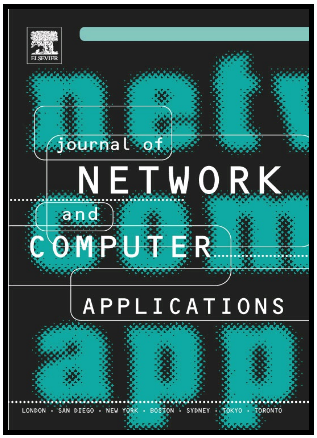
journal of
NETWORK
and
COMPUTER
APPLICATIONS
LONDON • SAN DIEGO • NEW YORK • BOSTON • SYDNEY • TOKYO • TORONTO

www.elsevier.com/locate/jnca

A Survey of Intrusion Detection in I nternet of Things

Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, Sean Carlisto de Alvarenga

PII: S1084-8045(17)30080-2
DOI: http://dx.doi.org/10.1016/j.jnca.2017.02.009
Reference: YJNCA1865

To appear in: *Journal of Network and Computer Applications*

Received date: 21 September 2016
Revised date: 28 January 2017
Accepted date: 19 February 2017

Cite this article as: Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudi Toshio Kawakani and Sean Carlisto de Alvarenga, A Survey of Intrusior Detection in I nternet of Things, *Journal of Network and Computer Applications* http://dx.doi.org/10.1016/j.jnca.2017.02.009

A Survey of Intrusion Detection in Internet of Things

Bruno Bogaz Zarpelão[a,*], Rodrigo Sanches Miani[b], Cláudio Toshio Kawakani[a], Sean Carlisto de Alvarenga[a]

[a] Computer Science Department, State University of Londrina (UEL), Rodovia Celso Garcia Cid, S/N, 86057-970, Londrina, Brazil
[b] School of Computer Science (FACOM), Federal University of Uberlândia (UFU), Uberlândia, Brazil

Abstract

Internet of Things (IoT) is a new paradigm that integrates the Internet and physical objects belonging to different domains such as home automation, industrial process, human health and environmental monitoring. It deepens the presence of Internet-connected devices in our daily activities, bringing, in addition to many benefits, challenges related to security issues. For more than two decades, Intrusion Detection Systems (IDS) have been an important tool for the protection of networks and information systems. However, applying traditional IDS techniques to IoT is difficult due to its particular characteristics such as constrained-resource devices, specific protocol stacks, and standards. In this paper, we present a survey of IDS research efforts for IoT. Our objective is to identify leading trends, open issues, and future research possibilities. We classified the IDSs proposed in the literature according to the following attributes: detection method, IDS placement strategy, security threat and validation strategy. We also discussed the different possibilities for each attribute, detailing aspects of works that either propose specific IDS schemes for IoT or develop attack detection strategies for IoT threats that might be embedded in IDSs.

Keywords: Intrusion Detection System, Internet of Things, Cybersecurity

[*] Corresponding author
Email addresses: brunozarpelao@uel.br (Bruno Bogaz Zarpelão), miani@ufu.br (Rodrigo Sanches Miani), claudio.tk93@gmail.com (Cláudio Toshio Kawakani), sean@uel.br (Sean Carlisto de Alvarenga)

1. Introduction

Evolution of different technology areas such as sensors, automatic identification and tracking, embedded computing, wireless communications, broadband Internet access and distributed services has increased the potential of integrating smart objects into our daily activities through the Internet. Convergence of the Internet and smart objects that can communicate and interact with each other defines the Internet of Things (IoT). This new paradigm is recognized as one of the most important actors in the Information and Communication Technology (ICT) industry for next years [1]. According to Gartner Inc., the IoT may have 26 billion units by 2020. Cisco Systems predicted that the IoT would create $ 14.4 trillion as a result of the combination of increased revenues and lower costs for companies from 2013 to 2022 [2, 3, 4, 5].

Many application domains such as logistic, industrial process, public safety, home automation, environmental monitoring and healthcare may have significant benefits with IoT systems [6]. However, the integration of real-world objects with the Internet brings the cybersecurity threats to the most of our daily activities. Attacks against critical infrastructures, such as power plants and transportation system, may have terrible consequences for whole cities and countries. Household appliances may also be a primary target, threatening security and privacy of families. In [7], tests performed with three popular smart home devices showed different vulnerabilities related to users privacy, lack of encryption and authentication. Due to the different standards and communication stacks involved, the limited computing power and the high number of interconnected devices, traditional security countermeasures could not work efficiently in IoT systems. For this reason, developing specific security solutions for IoT is essential to let users and organizations catch all opportunities it offers [4].

Some ongoing projects for enhancing IoT security include methods for providing data confidentiality and authentication, access control within the IoT network, privacy and trust among users and things, and the enforcement of security and privacy policies [4]. However, even with these mechanisms, IoT networks are vulnerable to multiple attacks aimed to disrupt the network. For this reason, another line of defense, designed for detecting attackers is needed. Intrusion Detection Systems (IDSs) fulfill this purpose.

IDS is one of the primary tools used for protection of traditional networks and information systems. The IDS monitors the operations of a host or a network, alerting the system administrator when it detects a security violation. Research efforts about intrusion detection have been conducted since the beginning of the 1980s, when Anderson [8] published his seminal work about network security monitoring. Hence, the IDS has consolidated its position as a popular defense technology for traditional IP networks, with several solutions on the market [1], [2].

Despite the maturity of IDS technology for traditional networks, current solutions are inadequate for IoT systems, because of IoT particular characteristics that affect IDS development. At first, processing and storage capacity of network nodes that host IDS agents is an important issue. In traditional networks, the system administrator deploys IDS agents in nodes with higher computing capacity. IoT networks are usually composed of nodes with resource constraints. Therefore, finding nodes with the ability to support IDS agents is harder in IoT systems. The second particular characteristic is related to the network architecture. In traditional networks, end systems are directly connected to specific nodes (e.g., wireless access points, switches, and routers) that are responsible for forwarding the packets to the destination. IoT networks, on the other hand, are usually multi-hop. Then, regular nodes may simultaneously forward packets and work as end systems. For instance, in IoT-based street lighting systems, sensors with short-range communication capabilities are deployed on light poles [9, 10, 11]. Then, the data collected by a sensor is forwarded through a path of sensors deployed on different light poles until reaching a gateway to the Internet. This kind of architecture poses new challenges for IDSs. The last characteristic is related to specific network protocols. IoT networks use protocols that are not employed in traditional networks, such as IEEE 802.15.4, IPv6 over Low-power Wireless Personal Area Network (6LoWPAN), IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) and Constrained Application Protocol (CoAP). Different protocols bring original vulnera-

[1] https://www.sans.org/critical-security-controls/vendor-solutions/control/13

[2] http://www.scmagazine.com/intrusion-detection-systems/products/91/0/

bilities and new demands for IDS.

Considering that the development of IDSs for IoT represents a significant challenge for information security researchers, we present a survey about intrusion detection in IoT. Our objectives are threefold: 1) to learn how the researchers have addressed the challenges that IoT particularities pose for IDS development; 2) to propose a taxonomy to classify IDSs for IoT according to the following attributes: detection method, IDS placement strategy, security threat and validation strategy; 3) to identify open issues in IDS development for IoT, indicating future research directions. Since our literature review shows that the research in this area is still incipient, we believe that the most important contribution of this survey is to provide a detailed discussion about future research directions in IDSs for IoT. We argue that open issues related to topics such as selection of detection method, attack detection range, management and security of alert traffic, alert correlation and improvement of validation strategies must be addressed in the future.

The rest of this paper is organized as follows. Section 2 introduces some relevant terms regarding intrusion detection and IoT. Section 3 discusses relevant reviews that surveyed intrusion detection approaches for technologies related to IoT, such as mobile ad hoc networks, wireless sensor networks, cloud computing and cyber-physical systems. Section 4 presents the proposed taxonomy and shows an analysis of the literature of IDSs for IoT. One of the most relevant contributions of this work, a discussion of open issues and future research possibilities IDSs in IoT, is detailed at section 5. Finally, in section 6, we present some concluding remarks.

2. Relevant Terms

This section provides an introduction to the central concepts of this paper: intrusion detection and IoT.

2.1. Intrusion Detection

Intrusion detection is the activity of detecting actions that intruders carry out against information systems. These actions, known as intrusions, aim to obtain unauthorized access to a computer system. Intruders may be external or internal. Internal intruders are users inside the network with some degree of legitimate access that attempt to raise their access privileges to misuse non-authorized privileges. External intruders are users outside the target network trying to gain unauthorized access to system information [12, 13].

A typical IDS is composed of sensors, an analysis engine, and a reporting system. Sensors are deployed at different network places or hosts. Their task is to collect network or host data such as traffic statistics, packet headers, service requests, operating system calls, and file-system changes. The sensors send the collected data to the analysis engine, which is responsible to investigate the collected data and detect ongoing intrusions. When the analysis engine detects an intrusion, the reporting system generates an alert to the network administrator.

IDSs can be classified as Network-based IDS (NIDS) and Host-based IDS (HIDS). Network-based IDS (NIDS) connects to one or more network segments and monitors network traffic for malicious activities. Host-based IDS (HIDS) is attached to a computer device and monitors malicious activities occurring within the system. Unlike NIDS, the HIDS analyzes not only network traffic but also system calls, running processes, file-system changes, interprocess communication, and application logs.

IDS approaches may also be classified as signature-based, anomaly-based or specification based. Since these categories are part of the taxonomy proposed in this paper, more details about them will be provided in Section 4.

2.2. Internet of Things

IoT is a concept that gathers all sorts of different applications based on the convergence of smart objects and the Internet, establishing an integration between the physical and the cyber worlds. These applications may range from a simple appliance for a smart home to a sophisticated equipment for an industrial plant. Although IoT applications have very different objectives, they share some common characteristics. Generally speaking, IoT operations include three distinct phases: collection phase, transmission phase, and processing, management and utilization phase [6].

In the collection phase, the primary objective is to collect data about the physical environment. Sensing devices and technologies for short range communication are combined to reach this goal. Devices of the collection phase are usually small and resource-constrained. Communication protocols and technologies for this phase are designed to operate at limited data rates and short distances, with constrained memory capacity and low energy consumption. Due to these characteristics, collection phase networks often are referred to as LLN (Low-power and Lossy Networks). Solutions for error control, medium access control, routing and addressing in LLNs may be different from those used on the conventional Internet.

The transmission phase aims to transmit the data gathered during the collection phase to applications and, consequently, to users. In this phase, technologies such as Ethernet, WiFi, Hybrid Fiber Coaxial (HFC) and Digital Subscriber Line (DSL) are combined with TCP/IP protocols to build a network that interconnects objects and users across longer distances. Gateways are necessary to integrate LLN protocols of the collection phase with conventional Internet protocols employed in the transmission phase.

In the processing, management and utilization phase, applications process collect data to obtain useful information about the physical environment. These applications may take decisions based on this information, controlling the physical objects to act on the physical environment. This phase also includes a middleware, which is responsible for facilitating the integration and communication between different physical objects and multi-platform applications.

Different alliances, consortiums, special interest groups, and standard development organizations have proposed an overwhelming amount of communication technologies for IoT, what may pose a big challenge for end-to-end security in IoT applications [14]. Most popular technologies for IoT include IEEE 802.15.4, Bluetooth Low Energy (BLE), WirelessHART, Z-Wave, LoRaWAN, 6LoWPAN, RPL, CoAP, and MQTT (Message Queue Telemetry Transport).

IEEE 802.15.4 is a standard proposed by the Institute of Electrical and Electronics Engineers (IEEE) for physical and medium access control layers of low-rate wireless personal area networks. With the IEEE 802.15.4, devices can operate with data rates from 20 kbps to 250 kbps and transmission ranges from 10m to 100m. Medium access control uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) technique [15].

Internet Engineering Task Force (IETF) has proposed standards to work on top of IEEE 802.15.4 and facilitate the integration between LLNs and the Internet. 6LoWPAN standard [16] aims to adapt the IPv6 packet for IEEE 802.15.4, since the former one has a header of 40 bytes and the last one allows only 127 bytes per frame, including header and payload information. 6LoWPAN facilitates the interoperability between IPv6 and LLN nodes, but a gateway between these two networks is still necessary. IETF Routing over Low Power and Lossy Networks (ROLL) Working Group proposed a routing protocol for LLNs, named RPL [17]. It represents the sensor network topology as Destination Oriented Directed Acyclic Graphs (DODAG) to find the best paths according to an objective function and some metrics. It supports multipoint-to-point, point-to-multipoint and point-to-point traffic.

IoT community has proposed protocols for the application layer as well. CoAP and MQTT are two of the most widely discussed application protocols for IoT. IETF Constrained RESTful Environments (CoRE) Working Group proposed CoAP to be a transfer protocol (such as Hypertext Transfer Protocol - HTTP) for LLNs. CoAP allows request/response transactions in LLNs as they occur in the traditional Web, enabling transmissions of gathered data from devices to users [18]. MQTT is a message protocol based on the publish-subscribe pattern. OASIS (Organization for the Advancement of Structured Information Standards), a non-profit international consortium, standardized MQTT in 2013. It was designed to be a lightweight protocol suitable for networks with unreliable or low bandwidth links. Three components are involved in the MQTT publish-subscribe process: the subscriber, the broker, and the publisher. The publisher sends data to the broker. The broker has a list of subscribers, which receive the data of their interest that was sent by publishers [19, 20].

IEEE 802.15.4, 6LoWPAN, RPL, CoAP, and MQTT are standards designed to address specific layers of LLNs protocol stack. However, there are also IoT standards that specify vertically integrated architectures, such as BLE, WirelessHART, Z-Wave, and LoRaWAN.

BLE was developed by the Bluetooth Special Interest Group as an evolution of

Bluetooth technology for low power devices. With BLE, devices can operate at 1 kbps in the 2.4 GHz band. The distance between two BLE nodes is up to 100 meters. The lower layers of BLE protocol stack include a physical layer, responsible for bits transmission and modulation, and a link layer, responsible for medium access control and connection establishment. When the link layer establishes a connection, the devices may adopt the roles of master or slave. A BLE piconet is composed of a set of slaves connected to one master. The Logical Link Control and Adaptation Protocol (L2CAP) works on top of the link layer. The BLE L2CAP is a simplified version of the traditional Bluetooth L2CAP, being mainly responsible for multiplexing the data from upper layers. The upper layers include the Generic Attribute Profile (GATT) and the Generic Access Profile (GAP). The GATT allows service discovery and exchange of characteristics between two devices. The GAP defines some possible operation modes for BLE devices [19, 21].

WirelessHART is the result of the HART Communication Foundation efforts to transform the Highway Addressable Remote Transducer (HART) protocol into a wireless solution. Both HART and WirelessHART were designed for industrial process control. WirelessHART is organized according to a structure of five layers: physical, link, network, transport, and application layer. The physical layer is specified according to the physical layer of the IEEE 802.15.4 standard. The link layer implements medium access control, which is based on the Time Division Multiple Access (TDMA) technique, and error correction. The network layer is the core of the WirelessHART and is responsible for routing, topology control, end-to-end security and session management. The WirelessHART network layer supports the deployment of self-healing and self-organizing mesh networks. On top of the network layer, the transport layer provides end-to-end reliability and flow control. Finally, the application layer relies on command-response based applications to allow data exchange between the devices and the gateway [22, 23].

Z-wave is a low power protocol architecture for automation of homes and small businesses. It was developed by ZenSys, and it is promoted by Z-Wave Alliance. Z-wave devices operate in the 900 MHz band. Data rates are up to 40 kbps and the maximum distance between two nodes is about 30 meters. Z-wave medium access control layer relies on CSMA/CA technique and has an optional retransmission mechanism for reliability. A Z-wave network has two types of devices: controllers and slaves. Controllers send commands and requests for slaves, which execute the commands or send replies to the controllers. Routing in Z-wave networks is performed by controllers, which keep a table with information about the entire topology. When a controller sends a packet, it includes information about the path that must be followed in the packet [19, 24].

LoRaWAN is a technology developed by the LoRa Alliance, a non-profit foundation. Unlike technologies such as IEEE 802.15.4, BLE, WirelessHART, and Z-Wave, which aim to operate at short distances, LoRaWAN is a technology for Low Power Wide Area Networks (LPWANs). In LoRaWAN networks, end devices communicate to a central network server through a gateway. End devices are directly connected to gateways through single hop wireless links, while gateways use traditional IP networks to connect to central servers. A single end device may transmit data for multiple gateways, and the network server is responsible for discarding redundant packets. Data rate per terminal ranges from 0.3 kbps to 50 kbps. Covered distance in urban areas may range from 2 km to 5 km, while in rural areas it may range from 10 km to 15 km. [25, 26].

3. Relevant Reviews

Over the recent years, several review articles have been published on IDSs for technologies related to IoT such as mobile ad hoc networks (MANETs) [27, 28, 29], wireless sensor networks (WSNs) [30, 31, 32], cloud computing [33] and cyber-physical systems [34].

Mishra et al. [27] point out that applying the research of wired networks to wireless networks is not an easy task due to the fundamental architectural differences, especially the lack of fixed infrastructure. The authors argue that the type of intrusion response for wireless ad hoc networks depends on the type of intrusion, the network protocols and applications in use, and the confidence in the evidence. Some of the likely responses include reinitializing communication channels between nodes, identifying the compromised nodes and reorganizing the network to cease the compromised nodes and initiating a re-authentication request to all nodes in the network. The authors also present a detailed discussion of seven IDSs proposals for MANETs according to the following methodologies: distributed anomaly detection and mobile-agent-based detection. In both cases, an IDS agent runs at each mobile node and performs local data collection and local detection. The difference between the two methodologies lies in the global detection: the distributed anomaly detection uses information from neighboring nodes to build a cooperative detection engine while the mobile-agent-based detection employs mobile agents technology for intrusion detection and response.

Anantvalee and Wu [28] present a study about network infrastructure for IDS in MANETs. The authors describe three architectures for IDS in MANETs: Distributed and Cooperative Intrusion Detection Systems (flat network infrastructure), Hierarchical Intrusion Detection Systems (multi-layered network infrastructure) and Mobile Agent for Intrusion Detection Systems (flat and multi-layered network infrastructure). Due to the nature of MANETs, the authors report that almost all of the surveyed IDSs are structured to be distributed and have a cooperative architecture. The authors also present a taxonomy of misbehaving nodes detection in MANETs concerning architecture, type of data collection, data distribution, observation, misbehavior detection, punishment and route discovery.

Kumar and Dutta [29] present an overview of intrusion detection techniques for MANETs focusing on the detection algorithms. The authors introduce a classification tree for intrusion detection techniques by the nature of processing mechanism involved in the detection method. The intrusion detection techniques are divided in statistical based, heuristics techniques based, rule based, state based, signature based, reputation based, routing information based, cross-layer based and graph theory based. For every intrusion detection technique studied, the authors propose a detailed classification of the system according to the detection technique (misuse, anomaly-based, specification or hybrid), architecture (standalone, distributed and cooperative, mobile agent-based and hierarchical IDS), time of detection (real-time or offline), routing protocol, type of attacks addressed, performance, effect of mobility, robustness, flexibility, scalability, speed, and reliability. Further, they enumerate research challenges and highlight open issues in intrusion detection for MANETs. One significant challenge is related to the dynamic environment. Both the intrusive behavior and benign behavior of users, systems, or network change over time. The IDS should be self-managed and self-configured to handle the continuous changing dynamic environment and respond more quickly to dynamically changing hardware and software sources on the network.

Farooqi and Khan [30] present a taxonomy of IDS for WSNs in terms of the way the IDS agent is deployed in the network: purely distributed (IDS agent is installed in each sensor node), purely centralized (IDS agent is installed at the base station) and distributed-centralized (IDS agent is installed in some monitor nodes). The authors also discuss the relationship between the IDS agent position in the WSN and energy consumption. They conclude that distributed-centralized IDS approach is a better fit for WSNs regarding power consumption and network complexity topology.

Abduvaliyev et al. [31] introduce a taxonomy of IDS for WSNs regarding the detection technique: misuse detection, anomaly detection, and specification-based detection. They also provide a detailed discussion of the IDS mechanisms concerning WSN structure, highlighting various vital areas that are currently underdeveloped. Some of the topics include lack of real-world implementations of IDS schemes in WSNs and developing IDS mechanisms that cope with the vision of the IoT. They also conclude that while the field of IDS for WSN has advanced significantly in the recent years, there are still various research areas (e.g. IDS architectures, the balance between accuracy and consumption of resources, better integration of underlying mechanisms) that need to be further developed.

Butun et al. [32] conduct an extensive literature review of IDS for WSNs. They present a brief survey of IDSs proposed for MANETs and investigate their applicability to WSNs. According to the authors, some IDSs would be applicable directly (two proposals), some would be applicable with significant modifications (seven proposals), while the rest would not apply to WSNs (eight proposals), simply due to the particular design requirements of WSNs. The authors also propose a comparison among the IDSs proposed for WSNs according to the network architecture and the detection technique. Finally, the work highlights the energy consumption of the IDSs due to the low power consumption requirement of WSNs.

Modi et al. [33] report several intrusions that affect availability, confidentiality, and integrity of Cloud Computing. The authors summarize and classify IDSs used in Cloud into three categories: IDS technology (Host-based intrusion detection system (HIDS), Network-based intrusion detection system (NIDS), Hypervisor based intrusion detection system and Distributed intrusion detection system (DIDS)), detection technique and network positioning. They also discuss advantages and disadvantages of each proposal and identify challenges to make Cloud Computing a trusted platform for delivering IoT services. Most of the proposed intrusion detection techniques in Cloud cannot deal with recurrent attacks in this environment such as the insider attacks and attacks on the virtual machine or hypervisor.

According to Mitchell and Chen [34], Cyber-Physical Systems (CPSs) are large-scale, geographically dispersed, federated, heterogeneous, life-critical systems that comprise sensors, actuators, and control and networking components. The authors present a taxonomy of modern IDSs for CPSs based on two design dimensions: detection technique and audit material (host based or network based). First, they provide a comprehensive analysis of the differences between traditional IDSs and IDSs for CPSs, which include dealing with physical process monitoring, sophisticated attacks, and legacy technology. Then, the authors summarize existing work in IDSs for CPSs design in terms of CPS application, attack type, audit features and dataset quality. The authors also enumerate research challenges and highlight future trends in the area of IDSs for CPSs.

Although these articles primarily focus on the design of IDSs for several IoT related elements, none of them provide a study of IDS techniques specific for the IoT paradigm. In this survey article, we discuss placement strategies and detection methods of IDSs designed specifically for IoT. We also present common threats for IoT security and how IDSs might be used to detect them. Furthermore, we present a review of the common validation strategies employed in the intrusion detection methods for IoT and discuss open research issues and future trends.

4. Intrusion Detection in Internet of Things

In this section, we conduct a literature review of IDS proposals for IoT. Every work was classified regarding the following attributes: IDS placement strategy, detection method, security threat and validation strategy. Figure 1 illustrates the proposed taxonomy for Intrusion Detection in IoT and Table 1 summarizes the investigated efforts to design IDS for IoT ("-" stands for an unspecified attribute).

4.1. IDS placement strategies

Before starting to discuss the placement strategies for IDSs in IoT networks, it is necessary to present an overview of the IoT networks architecture and the main elements that are part of it.

In recent years, researchers have shown different architectures for IoT [53, 54, 55, 56], which are strongly associated with the collection, transmission, and processing, management and utilization phases presented in Section 2.2. Although these proposals vary slightly in some aspects, they similarly organize IoT scenarios in three broad domains: physical domain, network domain, and application domain. The physical domain is related to the collection phase and includes devices that sense and act over the physical environment, often composing an LLN. The network domain, which relies on transmission phase, gathers conventional network solutions and protocols to carry the data from the physical environment to applications and users. A border router is necessarily placed between the physical and the network domains to integrate the LLN protocols with the conventional protocols of the network domain. Finally, the application domain includes the interfaces that allow users to handle the objects at the physical domain.

In IoT networks, the IDS can be placed in the border router, in one or more dedicated hosts, or in every physical object. The advantage of placing the IDS in the border router is the detection of intrusion attacks from the Internet against the objects in the physical domain. However, an IDS in the border router might generate communication overhead between the LLN nodes and the border router due to the IDS frequent querying of the network state. Placing the IDS in the LLN nodes might decrease the

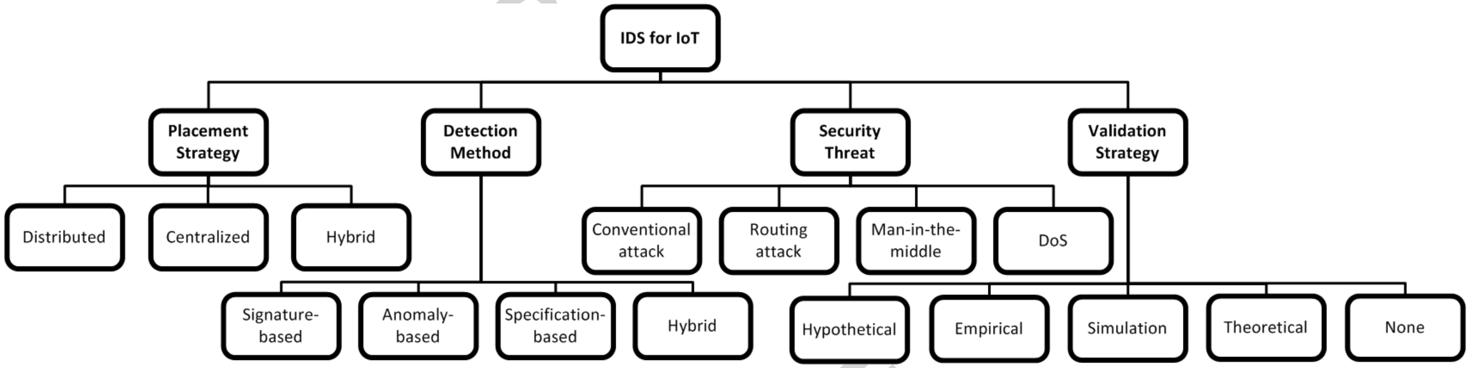

Figure 1: Taxonomy of IDSs for IoT.

Key references	Placement strategy	Detection method	Security threat	Validation strategy
Cho et al. [35]	Centralized	Anomaly-based	Man-in-the-middle	Simulation
Liu et al. [36]	-	Signature-based	-	None
Le et al. [37]	Hybrid	Specification-based	Routing attack	None
Misra et al. [38]	-	Specification-based	DoS	Simulation
Kasinathan et al. [39]	Centralized	Signature-based	DoS	Empirical
Wallgren et al. [40]	Centralized	-	Routing attack	Simulation
Raza et al. [41]	Hybrid	Hybrid	Routing attack	Simulation
Gupta et al. [42]	-	Anomaly-based	-	None
Kasinathan et al. [43]	Centralized	Signature-based	-	Hypothetical example
Amaral et al. [44]	Hybrid	Specification-based	-	Empirical
Oh et al. [45]	Distributed	Signature-based	Multiple conventional attacks (Snort and Clamav database)	Empirical
Lee et al. [46]	Distributed	Anomaly-based	DoS	Simulation
Krimmling & Peter [47]	-	Hybrid	Routing attack and Man-in-the-middle	Simulation
Cervantes et al. [48]	Distributed	Hybrid	Routing attack	Simulation
Summerville et al. [49]	-	Anomaly-based	Conventional	Empirical
Thanigaivelan et al. [50]	Hybrid	Anomaly-based	-	None
Le et al. [51]	Hybrid	Specification-based	Routing attack	Simulation
Pongle & Chavan [52]	Hybrid	Anomaly-based	Routing attack	Simulation

Table 1: Summary of the IDS for IoT literature

communication overhead associated with network monitoring, but requires more resources (processing, storage, and energy) from them [40]. This might be a problem due to resource limitations of LLN nodes. Distributing IDS agents across some dedicated nodes might be a solution to meet the requirements for less monitoring traffic and more processing capacity. However, this solution demands the organization of the network into different regions, what might be a challenge.

The following subsections describe three possible placement strategies for IDSs, presenting advantages and drawbacks of each one.

4.1.1. Distributed IDS placement

In this placement strategy, IDSs are placed in every physical object of the LLN. The IDS deployed in each node must be optimized since these nodes are resource-constrained. To address this issue, Oh et al. [45] and Lee et al. [46] proposed distributed lightweight IDSs. Oh et al. defined a lightweight algorithm to match attack signatures and packet payloads. They suggested two techniques, auxiliary shifting and early decision, which has an objective to decrease the number of matches needed for detecting attacks. They compared their approach with the Wu–Manber (WM) algorithm, which is one of the fastest pattern-matching algorithms. According to the authors, the proposed method is faster than the Wu-Manber algorithm, running on a resource-constrained platform. Lee et al. in turn suggested a lightweight method that monitors the node energy consumption for detecting intrusions. By focusing only on a single node parameter, the authors attempted to minimize the computational resources needed for intrusion detection.

In the distributed placement, the nodes may also be responsible for monitoring their neighbors. Nodes that monitor their neighbors are referred to as watchdogs. Cervantes et al. [48] proposed a solution called INTI (Intrusion detection of Sinkhole attacks on 6LoWPAN for InterneT of ThIngs) that combined concepts of trust and reputation with watchdogs for detecting and mitigating attacks. First, nodes are classified as leader, associated or member nodes, composing a hierarchical structure. The role of each node can change over time due to the network reconfiguration or an attack event. Then, each node monitors a superior node by estimating its inbound and outbound traffic. When a node detects an attack, it broadcasts a message to alert the other nodes and to isolate the attacker. The authors did not discuss the impact of the solution in low capacity nodes.

4.1.2. Centralized IDS placement

In the centralized IDS placement, the IDS is placed in a centralized component, for example, in the border router or a dedicated host. All the data that the LLN nodes gather and transmit to the Internet cross the border router as well as the requests that Internet clients send to the LLN nodes. Therefore, the IDS placed in a border router can analyze all the traffic exchanged between the LLN and the Internet [41, 30]. However, analyzing the traffic that traverses the border router is not enough to detect attacks that involve only nodes within the LLN. Then, researchers must design IDSs that can monitor the traffic exchanged between LLN nodes, without ignoring the impact that this monitoring activity may have on low capacity nodes operation. Also, the centralized IDS may have difficulty in monitoring the nodes during an attack that compromises part of the network.

Cho et al. [35] proposed a solution for analyzing the packets that pass through the border router between the physical and the network domain. The work focused on botnet attacks, what explains their choice for monitoring only the border router traffic. Kasinathan et al. [39, 43] also employed the centralized placement, but they took into consideration the IDS protection against a DoS (Denial of Service) attack. This way, the authors decided to deploy the IDS analysis engine and the IDS reporting system in a powerful dedicated host. They deployed the IDS sensors in the LLN, which were responsible for sniffing the network traffic and sending this data to the IDS analysis engine. The IDS dedicated host is wire connected to the IDS sensors, avoiding the transmission of IDS data and network regular data in the same wireless network. Therefore, if a DoS attack degrades the wireless transmission quality, IDS data transmission would not be affected.

Wallgren et al. [40] proposed a centralized approach in which the IDS is placed in the border router. The objective of the proposed solution is to detect attacks within the physical domain. Then, instead of monitoring the traffic crossing the border router, the authors suggested a heartbeat protocol. According to the proposed protocol, the border router sends ICMPv6 echo requests to all LLN nodes at regular intervals and expects the responses to detect attacks or availability issues. Although the solution creates additional traffic in the network, the authors showed in the experiments that the LLN nodes would not need to allocate additional memory to run the heartbeat algorithm, and the energy overhead was minimal.

4.1.3. Hybrid IDS placement

Hybrid IDS placement combines concepts of centralized and distributed placement to take advantage of their strong points and avoid their drawbacks.

The first approach for hybrid placement organizes the network into clusters or regions, and only the main node of each cluster hosts an IDS instance. Then, this node becomes responsible for monitoring the other nodes of its cluster. At first sight, this definition seems to match Cervantes et al.'s work [48], presented in Section 4.1.1 as an example of distributed placement. Although Cervantes et al.'s approach organized the networks into clusters and elected cluster leaders, any node, being a leader or not, could monitor its neighbor. In hybrid approaches, only selected nodes, which are often more robust, host IDS instances. Hence, hybrid placement IDSs may be designed to consume more resources than distributed placement IDSs.

Amaral et al. [44] proposed an IDS for IoT using this approach. In this work, selected nodes in the network host an IDS. These selected nodes (watchdogs) aim to identify intrusions by eavesdropping the exchanged packets in their neighborhood. The watchdog decides whether a node is compromised according to a set of rules. Each watchdog has a particular set of rules because each component in the network might have a different behavior. For example, a border router usually experiences higher rates of messages than a regular node. The advantage of this approach relies on allowing the construction of a different set of rules for each area of the network.

Le et al. [37] also followed the approach of organizing the network in regions. They used the hybrid placement by building a backbone of monitor nodes. With a minimal number of monitor nodes that cover the whole network, a monitor node sniffs the communication from its neighbors and defines whether a node is compromised. This solution has the advantage of not generating more communication overhead since the monitor nodes only sniff the transmissions among their neighbors. In a more recent work, Le et al. [51] organized the network into small clusters with a similar number of nodes. Each cluster has a cluster head, which is a node that had direct communication with all the cluster members. An IDS instance is placed in each cluster head which monitors the cluster members by sniffing their communication. Cluster members should report related information about itself and other neighbors to the cluster head. Even though the authors considered the cluster head might be a more powerful node, they chose to design a lightweight IDS solution.

In the second approach for hybrid placement, IDS modules are placed both in the border router and in the other network nodes. The main difference of this approach to the first one is the presence of a central component. The IDS modules in the border router are responsible for tasks that demand more resource capacity, while the IDS modules in regular nodes are usually lightweight. Raza et al. [41] proposed an IDS named SVELTE. On their work, the border router hosts process intensive IDS modules such as the one responsible for detecting intrusions by analyzing RPL network data. Network nodes are responsible for lightweight tasks such as sending RPL network data to the border router and notifying the border router about the malicious traffic they receive.

In Pongle and Chavan's work [52], network nodes are responsible for detecting changes in their neighborhood and sending information about neighbors to centralized modules, which are deployed in the border router. The centralized modules, in turn, are responsible for storing and analyzing this data to detect intrusions and identify the possible attackers. Though the IDS description might indicate an architecture that demands an intense traffic exchange to detect intrusions, the results showed that the energy overhead, the packet overhead, and the memory consumption were adequate to an environment with constrained nodes.

Thanigaivelan et al. [50] proposed an IDS that also allocates different responsibilities to the border router and the network nodes, making them work cooperatively. The IDS module in the node monitors node neighbors, detecting possible intrusions. When an event is detected, the node sends a notification to the IDS module on the border router. Then, the border router module correlates notifications from different nodes to make a final decision regarding the intrusion. Thanigaivelan et al. classified their IDS as a distributed IDS. However, the central role of the border router in taking the final decision about the intrusion detection makes the proposed IDS a hybrid approach.

4.2. Detection methods

Intrusion detection techniques are classified into four categories depending upon the detection mechanism used in the system: anomaly-based, signature-based, specification-based and hybrid.

The objective of this section is to discuss how these techniques have been used to develop IDSs for IoT.

4.2.1. Signature-based approaches

In signature-based approaches, IDSs detect attacks when system or network behavior matches an attack signature stored in the IDS internal databases. If any system or network activity matches with stored patterns/signatures, then an alert will be triggered.

Signature-based IDSs are accurate and very effective at detecting known threats, and their mechanism is easy to understand. However, this approach is ineffective to detect new attacks and variants of known attacks, because a matching signature for these attacks is still unknown [12, 57].

In [36], Liu et al. proposed a signature-based IDS that employs Artificial Immune System mechanisms. Detectors with attack signatures were modeled as immune cells that can classify datagrams as malicious (non-self element) or normal (self-element). Moreover, detectors can evolve to adapt to new conditions in the monitored environment. The paper does not discuss how this approach would be deployed in IoT networks with low capacity nodes. The computational cost of storing attack signatures and running learning algorithms might also be a problem.

Kasinathan et al. [39] integrated a signature-based IDS into the network framework developed within ebbits project[3]. Their main objective is to detect DoS attacks in 6LoWPAN-based networks. To implement the IDS, the authors adapted the Suricata[4],

[3] http://www.ebbits-project.eu/

[4] http://suricata-ids.org/

a signature-based IDS, to 6LoWPAN networks. The IDS sends the alerts to a DoS protection manager that analyzes additional information such as channel interference rate and packet dropping rate to confirm the attack. The objective of this verification is to reduce the false alarm rate. The proposed architecture was designed to allow the IDS deployment on a dedicated Linux host, avoiding problems related to low capacity nodes. However, it is not clear how the signatures database will be updated. Ref. [43] also presented a signature-based approach, extending the approach proposed in [39].

In their work, Oh et al. [45] aimed to reduce the computational cost of the comparison between packet payloads and attack signatures, since IoT nodes with low capacity may not support this process. The proposed scheme is based on a multiple pattern-detection algorithm. The idea is to skip a large number of unnecessary matching operations through auxiliary shift values. The authors evaluate the proposed algorithm using a Raspberry Pi computing unit integrating the Omnivision 5647 sensor. The main goal of the device was to capture images by the embedded sensor and to transmit these images to the central server. Three algorithms were tested using intrusion pattern sets from Snort and ClamAV. In a best case scenario, the proposed method achieved a speedup of up to 2.14 compared to the traditional pattern-matching algorithm, given restricted resources.

4.2.2. Anomaly-based approaches

Anomaly-based IDSs compare the activities of a system at an instant against a normal behavior profile and generates the alert whenever a deviation from normal behavior exceeds a threshold. This approach is efficient to detect new attacks, in particular, those attacks related to abuse of resources. However, anything that does not match to a normal behavior is considered an intrusion and learning the entire scope of the normal behavior is not a simple task. Thereby, this method usually has high false positive rates [34, 58, 59].

To construct the normal behavior profile, researchers usually employ statistical techniques or machine learning algorithms that may be too heavy for low capacity nodes of IoT networks. Therefore, anomaly-based approaches for IoT networks should take this particularity into account.

In [35], Cho et al. proposed a detection scheme for botnets using the anomaly-based method. The authors assumed that botnets cause unexpected changes in the traffic of 6LoWPAN sensor nodes. The proposed solution computes the average for three metrics to compose the normal behavior profile: the sum of TCP control field, packet length, and the number of connections of each sensor. Then, the system monitors network traffic and raises an alert when metrics for any node violate the computed averages.

Gupta et al. [42] proposed an architecture for a wireless IDS. According to the proposed architecture, the IDS would apply Computational Intelligence algorithms to construct normal behavior profiles for network devices. For each different IP address assigned to a device, there would be a distinct normal behavior profile. The authors did not consider the possibility of deploying the proposed IDS in networks with low capacity devices.

In [46], Lee et al. assumed the energy consumption as a parameter to analyze nodes behavior. They defined models of regular energy consumption for mesh-under routing scheme and route-over routing scheme. Then, each node monitors its energy consumption at a sampling rate of 0.5 seconds. When the energy consumption deviates from the expected value, the IDS classifies the node as malicious and removes it from the route table in 6LoWPAN. The authors claimed that it is a lightweight approach, specifically developed for low capacity networks. However, they did not present results related to false positive rates, which are necessary to take more precise conclusions about the approach.

Summerville et al. [49] developed a deep-packet anomaly detection approach that aims to run on resource constrained IoT devices. The authors argue that small IoT devices use few and relatively simple protocols, resulting in network payloads that are highly similar. Based on this idea, they use a technique called bit-pattern matching to perform feature selection. Network payloads are treated as a sequence of bytes, and the feature selection operates on overlapping tuples of bytes, called n-grams. A match between a bit-pattern and an n-gram occurs when the corresponding bits match in all positions. The authors propose an experimental evaluation using two Internet-enabled devices and the false-positive rates for the four attack types (worm propagation, tunneling, SQL code injection, and directory traversal attacks) were very low.

Thanigaivelan et al. [50] briefly introduced a distributed internal anomaly detection system for IoT. The principle of the proposed IDS is to look for any discrepancies in the network by monitoring the characteristics of one-hop neighbor nodes such as packet size and data rate. According to the authors, the system learns and derives the normal behaviors from the monitored information. However, no details about the method used to construct the normal behavior profile are provided. It is also unclear how the detection algorithm would work on IoT low capacity nodes.

Pongle and Chavan [52] presented an IDS designed to detect wormhole attacks in IoT devices. The authors assume that the wormhole attack always leaves its symptoms on the system, for example, a high number of control packets are exchanged between the two ends of the tunnel, or a high number of neighbors get formed after a successful attack. Using this logic, the authors propose three algorithms to detect such anomalies in the network. According to their experimentation, the system achieved a true positive rate of 94% for wormhole detection and 87% for detecting both the attacker and the attack. However, no details of false positive rates are provided. The authors also performed a study on power and memory consumption of the nodes. Apparently, the proposed system is suitable for IoT devices, since its power and memory consumption are low. On the other hand, the achieved results should be compared to the literature for establishing a baseline between them.

4.2.3. Specification-based approaches

Specification is a set of rules and thresholds that define the expected behavior for network components such as nodes, protocols, and routing tables. Specification-based approaches detect intrusions when network behavior deviates from specification definitions. Therefore, specification-based detection has the same purpose of anomaly-based detection: identifying deviations from normal behavior. However, there is one important difference between these methods: in specification-based approaches, a human expert should manually define the rules of each specification [34, 44, 60]. Manually defined specifications usually provide lower false positive rates in comparison with the anomaly-based detection [34, 44, 60]. Besides, Specification-based detection systems do not need a training phase, since they can start working immediately after specification setup [44]. However, manually defined specifications may not adapt to different environments and could be time-consuming and error-prone [34, 44, 60].

Misra et al. [38] presented an approach to prevent IoT middleware from DDoS (Distributed Denial of Service) attacks. To detect the attacks, the maximum capacity of each middleware layer is specified. When the number of requests to a layer exceeds the specified threshold, the system generates an alert.

In [37], Le et al. proposed other specification-based approach, focused on detecting RPL attacks. They specified the RPL behavior in a finite state machine, which is used to monitor the network and detect malicious actions. This work is extended in [51] where the authors use simulation trace files (Contiki-Cooja platform) to generate the finite state machine for the RPL protocol. This profile was transformed into a set of rules applied for checking monitoring data from the network nodes. According to their experimentation, the true positive rates are very high and in some cases could reach 100% while the false positive rates are low, varying from 0 to 6.78%. Besides, the proposed scheme has an energy overhead of 6.3% when compared to a typical RPL network.

Amaral et al. [44] proposed a specification-based IDS that allows the network administrator to create rules for attack detection. When one of these rules is violated, the IDS sends an alert to the Event Management System (EMS). The EMS runs on a node without resource constraints to correlate the alerts for different nodes in the network.

The success of Misra et al. [38] and Amaral et al. [44] approaches strongly depends on the expertise of the network administrator, which is a characteristic of the specification-based method. Wrong specifications may cause excessive false positives and false negatives, representing a considerable risk to network security.

4.2.4. Hybrid approaches

Hybrid approaches use concepts of signature-based, specification-based and anomaly-based detection to maximize their advantages and minimize the impact of their drawbacks.

SVELTE is a hybrid IDS that Raza et al. proposed in [41]. The objective of this hybrid IDS is to offer a satisfactory trade-off between storage cost of the signature-based method and computing cost of the anomaly-based method. In [47], Krimmling and Peter tested anomaly and signature-based IDSs using the IDS evaluation framework that they proposed. The results showed that each approach failed in detecting some kinds of attacks. According to the authors, a combination of these approaches could address a wider range of attacks with a single IDS. INTI IDS, proposed by Cervantes et al. [48] for detection and isolation of sinkhole attacks, combines anomaly-based concepts to monitor the exchange of packets between nodes and specification-based method to extract two kinds of node evaluation: reputation and trust. Values vary between 0 and 1. When the reputation or trust values are above 0.5, the node is assumed as good. INTI is evaluated and compared to SVELTE regarding its effectiveness and efficiency to mitigate sinkhole attacks. The authors proposed a simulation scenario and the results show that INTI achieves a sinkhole detection rate up to 92% in a fixed scenario and 75% in a mobile scenario. Moreover, INTI showed a low rate of false positives and negatives than SVELTE in both scenarios.

4.3. Security threats

The objective of this subsection is to discuss how different attack types have been addressed in the IDS proposals for IoT. Enabling IoT solutions involves a composition of several technologies, services, and standards, each one with its security and privacy requirements. With this in mind, it is reasonable to assume that the IoT paradigm has at least the same security issues as mobile communication networks (e.g., WSNs), cloud services and the Internet. However, as noted by [4], traditional security countermeasures, and privacy enforcement cannot be directly applied to IoT technologies due to three fundamental aspects: the limited computing power of IoT components, the high number of interconnected devices, and sharing of data among objects and users.

One example of how IoT devices are susceptible to attacks is described in [7]. The authors studied the network activity of three IoT devices (the Phillips Hue lightbulb, the Belkin WeMo power switch, and the Nest smoke alarm), and demonstrated the ease with which security and privacy can be compromised for these devices. For the Phillips Hue lightbulb, the authors managed to discover a flaw in the request/response message exchange between the bridge (a wireless router, for example) and the Phillips Hue App. The communication between them is in plain text, allowing the attacker to discover the whitelisted usernames and the bridge IP address. The attacker can also take full control of the bridge by making HTTP PUT requests, using a Python code developed by the authors.

According to Kolias et al. [61], the fast productization of IoT technologies might leave IoT networks vulnerable to security and privacy risks. The authors discovered several security vulnerabilities by building IoT use-cases using popular commercial off-the-shelf products and services. The authors assembled a smart watering system composed of a component that provided environmental readings, a module that implemented user decisions, and a unit that connected the user to the rest of the scheme. They also used a single-board computer (Arduino Uno) to execute all the sensing and actuating functionality and a Web application. Some points of failure identified by the authors are insecure Web application counterparts, leading to XSS and SQL injection attacks and insecure wireless communications. As an example, the authors describe the following attack: an intruder can create a software-enabled access point (SoftAP), bearing the same service set identifier (SSID) as the real network, but without protection. Then, it can temporarily shut down all IoT devices by spoofing broadcast deauthentication packets. At this point, IoT devices will attempt to reconnect to the SoftAP that has the same SSID and the strongest signal. The authors argue that advanced OSs might avoid the attack, but the less feature-rich OSs of many IoT devices will not understand the difference and will connect to the SoftAP forged by the attacker. From this point on, attackers will be able to eavesdrop on the network traffic and also send remote commands to the IoT devices.

Of course, IoT technology vendors must release patches for all these vulnerabilities as vendors of conventional software and hardware have done for their products. Moreover, the development of new IoT products must have the protection of interactions between IoT entities as a concern. These measures will improve the security of IoT systems. However, auxiliary lines of defense like IDSs are still necessary, since attackers may attempt to explore new vulnerabilities or known ones that were not properly patched.

Garcia-Morchon et al. [62] organize the security threats that can affect IoT entities into the following categories: cloning of things, malicious substitution of things, firmware replacement, extraction of security parameters, eavesdropping, man-in-the-middle, routing attack and DoS. Next, we briefly describe these threats.

Cloning of things, malicious substitution of things, firmware replacement and extraction of security parameters can be organized according to the process phase in which the attacker acts - manufacturing (cloning of things), installing (malicious substitution of things), operation (firmware replacement and extraction of security parameters) or maintenance (firmware replacement). Cloning of things usually happens during the manufacturing process of a physical object, when an untrusted manufacturer can easily clone the physical characteristics, firmware/software, or security configuration of the object, implementing additional functionality with the cloned physical object, such as a backdoor. During the installation of a physical object, a genuine one may be maliciously substituted with a similar variant of lower quality without being detected. When a physical object is in operation or maintenance phase, new features could be provided by upgrading its firmware. An attacker may be able to exploit such a firmware upgrade by replacing the physical object with malicious software. Also during the operation phase, an attacker may exploit the physically exposed environment where the object is deployed to extract security information such as keys (e.g., device key, private key, group key) from this object or try and re-program it to serve his needs. IDS solutions for IoT surveyed in our work do not address these types of threats.

Passive attackers can eavesdrop communication channels to extract security parameters, configuration settings or application data from the information flow. A man-in-the-middle attack is performed when an attacker node modifies communications from an entity A to another entity B without both A and B noticing it. Routing attacks consist of spoofing, modifying or replaying routing information to create routing loops, attract or repel network traffic, extend or shorten source routes and so on. Other possible routing attacks include sinkhole attack, selective forwarding, wormhole attack, and sybil attack [62]. Specific attacks to RPL, primarily used in a 6LoWPAN network, are also possible such as packet fragmentation attacks and rank attacks [39]. At last, physical objects usually have tight memory and limited computation capacity so that they might be vulnerable to DoS attacks. DoS attacks can be launched in a traditional way, exhausting service provider resources and network bandwidth or targeting the wireless communication infrastructure, jamming the communication channel.

Table 2 organizes the IDS proposals for IoT according to attacks that can be detected (claimed by the authors) and the categories of each attack proposed by Garcia-Morchon et al. [62]. As noted by [62], security threats related to conventional technologies and middlewares used to build the IoT environment might also apply to IoT systems, for instance, unsecured connections over HTTP and injection of malicious code. From now on, we refer to this type of attack as a conventional attack.

Table 2: IDS proposals for IoT - Security threats.

Proposed system	Detected attacks	Category
Le et al. [37]	Topology attacks on RPL - rank attack and local repair attack	Routing attack
Raza et al. [41]	Sinkhole and selective-forwarding attacks	Routing attack
Wallgren et al. [40]	Selective-forwarding attacks	Routing attack
Cervantes et al. [48]	Sinkhole attacks	Routing attack
Pongle and Chavan [52]	Wormhole attacks	Routing attack
Le et al. [51]	Topology attacks on RPL - rank, sinkhole, neighbor, local repair, and DIS attacks	Routing attack
Krimmling and Peter [47]	Simple routing attacks (replay, drop and insertion) and bit flip, byte change and field change combined with a routing attack to simulate a man-in-the-middle	Routing attack and Man-in-the-middle
Oh et al. [45]	Intrusion pattern sets from Snort and ClamAV	Conventional attack
Summerville et al. [49]	Worm propagation, tunneling, SQL code injection, and directory traversal attacks	Conventional attack
Cho et al. [35]	Botnet on 6LoWPAN	Man-in-the-middle
Misra et al. [38]	DDoS	DoS
Kasinathan et al. [39]	IPv6 UDP flooding attack	DoS
Lee et al. [46]	DoS detection using an energy consumption model	DoS

As shown in Table 2, IDS proposals for IoT can be divided into two big groups: methods to detect routing attacks and methods do detect DoS attacks. Man-in-the-middle and conventional attacks are the other threats that appear in our analysis.

Detection of routing attacks in the IoT are proposed in [37], [40], [41], [47], [48], [51], and [52]. Four of them focused only on one or, at most, two types of routing attacks: [40], [41], [48], and [52]. Wallgren et al. [40] investigated the protection capabilities of the RPL against many types of routing attacks: sinkhole, selective forwarding, hello flood, wormhole, clone ID, and sybil attacks. However, Wallgren et al.'s IDS focused only on selective forwarding attacks. Raza et al. [41] proposed an IDS to detect sinkhole and selective forwarding attacks. Cervantes et al. [48] also developed a system to detect sinkhole attacks. In their work, Cervantes et al. addressed nodes mobility and network self-repair, which are two significant contributions regarding Raza et al.'s work. Pongle and Chavan [52] proposed an IDS to detect wormhole attacks.

Le et al. [37] introduced two new topology attacks called rank and repair attacks. In a more recent work, Le et al. [51] focused on different routing attacks such as rank, sinkhole, local repair, neighbor, and DIS (DODAG Information Solicitation) attacks.

Krimmling and Peter [47] investigated how IDSs can be applied to IoT environments that use CoAP. They implemented some simple routing attacks (replay, drop, and insertion attacks) and situations such as bit flips, byte exchanges, and modifications of entire data fields that can be related to a man-in-the-middle attack. Cho et al. [35] proposed a system to detect man-in-the-middle attacks based on botnets. More specifically, in these attacks, an LLN node is compromised, becoming a bot. Then, this bot receives commands from an external controller to forge data that it forwards.

Three works discussed ways of detecting DoS attacks in the context of the IoT: [38], [39] and [46]. Misra et al. [38] used the concept of learning automata to devise a strategy for the prevention of DDoS attacks in the context of Service Oriented Architecture (SOA) for IoT. The authors defined thresholds for each network layer, and the learning automata helped to identify which packets would be discarded. A DoS detection architecture for 6LoWPAN in the form of an IDS was proposed by [39]. The system monitors the network traffic of 6LoWPAN through one or more IDS probes operating in promiscuous mode and detects the attack by using signature-based IDS detection method. A lightweight intrusion detection scheme for 6LoWPAN is developed in [46]. The system is based on analyzing energy consumption of nodes to detect possible DoS attacks.

Oh et al. [45] and Summerville et al. [49] focused on conventional attacks. Oh et al. evaluated their approach with intrusion pattern sets from Snort, a traditional open-source IDS for conventional networks, and ClamAV, an open-source anti-virus for conventional operating systems. Summerville et al. assessed the performance of their IDS with conventional attack scenarios that included worm propagation, tunneling, SQL code injection, and directory traversal attacks.

4.4. Validation Strategy

According to Balci [63], validation consists of checking that the built model behaves with satisfactory accuracy within the study objectives. There are many validation techniques, and they may be distinguished by two sources of information: experts and data. While the use of experts provides a subjective and often qualitative model validation, the use of data may allow a quantitative and more objective validation [64].

Our goal here is to investigate the validation strategy employed in the intrusion detection methods for IoT. Such criteria could be a starting point for evaluating the maturity level of this field. For this purpose, the classification of validation methods proposed by [65] is used:

- Hypothetical: hypothetical examples, having unclear relation to actual phenomena and degree of realism;
- Empirical: empirical methods, such as systematic experimental gathering of data from operational settings;
- Simulation: simulation methods of some IoT scenario;
- Theoretical: formal or precise theoretical arguments to support results.
- None: no validation methods are employed.

Scientific advances rely on reproducibility of results so that they can be independently validated and compared by repeated large-sample tests [66]. Much of the evaluation in traditional IDSs has been based on data from the experiments performed by the Lincoln Laboratory/DARPA in 1998 and 1999. This effort is considered the most comprehensive evaluation of research on IDSs that has ever been performed [67]. While several works criticize and point out that this is a very outdated dataset, unable to accommodate the latest trend in attacks [68], [69] and [70], having an evaluation dataset is crucial to learn about the correctness of a model.

Out of the 18 investigated works, 4 conducted the validation using empirical methods and operational settings [39, 45, 44, 49]. In all cases, the authors developed unique physical testbeds using a combination of specific IoT software/hardware components such as TinyOS, Raspberry Pi, Contiki and sensors to evaluate their proposals. Refs. [35, 38, 41, 40, 47, 46, 48, 51, 52] used simulation as their validation strategy. Again, different network configurations and tools simulators are used: Cooja [41, 40, 52, 51], OMNeT [47] and Qualnet [46]. Besides, it is unclear which simulators tools were used in [35] and [38]. Finally, one work was validated using a simple hypothetical example [43] and no validation efforts were found in 4 papers [37, 36, 42, 50].

The results show that there are no standardized validation efforts for intrusion detection in IoT: evaluation testbeds are created with soil purposes, simulation and software/hardware tools are chosen without clear reasons, and models to detect similar threats (e.g., DoS) are validated using completely different network parameters [35] and [46]. It is also important to note that only one work empirically evaluated and compared different IDS schemes [48]. The authors proposed an evaluation using the Cooja simulator between SVELTE and their scheme, called INTI.

5. Issues, concerns and future research directions

Research efforts in IDS for IoT are still incipient. After classifying the papers in Section 4, we observed that the proposed solutions do not investigate the strong and weak points of each possible detection method and placement strategy deeply. The authors also have focused on few attack types and IoT technologies. Finally, validation strategies are very simple, complicating the comparison and reproduction of the proposed approaches. Next, we provide a detailed view of some issues and concerns in IDS research for IoT, also highlighting possible future research directions.

Investigating pros and cons of detection methods and placement strategies. Detection method and placement strategy are important characteristics of IDSs. The 18 analyzed works do not reach a consensus on which are the most proper options for detection method and placement strategy for IDSs in IoT. Regarding detection methods, only

Krimmling and Peter [47] conducted tests to compare different approaches. They concluded that hybrid detection would be the best option. However, despite their importance, these results are not definitive. Summerville et al. [49] argue that zero-day threats and the lack of resources for a potentially large database of known attacks make the use of signature-based detection approaches unsuitable in an IoT environment. According to them, small resource constrained devices execute fewer and potentially less complex network protocols than general purpose computing platforms, making it easier to use anomaly based detection methods to identify deviations from normal behavior. However, the computational requirements for running such methods in resource constrained systems could be high. In fact, only one anomaly-based approach [52] evaluated the impact of IDS on the nodes energy consumption. Researchers should conduct more experiments to investigate the strong and weak points of each detection method in several situations and IoT applications. These experiments should show, for example, how different detection methods affect IDS properties such as attack detection accuracy, attack detection and reporting speed, energy consumption of network nodes and performance overhead [71]. For the discussion about IDS placement strategies, there is a starting point: the IDS should be able to monitor the traffic that physical objects exchange within the physical domain and the traffic that flows between physical objects and hosts on the Internet. Nodes in the physical domain of IoT systems may operate in a mesh topology, assuming other networking functions (e.g., routing). Consequently, monitoring these nodes is essential to detect, for example, routing attacks. Physical objects also deliver services to users on the Internet, which is a particularity of IoT. Detecting attacks in the traffic that flows through the boundary between the Internet and the physical domain is also very important. Based on these assumptions, researchers should propose more experiments to evaluate the pros and cons of each IDS placement strategy for different IoT applications.

Increasing attack detection range. Despite their differences, intrusion detection proposals for IoT have many similarities with intrusion detection in WSNs. One of them is related to the attack detection range. In both cases (IoT and WSNs), research efforts are focused on developing detection systems for specific attack types, especially for routing attacks and DoS. However, it is still unclear how these systems can be combined to each other to be properly employed in real world environments. There are some potential attacks against the IoT, but the proposals can detect only a few attacks at the same time. Kasinathan et al. [43] indicate that their architecture could be integrated with SVELTE, a system proposed by Raza et al. [41] and that additional attacks could be detected by developing specific modules for Suricata. However, there are not further guidelines about this subject. Therefore, the evaluation of different attack detection schemes running under the same operational settings would be a good topic of research. Energy consumption, interoperability between the schemes and the scalability are some of the features that would be studied in this context. Another issue found in the analyzed works is the absence of clear instructions for adding more attacks to the detection engine, also called extendability. Raza et al. [41] and Cervantes et al. [48] mention this fact but, again, there are not indications toward accomplishing this issue. As previously discussed, most of the analyzed papers covered only three attack types: routing attack, man-in-the-middle, and DoS. Tests in popular IoT devices used in today home environments (Phillips Hue lightbulb, Belkin WeMo power switch, and Nest smoke alarm) [7] show some examples of application attacks that do not fit in those three cited attack types. Future IDSs for IoT should expand the attack detection range and also consider the requirements of the intended application. The security level of healthcare applications might be different from the smart home domain, for instance.

Addressing more IoT technologies. 6LoWPAN is often cited as a typical IoT network technology, what may explain why most of the analyzed papers propose IDS for 6LoWPAN. However, since IoT will be used in many application domains with different technologies, development of IDSs only for 6LoWPAN is insufficient to meet the security needs of every IoT system. BLE and Z-Wave, for instance, are technologies frequently associated to IoT, but researchers have not proposed IDSs for BLE or Z-Wave based systems. CoAP is another technology that security researchers should address in future. As CoAP allows physical objects to deliver services to users on the Internet, it may be the source of several vulnerabilities. Krimmling and Peter approached the intrusion detection for applications that use CoAP in [47], but more work is necessary to deepen this discussion. Finally, IDSs that address other technologies, such as WiFi, NFC (Near Field Communication) and Bluetooth, should be studied. Household appliances that are currently available on the market use these technologies [7]. Therefore, users need to be protected against intrusions in these applications urgently.

Improving validation strategies. The most idealistic methodology for evaluating IDSs is running the system over real labeled network traces with an extensive set of intrusions [67]. One of the most popular IDS tests to date was conducted by the MIT Lincoln Laboratory and Defense Advanced Research Projects Agency (DARPA). The generated dataset includes some injected attacks at well-defined points, Windows NT audit data, process and file system information. Although this dataset is widely used by the IDS research community, their precision and ability to reveal real-world characteristics have been extensively criticized in [68], [69] and [70]. Shiravi et al. [67] propose a systematic approach to generate benchmark datasets for IDS. According to them, a qualifying dataset should have the following set of features: realistic network configuration, realistic traffic, labeled dataset, total interaction capture, full capture and multiple attack scenarios. These characteristics are suitable for traditional networks, where concepts like network perimeter and external/internal attackers could be clearly defined. The same could not be said for IoT environments. Studies should be conducted to verify whether those set of features can be applied to intrusion detection for IoT. For example, the authors simulated the natural behavior of network connected nodes by implementing a physical testbed with real live devices. User behavior was created by mimicking user activity from an operational network. Evaluating and creating similar network testbeds for IoT systems could be a viable starting point. Initiatives such as the SmartSantander [72] which is a is city-scale experimental research facility deployed in Santander, Spain, could be used as a role model towards developing robust strategies for IDS validation in IoT. Other testbeds for IoT experimentation can be found in [73].

Secure alert traffic and management. A constant concern related to IDSs is the protection of IDS communications. Conventional networks adopt management networks or Virtual Local Area Networks (VLANs) for protecting the communication between nodes and the IDS components. However, in IoT scenarios, the particular characteristics of nodes impose several difficulties for protecting IDS communication. In case weak security methods are used to protect the communication between IDS sensors and nodes, the attacker can passively monitor network traffic and decrypt the IDS traffic. Attackers can also use evasion techniques to discover channels that are not currently monitored and launch attacks on those channels. Kasinathan et al. [39, 43] suggest using a wired connection between the IDS sensors and the IDS itself. In [37], [41] and [40], the authors acknowledge the importance of protecting the IDS communication, but, in their proposals, they assume that the communication between the IoT nodes is secured. Some methods to protect communication between IoT nodes are proposed in [74] and [75] and usually involve lightweight encryption and authentication methods. Another important research topic is the Privacy-preserving Intrusion Detection (PPID) [76]. IoT nodes should avoid disclosing private information such as "being intruded or not" even when they share intrusion detection information with other parties. As discussed in [76], current literature has not seriously studied how to preserve the privacy of intrusion detection information. PPID schemes should be proposed based not only on the IDS placement strategy but also according to the IoT application domain.

Addressing further issues of IDSs. Adoption of IDSs in IoT networks may introduce new challenges for network administrators and users. In traditional networks, IDSs generate huge amounts of alerts, including many false positives and low priority alerts. Human network operators cannot manually analyze these alerts to figure out attack strategies, identify high priority alerts, discard false positives and mitigate attack consequences. IDSs for IoT systems may experience this issue as well. Therefore, researchers should propose post-processing approaches for IDS alerts in IoT systems. Alerts post processing includes techniques for alert correlation, false positives reduction and data visualization, which aid network administrators to extract useful information from huge volumes of alerts. Recent research studies published by [77, 78, 79, 80] may help security researchers developing novel post-processing techniques for IDS alerts in IoT. IDS administration is also a challenge for network administrators and users. In traditional networks, IDS installation, configuration and maintenance are complex, labor-intensive and error-prone processes. In IoT, IDSs may get even harder to manage. The most remarkable IoT aspect is its ability to transform everything in our lives, from a household appliance to a sophisticated industrial machine, into an Internet host. IoT systems will be ubiquitous and large-scaled. Therefore, IDS administration in IoT systems cannot depend on constant human intervention. To address this issue, security researchers should study the development of autonomic IDSs. Autonomic systems follow the self-* paradigm. According to this paradigm, systems can perform configuration, adaptation and repairing functions, among others, with minimal human intervention. Ashraf and Habaebi [81] present a survey about autonomic schemes for threat mitigation in IoT that may be valuable as a starting point to autonomic IDS research.

6. Conclusion

IoT has created high expectations due to its capacity of transforming physical objects of different application domains into Internet hosts. However, attackers may also take advantage of the IoT great potential as a new way to threaten users' privacy and security. Therefore, security solutions for IoT should be developed. As in traditional networks, the IDS is one of the most important security tools for IoT.

In this paper, we presented a survey about IDS research efforts for IoT. We selected 18 papers in the literature that proposed specific IDS schemes for IoT or developed attack detection strategies for IoT that could be part of an IDS. These papers were published between 2009 and 2016. We proposed a taxonomy to classify these papers, which is based on the following attributes: detection method, IDS placement strategy, security threat, and validation strategy. We observed that the research of IDS schemes for IoT is still incipient. The proposed solutions do not cover a wide range of attacks and IoT technologies. Moreover, it is not clear which detection method and placement strategies are more adequate for IoT systems. Finally, validation strategies are not well consolidated.

As future research, researchers may focus on the following issues: 1) to investigate strong and weak points of different detection methods and placement strategies; 2) to increase the attack detection range; 3) to address more IoT technologies; 4) to improve validation strategies; 5) to improve security of alert traffic and management; and 6) to develop further applications such as alert correlation and autonomic management systems.

References

[1] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497–1516.

[2] I. Lee, K. Lee, The internet of things (IoT): Applications, investments, and challenges for enterprises, Business Horizons 58 (4) (2015) 431 – 440.

[3] J. Bradley, J. Barbier, D. Handler, Embracing the Internet of Everything to capture your share of $14.4 trillion, Tech. rep., Cisco White Paper (2013).

[4] S. Sicari, A. Rizzardi, L. Grieco, A. Coen-Porisini, Security, privacy and trust in Internet of Things: The road ahead, Computer Networks 76 (0) (2015) 146 – 164.

[5] D. Singh, G. Tripathi, A. J. Jara, A survey of Internet-of-things: Future vision, architecture, challenges and services, in: Internet of Things (WF-IoT), 2014 IEEE World Forum on, IEEE, 2014, pp. 287–292.

[6] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Computer Communications 54 (2014) 1–31.

[7] S. Notra, M. Siddiqi, H. Gharakheili, V. Sivaraman, R. Boreli, An experimental study of security and privacy risks with emerging household appliances, in: Communications and Network Security (CNS), 2014 IEEE Conference on, 2014, pp. 79–84.

[8] J. P. Anderson, Computer security threat monitoring and surveillance, Tech. rep., Technical report, James P. Anderson Company, Fort Washington, Pennsylvania (1980).

[9] R. Pantoni, C. Fonseca, D. Brandão, Street lighting system based on wireless sensor networks, in: M. Eissa (Ed.), Energy Efficiency - The Innovative Ways for Smart Energy, the Future Towards Modern Utilities, INTECH Open Access Publisher, 2012, Ch. 16, pp. 337–352.

[10] P. Elejoste, I. Angulo, A. Perallos, A. Chertudi, I. J. G. Zuazola, A. Moreno, L. Azpilicueta, J. J. Astrain, F. Falcone, J. Villadangos, An easy to deploy street light control system based on wireless communication and LED technology, Sensors 13 (5) (2013) 6492–6523.

[11] G. Shahzad, H. Yang, A. W. Ahmad, C. Lee, Energy-Efficient Intelligent Street Lighting System Using Traffic-Adaptive Control, IEEE Sensors Journal 16 (13) (2016) 5397–5405.

[12] J. Vacca, Computer and information security handbook, Morgan Kaufmann, Amsterdam, 2013.

[13] A. Patel, Q. Qassim, C. Wills, A survey of intrusion detection and prevention systems, Information Management & Computer Security 18 (4) (2010) 277–290.

[14] A. Meddeb, Internet of things standards: who stands out from the crowd?, IEEE Communications Magazine 54 (7) (2016) 40–47.

[15] IEEE Standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (LR-WPANs), IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006) (2011) 1–314.

[16] J. Hui, P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, RFC 6282 (Proposed Standard) (Sep. 2011).
URL http://www.ietf.org/rfc/rfc6282.txt

[17] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550 (Proposed Standard) (Mar. 2012).
URL http://www.ietf.org/rfc/rfc6550.txt

[18] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol (CoAP), RFC 7252 (Proposed Standard) (Jun. 2014).
URL `http://www.ietf.org/rfc/rfc7252.txt`

[19] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Communications Surveys Tutorials 17 (4) (2015) 2347–2376.

[20] A. Banks, R. Gupta, MQTT version 3.1.1, OASIS Standard (2014).

[21] C. Gomez, J. Oller, J. Paradells, Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology, Sensors 12 (9) (2012) 11734.

[22] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, WirelessHART: Applying wireless technology in real-time industrial process control, in: 2008 IEEE Real-Time and Embedded Technology and Applications Symposium, 2008, pp. 377–386.

[23] A. N. Kim, F. Hekland, S. Petersen, P. Doyle, When HART goes wireless: Understanding and implementing the WirelessHART standard, in: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, 2008, pp. 899–907.

[24] C. Gomez, J. Paradells, Wireless home automation networks: A survey of architectures and technologies, IEEE Communications Magazine 48 (6) (2010) 92–101.

[25] A technical overview of LoRa and LoRaWAN, White paper, LoRa Alliance (2015).

[26] H. G. S. Filho, J. P. Filho, V. L. Moreli, The adequacy of LoRaWAN on smart grids: A comparison with RF mesh technology, in: 2016 IEEE International Smart Cities Conference (ISC2), 2016, pp. 1–6.

[27] A. Mishra, K. Nadkarni, A. Patcha, Intrusion detection in wireless ad hoc networks, IEEE Wireless Communications 11 (1) (2004) 48–60.

[28] T. Anantvalee, W. Jie, A Survey on Intrusion Detection Systems in Mobile Ad Hoc Networks, Wireless Network Security 2 (2007) 159–180.

[29] S. Kumar, K. Dutta, Intrusion detection in mobile ad hoc networks: techniques, systems, and future challenges, Security and Communication Networks 9 (14) (2016) 2484–2556.

[30] A. Farooqi, F. Khan, Intrusion detection systems for wireless sensor networks: A survey, in: D. Ślęzak, T.-h. Kim, A.-C. Chang, T. Vasilakos, M. Li, K. Sakurai (Eds.), Communication and Networking, Vol. 56 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2009, pp. 234–241.

[31] A. Abduvaliyev, A. S. K. Pathan, Z. Jianying, R. Roman, W. Wai-Choong, On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks, Communications Surveys & Tutorials, IEEE 15 (3) (2013) 1223–1237.

[32] I. Butun, S. D. Morgera, R. Sankar, A Survey of Intrusion Detection Systems in Wireless Sensor Networks, IEEE Communications Surveys & Tutorials 16 (1) (2014) 266–282.

[33] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in Cloud, Journal of Network and Computer Applications 36 (1) (2013) 42–57.

[34] R. Mitchell, I.-R. Chen, A survey of intrusion detection techniques for Cyber-Physical Systems, ACM Computing Surveys (CSUR) 46 (4) (2014) 55.

[35] E. Cho, J. Kim, C. Hong, Attack model and detection scheme for botnet on 6LoWPAN, in: C. Hong, T. Tonouchi, Y. Ma, C.-S. Chao (Eds.), Management Enabling the Future Internet for Changing Business and New Computing Services, Vol. 5787 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 515–518.

[36] C. Liu, J. Yang, Y. Zhang, R. Chen, J. Zeng, Research on immunity-based intrusion detection technology for the Internet of Things, in: Natural Computation (ICNC), 2011 Seventh International Conference on, Vol. 1, 2011, pp. 212–216.

[37] A. Le, J. Loo, Y. Luo, A. Lasebae, Specification-based IDS for securing RPL from topology attacks, in: Wireless Days (WD), 2011 IFIP, 2011, pp. 1–3.

[38] S. Misra, P. Krishna, H. Agarwal, A. Saxena, M. Obaidat, A learning automata based solution for preventing Distributed Denial of Service in Internet of Things, in: Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing, 2011, pp. 114–122.

[39] P. Kasinathan, C. Pastrone, M. Spirito, M. Vinkovits, Denial-of-service detection in 6LoWPAN based Internet of Things, in: Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on, 2013, pp. 600–607.

[40] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the RPL-based Internet of Things, International Journal of Distributed Sensor Networks 2013.

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

[42] A. Gupta, O. Pandey, M. Shukla, A. Dadhich, S. Mathur, A. Ingle, Computational intelligence based intrusion detection systems for wireless communication and pervasive computing networks, in: Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on, 2013, pp. 1–7.

[43] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, M. A. Spirito, DEMO: An IDS framework for internet of things empowered by 6LoWPAN, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS '13, ACM, New York, NY, USA, 2013, pp. 1337–1340.

[44] J. Amaral, L. Oliveira, J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, in: Communications (ICC), 2014 IEEE International Conference on, 2014, pp. 1796–1801.

[45] D. Oh, D. Kim, W. W. Ro, A malicious pattern detection engine for embedded security systems in the Internet of Things, Sensors 14 (12) (2014) 24188–24211.

[46] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, M.-C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN, in: Y.-M. Huang, H.-C. Chao, D.-J. Deng, J. J. J. H. Park (Eds.), Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Vol. 260 of Lecture Notes in Electrical Engineering, Springer Netherlands, 2014, pp. 1205–1213.

[47] J. Krimmling, S. Peter, Integration and evaluation of intrusion detection for CoAP in smart city applications, in: Communications and Network Security (CNS), 2014 IEEE Conference on, 2014, pp. 73–78.

[48] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 606–611.

[49] D. H. Summerville, K. M. Zach, Y. Chen, Ultra-lightweight deep packet anomaly detection for Internet of Things devices, in: 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), IEEE, 2015, pp. 1–8.

[50] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, J. Isoaho, Distributed internal anomaly detection system for Internet-of-Things, in: 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), 2016, pp. 319–320.

[51] A. Le, J. Loo, K. K. Chai, M. Aiash, A specification-based IDS for detecting attacks on RPL-based network topology, Information 7 (2) (2016) 25.

[52] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in Internet of Things, International Journal of Computer Applications 121 (9) (2015) 1–9.

[53] D. Bandyopadhyay, J. Sen, Internet of Things: Applications and challenges in technology and standardization, Wireless Personal Communications 58 (1) (2011) 49–69.

[54] R. Khan, S. U. Khan, R. Zaheer, S. Khan, Future Internet: The Internet of Things architecture, possible applications and key challenges., in: FIT, 2012, pp. 257–260.

[55] C. Han, J. M. Jornet, E. Fadel, I. F. Akyildiz, A cross-layer communication module for the Internet of Things, Computer Networks 57 (3) (2013) 622–633.

[56] T. ETSI, 102 690,"machine-to-machine communications (M2M); functional architecture.", European Telecommunications Standards Institute (ETSI) 20 (2011) 332.

[57] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 (1) (2013) 16 – 24.

[58] H. Debar, An introduction to intrusion-detection systems, in: Proceedings of Connect'2000, 2002, pp. 1–18.

[59] K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (IDPS), Tech. rep., National Institute of Standards and Technology, special Publication 800-94 (2007).

[60] I. Butun, S. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, Communications Surveys Tutorials, IEEE 16 (1) (2014) 266–282.

[61] C. Kolias, A. Stavrou, J. Voas, I. Bojanova, R. Kuhn, Learning Internet-of- things Security "Hands-on", IEEE Security and Privacy Jan/Feb 20 (February) (2016) 2–11. doi:10.1109/MSP.2016.4.

[62] O. Garcia-Morchon, S. Kumar, R. Struik, S. Keoh, R. Hummen, Security considerations in the IP-based Internet of Things, IETF Internet-Draft (2013).

[63] O. Balci, Verification, validation, and accreditation, in: Proceedings of the 30th Conference on Winter Simulation, WSC '98, IEEE Computer Society Press, Los Alamitos, CA, USA, 1998, pp. 41–4.

[64] D. Chrun, Model-Based Support for Information Technology Security Decision Making, Ph.D. thesis, University of Maryland (2011).

[65] V. Verendel, Quantified security is a weak hypothesis, in: Proceedings of the 2009 workshop on New security paradigms workshop - NSPW '09, 2009, pp. 37–49.

[66] M. V. Mahoney, P. K. Chan, An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection (2003) 220–237.

[67] A. Shiravi, H. Shiravi, M. Tavallaee, A. a. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Computers & Security 31 (3) (2012) 357–374.

[68] J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory, ACM Transactions on Information and System Security 3 (4) (2000) 262–294.

[69] C. Brown, A. Cowperthwaite, A. Hijazi, A. Somayaji, Analysis of the 1999 DARPA/Lincoln Laboratory IDS evaluation data with NetADHICT, in: IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, no. Cisda, 2009.

[70] J. O. Nehinbe, A critical evaluation of datasets for investigating IDSs and IPSs researches, in: Proceedings of 2011, 10th IEEE International Conference on Cybernetic Intelligent Systems, CIS 2011, 2011, pp. 92–97.

[71] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, B. D. Payne, Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices, ACM Computing Surveys 48 (1) (2015) 1–41.

[72] L. Sanchez, J. A. Galache, V. Gutierrez, J. M. Hernandez, J. Bernat, A. Gluhak, T. Garcia, SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities, 2011 Future Network & Mobile Summit (2011) 1–8.

[73] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, T. Razafindralambo, A survey on facilities for experimental Internet of Things research, IEEE Communications Magazine 49 (11) (2011) 58–67.

[74] J. Granjal, E. Monteiro, J. S. Silva, On the effectiveness of end-to-end security for Internet-integrated sensing applications, in: Green Computing and Communications (GreenCom), 2012 IEEE International Conference on, 2012, pp. 87–93.

[75] M. A. M. Isa, N. N. Mohamed, H. Hashim, S. F. S. Adnan, J. A. Manan, R. Mahmod, A lightweight and secure TFTP protocol for smart environment, in: Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on, 2012, pp. 302–306.

[76] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for Internet of Things, Journal of Network and Computer Applications 42 (0) (2014) 120 – 134.

[77] G. Spathoulas, S. Katsikas, Methods for post-processing of alerts in intrusion detection: A survey, International Journal of Information Security Science 2 (2) (2013) 64–80.

[78] G. P. Spathoulas, S. K. Katsikas, Enhancing IDS performance through comprehensive alert post-processing, Computers & Security 37 (2013) 176–196.

[79] N. Hubballi, V. Suryanarayanan, False alarm minimization techniques in signature-based intrusion detection systems: A survey, Computer Communications 49 (2014) 1–17.

[80] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, M. Rajarajan, Intrusion alert prioritisation and attack detection using post-correlation analysis, Computers & Security 50 (2015) 1–15.

[81] Q. M. Ashraf, M. H. Habaebi, Autonomic schemes for threat mitigation in internet of things, Journal of Network and Computer Applications 49 (2015) 112–127.