN	NOM	MAX	UNIT
Supply voltage	2.1		3.6	V
Supply voltage during flash memory programming	2.7		3.6	V
Current Consumption: MCU on, Radio RX		21.8	23	mA
Current Consumption: MCU on, Radio TX		19.5	21	mA
Current Consumption: MCU on, Radio off		1800	2400	μA
Current Consumption: MCU idle, Radio off		54.5	1200	μA
Current Consumption: MCU standby		5.1	21.0	μA

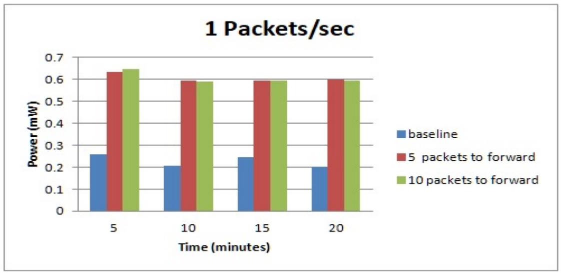

Fig. 3: Power Consumption vs Time of the IDS node when the transmission rate is 1 packet/sec with duty cycling

The average power consumption of a single node can be calculated using the following formula.

$$Power(mW) = \frac{Energy(mWs)}{Time(s)} \tag{2}$$

Which takes into consideration the real time each node was active. Results obtained are discussed in Section V.

C. RAM and ROM usage

Another scarce resource in IoT is the memory of the nodes. As these nodes are cheap, small, and usually expendable, they usually do not have memory size akin to personal computers. For example, the Tmote Sky has only 48kb of flash and 10kb of RAM. Therefore, we have to measure the footprint of the code for the baseline setup and for the IDS setup to assess the extra resources required for our proposed system. Results for the baseline power and energy consumptions are presented in the next section for the cases with and without duty cycling.

V. Evaluation

The simulations were performed using the network topology shown in the previous section where malicious nodes continuously send packets to the BR. We have tested different values of the transmission rate primarily for 1, 10, 100, and 1000 packets per second. The IDS nodes will collect information on the malicious packet. Mainly the source and destination IP:Port of the malicious message as well as its size. The IDS will aggregate this information into one packet and send it to the BR. Two variations on the performance of the IDS node were tested: Firstly when the nodes send information to the BR each time they receive 5 packets from the malicious node. Secondly, when the number of received packets is 10.
The figures below show the power consumption of the IDS node for the different scenarios simulated. The consumed power was measured after 5, 10, 15 and 20 minutes of run time with duty cycling enabled.

The figures 3,4,5 and 6 presented above show that the power consumption increases as the size of the IDS packet increases.

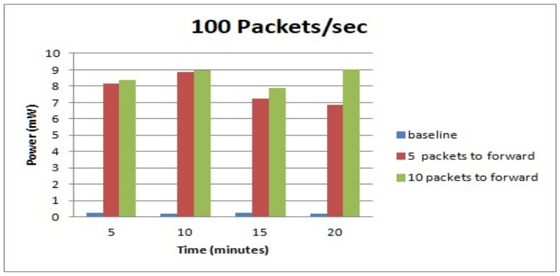

Fig. 4: Power Consumption vs Time of the IDS node when the transmission rate is 100packet/sec with duty cycling

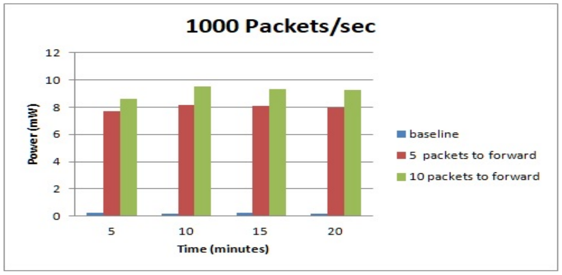

Fig. 5: Power Consumption vs Time of the IDS node when the transmission rate is 1000 packet/sec with duty cycling

However, as is evident from the figures, it is not a significant increase that may affect the performance of the system greatly. Additionally, we have tested the system without duty cycling, i.e when the radio is always on and nodes never sleep. As expected the results are same for all the scenarios tested as the CPU consumes negligible energy when compared with the energy consumed by the radio. And since the radio is always turned on, the difference in energy consumption between the different scenarios is not significant which makes the results for different rates virtually identical to each others. Further experimentation has been conducted to assess the memory overhead for the proposed scheme however details of this evaluation will be included as part of future work.

VI. Discussion

In IoT networks, the IDS can be placed in two places, at the edge router or at the end host. The IDS placed at the edge router has the capability of blocking malicious traffic at the network entry point thus protect the end nodes from the malicious traffic. However, an IDS at the edge router might not consider the behaviour of the devices themselves and may lead to high communication overheads between

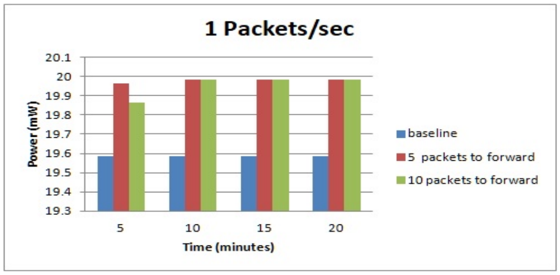

Fig. 6: Power Consumption vs Time of the IDS node when the transmission rate is 1 packet/sec without duty cycling.

nodes and the edge router. On the other hand, an IDS at the end nodes can monitor the performance of end nodes but it requires high processing overheads resources (processing, storage, and energy). The existing IDSs for IoT networks are mostly isolated and monitor a single device by performing analysis for the attacks on a local device. There is no communication taking place between the nodes to make a collaborative detection. The standalone system will not detect most advance and distributed attacks. Moreover, standalone systems will not be able to correlate the traffic statistics or malicious traffic passing through a number of devices at the same time. Furthermore, the standalone IDS system always perform the detection function with respect to seen traffic on its deployed node, thus allows the intruder to misuse the device for longer time periods. Naturally, collaboration among devices could provide the effective defence. In collaboration, the end devices or IDS at the nodes monitors the traffic patterns and reports the events to the centralized or distributed system for event correlation and feedback aggregation. This paper is a first attempt towards the design of collaborative intrusion detection system for the IoT network without incurring high communication or computation overheads. Furthermore, we have evaluated here only the node level detection. The edge router detection will be evaluated as part of future work.

VII. Conclusion

This paper has focused on the challenge of intrusion detection for 6LoWPAN based networks taking into account characteristics such as resource constraints and the M2M communication among these devices. The paper has proposed a novel framework for intrusion detection which combines host and network based approaches to achieve efficient intrusion detection for IoT using 6LoWPAN. We have implemented and evaluated the performance of proposed system by performing simulation for different network scenario in the Contiki operating system. Our results show that the proposed approach has small overheads in terms of energy consumption and memory, and is feasible to use in low energy intrusion detection such as an IoT scenario. As part of the future work, we envisage expanding the evaluation to the edge router module and experimentation involving multi-stage attacks.

References

[1] Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[2] H. J. W. and T. P., ",compression format for ipv6 datagrams over ieee 802.15.4-based networks," 2011. [Online]. Available: http://tools.ietf.org/html/rfc6282, Sept. 2011.

[3] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors*, ser. EmNets '07. New York, NY, USA: ACM, 2007, pp. 78–82.

[4] J. Olsson, "6lowpan demystified." [Online]. Available: http://www.ti.com/lit/wp/swry013/swry013.pdf

[5] G. Montenegro, N. Kushalnagar, and D. Culler, "Transmission of ipv6 packets over ieee 802.15.4 networks." [Online]. Available: https://tools.ietf.org/html/rfc4944

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

[7] T. W. et.al, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[8] M. Sheikhan and H. Bostani, "A hybrid intrusion detection architecture for internet of things," in *2016 8th International Symposium on Telecommunications (IST)*, Sept 2016, pp. 601–606.

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

[10] C. Jun and C. Chi, "Design of complex event-processing ids in internet of things," in *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, Jan 2014, pp. 226–229.

[11] A. Le, J. Loo, Y. Luo, and A. Lasebae, "Specification-based ids for securing rpl from topology attacks," in *2011 IFIP Wireless Days (WD)*, Oct 2011, pp. 1–3.

[12] A. S. Obaid, S. Muhammad Shoaib, H. Choong Seon, and L. Sungwon, "Rides: Robust intrusion detection system for ip-based ubiquitous sensor networks," 2009.

[13] A. Abduvaliyev, S. Lee, and Y. K. Lee, "Energy efficient hybrid intrusion detection system for wireless sensor networks," in *2010 International Conference on Electronics and Information Engineering*, vol. 2, Aug 2010, pp. V2–25–V2–29.

[14] L. Wenchao, Y. Ping, W. Yue, P. Li, and L. Jianhua, "A new intrusion detection system based on knn classification algorithm in wireless sensor network," 2014.

[15] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani, "Intelligent intrusion detection in low-power iots," *ACM Trans. Internet Technol.*, vol. 16, no. 4, pp. 27:1–27:25, Dec. 2016.

[16] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, thirdquarter 2015.

[17] M. A. Azad, S. Bag, and F. Hao, "M2m-rep: Reputation of machines in the internet of things," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17. New York, NY, USA: ACM, 2017, pp. 28:1–28:7.

[18] C. V. Zhou, C. Leckie, and S. Karunasekera, "Decentralized multi-dimensional alert correlation for collaborative intrusion detection," *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1106 – 1123, 2009.

[19] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "Kalis; a system for knowledge-driven adaptable intrusion detection for the internet of things," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 656–666.

[20] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 570–577, Dec 2014.

[21] S. Alessandro, G. Felix, C. Mauro, and B. Jens-Matthias, "Raspberry pi ids: A fruitful intrusion detection system for iot," in *2017 13th IEEE International Conference on Advanced and Trusted Computing (ATC 2016)*, 2016, pp. 1–9.

[22] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, "An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2704–2713, Oct 2017.

[23] P. Ning and D. Xu, "Learning attack strategies from intrusion alerts," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 200–209.

[24] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, Nov 2004, pp. 455–462.

[25] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Local computer networks, proceedings 2006 31st IEEE conference on*. IEEE, 2006, pp. 641–648.

[26] "Tmote Sky," in *http://www.snm.ethz.ch/Projects/TmoteSky*.

[27] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 28–32.