An Intrusion Detection Framework for Energy Constrained IoT Devices

Junaid Arshad[a], Muhammad Ajmal Azad[b], Muhammad Mahmoud Abdeltaif[c], Khaled Salah[d]

[a]*School of Computing and Engineering, University of West London, London, UK*
[b]*Department of Computer Science and Mathematics, The University of Derby, Derby, United Kingdom*
[c]*Electrical Engineering Department, Faculty of Engineering, The British University in Egypt, Cairo, Egypt*
[d]*Department of Electrical Engineering and Computer Science, Khalifa University, UAE*

Abstract

Industrial Internet of Things (IIoT) exemplifies IoT with applications in manufacturing, surveillance, automotive, smart buildings, homes and transport. It leverages sensor technology, cutting edge communication and data analytics technologies and the open Internet to consolidate IT and operational technology (OT) aiming to achieve cost and performance benefits. However, the underlying resource constraints and ad-hoc nature of such systems have significant implications especially in achieving effective intrusion detection. Consequently, contemporary solutions requiring a stable infrastructure and extensive computational resources are inadequate to fulfill these characteristics of an IIoT system. In this paper, we propose an intrusion detection framework for the energy-constrained IoT devices which form the foundation of an IIoT ecosystem. In view of the ad-hoc nature of such systems as well as emerging complex threats such as botnets, we assess the feasibility of collaboration between the host (IoT devices) and the edge devices for effective intrusion detection whilst minimizing energy consumption and communication overhead. We implemented the proposed framework with Contiki operating system and conducted rigorous evaluation to identify potential performance trade-offs. The evaluation results demonstrate that the proposed framework can minimize energy and communication overheads whilst achieving an effective collaborative intrusion detection for IIoT systems.

Keywords: Internet of Things (IoT), Industrial IoT, Intrusion Detection, Constrained IoT Devices, Performance Evaluation

1. Introduction

Internet of Things (IoT) is a network of connected devices that are equipped with specialized software, actuators, sensors, electronic, and communication systems. IoT devices are capable of monitoring physical environment, reporting events and information from the physical world to a remote system for meaningful decisions. IoT has received significant attention over the recent years and has been a building block for number of emerging domains such as smart cities, smart industries and smart homes. It is estimated that the number of IoT devices across the world will exceed more than 26 billion by the year 2020 [15, 4, 2]. A predominant area of IoT applications is industrial settings, Industrial IoT (IIoT), such as manufacturing, surveillance, automotive, smart buildings, homes and transport as illustrated by [47] and [45]. A study by Forbes [1] has estimated manufacturing, transportation and logistics industries to spend more than $40 billion by 2020 to adopt IoTs for improved services and functions. The extraordinary growth predicted by such studies is profoundly due to widespread use of embedded controllers in such industries to achieve automation. The significance of IoT within industrial applications is further heightened due to forthcoming adoption of 5G technologies which will strengthen the impact of IoT by virtue of seamless integration with back-end services such as cloud computing [23].

Typically, devices in an IoT network are autonomous and connected to each other as well as physical systems such as grids, automobiles and industrial systems converging into *Cyber-Physical Systems (CPS)* [24]. This creates an open architecture for the IoT making them an attractive target for malicious actors intending to break the secure network infrastructure, or compromise sensitive information about behaviour of the users as well as attempting to undermine critical national and international infrastructures. These factors intensify the significance of security and privacy mechanisms for IoT systems and therefore mandate explicit efforts to address them.

A typical IoT device has limited processing power, energy resources, and communication range. In view of these characteristics and the need for connectivity, 6LoWPAN standard (IPv6 over Low power Wireless Personal Area Networks) [20], [31],[33] has been developed to enable such resource-constrained devices to communicate with each over the existing IPv6 network. This standard facilitates communication between devices by performing header compression and fragmentations to fit the large sized IPv6 packets into smaller link layer frames such as those defined in IEEE 802.15.4 [30]. For a typical LoWPAN, this connectivity is achieved by using an edge router which facili-

tates connectivity among the devices participating within a LoWPAN as well as with the Internet. However, an IoT network is vulnerable to security threats because of its openness, larger foot print, continuously changing topology, real-time joining and leaving of nodes in the network and lack of centralized network management systems. Furthermore, IoT devices have unique features such as limited power supply, limited energy and communication bandwidth, small memory size and data storage. These constraints have significant impact on the effectiveness of security solutions for IoT infrastructures both in terms of scalability [25] as well as performance efficiency of the solution [23]. Hence designing an effective intrusion detection system for the IoT network is non-trivial motivating us to investigate challenges to achieve an effective intrusion detection system for IoT without incurring significant computation and communication overheads.

An Intrusion detection system (IDS) can be distinguished in two fundamental system architectures [46, 28, 49]: standalone, and collaborative systems. A standalone system uses information from a single source, whereas a collaborative system uses information from diverse set of devices across network. Furthermore, an IDS can be either placed at the device level (host-level) or the edge router level (network base). Within this context, standalone systems are not ideal for resource-constrained devices because of high energy and memory overheads. Additionally, a standalone system can be easily circumvented by sophisticated attackers rendering their monitoring ineffective. The objective of our research is to investigate challenges within intrusion detection for resource-constrained IoT devices with two-fold motivation: firstly, IoT devices are typically resource constrained thereby limiting their ability to host sophisticated security system and secondly, 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. We believe transferring computation workload from device level to edge router will not only benefit the resource-constrained devices but also have the advantages of collaborative defense against malicious actors. 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.

In this paper we propose a framework for *COLlaborative Intrusion DEtection for IoT (COLIDE)* which leverages the fundamental principle of collaboration between individual sensor nodes and the edge router to achieve efficient and cost-effective intrusion detection for IoT systems. The framework is composed of a device level and a edge router component. The device-level component is responsible for detection at the device level, monitoring events visible at the device level whereas the edge router module is responsible for processing security events from the IoT devices for the meaningful decision (malicious or non-malicious), which is then communicated back to the IoT device to block or allow events of specific patterns. By adopting this approach, COLIDE is able to enhance defence for a LoWPAN by using security events gathered at all the participating node therefore protecting against complex, multi-stage attacks. We also 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. We have implemented the framework in the Contiki operating system simulating real-lfie scenarios such as Denial of Service (DoS) and Botnet attacks. The evaluation results demonstrate that COLIDE is effective against such threats whilst minimizing communication, processing and memory overheads.

The rest of the paper is structured as follows. Section 2 describes the related work regarding intrusion detection in an IoT network. Section 3 presents our proposed intrusion detection approach for IoT including its description and formal analysis using Z-notations. A detailed description of the experimentation setup is presented in section 4 with thorough evaluation presented in section 5. Section 6 presents an overall discussion about the effectiveness of the scheme highlighting its applicability and effectiveness for IoT systems. Section 7 concludes the paper.

2. Related work

Intrusion detection within IoT systems has received significant attention in recent years. Reviewing the literature, we identified significant overlap with existing IDS efforts for wireless sensor networks as well as SCADA system however we only include efforts focusing specifically on IoT systems here. Furthermore, we have organized the selected efforts into categories based on the approach used for intrusion detection i.e. signature, anomaly or hybrid.

2.1. Signature based intrusion detection

Sheikhan and Bostani [41] presented an effort similar to [35] such that; both focus on using 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 COLIDE framework proposed in this paper 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 participating devices. Forzin et al. [40] proposed leveraging Snort, a contemporary signature based network IDS, to establish a portable, easy to use and versatile intrusion detection system for IoT networks. The resultant IDS is packaged within a Raspberry Pi so it can be transported with the device to any network the hardware travels to. Consequently, the device is not dependent on a centralized IDS which is an advantage of a host based intrusion detection system.

Kasinathan et al. [21] presented an IDS framework for 6LoWPAN able to detect denial of service attacks by monitoring physical parameters of the device. The proposed IDS is included into an IoT network and monitors network traffic for both signatures and abnormal behaviour to identify malicious users. The intrusion detection component is implemented using an open source IDS Suricata which has complete IPv6 support, multithreading, automatic protocol detection and a built-in intrusion prevention system. Sedjelmaci et al. proposed an efficient and lightweight intrusion detection mechanism for securing the vehicular network in [39]. The approach utilizes the rule-based intrusion detection to identify different type of attacks. In [6], Alessandro et al. proposed an IDS architecture for the IoT that uses the Raspberry Pi equipped with Snort intrusion detection system. The authors in [19] 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.

2.2. Anomaly based intrusion detection

In [11], Chordia and Gupta proposed an anomaly based IDS focused on four attacks namely; U2R, R2R, DoS and Probe. The proposed system aims to monitor network traffic and uses techniques such as K-NN, K-Means and Decision Table Majority Rule Based scheme. Khan and Herrmann [22] proposed an IDS for IoT by using trust management mechanism which collects information about neighboring devices and their reputation. Authors investigated the patterns of normal use for the RPL protocol using them as a foundation to devise trust among a sensor device and the edge routers. The proposed approach is aimed at routing-specific attacks such as sinkhole, selective forwarding and version number. [42] presents an IDS where each device monitors nearby devices for abnormal activity. If an abnormality is detected, packets for malicious node are blocked and reported to the parent node or root node by Distress Propagation Object (DPO). Zhang et al. [48] proposed a hierarchical and distributed IDS (SGDIDS) for smart grids which is applied to three layer communication network i.e. Home Area Network (HAN), Neighbourhood Area Network (NAN) and Wide Area Network (WAN) and is envisioned to address the cyber-physical nature of smart grids protecting against both physical and cyber attacks. It leverages classification algorithms such as support vector machine (SVM) and artificial immune system (AIS) in order to determine occurrence of an attack, the attack type, and its origin. Similarly, Saeed et al. [38] used random neural networks to achieve intrusion detection for IoT systems.

Due to the emergence of recent IoT Botnet threats such as Mirai [17], recent approaches have focused on protection against such attacks. In this context, [27] present one of the first efforts focusing on protection against the IoT botnets which are one of the emerging threats for IoT systems. The authors use deep learning autoencoders to establish an anomaly detection engine which is evaluated against two major botnets i.e. Mirai and BASHLITE. With respect to placement of the IDS, authors use a hybrid approach where by a central unit coordinates with device level encoders (each encoder is responsible for profiling individual IoT device). Similarly, [26] use deep learning to achieve effective detection of IoT botnets. Furthermore, both approaches are similar in that they use patterns within network traffic to discover anomalous behavior representing infection caused by botnets. In terms of placement of the intrusion detection system, the authors have adopted a network based approach employing deep packet inspection techniques.

2.3. Hybrid intrusion detection approaches

Raza et al. [35] proposed a hybrid intrusion detection system for IoT taking into account unique network elements of IoT i.e. network protocols developed for the constrained devices including RPL and 6LowPAN. Furthermore, Nobakht et al. [32] proposed a host based IDS using Software Defined Networks (SDN) for smart homes. The authors define three requirements for an efficient IDS for IoT i.e. unobtrusive approach, negligible overheads, and scalability and aim to achieve these with their proposed approach. Obaid et al. [37] 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 [37] in that it also uses both signature and anomaly based detection systems, however [37] proposed to use both detection engines at device and edge router level. We believe this has significant performance overheads especially at device level due to limited resources. On the contrary, we propose using signature based intrusion detection system at the device level and the anomaly based IDS at the edge router thereby significantly reducing the overall performance overhead. In [5], Abduvaliyev et al. presented another hybrid intrusion detection system which combines 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.

2.4. Other intrusion detection approaches

In [50], Zhou et al. proposed a decentralized multidimensional alert correlation system for the collaborative intrusion detection. The system consists of two algorithms implemented in a distributed CIDS, first algorithm clusters alerts locally at each device, before reporting significant alert patterns to a global correlation stage. The authors in [29] proposed a self-adapting, knowledge-driven IDS for IoT network running different communication protocols. Golomb et al. [16] present an innovative approach, CIOTA, to intrusion detection for IoT leveraging blockchain technology. The proposed approach is comprised of local

agents and a central component which coordinates information (alerts) received from these agents. Authors use blockchain technology to achieve assurances about the authenticity of alerts generated by local agents.

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 device-level resources and the inherent flexibility of the IoT networks.

3. Intrusion detection framework for Constrained IoT Devices

We have proposed a novel intrusion detection system - named COLIDE, for constrained IoT devices as introduced in [7]. COLIDE is a collaborative IDS 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 presenting its different components and interactions between them.

As presented in Fig 1, intrusion detection is performed at two levels, the IoT device level and the edge router level. A detailed description of these components along with motivation for the framework and formal representations is presented in the following sub-sections. In order to to aid the formal notations for different components of the systems, we define the following data models:
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
MEH:a set of events identified as malicious by device-level components for a host H
PRE-COND: a set of events classified as a pre-condition for a complex attack
POST-COND: a set of events classified as a post-condition for a complex attack

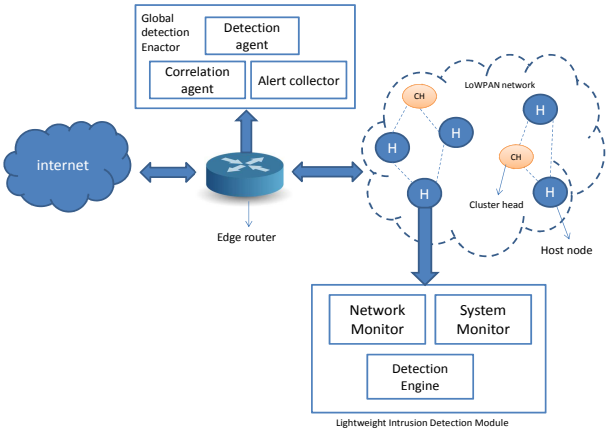


Figure 1 An overview of collaborative intrusion detection system for IoT

3.1. Device level detection

The device level detection module is envisaged to be a light-weight module present at all endpoints (IoT devices). This module is envisioned to take advantage of the unique visibility offered by a device level system to improve the overall intrusion detection. In this regard, we propose using signature based intrusion detection within devices 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 device level is rendered application specific as it will influence the types of attack that the system can mitigate with. For instance, a Denial of Service (DoS) attack targeted to flood specific devices within a LoWPAN can be detected by monitoring network traffic whereas a backdoor channel attack targeted at gaining unauthorized access to a device can be detected by monitoring system events. The detection engine for the device level module therefore processes the information generated by the monitor(s) using existing signatures to detect attack attempts.

To present device level component in formal notations, let us define an event within a given host H as Eh_i. In view of the flexibility of implementation (network or system) offered by this 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, Eh_i 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 and packet size. Therefore, Eh_i can be represented as:

$$Eh_i : \{pr_1, pr_2, pr_3, \cdots pr_n\}$$

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

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

[Device Level Intrusion Detection]

$\Delta Host_{ID}$
$Eh_{ID}? : \mathbb{N}$
$Eh_i? : EH$
$PH_i? : PH$
$SEH_i? : SE$

$Eh_{ID} > 0$
$EH \neq <>$
$PH \neq <>$
$SH \neq <>$
if $Eh_i \in EH$ *then*
$SEH_i = DE(Eh_i, PH_i)$
if $SEH_i = Malicious$ *then*
$IntrusionResponse(Eh_i, SEH_i)$

3.2. 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 leveraging enhanced computational capabilities of edge routers to achieve rigorous intrusion detection for IoT systems as highlighted in [18]. In particular, the edge router detection module is envisaged to monitor traffic for LoWPAN(s) connected to it thereby achieving an extra layer of protection for the infrastructure. Among other benefits, it 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 also called the Global Detection Enactor (GDE), has three components: Alert Collector (AC), Correlation Agent (CA), and Detection Agent (DA). As the edge router is envisaged to monitor all the devices within a LoWPAN, it requires a mechanism to communicate with individual IoT devices and be able to identify threats/alerts respectively. In the proposed system, this function is achieved within the GDE by the *Alert Collector* which communicates with individual IoT devices to gather alerts from the device-level monitoring agents. This can be achieved through an established lightweight protocol such as the Message Queuing Telemetry Transport (MQTT) protocol. However, A typical intrusion is usually not an isolated malicious event which can be achieved within a single transaction or network event but is usually a series of steps each of which may target a specific vulnerability aiming to achieve overall successful intrusion. In this regard, the proposed system implements the *Correlation Agent* component which facilitates countermeasures for such attacks by correlating malicious events at network and system levels as monitored by the device level monitors. Correlation of security events is a well researched area with recent approaches focusing on use of association rule mining and deep learning techniques to achieve this effectively. This enables improved visibility into the events within IoT devices and facilitates detection of complex attacks which involve multiple different steps thereby improving the effectiveness of the intrusion detection process.

As a snapshot of formal representation of the GDE, we include the formal notation for the Correlation Agent below with details of other components provided in [8]. This formal notation is envisaged to help development of the GDE for specific IoT environments and ensures correctness of the scheme against design flaws. As presented in the formal description, the correlation agent takes a set of events which have been identified as malicious (M_EVENTS) by device level agents (line1). It also defines the set of all events and identifies dependencies between them (line 2-4) which will indicate a potential correlation between them. Finally, the CA iterates through the malicious events set to identify temporal relationship between individual events. If a temporal relationship between two events is identified, the CA then triggers intrusion response system with the chain of events so to evaluate appropriate response required for the intrusion.

[Global Detection Enactor]

ΔCorrelation Agent
$MEH_i?, MEH_j? : M_{EVENTS}$
$EH_i? : EVENTS$
$dependencies - evaluation :$
$(EH_i, PRE - COND, POST - COND) \rightarrow \mathbb{P}\, DEPEND$
$d_{event} : DEPEND$

$M_{EVENTS} \neq <>$
$DEPEND \neq <>$
IF $(MEH_i.post - cond = MEH_j.pre - cond)$
$\wedge\ (t_i < t_j)$ *THEN*
Intrusion Response(MEH_i, MEH_j)

Furthermore, anomaly based intrusion detection approaches have historically 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 through the alerts collected and correlated by the Alert Collection and Alert Correlation components.

3.3. Formal analysis of the scheme

A key characteristic of the proposed scheme is its ability to protect against multi-stage attacks. In this section we evaluate correctness of the scheme to evaluate dependencies among malicious events and detection accuracy for multi-stage attacks.

a. Analyzing correctness of dependencies evaluation:

Let $PRE - COND = pre_1, pre_2, ..., pre_n$ represent a set of possible *pre conditions* for respective malicious events.

Let $POST - COND = post_1, post_2, ..., post_n$ represent a set of possible *post conditions* for respective malicious events.

$\exists_1\, Eh_i, Eh_j \mid (post_i \neq pre_j \wedge MEd_i.post = MEd_j.pre)\, \forall (post_i = pre_j \wedge MEd_i.post \neq MEd_j.pre) \bullet j > i \bullet Eh_i \in M_{EVENTS} \bullet MEd_i \in DEPEND$

From system specification,

$\forall\, Eh_i \in M_{EVENTS}$, $dependency_{evaluation} : (Eh_i, pre, post) \rightarrow DEPEND \bullet pre \in PRE - COND \bullet post \in POST - COND$

$\wedge PRE - COND \neq <>$

$\wedge POST - COND \neq <>$

$\Rightarrow dependencies' = dependencies \cup Eh_i \Leftrightarrow pre = post \wedge (t_e > t_{cp}) \leftrightarrow \neg(\exists\, Eh_i, Eh_j, \mid (post_i, \neq pre_j \wedge MEd_i.post = MEd_j.pre)\, V\, (post_i = pre_j \wedge MEd_i.post \neq MEd_j.pre) \bullet j > i \bullet Eh_i \in M_{EVENTS} \bullet MEd_i \in DEPEND$

$\Leftrightarrow false$

The above proof verifies the ability of the proposed intrusion detection scheme to correctly identify and match dependencies between individual attack steps.

b. Correctness to detect multi-stage attacks:

The property of the scheme to mitigate against multi-stage attacks can be formally represented as under.

$((\exists_1\, Eh_i \mid DE(Eh_i)) \wedge (\neg\Sigma DE(Eh_i))) \bullet Eh_i \in M_{EVENTS}$

In order to prove this, we can divide this into two parts. For the first part,

$\Leftrightarrow (\exists_1\, Eh_i — (Eh_i)) \bullet Eh_i \in M_{EVENTS}$

From specification,

$\forall\, Eh_i \mid Eh_i \in M_{EVENTS} \bullet (MEH_i.post_{cond} = MEH_j.pre_{cond}) \wedge (t_i < t_j) \mid (SHE_i = Malicious \wedge SHE_i$=Malicious)

$\Leftrightarrow DE(Eh_i) \wedge SHE_i$= Malicious

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge SHE_i$=Malicious

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge DE(Eh_i) \mid Eh_i \in$ DEPEND

$\Leftrightarrow DE(EH_i) \mid Eh_i \in M_{EVENTS} \wedge Eh_i \in DEPEND \wedge DE(Eh_i)$

$\in DEPEND \bullet dependencies' = dependencies \cup (Eh_i)$

$\Leftrightarrow (Eh_i.post_{cond} = Eh_j.pre_{cond}) \wedge j > i$

$\Leftrightarrow \neg(\exists_1\, Eh_i \mid (Eh_i))$

$\Leftrightarrow false$

The above analysis proves that the hypothesis $\Leftrightarrow \neg(\exists_1\, Eh_i \mid DE(Eh_i)) \bullet Eh_i \in M_{EVENTS}$ i.e. the intrusion detection performed at the edge router level takes into account multiple events and the dependencies among them.

Now the second part can be written as under.

$(\neg\Sigma DE(Eh_i)) \bullet Eh_i \in M_{EVENTS}$

From specification,

$\forall\, Ehi \mid Ehi \in M_{EVENTS} \bullet (MEH_i.post_{cond} = MEH_j.pre_{cond}) \wedge (t_i < t_j) \mid (SHE_i = Malicious \wedge SHE_i = Malicious)$

$\Leftrightarrow DE(Eh_i) \bullet Eh_i \in M_{EVENTS} \Leftrightarrow DE(Eh_i) + DE(dependencies \rightarrow Indiv_{damage})$

$\Leftrightarrow DE(Eh_i) + DE(dependencies \rightarrow Indiv_{damage}) \wedge : \nu FDEPEND \bullet DEPEND \neq <>$

$\Leftrightarrow DE(Eh_i) + \Sigma MEd_i, DE(Med_i) \bullet Med_i \in dependencies \bullet DEPEND\ neq <>$

$\Leftrightarrow DE(Eh_i) \bullet Eh_i \in M_{EVENTS}$

$\Leftrightarrow false$

The analysis presented above highlights the ability of the proposed scheme to correctly evaluate dependencies among different events to be a part of a complex attack attempt. This is proved as both conditions necessary for the hypothesis to be true have been found false in accordance with the formal specification of the intrusion detection scheme. However, as has been highlighted by this analysis, the correct operation of the scheme depends on the accuracy of the events data and the process of identifying dependencies between individual events. The completeness, coverage and accuracy of security events is a well-established challenge with implications across wide application domains. Within the context of research presented here, dedicated measures can be adopted at the event capturing (at the device level component), event transmission (between the device and edge router components) and event correlation (at the alert correaltor) stages with each of these phases introducing specific challenges. For instance, the alert correlator component is envisaged to produce a directed acylic graph by utilizing techniques such as association rule mining to identify relationships between different events whilst taking into account temporal relations between events. This will not only identify relationships between different events but also the sequence of events so as to present an accurate picture of cause and effect which will be beneficial in classifying events as malicious or non-malicious.

4. Implementation and Experimentation

The implementation for the COLIDE framework was achieved in Contiki OS, the operating system for IoT used widely in research and industry [13]. The evaluation was performed using Contiki v2.7 and its built-in emulator Cooja [34]. A preliminary evaluation of the proposed scheme was presented in [7] focusing on the overheads at the device level. This section presents further experimentation and evaluation of the scheme taking into account the role of edge router and focusing at performance metrics to address the overall performance impact of the proposed scheme.

4.1. Experimental setup

In continuation to our efforts presented in [7], experimentation is focused at achieving a thorough evaluation of our proposed approach with respect to performance overhead caused as a consequence of achieving effective intrusion detection. In this respect, experimentation has simulated scenarios where multiple IoT devices from within an IoT network are targeted as part of a large-scale attack such as a Denial of Service or a Botnet. Such attacks are an important threat to IoT system as identified by [DCMS] with Miari [17], Reaper [44] and Brickerbot [43] as recent examples of such attacks on IoT systems. One of the key characteristics of the proposed scheme is the collaboration between intrusion detection components at the device and edge router levels. Therefore, the experimentation simulates scenarios involving collaboration between the device-level and edge-router IDS components to achieve effective intrusion detection. Within this context, each device-level IDS agent implements signature based approach to monitor traffic using predefined intrusion signatures. Upon

detecting an intrusion attempt, each device-level component is programmed to transmit intrusion alerts to the edge router component.

Each alert contains threat-specific information such as device identifier, threat type and time stamp. The edge-router component receives alerts from all the device-level components within a LoWPAN and performs a correlation to evaluate the overall significance of a threat. Within the current experimentation setup, the threat-specific information transmitted through alerts is used to achieve correlation. Furthermore, the frequency of a threat pattern is used to evaluate the overall impact of a threat which triggers a mechanism within the edge-router component to report the alert to a system administrator. Alert correlation is an established research area and we consider more sophisticated alert correlation experiments an avenue for future work.

4.2. 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 which has higher processing power and memory than the devices in the 6LoWPAN network, will be acting as the edge router.

The first tier of nodes that are a part of the 6LoWPAN are referred to as routers/IDS nodes. These IDS nodes will be responsible for forwarding packets to and from the edge router. Additionally, they are responsible for monitoring the packets passing through them for certain malicious patterns and report to the edge router by sending periodic alerts regarding any malicious behavior that may occur in the network. In order to capture any malicious packet being sent to the edge router, the IDS nodes are always placed one hop away from the edge router.

Malicious nodes are located in the second tier. This is done to make sure they can only join the network through one of the IDS nodes in the first tier which means that their packets have to pass through an IDS node before reaching the edge router. This setting enables us to simulate different scenarios to aid evaluation. Figure 2 shows an example of a network with 5 IDS nodes and 3 malicious nodes. In order to evaluate the work done in this paper, a set of simulations have been performed with Cooja using Tmote Sky motes [3]. Tmote Sky uses a CC2420 IEEE802.15.4 transceiver with 250kbs and has 48kb of flash and 10kb of RAM. The simulated network was created with one Ede router, 5 IDS nodes, and 3 malicious nodes. The location of these malicious nodes was selected at random to make the simulation more realistic as the location of the malicious node is not known in real life scenarios. Since we aim to test the impact of the proposed IDS on the devices, we have started by running some basic experiments in order to measure a baseline which will be used as a reference for the comparison with the IDS system. This is done by simulating the network shown in the above mentioned figure

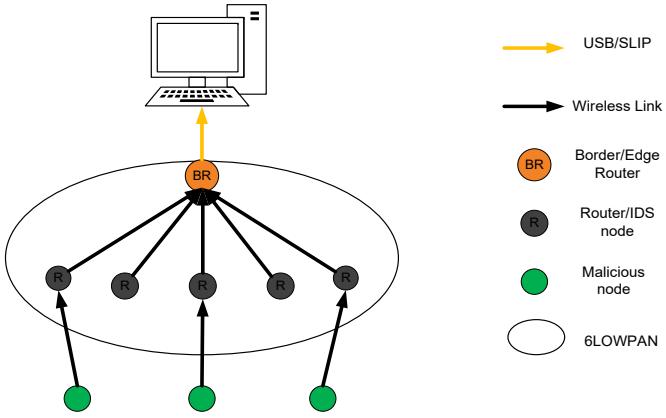


Figure 2: Experimental Setup

without the presence of the malicious node, and with the IDS nodes acting only as RPL routers and exchange only RPL control messages among themselves and the Edge router.The Edge router in the baseline experiment only runs the basic RPL root functionalities.

Another issue that usually rises when dealing with IoT is the Radio Duty Cycle (RDC). In IoT, 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. However, when using NullRDC, the power consumption from the radio is much more than that of the other factors including the additional computation from the proposed IDS system. And so, we have only performed test using ContikiMAC in order for the effect of the IDS system to appear clearly.

We evaluated the performance measurement for two parameters: the power consumption of the intrusion detection system, and the memory overhead caused by adding the IDS features to the device nodes and to the Edge router. These two metrics are discussed in more detail in the following subsections.

4.3. 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 [14]. 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

$$E(mWs) = V*(T_x*19.5+R_x*21.8+LPM*0.0545+CPU*1.8) \quad (1)$$

The total current is calculated by multiplying the time spent in each state by the current consumed during such a state and adding for all the four states. The energy is then calculated by multiplying the total current with the nominal voltage V which is 3 volts in the case of Tmote Sky. The values for each current consumption per state are taken from the Tmote Sky data sheet and they are shown in Table 1.

Table 1: 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

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

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

Which takes into consideration the real time each node was active.

4.4. 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.

5. Evaluation

The simulations were performed using the network topology shown in the previous section.Figure 3 shows the topology of our experimental setup.

The network consists of an edge router (node 1), 5 IDS nodes (nodes 2-6), and three malicious nodes (nodes 7-9). In each simulation run, malicious nodes continuously send malicious packets to the edge router. Different values of the transmission rate were tested, mainly 1, 10, 100, and

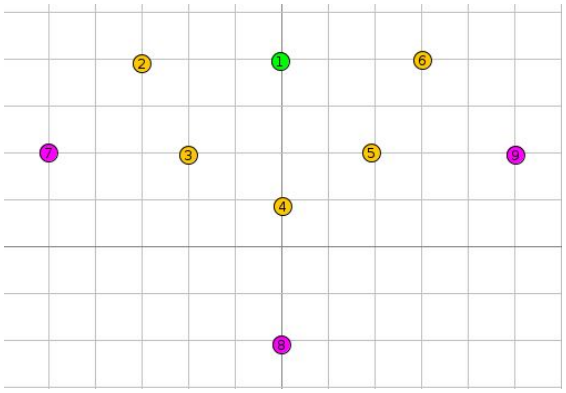


Figure 3: Simulation Topology in COOJA

1000 packets per second. Additionally, malicious nodes alternate between sending one of two predefined malicious patterns simulating attack signatures. The IDS nodes scan the payload of all packets passing through them for malicious patterns, and if found, the IDS node collects information on that packet such as the source and destination IP addresses and port numbers of the malicious packet as well as the attack signature. The IDS nodes will aggregate this information into an alert packet and send it to the edge router for further processing simulating anomaly detection and alert correlation. Two variations on the performance of the IDS node were tested: when IDS nodes send an alert to the edge router each time they receive 5 malicious packets with a certain pattern from a malicious node which we will refer to as (IDS mode 5). Or when the number of received malicious packets is 10 (IDS mode 10). The power consumption and memory foot print of the IDS framework for the IDS nodes is measured.

The experimentation presented here specifically targets performance evaluation of the edge router component (Global Detection Enactor) of the proposed intrusion detection system. In the simulation, the GDE receives alerts from the IDS nodes. It then collects certain information from these alerts, mainly the number of alerts, their source, and time of occurrence. Based on these information, it attempts to identify correlation between these alerts in order to decide the course of action for a malicious node. For instance, depending upon the strength of correlation and the confidence value produced by the correlation agent, a malicious node can be suspended from the network. We have also looked at the power consumption and the memory overhead of the edge router. However, since the edge router is always connected to a consistent power supply, power consumption is not envisaged to be of notable concern. Therefore, in order to demonstrate the effect of the IDS system on the power consumption of the edge router, we only show the power consumption that comes from the additional CPU active time of the edge router.

Specifically, figure 4 shows the power consumption of the IDS node for the various scenarios simulated. The consumed power was measured after 5, 10, 15 and 20 minutes

of run time. As illustrated by this figure, the power consumption for different modes of intrusion detection generally decreases as the size of the IDS alert increases primarily due to the small size of the packets. However, the difference observed is not deemed significant that may affect the performance of the system greatly.

Additionally, as we are varying the transmission rate of the malicious node, the simulation time may not be best option to use as. Therefore, we have measured the power consumption of the nodes vs the number of malicious packets received. The results for these experiments are shown in Figure 5 for 10, 100, and 1000 received malicious packets. They show a similar trend to those results vs time where the difference between the consumption is minimal between the two IDS mode tested.

Figure 6 shows the power consumed by the CPU of the edge router for different values of the transmission rate of the malicious node. From Figure 6, it can be observed that the CPU power consumption of the edge router does not change much between the two different IDS modes regardless of the transmission rate. Similar to the IDS node level, Figure 7 show the CPU power consumption vs the number of received alerts from the IDS nodes. It is clear that the consumption due to the extra computation does not change greatly with the IDS mode change.

Additionally, the Memory footprint of the code has been measured for both the baseline setup and the IDS setup. Table 2 presents the memory overhead caused by adding the IDS functionality to the IDS nodes and the edge router. From the table, we can conclude that the ROM overhead does not change as the size of the IDS alert change. This is because the program itself does not rely on that size. However, the RAM requirements increase with the IDS packet size as more information will be saved in the memory to be sent at once.

Table 2: Memory Overhead caused by the IDS functionality to the IDS node and the Edge router

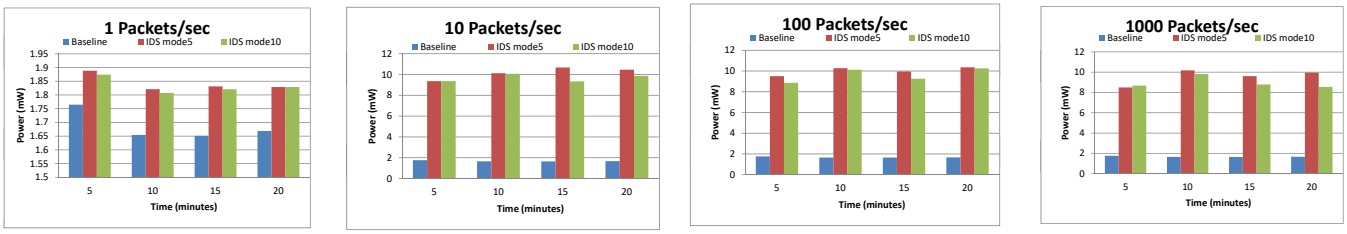
	IDS Mode	RAM overhead	ROM overhead
IDS Node	5	368	274
	10	728	274
Edge Router	5	198	206
	10	378	206

6. Discussion

There are three ways an IDS system can be implemented within an IoT system: as a device-based system, as an edge router-based system, or as a distributed system. Additionally, an IDS system can operate either as a standalone system or as a collaborative system. Existing IDSs for IoT mostly operate in the standalone setting, i.e. using data from a single source (edge router or sensor device). However, standalone systems cannot detect sophisticated attacks such as stealthy attacks or distributed denial of service attacks. On the other hand, distributed collaborative systems are the more suitable way to implement the IDS system for the energy constrained IoT devices. In a collaborative system, the IDS nodes monitor traffic patterns or system events and report to other IDS nodes in a distributed system settings. This paper extends our attempt presented in [8] towards the design of a collaborative intrusion detection system for IoT networks without incurring high power consumption or memory overhead. The distributed collaboration between the IDS nodes and the edge router has the following benefits: 1) it shifts the computation load from the resource and energy-constrained IoT devices to the resource rich edge router thus increase life time of network, and 2) collaboration minimizes the detection time while correlating information from the different devices in the network.

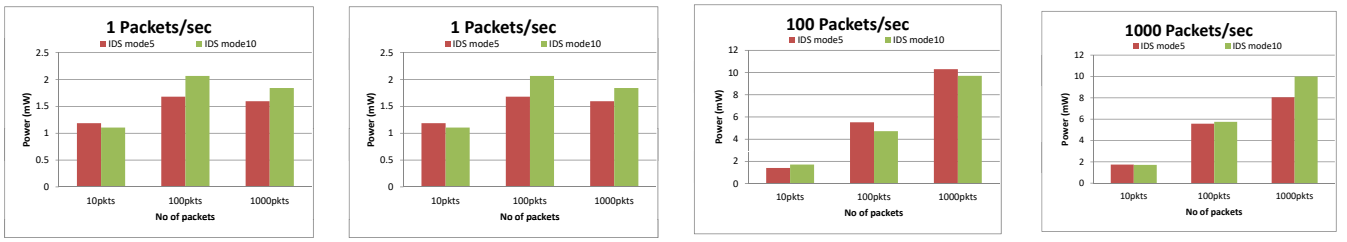
The approach presented in this paper segregates intrusion detection task among the IoT devices and the edge router. IoT devices acting as IDS nodes scan any packet passing through them to decide if they are malicious or legitimate and send periodic alerts to the edge router regarding malicious activity. This enables the Edge router a comprehensive view of the network allowing it to identify intruders. The design choice increases processing load at edge router, but this is not envisaged to create bottleneck as the edge router is equipped with more resources in terms of memory and computation power than the other IoT devices in the network. The major limitation of our design is that it does not ensure the privacy of the host node as it forwards the raw packets to the edge router which may contain sensitive information of host nodes. This becomes a significant challenge when host devices collaborate with devices belonging to other networks. However, in our design choice, we assume that the edge router and devices are placed within the same organization or home. The privacy of host device in cross-network collaboration can be protected in two ways: using the cryptographic measures [9, 10] and secondly exchanging the security model rather than the raw data.

In this paper, we provided a framework for the collaborative intrusion detection in the the resource-constrained devices. We have used formal notations to present a description of the system and to evaluate its accuracy with respect to intrusion detection. The formal notation provides a template for the system whilst enabling flexibility in achieving a custom implementation. However, the framework can be easily extended to apply the machine learning and deep neural networks [36, 12] to identify malicious traffic patterns. To this extent the deep neural networks can be applied in two ways: firstly by deploying the neural network model at the device level and secondly by deploying the model at the edge router. The former would have some overheads, and the latter is not only suitable in terms of overheads but could have high detection accuracy and classification. Furthermore, the advanced learning techniques can also be applied as part of the correlation



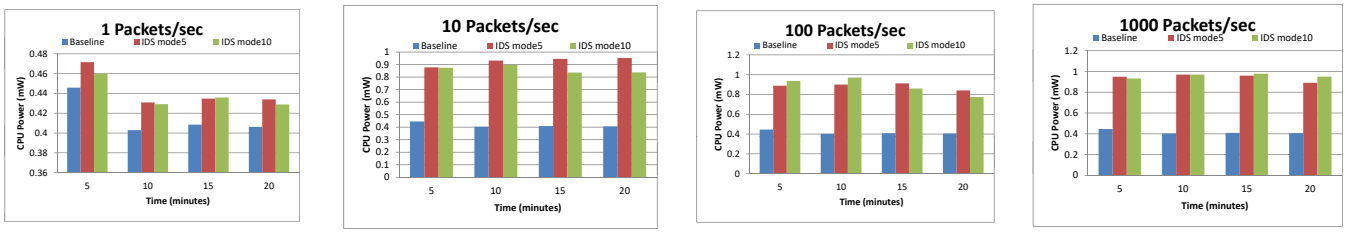
((a)) Tx rate is 1 packet/sec ((b)) Tx rate is 10 packet/sec ((c)) Tx rate is 100 packet/sec ((d)) Tx rate is 1000 packet/sec

Figure 4: Power Consumption of an IDS node vs Time



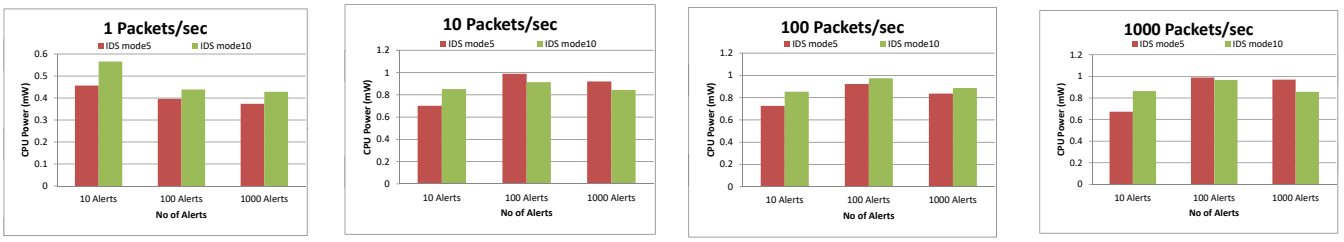
((a)) Tx rate is 1 packet/sec ((b)) Tx rate is 10 packet/sec ((c)) Tx rate is 100 packet/sec ((d)) Tx rate is 1000 packet/sec

Figure 5: Power Consumption of an IDS node vs the Number of malicious packets received



((a)) Tx rate is 1 packet/sec ((b)) Tx rate is 10 packet/sec ((c)) Tx rate is 100 packet/sec ((d)) Tx rate is 1000 packet/sec

Figure 6: CPU Power Consumption of the Edge router vs Time



((a)) Tx rate is 1 packet/sec ((b)) Tx rate is 10 packet/sec ((c)) Tx rate is 100 packet/sec ((d)) Tx rate is 1000 packet/sec

Figure 7: CPU Power Consumption of the Edge router vs the Number of received alerts

agent of the COLIDE model to enhance the accuracy of the correlation task which directly affects the effectiveness of the overall intrusion detection process.

7. Conclusion

The performance efficiency of security measures such as intrusion detection is paramount for Industrial IoT infrastructures primarily due to the resource constraints of the devices participating in such infrastructures. This paper has presented a novel framework for intrusion detection which combines host and network based approaches to achieve efficient intrusion detection for IoT. The proposed system adopts a collaborative approach to address the intrusion detection challenge whilst minimising the performance overhead to conserve resources available at the IoT devices. We have implemented and evaluated the performance of proposed system by simulating different network scenarios using Contiki operating system. Our results show that the proposed approach minimizes the overall overhead in terms of energy consumption and memory, and is effective within constrained devices such as the IoT. As part of the future work, we envisage expanding evaluation to assess effectiveness of machine learning ap-

proaches to enhance performance with respect to accuracy of detecting large scale complex attacks.

References

[1] 2017 roundup of internet of things forecasts.

[2] Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016.

[3] Tmote Sky. In *http://www.snm.ethz.ch/Projects/TmoteSky*.

[4] Gartner says the internet of things installed base will grow to 26 billion units by 2020., 2017.

[5] 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*, volume 2, pages V2–25–V2–29, Aug 2010.

[6] 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)*, pages 1–9, 2016.

[7] J. Arshad, M. Abdellatif, M. Khan, and M. Azad. A novel framework for collaborative intrusion detection for m2m networks. In *The 9th International Conference on Information and Communication Systems*, 2018.

[8] J. Arshad, M. A. Azad, M. Mahmoud Abdellatif, M. H. Ur-Rehman, and K. Salah. Colide: A collaborative intrusion detection framework for internet of things. *IET Networks*, 2018.

[9] 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*, ARES '17, pages 28:1–28:7, New York, NY, USA, 2017. ACM.

[10] M. A. Azad, S. Bag, F. Hao, and K. Salah. M2m-rep: Reputation system for machines in the internet of things. *Computers & Security*, 2018.

[11] A. S. Chordia and S. Gupta. An effective model for anomaly ids to improve the efficiency. In *International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015.

[12] A. A. Diro and N. Chilamkurti. Distributed attack detection scheme using deep learning approach for internet of things. *Future Generation Computer Systems*, 82:761 – 768, 2018.

[13] 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*, pages 455–462, Nov 2004.

[14] 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*, pages 28–32. ACM, 2007.

[15] D. Evans. The internet of things how the next evolution of the internet is changing everything. *White Paper Cisco Internet Business Solutions Group (IBSG)*, 2011.

[16] T. Golomb, Y. Mirsky, and Y. Elovici. Ciota: Collaborative iot anomaly detection via blockchain. In *Workshop on Decentralised IoT Security and Standards*, 2018.

[17] R. Graham. Mirai and iot botnet analysis. In *2017 RSA Conference*, 2017.

[18] M. Habib ur Rehman, A. Batool, and K. Salah. The rise of proximal mobile edge servers. *IT Professional*, 21(3):26–32, May 2019.

[19] 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*, 47(10):2704–2713, Oct 2017.

[20] H. J. W. and T. P. ,compression format for ipv6 datagrams over ieee 802.15.4-based networks. 2011.

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

[22] Z. Khan and P. Herrmann. Hive: Home automation system for intrusion detection. In *IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*., 2017.

[23] S. Li, L. D. Xu, and S. Zhao. 5g internet of things: A survey. *Journal of Industrial Information Integration*, 10:1 – 9, 2018.

[24] Y. Lu. Cyber physical system (CPS)-based industry 4.0: A survey. *Journal of Industrial Integration and Management*, 02(03):1750014, sep 2017.

[25] Y. Lu and L. D. Xu. Internet of things (iot) cybersecurity research: A review of current research topics. *IEEE Internet of Things Journal*, 6(2):2103–2115, April 2019.

[26] C. D. McDermott, F. Majdani, and A. V. Petrovski. Botnet detection in the internet of things using deep learning approaches. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018.

[27] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici. N-baiotnetwork-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, Jul 2018.

[28] G. Meng, Y. Liu, J. Zhang, A. Pokluda, and R. Boutaba. Collaborative security: A survey and taxonomy. *ACM Computing Survey*, 48(1):1:1–1:42, July 2015.

[29] 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)*, pages 656–666, June 2017.

[30] G. Montenegro, N. Kushalnagar, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks.

[31] G. Mulligan. The 6lowpan architecture. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, pages 78–82, New York, NY, USA, 2007. ACM.

[32] M. Nobakht, V. Sivaraman, and R. Boreli. A host-based intrusion detection and mitigation framework for smart home iot using openflow. In *11th International Conference on Availability, Reliability and Security (ARES)*, 2016.

[33] J. Olsson. 6lowpan demystified.

[34] 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*, pages 641–648. IEEE, 2006.

[35] S. Raza, L. Wallgren, and T. Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad Hoc Networks*, 11(8):2661 – 2674, 2013.

[36] P. A. A. Resende and A. C. Drummond. A survey of random forest based methods for intrusion detection systems. *ACM Comput. Surv.*, 51(3):48:1–48:36, May 2018.

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

[38] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani. Intelligent intrusion detection in low-power iots. *ACM Trans. Internet Technol.*, 16(4):27:1–27:25, Dec. 2016.

[39] 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*, 1(6):570–577, Dec 2014.

[40] A. Sforzin, F. Marmol, M. Conti, and J. Bohli. Rpids: Raspberry pi ids ? a fruitful intrusion detection system for iot. In *Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*., 2016.

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

[42] N. Thanigaivelan, E. Nigussie, R. Kanth, S. Virtanen, and J. Isoaho. Distributed internal anomaly detection system for internet-of-things. In *13th IEEE Annual Consumer Communi-*

cations & Networking Conference (CCNC), 2016.

[43] I. Thomson. "forget mirai brickerbot malware will kill your crap iot devices".

[44] K. Townsend. Financial services ddos attacks tied to reaper botnet. In *available at: https://www.securityweek.com/financial-services-ddos-attacks-tied-reaper-botnet*, 2018.

[45] M. H. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera. The role of big data analytics in industrial internet of things. *Future Generation Computer Systems*, 99:247 – 259, 2019.

[46] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer. Taxonomy and survey of collaborative intrusion detection. *ACM Comput. Surv.*, 47(4):55:1–55:33, May 2015.

[47] L. D. Xu, W. He, and S. Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, Nov 2014.

[48] Y. Zhang, L. Wang, W. Sun, R. Green II, and M. Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. In *IEEE Transactions on Smart Grid*, volume 2 of *4*, pages 796–808, 2011.

[49] C. Zhou, C.Leckie, and S.Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124 – 140, 2010.

[50] C. V. Zhou, C. Leckie, and S. Karunasekera. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. *Journal of Network and Computer Applications*, 32(5):1106 – 1123, 2009.