A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Model

Hezam Akram Abdul-Ghani, Dimitri Konstantas
Geneva School of Economics and Management
Geneva University, Switzerland

Mohammed Mahyoub
King Fahd University of
Petroleum and Minerals, KSA

***Abstract*—Internet of Things (IoT) has not yet reached a distinctive definition. A generic understanding of IoT is that it offers numerous services in many domains, utilizing conventional internet infrastructure by enabling different communication patterns such as human-to-object, object-to-objects, and object-to-object. Integrating IoT objects into the standard Internet, however, has unlocked several security challenges, as most internet technologies and connectivity protocols have been specifically designed for unconstrained objects. Moreover, IoT objects have their own limitations in terms of computation power, memory and bandwidth. IoT vision, therefore, has suffered from unprecedented attacks targeting not only individuals but also enterprises, some examples of these attacks are loss of privacy, organized crime, mental suffering, and the probability of jeopardizing human lives. Hence, providing a comprehensive classification of IoT attacks and their available countermeasures is an indispensable requirement. In this paper, we propose a novel four-layered IoT reference model based on building blocks strategy, in which we develop a comprehensive IoT attack model composed of four key phases. First, we have proposed IoT asset-based attack surface, which consists of four main components: 1) physical objects, 2) protocols covering whole IoT stack, 3) data, and 4) software. Second, we describe a set of IoT security goals. Third, we identify IoT attack taxonomy for each asset. Finally, we show the relationship between each attack and its violated security goals, and identify a set of countermeasures to protect each asset as well. To the best of our knowledge, this is the first paper that attempts to provide a comprehensive IoT attacks model based on a building-blocked reference model.**

***Keywords*—Internet of Things (IoT); building block; security and privacy; reference model**

I. Introduction

Flooding a huge number of the physical objects into the Internet at an unprecedented scale is a consequence of the Internet of Things (IoT)[1], [2]. These physical objects include, but not limited to, temperature sensors, smart phones, air conditioning, medical equipment, light bulbs, smart grid, thermostats, and TVs. Being communicated directly without human intervention, physical objects are enabled not only to monitor their environments, but also to execute shared tasks and coordinate their decisions autonomously [3].

The importance of IoT systems in different aspects of our lives has been elucidating in many research studies [4], [5] associated with fetching a networked intelligence to the physical objects world-wide, allowing them to sense and collect environmental data. Furthermore, human lives seriously depend on transportation facilities traveling us every day, civil infrastructure systems such as electric power and water, and critical healthcare infrastructure systems, all of them have created a proper environment a round us. Being tightly coupled with human beings and their environment, a single vulnerability in such systems could lead to harmful consequences, ranging from loss of privacy, physical damage, financial losses, and the possibility of endangering humans' lives [6]. To this end, IoT security is the biggest concern, for citizens, consumers, organizations, and governments wanting to protect their objects from being hacked or compromised, and must be addressed with caution [7].

Protecting IoT objects necessitates a general security framework - which is a challenging task indeed - covering all IoT assets and their corresponding possible attacks in more details. Therefore, it is absolutely essential to identify all attacks against security or privacy of IoT assets, which is the first step towards developing such framework. Having said that, IoT ecosystem, without doubt, is very complex and confusing, especially when it comes to precisely defining its main assets. Literature, however, has shown several IoT threat models based on IoT assets, none of which has introduced a comprehensive IoT attack model along with compromised security goals for such a highly intricate system [8]. This paper has investigated all possible IoT security attacks and countermeasures in each IoT asset. More particularly, it:

- states a novel IoT reference model, comprising of four main layers and their corresponding building blocks. This kind of combination would play a crucial role in identifying IoT components or assets;
- approaches a great enhancement to IoT Reference Models (RMs) since the IoT RMs currently published have not addressed IoT attacks and threats, nor described required building blocks for each layer as this paper did;
- defines a set of IoT security goals, security attacks, and a secure object;
- proposes a comprehensive IoT attack model which consists of four main phases; Mainly, it could be used to support the creation of a secure IoT-related system. Application designers willing to develop secure IoT systems could integrate mitigation techniques explained in this paper with a list of common IoT attacks targeting each asset from the early stages of IoT development; and
- establishes what type of security goals has been violated for each addressed asset, such as privacy,

confidentiality, auditability, integrity, accountability, availability, trustworthiness, and non-repudiation;

As a summary, this comprehensive survey would be useful for academic and industry based researchers, who are engaged in design of secure IoT systems by examining which attacks have been investigated, how such attacks have been handled, and which attacks remain untouched.

The rest of the work has been organized as follows. The proposed IoT reference model is given in Section II. Section III shows the related work presented in the state-of-the art. The proposed IoT attack model is discussed in details in Section IV, defining all possible attacks and their corresponding countermeasures on IoT physical objects, protocols, software, and data. Final remarks and future work conclude this paper are given in Section V.

II. IoT Reference Models

The state-of-the-art has shown that there is a lack of standardized approaches for understating and modeling IoT vision in many aspects [12].

First, differentiating between an IoT system and a non-IoT system is not absolutely clear. It is worth noting that not every system is the IoT system. In fact, when data is created under the control of objects or entities and forwarded or sent across a network, it can be considered as the IoT system [11]. Second, identifying precisely IoT assets and its components is very confusing due to the complexity of IoT ecosystem, varying from physical objects placed in the environments until their data and applications resided in the cloud. As a result of this complexity, they are susceptible to many attacks and threats [13].

Third, IoT umbrella covers different applications, development stages or cycles, middleware, fog computing, software platfrom , protocols and hardware platforms. That said, it lacks a common ground to be understood by researchers or even IoT developers [14].

Motivated by above mentioned aspects, handful of papers have been proposed to establish a common ground of understanding IoT paradigm known as IoT reference models, the most dominant of which are the following:

1) The three-layer model as shown in Fig. 1 represents IoT system as an extension to wireless sensor networks (WSN) [9]. In other words, it can be considered as an integration of WSNs and cloud severs providing several services to the users.

2) The five-layer model as depicted in Fig. 2 is relatively more structured suggested to ease the communications among several components of IoT system by dividing the complex system into a well-defined part [15], compared to the previous one.

3) The seven-layer developed by Cisco as shown in Fig. 3 extends both the three-layer and the five-layer models, trying to create a comprehensive and agreeable IoT reference model[11]. Its capability of standardization makes it ideal for IoT system.

Despite the simplicity of RMs mentioned above which breaks down the complexity of IoT ecosystem into different layers, they lack the required building blocks for their layers. An IoT building block is an essential unit or an enabler technology on which IoT system is constructed. In the context of IoT vision, building blocks are nowadays receiving more attention to provide a better understanding of IoT. Authors in [16] have described different building blocks. The most important ones are the identification, sensing, and communication.

To this end, we propose a four-layered reference model based on building blocks strategy as shown in Fig. 4, the main contributions of such model are the following:

First, the great contribution we intend to produce lies in merging each layer of IoT RMs with the required building blocks. This kind of combination would greatly help IoT stakeholders, paving the road for precisely identifying IoT components and assets. Second, we believe that building blocks would lead to a huge change in the mentality of security analysts who used to address security issues as a whole for each layer, making them address security issues of specific enablers technologies at each layer. Third, equipped with a set of building blocks at each layer, it introduces a new classification of IoT assets composed of four main components, hardware components, protocols, data at rest, and software including operating systems, firmware, and applications. These components will be used as a staring point in our attack model proposed in Section IV.

The proposed IoT reference model classifies building blocks broadly into three categories, protocols, hardware components, and software. In general, protocols fall under five building blocks in our model: connectivity, routing and networking, service discovery, communication, and web service and web servers protocols. Hardware components consists of two main building blocks: 1) sensing, which includes sensors, actuators, and RFID; and 2) hardware platforms or micro-controllers, which include different types of micro-controllers. Finally, software components are composed of four building blocks: operating systems, fog computing, middle-ware, and could solutions.

Despite the difference in the number of layers in both our model and Cisco reference models, they have the same IoT components. To validate that our reference model have covered the most important IoT components, we compare it with Cisco's reference model. Unlike Cisco's model, its first layer is specialized for physical devices, perception level in our model includes not only physical objects but also connectivity technologies represented in different layers in Cisco's model. By observing Fig. 3, it is obvious that layer 1 and layer 2 in Cisco's model have been merged into one layer in our reference model shown in Fig. 4. In contrast, the layer 3 in Cisco's model has been divided into two layers (2 and 3) in our proposed reference model. Finally, our last top layer, the cloud layer, includes four layers of Cisco model, starting from layer 4(Data accumulation) until the last one known as users and data centers.

It is worth noting that this paper is not meant to give a detailed explanation of the previous IoT reference models, because they are beyond the scope of this work. Each layer, in our model, is associated with specific tasks and functions and the data movement is often bidirectional, either from the cloud layer to the perception layer in the controlling mode or from

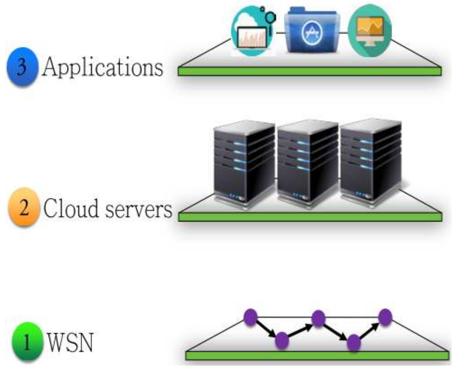


Fig. 1. Three-layer model [9].

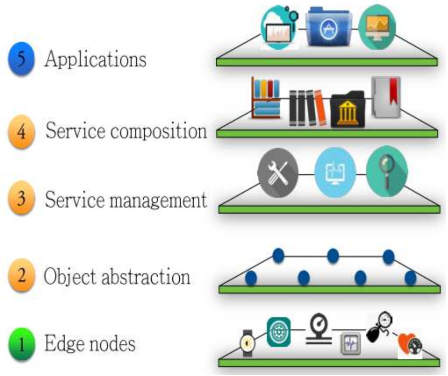


Fig. 2. Five-layer model [10].

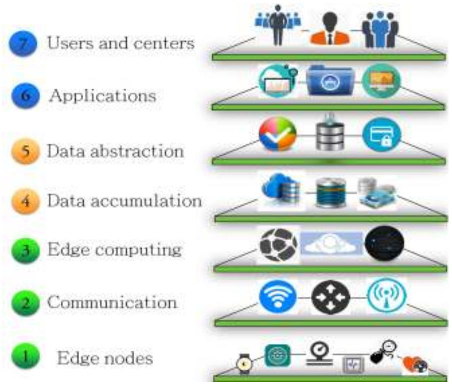
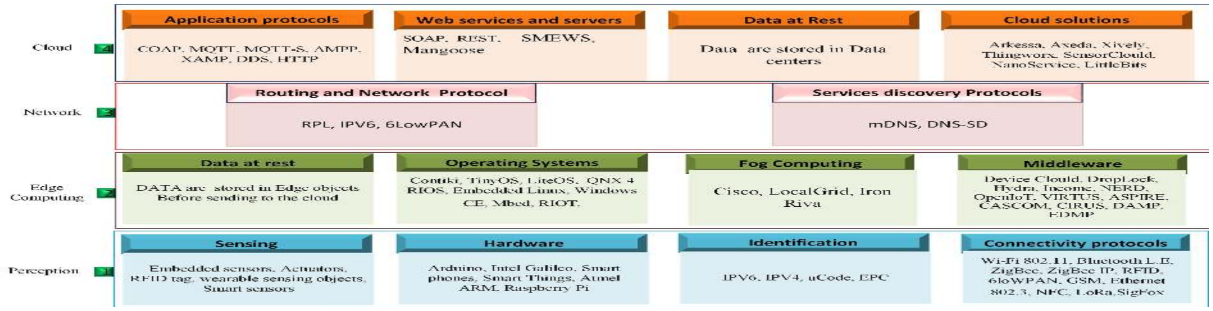


Fig. 3. CISCO's seven-layer model [11].



Cloud 4

Application protocols: COAP, MQTT, MQTT-S, AMPP, XAMP, DDS, HTTP

Web services and servers: SOAP, REST, SMEWS, Mangoose

Data at Rest: Data are stored in Data centers

Cloud solutions: Arkessa, Axeda, Xively, Thingworx, SensorClould, NanoService, LittleBits

Network 3

Routing and Network Protocol: RPL, IPV6, 6LowPAN

Services discovery Protocols: mDNS, DNS-SD

Edge Computing 2

Data at rest: DATA are stored in Edge objects Before sending to the cloud

Operating Systems: Contiki, TinyOS, LiteOS, QNX 4 RIOS, Embedded Linux, Windows CE, Mbed, RIOT.

Fog Computing: Cisco, LocalGrid, Iron Riva

Middleware: Device Clould, DropLock, Hydra, Income, NERD, OpenIoT, VIRTUS, ASPIRE, CASCOM, CIRUS, DAMP, EDMP

Perception 1

Sensing: Embedded sensors, Actuators, RFID tag, wearable sensing objects, Smart sensors

Hardware: Arduino, Intel Galileo, Smart phones, Smart Things, Atmel ARM, Raspberry Pi

Identification: IPV6, IPV4, uCode, EPC

Connectivity protocols: Wi-Fi 802.11, Bluetooth L.E, ZigBee, ZigBee IP, RFID, 6loWPAN, GSM, Ethernet 802.3, NFC, LoRa,SigFox

Fig. 4. An overview of the proposed IoT reference model and its building blocks.

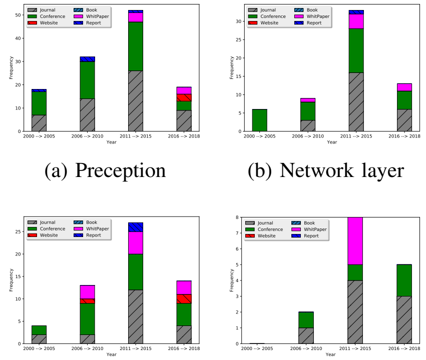
the perception layer to the cloud in the monitoring mode.

III. Related Work

The methodology followed to execute the conducted survey is illustrated here for the purpose of evaluating the research works that have been done in literature and to determine if the topic has been completely investigated. As IoT vision and its security is relatively new, our concentration was on the publications that were released in the period 2000-2017. These publications include books, journals, conferences, websites, white-papers, and reports. Fig. 5 provides the time period of this survey, and the number of published papers at each layer of the proposed reference model in that period. Perception, network, cloud, and edge computing layers are represented in Fig. 5 as a, b, c, and d bar charts, respectively.

According to Fig. 5, the key observation is that there has been an increase in the number of published papers addressing security attacks on all layers nearly from 2006 to 2015. This is, in our opinion, because of the rapid growth of IoT in a huge number of application domains such as critical infrastructure systems and the appearance of different attacks that threaten human lives and hamper the realization of IoT, which require a lot of research to be solved. As samples of our searching keywords, we have used "IoT security", "IoT countermeasures", "IoT security challenges", "attacks on IoT", "IoT security goals", "IoT privacy".

Although a huge number of research works have conducted to address security attacks of IoT systems in the state-of-the-art, handful of papers have attempted to investigate IoT attacks in a comprehensive approach, the most popular of which are the following:



(a) Preception

(b) Network layer

(c) Cloud layer

(d) Edge Computing

Fig. 5. Published paper frequency corresponding to different layers.

In [17], authors have proposed a new approach of addressing IoT threats and attacks based on a four-layer model composed of objects, interfaces, storage, and transport. Although this paper described some attacks on these levels, it did not comprehend all possible attacks. For example, firmware tampering has only been discussed as an example of physical attacks against IoT objects.

In [8], edge nodes, edge computing , and communication have been investigated in details by identifying all possible threats and attacks on each level. Moreover, this paper has introduced a set of countermeasures to mitigate such attacks. In spite of identifying possible attacks and their countermeasures in these levels, it untouched other important components in IoT systems. For example, attacks on data at rest either locally on IoT objects or remotely on the cloud have been completely uncovered.

In [18], IoT architecture has been divided into four layers: 1) application layer; 2) adaptation or support layer; 3) network layer; and 4) perception layer. Even though this approach described security threats in each layer, it lacks a comprehensive set of attacks of each layer. For example, it identified security attacks against IoT network in high level without analyzing the attacks against each network protocol. Furthermore, it uncovered the relationship between IoT attacks and their compromised security goals. IoT ecosystem presented in [19] has been divided into three levels, namely back-end system, network, and front-end sensors. The authors did not identify all attacks for each level. For example, only two types of attacks, management of the code and replacement of operator, have been identified in the network layer.

The authors in [20] divided IoT attack taxonomy into six categories, namely storage management, identity management, dynamic biding, physical threats, communication threat, and embedded security. However, security attacks have been identified at high level in each category. Only three types of attacks on communication, denial of service, spoofing, and network injection, have been introduced.

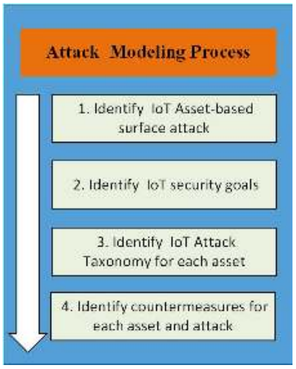

Attack Modeling Process
1. Identify IoT Asset-based surface attack
2. Identify IoT security goals
3. Identify IoT Attack Taxonomy for each asset
4. Identify countermeasures for each asset and attack

Fig. 6. An overview of the proposed attack model.

IV. Our Proposed IoT Attack Model

In this section, we will explain the proposed methodology used to create a comprehensive IoT attack model for Internet of Things. The proposed methodology for developing IoT attack model consists of four main phases. An overview of the whole methodology is shown in Fig. 6, starting from phase one, which suggests a new IoT asset-based attack surface based on the proposed building-blocked reference model in section II, down to phase four, which identifies a set of countermeasures to protect each IoT asset. The main phases of the proposed approach are described in greater detail below:

A. Identify IoT Asset-based Attack Surface

By observing the proposed IoT reference model and its companion building blocks so far, we classify IoT asset according to its threats and attacks possibilities on its building blocks into four categories: 1) physical objects; 2) protocols; 3) data; and 4) software. In other words, IoT attack surface, in the proposed IoT attack model, will be analyzed from a multi-layer perspective as shown in Fig. 7 and described as follows:

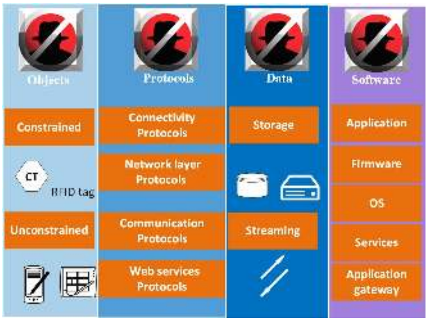


Fig. 7. IoT attack surface.

1) **Physical objects:** This category will focus on identifying all physical attacks targeting the hardware components of both constrained and unconstrained objects, resided in the perception and the edge computing layers, respectively. RFID tags, RFID readers, micro-controllers, actuators, and sensor nodes are examples of such components.
2) **Protocols:** This category is devoted to discover all potential attacks on IoT protocols. These protocols are connectivity, networking and routing, application and transport layers protocols known as communication protocols in the proposed reference model, and web services protocols. In other words, all possible attacks on IoT stack will be investigated.
3) **Data:** This category investigates the main attacks only on data at rest located either in IoT objects or in the cloud. This is because attacks on data in motion will be discussed on protocols' attacks as shown in Fig. 7.
4) **Software:** This category focuses on identifying all possible attacks on IoT software, including IoT applications located either in IoT objects or in cloud, firmware, operating systems, application gateway and services. [21].

B. Identify Security Goals and Security Attack

In this section, we will explain the two most common concepts used in IoT domain: a secure object and a security attack [8]. In order to define the secure object, it is mandatory to comprehend the security goals in which we can distinguish security. In the state-of-the-art, conventional security goals are divided into three key categories known as the CIA triad: confidentiality, integrity, and availability. Confidentiality is associated with a set of guidelines in which only authorized entities can get access to information. With the advent of Internet of things paradigm, it is important to ensure the confidentiality of IoT objects, since such objects may deal with sensitive data like medical records. Providing reliable services in the IoT requires integrity to ensure that IoT objects have received only legitimate commands and data. IoT availability ensures that IoT services are accessible only by authorized

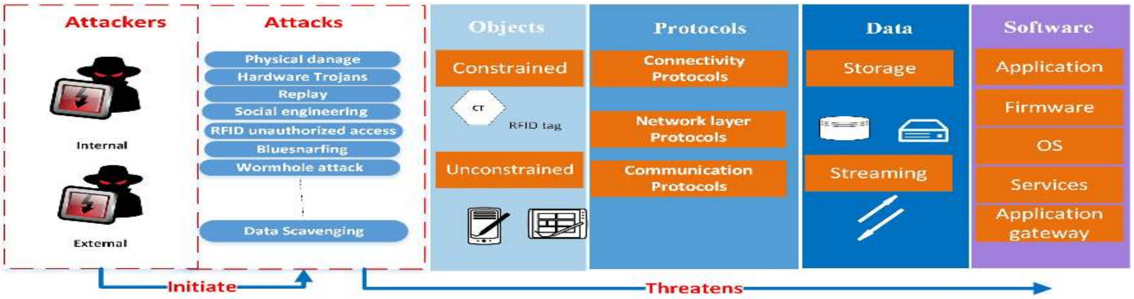


Fig. 8. IoT attack taxonomy.

TABLE I. IoT Security Requirements

Security Requirements	Definition	Abbreviations
Confidentility	The process in which only authorized objects or users can get access to the data	C
Integrity	The process in which data completeness, and accuracy is preserved	I
Non-repudiation	The process in which an IoT system can validate the incident or non-incident of an event	NR
Availability	An ability of an IoT system to make sure its services are accessible, when demanded by authorized objects or users	A
Privacy	The process in which an IoT system follows privacy rules or policies and allowing users to control their sensitive data	P
Auditability	Ensuring the ability of an IoT system to perform firm monitoring on its actions	AU
Accountability	The process in which an IoT system holds users taking charge of their actions.	AC
Trustworthiness	Ensuring the ability of an IoT system to prove identity and confirm trust in third party	TW

users or objects. In spite of the popularity of CIA-triad, authors in [22] have proven that the CIA-triad fails in addressing novel threats, which emerge in a cooperating security environment. To fill this gap, they provide a comprehensive set of security goals known as an IAS-octave, referred to the Information Assurance and Security, by investigating a large number of information systems in terms of security and assurance. Table I outlines the security goals proposed by the IAS-octave, along with their definitions and abbreviations. Once the main security goals are identified, then the secure object and the security attacks can be defined as follows:

- Secure object is an object that matches or meets all the security goals shown in Table I.
- Security attack is an attack that compromises at least one of the security goals.

C. IoT Attack Taxonomy and Countermeasures for Each Asset

The proposed IoT attack taxonomy, as depicted in Fig. 8, shows different attacks launched either internally or externally, such as hardware trojans, viruses, and physical damage [21]; the list is almost endless. Such attacks target four asset categories mentioned in the asset-based attack surface. In other words, this attack taxonomy will be analyzed from multi-layer perspectives as follows:

*1) **Physical-based attacks**:* IoT software are subjected to so many attacks. Similarly, hardware components of IoT systems, such as controllers, RFID readers, sensors, and different types of RFID tags, are vulnerable to different physical attacks, [23]. In this section, the main attacks targeting the hardware components of IoT systems as depicted in Fig. 9 are described in greater detail below.

Object replication attacks: An attacker, in this type of attack, has a capability to add physically a new object to the network. For example, a malicious object could be added by replicating object's identification. Such an attack, therefore, could cause a huge drop in the network performance. In addition to performance degradation, corrupting or misdirecting the received packets can easily be fulfilled by the malicious object, allowing the attacker to get access to sensitive data and extract the secret keys [24].

RF Interference on RFID: Sending a huge number of noise signals over radio frequencies, which are mainly used for RFID' communication, is the main goal of this type of attack [34].

Hardware Trojan: A number of research works have shown that the main security issue in an integrated circuit is its vulnerability to a hardware trojan attack. The main purpose of such attack is to maliciously modify the integrated circuit to gain access to its sensitive data and frimware. Hardware trojan attack takes place at the design phase and remains dormant until receiving a trigger or an event from its designer [35]. **Outage attacks:** In some situations, a group of IoT objects placed in unattached environments may stop operating due to either turning off their power or using much power by an attacker.

Object jamming: In spite of the benefits of using wireless technology in IoT vision, its signals can easily be hindered using a jammer [36].

Physical damage: Being deployed in unattended environments, IoT objects are significantly susceptible to physical attacks, the easiest one of which is a direct harm of its components [36].

Camouflage: Physically inserting a counterfeit edge object to a network by an attacker, to be hidden among other objects so that it could be used as the normal object to process and redirect the packets, is the main idea behind this attack [37].

Malicious node injection: To gain an unauthorized access to an IoT network, the attacker could insert a malicious object

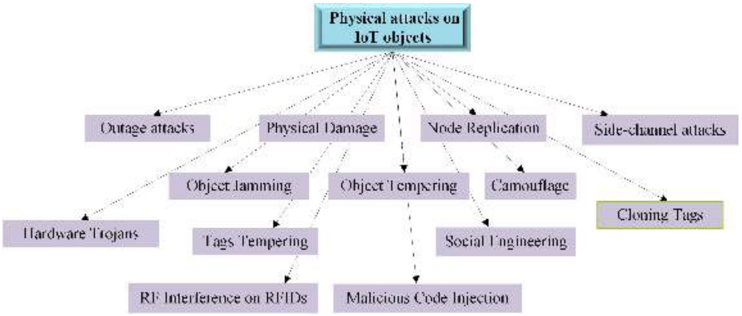


Fig. 9. Taxonomy of physical attacks against IoT objects.

TABLE II. PHYSICAL ATTACKS WITH COMPROMISED SECURITY GOALS AND COUNTERMEASURES

Physical attacks	Compromised security requirements	Countermeasures
Object tempering	ALL	Tamper proofing and self-destruction, minimizing information leakage [25] (adding randomized delay, intentionally-generated noise , balancing hamming weights , improving the cache architecture, shielding), integrating Physically Unclonable Function (PUF) into object [26]
Outage attack	A,AC,P,AU,NP	Secure physical design [27]
Object replication	ALL	Encryption, Lightweight cartographic mechanisms, Hash-based techniques [8]
Camouflage	ALL	Securing firmware update, Encryption, hash-based schemes, authentication technique [8]
Side-channel attacks	C, AU, NR, P	Blocking, isolation, kill command, sleep Command, tamper proofing and self-destruction, mimimizing information leakage, obfuscating techniques [8]
Tag cloning	ALL	Encryption, hash-based schemes [28], authentication technique, kill sleep command, isolation, blocking, distance estimation. 8. Integrating PUFs into RFID tags [29]
Social engineering	ALL	Back up techniques, education of IoT users, tamper proofing and self-destruction [30]
Physical damage	ALL	Secure physical design, tamper proofing and self-destruction [8]
Malicious Code Injection	ALL	Tamper proofing and self-destruction, IDS [8]
Hardware Trojans	ALL	Side-channel signal analysis (based on path-delay fingerprint, based on symmetry breaking, based on thermal and power, based on machine learning), trojan activation [31]
Object jamming	ALL	Spread Spectrum, priority messages, lower duty cycle, region mapping, [32]
Tag Tempering	ALL	Integrating PUFs into RFID tags, encryption, hash-based schemes [28], tamper-release layer RFID, alarm Function for active Tags[33]

among legitimate ones in the network. As a result, he could gain access to any object, insert false data to hamper messages delivery, and perhaps control the entire network. [37].

Object tampering: The possibility of accessing IoT objects physically by attackers is very high due to the fact that some IoT objects may be deployed in unfriendly environments. Therefore, such objects are vulnerable to hardware attack, the most notable ones are the extraction of cryptography keys, the alteration of operating system or firmware, and the circuit modification. The replacement of the Nest thermostat with malicious one is an example of such attacks[38].

Social engineering: Authors in [36] show that a social engineering attack can be considered as a physical attack, since an attacker could physically modify the users of IoT system in order to get their sensitive data.

Side-channel attack: Most IoT objects, for security purpose, will be integrated with some of security mechanisms such as an encryption to protect their sensitive data. Side-channel attack, however, is intended to break such mechanisms by analyzing side channel information emitted by IoT objects. Power, and time analysis attacks are some examples of such attacks [8].

Malicious code injection: An adversary, in this type of attack, could insert physically a malicious code into an IoT object. The main goal of such injection is to gain a full control of IoT system [36].

Tag cloning: Due to the deployment of tags on different objects, tags are vulnerable to physical attacks. An attacker could easily capture these tags and build a replica of them, which look like original ones to compromise a RFID system by deceiving even the RFID readers [8].

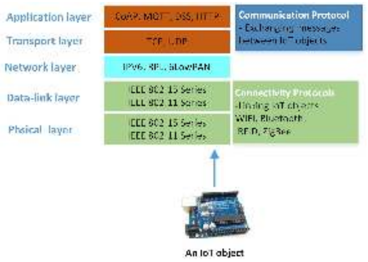


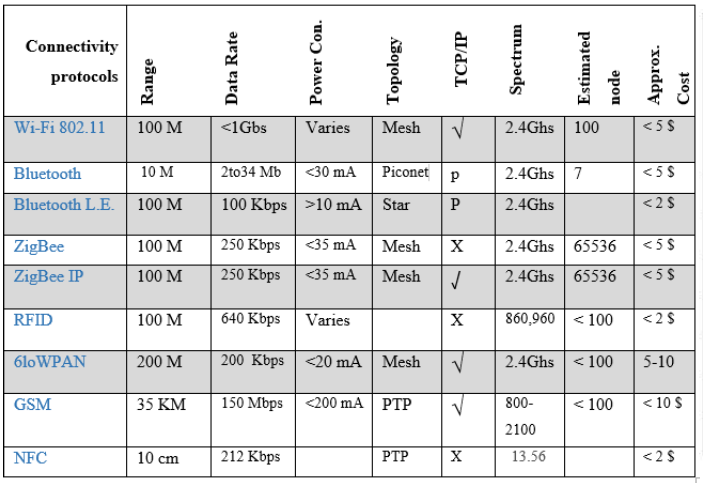
Fig. 10. The IoT stack.

An overview of all attack against hardware components, their compromised security goals, and their available defense mechanisms is presented in Table II.

2) ***Protocols-based attacks:*** Unlike traditional internet stack designed for unconstrained objects, IoT system has its own stack, as described in Fig. 10. IoT stack requires lightweight protocols such as 6loWPAN and IEEE 802.15.4 different from conventional internet protocols. For simplicity, we will classify IoT protocols into three groups known as connectivity protocols, communication protocols, and network protocols as shown in Fig. 10.

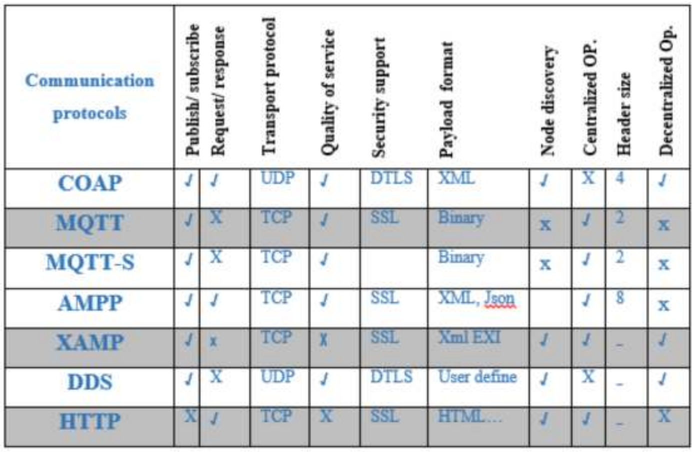
Connectivity protocols are used to link IoT objects with

TABLE III. CONNECTIVITY PROTOCOLS AND THEIR FEATURES



Connectivity protocols	Range	Data Rate	Power Con.	Topology	TCP/IP	Spectrum	Estimated node	Approx. Cost
Wi-Fi 802.11	100 M	<1Gbs	Varies	Mesh	√	2.4Ghs	100	< 5 $
Bluetooth	10 M	2to34 Mb	<30 mA	Piconet	p	2.4Ghs	7	< 5 $
Bluetooth L.E.	100 M	100 Kbps	>10 mA	Star	P	2.4Ghs		< 2 $
ZigBee	100 M	250 Kbps	<35 mA	Mesh	X	2.4Ghs	65536	< 5 $
ZigBee IP	100 M	250 Kbps	<35 mA	Mesh	√	2.4Ghs	65536	< 5 $
RFID	100 M	640 Kbps	Varies		X	860,960	< 100	< 2 $
6loWPAN	200 M	200 Kbps	<20 mA	Mesh	√	2.4Ghs	< 100	5-10
GSM	35 KM	150 Mbps	<200 mA	PTP	√	800-2100	< 100	< 10 $
NFC	10 cm	212 Kbps		PTP	X	13.56		< 2 $

TABLE IV. COMMUNICATION PROTOCOLS AND THEIR PROPERTIES



Communication protocols	Publish/ subscribe	Request/ response	Transport protocol	Quality of service	Security support	Payload format	Node discovery	Centralized OP.	Header size	Decentralized Op.
COAP	✓	✓	UDP	✓	DTLS	XML	✓	X	4	✓
MQTT	✓	X	TCP	✓	SSL	Binary	x	✓	2	x
MQTT-S	✓	X	TCP	✓		Binary	x	✓	2	x
AMPP	✓	✓	TCP	✓	SSL	XML, Json		✓	8	x
XAMP	✓	x	TCP	X	SSL	Xml EXI	✓	✓	-	✓
DDS	✓	X	UDP	✓	DTLS	User define	✓	X	-	✓
HTTP	X	✓	TCP	X	SSL	HTML…	✓	✓	-	X

each other, and implemented on data link and physical layers of IoT stack. Communication protocols are used to exchange messages between IoT objects, and implemented on application and transport layers of IoT stack.

2.1 Connectivity protocols-based attacks

IoT objects are armed with different connectivity protocols divided broadly into two main categories, wired and wireless protocols. The wired connection requires a physical medium between IoT objects, while wireless connection runs through radio waves. Both connectivity technologies have several key properties such as range, data rate, power consumption, spectrum, TCP/IP support, and topology. It is worth mentioning that this paper focuses only on wireless connectivity protocols, because most IoT objects are nowadays equipped with wireless connectivity protocols. Furthermore, attacks on wired connectivity protocols are adequately addressed in the context of traditional internet. An overview of the most popular wireless connectivity protocols and their properties is shown in Table III. In this section, the main attacks targeting the most common connectivity protocols as depicted in Fig. 11 are described in greater detail below.

2.1.1 RFID-based attacks: RFID technology facilitates automatic information exchange between tags and readers using radio waves. RFID uses the Automatic Identification and Data Capture (AIDC) technology. RFID tags, recently, have been utilized in many applications such as credit cards, assets tracking, and military [39]. However, RFID technology is vulnerable to many attacks, the most important of which are the following (Table V):

Replay: In this type of attacks, an attacker could use tags' responses to fake readers' challenges. In replay attacks, the transmitted signal between the reader and the tag is captured, documented, and repeated at a later time to the receiving object, resulting in counterfeiting the accessibility of the tag [39].

Spoofing: This type of attack happens when a malicious tag pretends to be a valid tag and obtains an unauthorized access. Spoofing attack used to eavesdrop the data coming from the valid tag, and copies the captured data to another one [39].

Tracking: Tracking attack can be considered as a direct attack against an individual or a victim. Within the next few years, companies may place RFID tags on many household items. Tracking products using RFID tags could be used to threat the privacy of human by tracking their movements, and generate an exact profile of their procurement [34].

Unauthorized access: Due to the lack of authentication in RFID system, tag could be vulnerable to an unauthorized attack. The main goal of such attack is to manipulate its sensitive data [40].

Virus: RFID system is not suitable environment for viruses as the tag has a small storage capacity of 128 bits. However, this situation has changed, as authors in [41] stated that RFID tags could be used as a medium to spread a computer virus. This paper also described how the RFID virus ran in supply chain products.

Eavesdropping: In RFID system, tags and readers are wirelessly connected and communicated without a human intervention. So, there is a possibility that their communication medium can be eavesdropped. In general, eavesdropping launches when an adversary captures data transmitted between tag and reader, since most RFID systems lack any encryption technique during transmission process due to the memory capacity. As a result, it is very easy for any attacker to obtain the sensitive data from RFID tags.

Man in the middle (MITM): MITM attack might be happened on RFID system during transmission of data between reader and tags. In this case, an attacker may intercept and modify the communication channel between the components of RFID system. This type of attack is considered as a real time attack, displaying and modifying the information before the legitimate object receiving it.

Killing Tag: Killing tag attack on RFID system could be launched to stop tags communication with their reader. Killing tags makes them impossible to be read, and therefore, it is absolutely essential to make sure that RFID tags are not killed by an illegal party.Kill command should be secured by a strong password as well[39].

2.1.2 NFC-based attacks: It uses in several payment

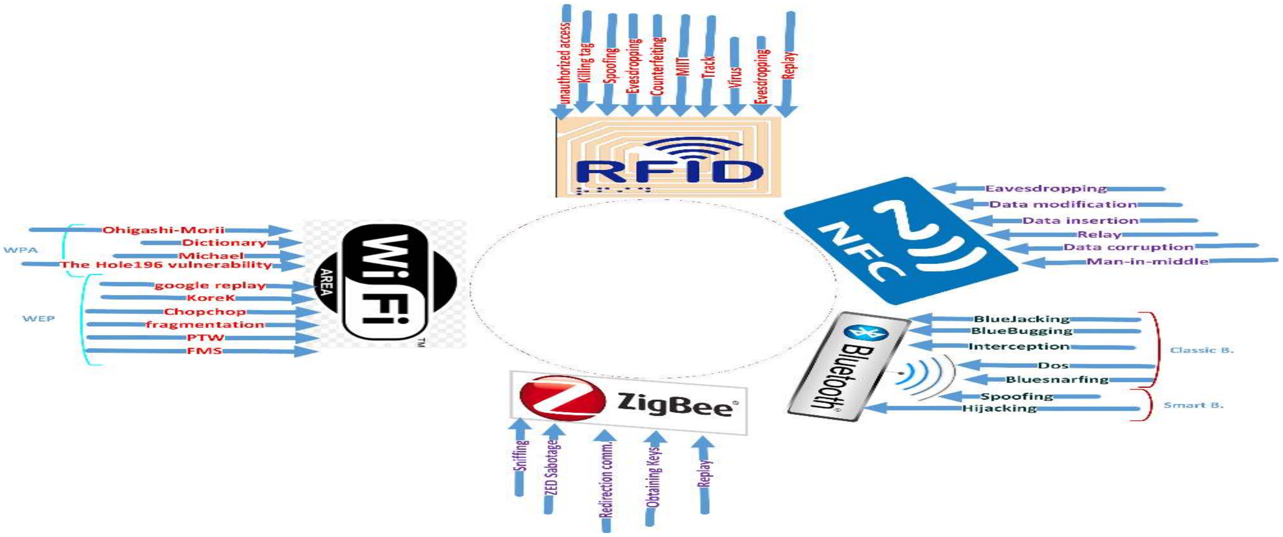


Fig. 11. Taxonomy of connectivity protocols attacks.

TABLE V. CONNECTIVITY PROTOCOLS AND THEIR SECURITY SUPPORTS

Connectivity Protocols	Security Modes	Reliability	Device type	Error Control
Wif-Fi 802.1X	WEP, WPA, WPA2 , Access Control List (ACL).	TCP/UDP others	Access point, devices	Frame Check Sequence (FCS)
Blurtooth	Three different security suites: null security 1, service level security 1, link level security 1	The acknowledge information (ACK or NAK bit)	Master and slave	1/3 rate FEC, 2/3 rate FEC, Automatic Repeat reQuest (ARQ)
ZigBee	Eight different security suites provided by IEEE 802.15.4 and key management	ACKS and control of duplicate packets	Coordinator and end device	Cyclic Redundancy Check (CRC)
Active RFID 802.15.4f	Eight different security suites provided by IEEE 802.15.4	TCP/UDP others	Tags and reader	CRC, ACKS optional
6LoWPAN	Eight different security suites provided by IEEE 802.15.4: null security 1, encryption only 1, authentication 3, authentication with encryption 3	TCP/UDP others	Edge router,mesh node(mesh under), router(route over), and host	CRC, ACKS optional

applications reaching almost 50 billions at the end of 2013. NFC was designed to allow different objects with the same technology to communicate securely with each other. However, this protocol suffers from several attacks [42]. The most important attacks are the following:

Eavesdropping: In NFC system, data exchange between two objects takes place in the close proximity. That said, such system is susceptible to an eavesdropping attack. Communication channel between two IoT objects equipped by NFC protocol is vulnerable to such attack, since NFC lacks any protection technique. An attacker could intercept the communication channel using a powerful antenna or be on close proximity of the communication range [43].

Relay attack: Performing this type of attacks relies heavily on the execution of the application protocol data unit instructions (ISO/IEC1443). Relay attack Forwards the request of victim's reader to a malicious one and replays back its response as fast as possible [44].

Man-in-middle: Although NFC protocol requires a close proximity between communicated objects, these objects are theoretically vulnerable to man in the middle attacks. An attacker could intercept the data, modifying and relaying it to malicious objects. Besides the close proximity that makes these attacks are very difficult, encryption techniques also make them so hard to success if they implemented properly [45].

Data corruption: Data corruption launches when an attacker has the capability to disturb communication channel between two objects by changing the transmitted data to an unreadable format, resulting in denial of services attack [46].

Data modification: Unlike data corruption, in which an attacker change only the format of transmitted data, data modification attack could alter the content of the data [47].

Data insertion: During the process of changing data transmitted between two bjects equipped with NFC protocol, an attacker could insert some data into this data only if the object requires a long time to reply. The successful insertion could only happen "if the inserted data can be transmitted, before the original device starts with the answer. If both data streams overlap, the data will be corrupted" [46].

2.1.3 Bluetooth-based attacks: In this section, we will identify the most popular attacks targeting Bluetooth protocol:

Bluesnarfing: The main goal of this attack is to get access illegally to Bluetooth devices so that the attacker could retrieve their information, and redirect the incoming calls to another [48].

BlueBugging: Bluetooth devices are vulnerable to many attacks, the most dangerous of which is bluedugging. In this type of attack, an adversary could be inside the victims device by exploiting some vulnerabilities in old devices firmware; hence, he could spy on phone calls, send and receive messages, and connect to the internet without legal users awareness [48].

bluejacking: Recently, the majority of Bluetooth devices have been designed to send a wireless business card. Consequently, a new attack has been designed to exploit this feature by sending an offensive card; however, such attack doesn't put information at risk. In this type of attack, the attacker should be very close-within 10meters- to the victim's device to establish this attack. To overcome such attack, it is recommended to put the devices armed with this protocol on nondiscoverable mode [48].

Denial of service (Dos): Repeatedly using his Bluetooth device to send a request pairing to the victim's device, an adversary could establish DOS attacks. Unlike a traditional Internet, where this kind of continuous request could shut down services, most of Bluetooth DoS attacks have been designed to create a nuisance because information in Bluetooth system can be transmitted without user's awareness. According to many research [49], performing DoS attacks is the simplest way to drain a device's battery.

Interception: Unencrypted transmission could be intercepted by a passive eavesdropper. Bluetooth interception does not require sophisticated nor expensive hardware. There are several affordable hardware options that help to accomplish this task, such as Ubertooth [50].

Hijacking: This type of attacks takes place when a configuration layer of the iBeacon has been compromised by an unauthorized third-part to control the beacon settings. DoS and spoofing might be happened as a consequence of such attack [51].

Spoofing: One of the most popular vulnerabilities in Bluetooth Low Energy is the spoofing, as the beacon is publicly broadcasted . A sniffing tool may be used to capture beacon's UUID by an attacker, imitate the beacon and break the rules made by the applications to verifiy the identity so that he could access to the services [51].

2.1.4 Wifi-based attacks: The development and the realization of IoT vision depends heavily on different enabler technologies. Wifi (IEEE 802.11) is one of such technologies . In this section, we identify possible attacks against the Wi-Fi.

FMS attack: This type of attack was released in 2001 by Fluhere, Shamir, and Mantin. The attackers have compromised the WEP protocol due to its vulnerabilities. It is a stream cipher attack in which attackers could recover the encryption key used to encrypt exchange by knowing Initialization vectors (IV). However, the possibility of this attack on RC4-based SSL(TLS) is very difficult as the key generation depends on a hash function [52].

Korek attack: Korek, an unknown participant on NetStumbler.org security forums, has discovered a new attack on Wired Equivalent Privacy (WEP) protocol [53]. Such attack depends on FMS-attack to find the key. Furthermore, he has released an A-neg attack through which the attackers could reduce the key generation possibilities to discover the key faster [54].

Chopchop attack: The Chopchop attack was developed by Korek. Instead of compromising a vulnerability in the RC4 algorithm, such attack focuses on the design defects in WEP protocol such as the vulnerability in CRC 32 check-sum and the absence of replay protection. Chopchop attack allows an attacker to encrypt the exchange messages without knowing the key [53].

Fragmentation attack: A fragmentation attack has been discussed on the context of WEP protocol and the first implementation of such attack was published by Bittqu et al. [55]. To successfully perform this attack, eavesdropping a packet is required. All packets transmitted over 802.11 network have homogeneous headers and, which helps the attacker to guess the first 8 bytes of the headers by XORing these 8 bytes and 8 bytes of cipher text, to get 8 bytes from the IV [55], [53] .

PTW Attack: The Pyshkin Tews Weinmann (PTW) attack was released in 2007. This attack has introduced two new principals: 1) Jenkins relationship proposed to guess the key with less minimum attempts; and 2) multiple bytes prediction instead of guessing bytes indvidually [55].

Google Replay Attack: By setting Google.com as a home page, an attacker could simply discover a part of key stream using Google log downloaded every time the users open the Google website. The main difficulty of this attack is how to know exactly when users will download the Google log [53].

Michael Attacks: Michael's algorithm is used to generate a hash function. However, Reck and Tews in [53] discovered a method in which they could reverse this algorithm. Also, Beck in [73] found a method to execute attack based on Michael's flaws by exploiting its internal state to be reset when it reaching a particular point. In this case, an attacker could inject some code in a packet.

Ohigashi-Morii Attack: This type of attack was introduced as an extension to Beck-Tews attack on WPA-TKIP. In fact, this attack is effective for all modes of WPA. The time of injecting a malicious packet is minimized approximately from 15 to one minute in the best case [53].

The Hole196 Vulnerability: This vulnerability has been discovered by Sohail Ahmad in [74]. Ahmed found that there is a hole in standard 802.11 protocols exactly on the page 196. An attacker, who is an unauthorized user of the network, could send a fake ARP request with access point MAC address and other users will update their ARP tables upon the request. After updating their ARP tables, users will transmit their packets to attacker's MAC address instead of access point. The attacker, in this scenario, can get the packets decrypted by the access point, read them, and re-encrypt these packets with his own key [53].

Dictionary Attack: A dictionary attack is a technique in which an attacker could breach into a password-protected WiFi by guessing its passphrase by attempting millions or billions of possibilities, for instance words in a dictionary. [53].

2.1.5 ZigBee-based attacks

Sniffing: Because most ZigBee networks do not use any encryption technique, they might be vulnerable to sniffing attacks. The attacker can intercept some packets to perform malicious activities using KillerBess's zbdump tool [62].

Replay attack: Replay attack depends heavily on network traffic interception. Being able to intercept the packets, the attacker could re-transmit the intercepted data as if they sent

TABLE VI. CONNECTIVITY ATTACKS WITH COMPROMISED SECURITY GOALS AND COUNTERMEASURES

Connectivity attacks	**Compromised security goals**	**Countermeasures**
Killing Tag	ALL	Users or objects authentication [56]
Spoofing	ALL	RFID authentication and encryption techniques [51]
Man in the middle	C, I, P, NR	Encryption of the RFID communication channel [45], authentication techniques
Tracking	P, NR	Kill/sleep command, isolation, anonymous tag, blocking[57]
Virus	P, I, AU, TW, NR, C	Blocking strange bits from the tag using well-developed middleware, bounds checking and parameter [41]
Evesdropping	C, NR, P	Encryption techniques, shift data to the back end
Replay	C,I,AC,NR,P	A challenge and response mechanism, the time-based or counter- based scheme [41]
RFID unauthorized access	All	Network authentication [40]
NFC		
Eavesdropping	C, NR, P	Secure channel (authentication and encryption) [43]
Data modification	ALL	Changing the baud rate(use of 106k Baud), the continuous monitoring of RF field, secure channel[43]
data corruption	A, AC, AU, NR	The detection of RF fields during data transmission [43]
Relay attack	C, I, AC, NR, P	Timing(enforcing stricter timing restraints on responses) [58], distance Bounding (Round-Trip-Time (RTT) of cryptographic challenge-response pairs [59]
Data insertion	P, I, AU, TW, NR	Objects reply with no delay, a secure channel between the two objects [46]
Man-in-the middle	C, I, P, NR	A secure channel between the NFC objects
ZigBee		
Sniffing	C, NR, P	Implementing high security by preinstalling the network key on the ZigBee devices [60]
Replay attack	C,I,AC,NR,P	The implementation of freshness counter (a 32-bit frame counter), [61]
ZED Sabotage attack	All	The remote alerting system for warning about power failures of ZigBee objects, configure the legitimate ZEDs in a cyclic sleep mode [61]
Obtaining keys	P,I,AU,TW,NR	Out-of-band key loading method Using [62]
Redirecting Communication	C, I, AC, NR, P	Secure network admission control, preconfigure nodes with the Trust Center address [63].
Bluetooth		
Bluejacking	NR, AU, TW, AU	Putting objects on nondiscoverable mode, stay offline [48]
Bluebugging	All	Firmware and software update, use of RF signatures [64]
Interception	C,NR,P	Data/voice encryption, increasing user understanding of security issues, minimization of transmit powers,using only long PIN codes [64], pairing process in private settings [48]
DoS	A AC, AU, NR, P	Keeping a list of suspicious devices [65]
Bluesnarfing	All	Putting phones on nondiscoverable mode [48], stay offline[64], verify incoming transmission
Spoofing	P,I,AU, TW, NR	Secure UUID - Rotating UUIDw/ limited token scope, Private Mode with Rotating UUID, Secure Shuffling randomly rotating UUID [66]
Hijacking	All	Cloud-based token authentication,Secure Communications, Software Lock[66]
WiFi		
FMS	P, I, AU, TW, NR, C	The use of RC4-based SSL (TLS), the use of higher-level security mechanisms such as IPsec [67]
Korek, Chopchop, Fragmentation, PTW, Google replay	P, I, AU, TW, NR, C	The use of a very short rekeying time, disabling the sending of MIC failure report ,disabling TKIP and using a CCMP only network [68], the use of higher-level security mechanisms such as IPsec, DTLS, HTTP/TLS or CoAP/DTLS, DTLS for CoAp[69]
Michael	P, I, AU, TW, NR, C	Deactivating QoS or settingthe rekeying timout to a low value[70], disable TKIP and switch to the more secure CCMP
Ohigashi-Morii	P, I, AU, TW, NR, C	Security protocols based on AES [71]
Dictionary Attack	P, I, AU, TW, NR, C	The use of salt technique [72]

by a legitimate user. The main consequence of such an attack relies on the content of the packets being re-transmitted [62].

Obtaining the key: What makes ZigBee protocol vulnerable to such an attack is that its keys need to be re- installed over the air if its objects are re-flashing [75].

Redirecting Communication: In the ZigBee network, an attacker could redirect and eavesdrop its packets. This attack attack could be used to launch a MITM attack, the main objective of this attack is intercepting and changing the transmitted data [76].

ZED sabotage attack: In [61], the authors have proposed a new attack against ZigBee protocol known as a ZigBee End-Device. The main goal of such attack is to vandalism the ZED by sending periodically a particular signal to wake up the object to drain its battery.

An overview of all attacks against wireless connectivity protocols, their compromised security goals, and their available defense mechanisms is presented in Table VI.

2.2 Network Protocols-based attacks

In this section, the main attacks targeting the network protocols as depicted in Fig. 12 are described in greater detail below.

2.2.1 RPL-based attacks

Routing Protocol for Low power and lossy network (RPL) has been designed to allow multiple-point to point, point to point, and point to multiple-point communication. Its topology depends heavily on the DODAG tree (Destination Orientation Directed Acyclic Graph) composed of one root known as a sink node [77]. The main attacks against RPL are the following:

Selective forward attack: Forwarding chosen packets by attacker to disturb routing paths is the main goal of this attack. Denial of service attacks may take place as a consequence of such attack. An attacker is capable of forwarding all RPL control packets and getting ride of the remaining traffic [78].

Sinkhole attack: In this attack, a malicious node may announce beneficial route or falsified path to attract so many nodes to redirect their packets through it. Despite not disruption the network, it could be dangerous if it is jointed with another attack [79].

Sybil attack: In this type of attack, a malicious object may use different identities in the same network. Such attack was designed to overcome the main goal of redundancy techniques in scattered data storage. Furthermore, it can be used to attack routing algorithms [80].

Wormhole attack: RPL is prone to the wormhole attack, which disturbs both network topology and traffic. This attack can be launched by creating private channel between two attackers in the network and forwarding the selective packets through it [78].

Blackhole attack: Like a hole, which absorbs everything, a blackhole attack has been designed to drop silently all data packets that are meant to it by maliciously advertising itself as the shortest path to the destination during the path-discovering mechanism [79].

Identity attack: In RPL network, identity attack is a combination of spoofing and sybil attacks. An attacker could illegally get access to packets intended to specific node by cloning its identity [78].

Hello flooding attack: Objects recently joining the network send broadcast packet known as a hello message. In this case, an attacker can represent himself as a neighbor object to several objects by broadcasting hello message with a high-powered antenna to deceive other objects to send their packet through it [80].

2.2.2 6loWPAN-based attacks

IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) enables the communication between resource constrained objects and IPv6 network. It performs as an adaptation layer between network and data link layers, offering many advantages such as encapsulation, header compression, fragmentation and reassembly mechanism. Despite the lack of security mechanisms in the 6LoWPAN, its security provide by the underlying layers such as IEEE 802.15.4. The main attacks against 6LoWPAN are the following:

Fragmentation Attack: Unlike IPv6, which has a minimum MTU of 1280 bytes, IoT object operated in IEEE 802.15.4 has Maximum Transmission Unit (MTU) of 127 bytes. Having designed with fragmentation mechanism, 6loWPAN allows the transmission of IPv6 packets over IEEE 802.15.4. Being designed without any type of authentication, an attacker can insert his fragment among fragmentation chain [79].

Authentication Attack: Due to the lack of authentication in 6loWPAN, any objects can join to the network and get an authorized access [79].

Confidentiality Attack: Due to the absence of encryption technique in 6loWPAN, many attacks can be launched such as MITM, eavesdropping, and spoofing [79].

An overview of all attacks targeting RPL and 6lowPAN, their compromised security goals, and their available defense mechanisms is presented in Table VII.

2.3 Communication protocols-based attacks

While connectivity protocols have been designed to link different IoT objects with each other, communication protocols have been engineered to exchange messages between them by providing a standard way for naming, messaging, and controlling [92]. Standard naming refers to the process via which each IoT object will be reached, referred, and recognized. Standard messaging defines how each IoT message is structured so that all IoT objects can easily understand it. Standard controls allow IoT objects to manage communication flow. In this section, the main attacks targeting the communication protocols as depicted in Fig. 13 are described in greater detail below.

2.3.1 TCP-UDP-based attacks

Unlike application layer of IoT stack, which has different protocols to choose from such as HTTP, CoAP, MQTT, and DDS, transport layer has only two standardized protocols TCP and UDP. The most common attacks targeting these protocols are:

TCP-UDP Port scan: One of the most popular methods used by attackers to explore services to compromise them is a port scan attack. If used a port scan tool, an attacker can send a message to each port to test if the port is working to discover some weaknesses [93].

UDP flood: It is a kind of DoS attack in which an attacker sends a huge number of UDP packets randomly to different ports to force so many objects to send back ICMP packets which may make some object unreachable [94].

TCP Hijacking: The first step to achieve such an attack is to monitor a TCP session. In this case, an attacker can detect and guess the sequence numbers and check-sums of the communicated entities. Then, the attacker can inject a malicious TCP packet containing the check-sum and sequence numbers expected by the receiver, who lacks a mechanism to validate the packet source deeming it as a legitimate one [95].

TCP SYN flooding: According to [96], more than 90 percent of the DoS attacks target the TCP protocol, and the most popular of which is SYN flooding attack. This attack consists of a set of eavesdropped TCP SYN packets directed to victim's port. Web servers such as Mail servers, and FTP servers and the connected objects are vulnerable to such attack[94].

TCP-UDP fragmentation: In general fragmentation attacks in both TCP and UDP protocols cause a DoS attack. In UDP, the main objective of such attacks is to re-transmit malicious UDP packets of size bigger than the network's MTU to consume server's resources as it is difficult to resemble these packets [97].

2.3.2 Application layer protocols-based attacks

Application protocols play a major role in the IoT context. The most dominant protocols are MQTT and CoAP [98]. Table IV gives the summary of all IoT communication protocols and their main properties. A brief overview of attacks targeting IoT communication protocols is shown below:

Pre-shared key attack: Security mechanism in some IoT application such as a CoAP protocol depends on pre-shared keys. In some cases, these keys are hard-coded within the code. Therefore, the attacker can easily get access to them if he has access the library files. [99].

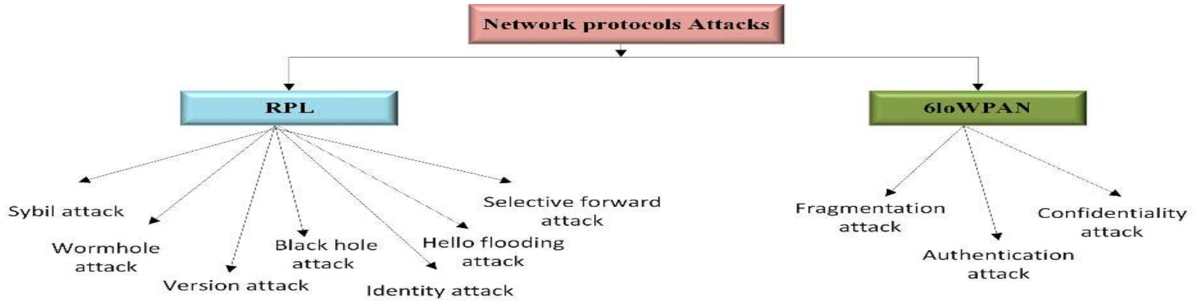


Fig. 12. Taxonomy of network attacks against IoT objects.

TABLE VII. NETWORK LAYER ATTACKS WITH ITS COMPROMISED SECURITY GOALS AND COUNTERMEASURES

Physical attacks	Compromised security goals	Countermeasures
Selective forward attack	C,I,AC,NR,P	Encryption technique , disjoint path or dynamic path between parent and children [79], heartbeat protocol, IDS solution.
Sniffing attack	C, NR, P	Encryption [81]
Sybil attack	C,I,AC,NR,P	Classification-based Sybil detection (BCSD) [82]
Wormhole attack	C,I,AC,NR,P	Markle tree authentication [82], binding geographic information [83]
Blackhole attack	C,I,AC,NR,P	The implementation of RPL in RIOT OS, Tiny OS, monitoring of counters [84], SEVELTE [85]
Identity attack	A, AC, I	Tracking number of instances of each identity, storing Identities of nodes in RPL, distributed hash table (DHT) [79]
Hello flood attack	C,I,AC,NR,P, A	link-layer metric as a parameter in the selection of the default route [86]
Version attack		Version Number and rank authentication, TRAIL [87]
Sinkhole attack	A, C, I	IDS solution [85], identity certificates, parent fail-over [88], and a rank authentication technique
Fragmentation attack	P,I,AU,TW,NR	Split buffer approach, content chaining approach [89], add new fields to the protocol fragmentation header
Authentication attack	C, I, P, NR	Authentication mechanism [90]
Confidentiality attack	C, I, P, NR	Moving Target IPv6 Defence in 6LoWPAN [91]

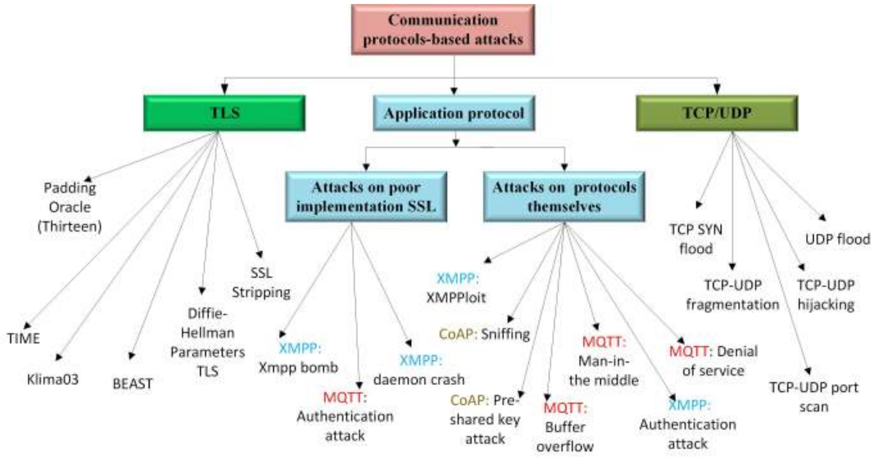


Fig. 13. Taxonomy of communication protocols attacks.

Sniffing attack: The use of sniffer applications may help sniffing or monitoring the network traffic to gain access to sensitive data especially if application protocols have been implemented without security mechanism such as CoAP with no-security mode [100].

SSL stripping: Secure Socket Layer (SSL) stripping was first developed by Moxie Marlinspike [101]. The main goal of such attack is to try to take out the use of SSL / Transport Layer Security (SSL/TLS) by manipulating unencrypted protocols to demand the use of TLS. More specifically, it manipulates both HTTP traffic and HTML pages while they are transmitted.

Beast: The beast attack depends heavily on exploiting the vulnerabilities in TLS 1.0, as it implements Cipher Block Chaining (CBC). Having used HTTP to run over TLS, the attacker can use the CBC to decrypt either parts of message or HTTP cookies [102].

Diffie-Hellman Parameters: All TLS versions are vulnerable to some attacks known as cross-protocol attacks when Diffie-Helman and Elliptic Curve Diffie-Hellman parameters are used to exchange the pre-shared key [103].

Klima03: It is a kind of Certificate and RSA-related attacks on TLS. The process of deriving all session keys depends entirely on premaster-secret value. So, the entire captured SSL/TLS could be decrypted once an attacker get the premaster-secret value [114].

Time: It is a type of compression attacks using TLS with TLS-level compression, which may help an active adversary to decrypt the cipher-text, particularly cookies [113].

Padding oracle(Thirteen): This type of attack is intro-

TABLE VIII. COMMUNICATION PROTOCOLS ATTACKS WITH ITS COMPROMISED SECURITY GOALS AND COUNTERMEASURES

Physical attacks	Compromised security goals	Countermeasures
TCP SYN flood	A,AC,AU,NR,P	SYN Cache mechanism [94], SYN cookies, firewalls , switches and routers with rate-limiting and ACL capability [94]
UDP flood	A,AC,AU,NR,P	Firewalls, deep Packet Inspection [104]
TCP-UDP Port scan	A,AC,AU,NR,P	Network intrusion detection system(NIDS), external firewall[93]
TCP-UDP session hijacking	P,I,AU,TW,NR, C	Encrypted transport protocols[105] such as Secure Shell (SSH), Secure Socket Layers (SSL), and Internet Protocol Security (IPSec)
TCP-UDP Fragmentation	A,AC,AU,NR,P	Blacklisting/whitelisting mechanisms, a secure proxy [97]
XMPPloit	P,I,AU,TW,NR	SSL
Sniffing	C, NR, P	DTLS [106]
Pre-shared key attack	P,I,AU,TW,NR, C	The use of the ephemeral keys as in ECDH key exchange guarantees PFS[99]
MITM	C, I, P, NR	Secure MQTT[107]
Buffer overflow	P,I,AU,TW,NR, C	Close the opening ports, awareness of security [40]
XMPP: Authentication attack	P,I,AU,TW,NR, C	Authentication mechanism [90]
Xmpp bomb	P,I,AU,TW,NR, C	Validating parsers using Document Type Definitions (DTD) and XML Schemas [108]
Daemon crash	P,I,AU,TW,NR, C	Good implementation of TLS
Padding oracle (Thirteen)	P,I,AU,TW,NR, C	The encryption-then-MAC instead of the TLS default of MAC-then-encryption [109].
Time	P,I,AU,TW,NR, C	Disabling TLS compression [110]
Klima03	P,I,AU,TW,NR, C	TLS 1.1, [111]
Beast	P,I,AU,TW,NR, C	Authenticated encryption algorithm like AES-GCM [109]
Diffie-Hellman parameters	P,I,AU,TW,NR, C	The of predefined DH groups [112]
SSL stripping	P,I,AU,TW,NR, C	HTTP Strict Transport Security (HSTS) [113]

duced as a result of using the MAC-then-encrypt in all TLS versions. A thirteen is a new type of such attack in which a timing side channel is utilizing to decrypt the ciphertext [115].

Xmpp bomb: This type of attack can be used to launch a DoS attack specially when the attacker sends a valid compressed request with lot of white spaces[116].

XMPPloit: This attack depends heavily on XMPP weaknesses and the main goal of which is to act as a gateway between clients and server forcing clients to send their messages without encryption.

Man-in-the middle (MITM): Because MQTT has been designed to send its usernames and passwords without any encryption, it is vulnerable to the MITM attack[116].

Buffer overflow: The buffer overflow attack can be happened as a consequence of opening a port on MQTT protocol [117].

An overview of all communication protocol attacks, their compromised security goals, and their available defense mechanisms is presented in Table VIII.

*3) **Data at rest-based attacks**:* In this section, we will identify all potential threats and possible attacks targeting only IoT data at rest resided either locally in IoT objects or remotely in the cloud, as most of the attacks targeting data in motion have been implicitly discussed in protocols attacks. A brief description of all attacks targeting IoT data at rest is presented below and depicted in Fig. 14.

Data exposure: An IoT data is subjected to several attacks due to storing them remotely on the data centers with no supervision of their holders. The number of attacks will be increased, as the malicious objects can get access to these data once they are not properly protected due to the lack of encryption and key management [118]. Additionally, data may place in different data centers distributed at different geographical countries, and have a high power to access this data without permission of their holders [129].

Data loss: IoT objects and cloud providers should be equipped with data loss prevention to deal with high possibility of losing data, causing harmful consequences such as a ransomware attack [130].

Account hijacking: Weak passwords and social engineering might be used to perform an account hijacking. An attacker may compromise, manipulate, and redirect the sensitive data [131]. In the cloud environment, application program interfaces such as SoAP, REST, and HTTP have been used to provides different services. However, many issues have been identified with such interfaces, and the most notable of which are week passwords, insufficient authorization inspections, and input data validation [132].

Data scavenging: Being recoverable, IoT data are vulnerable to many attacks if they are not properly destroyed or removed [133].

Data leakage The lack of the secure methods of processing, storing, and transmitting data is the main consequence of this attack, for example, storing unencrypted data either on the cloud or on IoT objects [134].

DoS: Making IoT data inaccessible by legitimate users is the main objective of such attack. Dos attack exploit the vulnerabilities of the application interface programs (API)[119], [132].

Data manipulation: Illegal manipulating of data at rest can be achieved in two ways: 1) exploiting different vulnerabilities in API like SQL injection, and cross site scripting; and 2) taking advantage of weak security mechanisms such as small passwords [134].

Virtual Machine (VM) Escape: VM escape exploits weaknesses of hyper-visor. The objective of such attack is to dominate the underlying infrastructure gready for its configuration flexibility and code complexity, which matches companies needs [133].

VM Hopping: Due to the hyper-visor complexity, the unlimited resource allocation, and its configuration flexibility on the cloud, attackers may be able to attack one VM to gain access to another one [135].

Malicious VM creation: As many VM images are de-

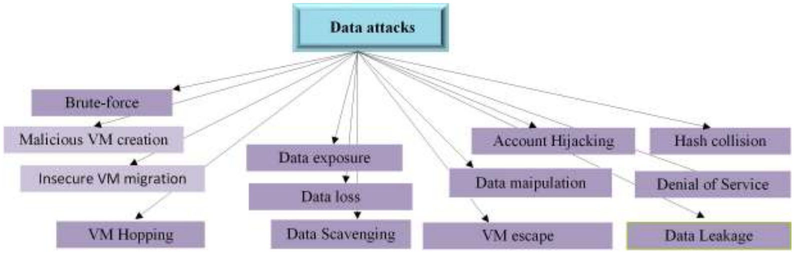


Fig. 14. Taxonomy of data at rest attacks.

TABLE IX. DATA AT REST ATTACKS WITH COMPROMISED SECURITY GOALS AND COUNTERMEASURES

Physical attacks	Compromised security goals	Countermeasures
DOS Exposure	C, I, PP	Strong encryption techniques, key management methods [118]
Data loss	ALL	Strong key generation, storage and management, and destruction practices [119], backup and retention strategies.
Data Scavenging	C, I, P	Symmetric key Cryptography [120]
VM Hopping	ALL	None [120]
Malicious VM Creation	ALL	Mirage [120]
Insecure VM Migration	All	Protection aegis for live migration of VMs(PALM) [121], VNSS offers protection through virtual machine live migration [122]
Account Hijacking	ALL	Identity and access management guidance, dynamic credentials [123]
Data Manipulation	ALL	Web application scanners (such as firewall) [124]
VM Escape	ALL	Trusted cloud computing platform, trusted Virtual Datacenter, hyperSafe, properly configuring the host/guest interaction [125] .
Data leakage	C, I	Digital Signature, fragmentation-redundancy-scattering (FRS) technique, homomorphic encryption [126], encryption [120]
Dos	P , A	Policies provided by providers [120]
Hash collision	C, I	Modern hashing algorithms like SHA-2, SHA-3, or bcrypt[127]
Brute-force	C, I	lockout mechanisms, IP address lock-out, detection tools, brute force site scanners[128]

ployed in unattended environments, an attacker could build legitimate VM account containing a malicious code like Trojan horse [134].

Insecure VM migration: An attacker could get access to data illegally during the immigration process of a VM to a malicious or a trusted host, which may expose its data to the network [136].

Brute-force attack: This type of attack depends on a trial and error method in order to get information such as user passwords or personal identification number (PIN). Brute force attack uses automated software to generate a huge number of sequential guesses to decrpypt the the ciphertext [137].

Hash collision: The main objective of the collision attack is to discover two input strings of a hash function that gives the same hash value. Because hash functions have variable input lengths and a short fixed length output, there is a possibility that two different inputs generate the same output and this case is known as a collision [138].

An overview of all data at rest attacks, their compromised security goals, and their available defense mechanisms is presented in Table IX.

*4) **IoT Software-based Attacks**:* In IoT system, data security for IoT is not equivalent to software security. In some cases, even if the attacker hacks IoT application, he will not get an access to the data if it is well encrypted, but he might be able to do other harmful actions such as control the IoT object or sending spam to other IoT objects.

A brief description of all attacks targeting IoT data at rest as depicted in Fig. 15 is presented below.

4.1 Application-based attacks

Nowadays, IoT web application is rarely developed to operate in a stand-alone mode. Each application is connected to other applications that may inflict harm, making them vulnerable to many attacks. It is worth mentioning that most attacks on web applications often occur in unconstrained IoT objects resided in layer two or layer four in the proposed reference model. The most popular attacks targeting web applications are the following:

Exploitation of a misconfiguration: In some cases, several components such as operating systems, databases, and servers can be used to support running IoT applications.Thus, improper configuration of such components may lead to security issues in IoT application.

Malicious code injection: In this type of attack, an attacker injects a spiteful code into some packets to either steal or modify sensitive data[100].

Path-based DoS attack: The main objective of this attack is to inject malicious code into the packets or replay some packets to the network. It could destroy or destruct an IoT network by sending a huge number of legitimate packets to exhaust network resources along path to a base station. This attack, therefore, may prevent other objects from sending messages to the base [139].

Reprogram attack: Reprogramming the IoT objects remotely as done in some environments can be achieved using a network programming system. Once the programming process is not protected, the attacker could hijack this procedure to control a large part of the network [140].

Malware: The process of infection web applications with a malicious program is known as a malware. Recently, a

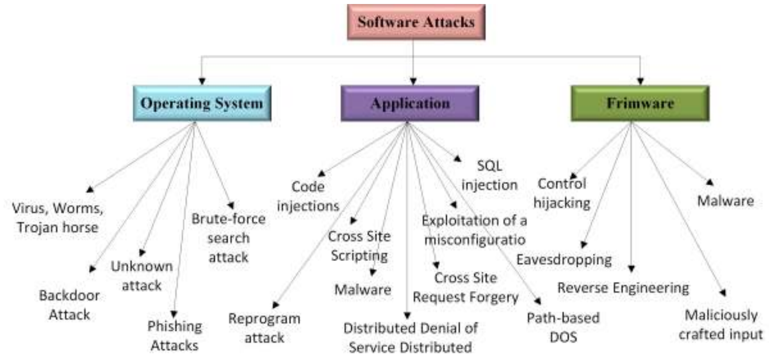


Fig. 15. Taxonomy of IoT software.

huge number of malware have been designed to attack IoT applications.

Distributed Denial of Service (DDoS): One of the main techniques that can be used to establish DDOS attack is a botnet. An example of this attack is the access prevention to a resource by flooding it with so many requests [141].

4.2 Operating system-based attacks

Phishing attack: It is one of the most common security challenges either to users or companies to keep their sensitive data secure. An attacker could get access to passwords , credit cards and other sensitive data via hacking an email, phones, or social media [142].

Backdoors: With the advent of IoT vision, many developers have proposed different IoT operating system like RTOS and Contik . Such operating systems may contain backdoor in which they could reprogram them to get access sensitive data anytime [143].

Virus, worm attack: Nowadays, many viruses and worms, like Mirai, Stuxnet, and Brickerbot, have been designed to attack some weaknesses such as lack of updat mechanisms found in IoT objects [144].

Brute-force search attack: This type of attack has been designed to hack an IoT system by breaking its security mechanisms such as cryptography and authentication using different techniques [145].

Unknown attack: The authors in [146] stated that some common vulnerabilities and exposures (CVE) records have not provided with adequate information to define the preconditions of an attack, which classify as as unknown attack.

4.3 Firmware-based attacks

Smart phones and computers systems have been designed to receive frequent updates to fix the future vulnerabilities or bugs. For example, companies such as Microsoft, Samsung, and Google have built in such a way that they update their vulnerabilities remotely when they are revealed. In contrast, IoT systems are rarely designed to receive regular updates as they are being created by offshore third parties. These third parties mostly don't have professional developers to secure these systems. Even worse, the majority of IoT devices lack any approach to be updated.

Control hijacking: The process of changing the normal flow control of the IoT object firmware by injecting a malicious code is known as a control hijacking attack [156], [146].

Reverse Engineering The main goal of this attack is to analyze the objects' firmware to get sensitive data such as credentials [146].

Eavesdropping: Unlike reverse engineering attacks, which is an active attack, eavesdropping is a passive attack. Eavesdropping attack monitors the packets transmitted between objects and servers during the firmware update process. The attacker could only get sensitive data if the packets are either weakly protected or not protected at all. Furthermore, the attacker could resend the packets to establish a replay attack [157].

Malware: Modifying the behavior of the IoT system after infecting its firmware with a malicious code is the main purpose of such an attack. Several malware have been found in the state-of-the-art such as BASHLITE, Hydra, and Darlloz [158].

An overview of all IoT software attacks, their compromised security goals, and their available defense mechanisms is presented in Table X.

V. Conclusion

The appearance of IoT paradigm in the last few years has unleashed so many threats and feasible attacks against security and privacy of IoT objects and individuals. These threats lead to hamper the realization of this paradigm if they have been left without proper countermeasures. Despite unprecedented number of security attacks generated on IoT domain, there is a lack of standard method to identify and address such attacks. This paper, therefore, makes a best effort to provide a comprehensive classification of IoT attacks based on a novel building-blocked reference model, along with proposed countermeasures to mitigate them. Given IoT developers and researchers, willing to develop a secure IoT system, a chance to investigate which attacks have been fired, how they have been mitigated, which attacks still stick around was the main objective of this paper. Moreover, if manufactures and academia have proactively and sharply targeted such attacks by leveraging their mitigation techniques from the ground up starting from the objects in the physical environment until the data centers in the cloud, the broad applicability of IoT will improve significantly.

TABLE X. IoT Software Attacks with Compromised Security Goals and Countermeasures

Physical attacks	**Compromised security requirements**	**Countermeasures**
Virus, worms	All	Security updates, side-channel analysis, verify software integrity [147], control flow [148]), protective Software
Backdoor attack	ALL	Circuit design modification
Malicious Scripts	ALL	Firewalls [149]
Phishing Attacks	ALL	Crptographic methods
Brute-force search attack	ALL	Securing firware update, cryptography methods
SQL injection	ALL	Data validation, pretesting [150], network-based intrusion detection (IDS)
Cross Site Scripting	P,I,AU,TW,NR, C	Data validation [17]
Cross Site Request Forgery	P,I,AU,TW,NR, C	including a unique, disposable and random token [17]
Exploitation of a misconfiguration	All	A strong application architecture, perform scans and audits continuously [151]
DoS attack	A,AC,AU,NR,P	Access Control Lists[152]
Malware	All	Security updates, side-channel analysis, verify software integrity, control flow [148]), IoT Scanner [153]
Path-based DOS attack	A,AC,AU,NR,P	Combining packet authentication and anti replay protection [154]
Reprogram attack	P,I,AU,TW,NR, C	secure the reprogramming process [154]
Control hijacking	All	Use Safe programming languages , audit software, add runtime code [155]
Reverse Engineering	All	Tamper proofing and self-destruction(obfuscation)
Eavesdropping	C, NR, P	A secure channel

References

[1] F. DaCosta, "Rethinking the Internet of Things: A scalable approach to connecting everything." *Apress Open*, p. 185, 2013.

[2] D. Konstantas, "An overview of wearable and implantable medical sensors." *Yearbook of medical informatics*, pp. 66–69, 2007.

[3] J. Pike, "Internet of Things - Standards for Things," 2014.

[4] E. Alsaadi and A. Tubaishat, "Internet of Things: Features, Challenges, and Vulnerabilities," *International Journal of Advanced Computer Science and Information Technology (IJACSIT)*, vol. 4, no. 1, pp. 1–13, 2015.

[5] S. Institute, "InfoSec Reading Room Securing the Internet of Things Survey," p. 22, 2014.

[6] E&Y, "Cybersecurity and the Internet of Things," *E&Y*, no. March, pp. 1–15, 2015.

[7] European Research Cluster on The Internet of Things (IERC), "Internet of Things: IoT Governance, Privacy and Security Issues," *European Research Cluster on the Internet of Things*, p. 128, 2015.

[8] A. Mohsen Nia and N. K. Jha, "A Comprehensive Study of Security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, p. d, 2016.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[10] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, oct 2010.

[11] Cisco, "The Internet of Things Reference Model," *Internet of Things World Forum*, pp. 1–12, 2014.

[12] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry, "IoT architecture challenges and issues: Lack of standardization," *FTC 2016 - Proceedings of Future Technologies Conference*, no. December, pp. 731–738, 2017.

[13] Z.-K. Zhang, M. C. Y. Cho, and S. Shieh, "Emerging Security Threats and Countermeasures in IoT," *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security - ASIA CCS '15*, pp. 1–6, 2015.

[14] Andreas Fink, *IoT: Lack of standards becoming a threat*.

[15] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," 2010.

[16] S. Agrawal, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *Abakos*, vol. 1, no. 2, pp. 78–95, 2013.

[17] B. Dorsemaine, J. P. Gaulier, J. P. Wary, N. Kheir, and P. Urien, "A new approach to investigate IoT threats based on a four layer model," *IEEE Transactions on Emerging Topics in Computing*, no. Notere, 2016.

[18] D. Kajaree and R. Behera, "A Survey on IoT Security Threats and Solutions," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 1302–1309, 2017.

[19] J. S. Kumar and D. R. Patel, "A Survey on Internet of Things : Security and Privacy Issues," *International Journal of Computer Applications*, vol. 90, no. 11, pp. 20–26, 2014.

[20] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the Internet of Things (IoT)," *International Conference on Network Security and Applications*, vol. 89 CCIS, pp. 420–429, 2010.

[21] J. Guo and I. R. Chen, "A Classification of Trust Computation Models for Service-Oriented Internet of Things Systems," *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, pp. 324–331, 2015.

[22] Y. Cherdantseva and J. Hilton, "A Reference Model of Information Assurance & Security*."

[23] Symantec, "An Internet of Things Reference Architecture," 2016.

[24] J. Sen, "A Survey on Wireless Sensor Network Security," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 1, no. 2, 2009.

[25] A. M. Nia, S. Member, S. Sur-kolay, and S. Member, "Physiological Information Leakage : A New Frontier in Health Information Security," vol. 4, no. 3, pp. 321–334, 2015.

[26] C. Wachsmann, "Physically Unclonable Functions (PUFs)," *Morgan & Claypool Publishers*, 2014.

[27] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016.

[28] W. I. Khedr, "SRFID: A hash-based security scheme for low cost RFID systems," *elsevier*, 2013.

[29] H.-h. Huang, L.-y. Yeh, and W.-j. Tsaur, "Ultra-Lightweight Mutual Authentication and Ownership Transfer Protocol with PUF for Gen2 v2 RFID Systems," vol. II, pp. 16–19, 2016.

[30] Nate Lord, "Social Engineering Attacks: Common Techniques & How to Prevent an Attack — Digital Guardian."

[31] N. Lesperance, S. Kulkarni, and K.-t. T. Cheng, "Hardware Trojan Detection Using Exhaustive Testing of k -bit Subspaces," *IEEE Access*, pp. 755–760, 2015.

[32] A. Davis, "A Survey of Wireless Sensor Network Architectures," *International Journal of Computer Science & Engineering Survey*, vol. 3, no. 6, pp. 1–22, 2012.

[33] Q. Xiao, T. Gibbons, and H. Lebrun, "RFID technology, security vulnerabilities and countermeasures," *Supply Chain, The Way to Flat Organisation*, no. December, pp. 357–382, 2009.

[34] H. Li and Y. Chen, "The Survey of RFID Attacks and Defenses," *ieee*, pp. 0–3, 2012.

[35] H. Salmani, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," *ieee*, pp. 247–276, 2014.

[36] J. Deogirikar, "Security Attacks inIoT : A Survey," *International conference on I-SMAC*, pp. 32–37, 2017.

[37] Walters, “Security in distributed, grid, mobile, and pervasive computing,” *ACM*, 2007.

[38] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, “Smart Nest Thermostat : A Smart Spy in Your Home,” *Black Hat USA*, pp. 1–8, 2014.

[39] M. Polytechnic and M. Polytechnic, “RFID Security Issues & Challenges,” *ieee*, 2014.

[40] M. M. Ahemd, M. A. Shah, and A. Wahid, “IoT security: A layered approach for attacks & defenses,” *2017 International Conference on Communication Technologies (ComTech)*, pp. 104–110, 2017.

[41] Q. Xiao, C. Boulet, and T. Gibbons, “RFID security issues in military supply chains,” *Proceedings - Second International Conference on Availability, Reliability and Security, ARES 2007*, pp. 599–605, 2007.

[42] A. Elbagoury, A. Mohsen, M. Ramadan, and M. Youssef, “Practical provably secure key sharing for near field communication devices,” in *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, jan 2013, pp. 750–755.

[43] N. B. Thorat and C. A. Laulkar, “Survey on Security Threats and Solutions for Near Field Communication,” pp. 291–295, 2014.

[44] M. Roland, J. Langer, and J. Scharinger, “Practical Attack Scenarios on Secure Element-Enabled Mobile Devices,” in *2012 4th International Workshop on Near Field Communication*. IEEE, mar 2012, pp. 19–24.

[45] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum, “Classifying RFID attacks and defenses,” *springer*, vol. 12, no. 5, pp. 491–505, 2010.

[46] E. Haselsteiner and K. Breitfuß, “Security in Near Near Field Communication (NFC) Strengths,” *Semiconductors*, vol. 11, no. 71, p. 71, 2006.

[47] C. H. Chen, I. C. Lin, and C. C. Yang, “NFC attacks analysis and survey,” *Proceedings - 2014 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2014*, pp. 458–462, 2014.

[48] N. Be-Nazir, I. Minar, and M. Tarique, “BLUETOOTH SECURITY THREATS AND SOLUTIONS: A SURVEY,” *International Journal of Distributed and Parallel Systems (IJDPS)*, vol. 3, no. 1, 2012.

[49] M. Tan, “An Investigation of Bluetooth Security Threats,” *ieee*, 2011.

[50] Ubertooth One, “Great Scott Gadgets - Ubertooth One.”

[51] H. Jun Tay, J. Tan, and P. Narasimhan, “A Survey of Security Vulnerabilities in Bluetooth Low Energy Beacons,” 2016.

[52] N. Chen, “Bluetooth Low Energy Based CoAP Communication in IoT CoAPNonIP: An Architecture Grants CoAP in Wireless Personal Area Network,” 2016.

[53] M. Caneill and J.-L. Gilis, “Attacks against the WiFi protocols WEP and WPA,” 2010.

[54] K., “Korek Attack,” 2004.

[55] A. Bittau, M. Handley, and J. Lackey, “The Final Nail in WEP’s Coffin,” 2006.

[56] I. R. Adeyemi Norafida Bt Ithnin, “Users Authentication and Privacy control of RFID Card,” 2012.

[57] Gan Yong, He Lei, Li Na-na, and Zhang Tao, “An improved forward secure RFID privacy protection scheme,” in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*. IEEE, mar 2010, pp. 273–276.

[58] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, “Practical relay attack on contactless transactions by using NFC mobile phones,” *Cryptology and Information Security Series*, vol. 8, pp. 21–32, 2012.

[59] G. P. Hancke and M. G. Kuhn, “Attacks on time-of-flight distance bounding channels,” *Proceedings of the first ACM conference on Wireless network security - WiSec ’08*, pp. 194–202, 2008.

[60] Cyber Security Community, “Different Attacks and Counter Measures Against ZigBee Networks — TCS Cyber Security Community.”

[61] N. Vidgren, K. Haataja, J. L. Pati??o-Andres, J. J. Ram??rez-Sanchis, and P. Toivanen, “Security threats in ZigBee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned,” *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 5132–5138, 2013.

[62] X. Fan, F. Susan, W. Long, and S. Li, “Security Analysis of Zigbee,” 2017.

[63] K. Masica, “Recommended Practices Guide For Securing ZigBee Wireless Networks in Process Control System Environments,” 2007.

[64] K. HAATAJA, “Security Threats and Countermeasures in Bluetooth-Enabled Systems,” 2009.

[65] M. Keijo and H. Senior, “Bluetooth network vulnerability to Disclosure, Integrity and Denial-of- Service attacks,” vol. 17, 2005.

[66] Gofor, “Common attacks and how Kontakt.io can protect you.”

[67] A. Stubblefield, J. Ioannidis, and A. Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP.” *Ndss*, no. Iv, 2002.

[68] E. Tews and M. Beck, “Practical Attacks Against WEP and WPA,” *Proceedings of the second ACM conference on Wireless network security*, pp. 79–85, 2009.

[69] C. Schmitt, T. Kothmayr, W. Hu, and B. Stiller, “Two-Way Authentication for the Internet-of-Things,” *springer*, vol. 25, pp. 27–57, 2017.

[70] M. Beck, “Enhanced TKIP Michael Attacks,” 2010.

[71] T. Mekhaznia and A. Zidani, “Wi-Fi Security Analysis,” *Procedia Computer Science*, vol. 73, no. Awict, pp. 172–178, 2015.

[72] Wikipedia, “Dictionary attack - Wikipedia.”

[73] M. Beck, “Enhanced TKIP Michael Attacks,” 2010.

[74] M. S. Ahmad, “WPA Too!” *Defcon 18*, p. 7, 2010.

[75] J. Wright, “KillerBee: Practical ZigBee Exploitation Framework or ”Wireless Hacking and the Kinetic World”,” 2009.

[76] J. Markert, M. Massoth, K.-P. Fischer-Hellmann, S. Furnell, and C. Bolan, “Attack Vectors to Wireless ZigBee Network Communications - Analysis and Countermeasures,” *Proceedings of the Seventh Collaborative Research Symposium on Security, E-learning, Internet and Networking (SEIN 2011), Furtwangen, Germany*, pp. 57–66, 2011.

[77] M. Asim, “IoT Operating Systems and Security Challenges,” vol. 14, no. 7, p. 5500, 2016.

[78] A. Mayzaud, R. Badonnel, and I. Chrisment, “A taxonomy of attacks in RPL-based internet of things,” *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.

[79] P. Pongle and G. Chavan, “A survey: Attacks on RPL and 6LoWPAN in IoT,” *2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015*, vol. 00, no. c, pp. 0–5, 2015.

[80] J. Sen, “Security in Wireless Sensor Networks.”

[81] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. Kenchiku Setsubi Iji Hozen Suishin Kyokai, 2004.

[82] K. Zhang, X. Liang, R. Lu, and X. Shen, “Sybil attacks and their defenses in the internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[83] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, “Preventing Wormhole Attacks on Wireless Ad Hoc Networks: A Graph Theoretic Approach,” *ieee*, 2005.

[84] K. Chugh, A. Lasebae, and J. Loo, “Case Study of a Black Hole Attack on 6LoWPAN-RPL,” *SECURWARE 2012, The Sixth International Conference on Emerging Security Information, Systems and Technologies*, no. c, pp. 157–162, 2012.

[85] S. Raza, L. Wallgren, and T. Voigt, “SVELTE: Real-time intrusion detection in the Internet of Things,” *Ad Hoc Networks*, vol. 11, pp. 2661–2674, 2013.

[86] L. Wallgren, S. Raza, and T. Voigt, “Routing Attacks and Countermeasures in the RPL-Based Internet of Things,” *International Journal of Distributed Sensor Networks*, vol. 794326, no. 11, 2013.

[87] H. Perrey, M. Landsmann, O. Ugus, T. C. Schmidt, and M. Wählisch, “TRAIL: Topology Authentication in RPL,” 2013.

[88] K. Weekly and K. Pister, “Evaluating sinkhole defense techniques in RPL networks,” *Proceedings - International Conference on Network Protocols, ICNP*, 2012.

[89] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, “6LoWPAN Fragmentation Attacks and Mitigation Mechanisms,” 2013.

[90] L. M. L. Oliveira, J. J. Rodrigues, A. F. De Sousa, and V. M. Denisov, “Network Admission Control Solution for 6LoWPAN Networks Based

on Symmetric Key Mechanisms,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2186–2195, 2016.

[91] M. Sherburne, R. Marchany, and J. Tront, “Implementing moving target IPv6 defense to secure 6LoWPAN in the internet of things and smart grid,” in *Proceedings of the 9th Annual Cyber and Information Security Research Conference on - CISR ’14*. New York, New York, USA: ACM Press, 2014, pp. 37–40.

[92] Stéphane GARCIA, “Wireless Security and the IEEE 802.11 Standards,” 2004.

[93] McMaster University, “The Five-Layer TCP/IP Model: Description/Attacks/Defense - Computing and Software Wiki,” 2008.

[94] S. Kumarasamy and G. A. Shankar, “An Active Defense Mechanism for TCP SYN flooding attacks,” *arXiv.org*, pp. 1–6, 2012.

[95] O. Zheng, J. Poon, and K. Beznosov, “Application-Based TCP Hijacking,” 2009.

[96] D. Moore, G. M. Voelker, and S. Savage, “Inferring Internet Denial-of-Service Activity,” 2000.

[97] Incapsula, “What is an IP Fragmentation Attack (Teardrop ICMP/UDP) — DDoS Attack Glossary — Incapsula.”

[98] Toby Jaffey, “MQTT and CoAP, IoT Protocols.”

[99] S. Jucker, “Master ’ s Thesis Securing the Constrained Application Protocol by Stefan Jucker,” no. October, pp. 1–103, 2012.

[100] S. N. Swamy, “Security Threats in the Application layer in IOT Applications,” pp. 477–480, 2017.

[101] Moxie Marlinspike, “SSLstrip.”

[102] T. D. Juliano Rizzo, “Browser Exploit Against SSL/TLS Packet Storm.”

[103] N. Mavrogiannopoulos, F. Vercauteren, V. Velichkov, and B. Preneel, “A cross-protocol attack on the TLS protocol,” *{ACM} Conference on Computer and Communications Security*, pp. 62–72, 2012.

[104] Incapsula, “What is a UDP Flood — DDoS Attack Glossary — Incapsula.”

[105] B. S. Kevin Lam, David LeBlanc, “Theft On The Web: Theft On The Web: Prevent Session Hijacking.”

[106] J. Granjal, E. Monteiro, and J. Sa Silva, “Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[107] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, “Secure MQTT for Internet of Things (IoT),” *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*, pp. 746–751, 2015.

[108] UsingXML, “White Space in XML Documents.”

[109] P. Gutmann and University, “Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS),” pp. 1–7, 2014.

[110] T. Be’ery and A. Shulman, “A Perfect CRIME? Only TIME Will Tell,” *BlackHat Europe 2013*, 2013.

[111] A. Choudhury and D. Mcgrew, “AES Galois Counter Mode (GCM) Cipher Suites for TLS Status,” pp. 1–8, 2008.

[112] D. Gillmor, “Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS),” pp. 1–29, 2016.

[113] P. S.-A. Y. Sheffer, R. Holz, “Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS),” pp. 1–13, 2015.

[114] V. Klima, O. Pokorny, and T. Rosa, “Attacking RSA-based sessions in SSL/TLS,” *Cryptographic Hardware and Embedded Systems Ches 2003, Proceedings*, vol. 2779, pp. 426–440, 2003.

[115] N. J. AlFardan and K. G. Paterson, “Lucky thirteen: Breaking the TLS and DTLS record protocols,” *Proceedings - IEEE Symposium on Security and Privacy*, pp. 526–540, 2013.

[116] M. Wang, “Understanding security flaws of IoT protocols through honeypot tech- nologies MengWang,” 2013.

[117] P. Du, “IoT Message Protocols: The Next Security Challenge for Service Providers? The State of IoT,” pp. 2–4, 2017.

[118] C. Rong, S. T. Nguyen, and M. G. Jaatun, “Beyond lightning: A survey on security challenges in cloud computing,” *Computers & Electrical Engineering*, vol. 39, no. 1, pp. 47–54, jan 2013.

[119] Cloud Security Alliance, “Cloud Security Alliance,” 2010.

[120] S. Chandna, R. Singh, and F. Akhtar, “Data scavenging threat in cloud computing,” no. August, pp. 17–22, 2014.

[121] Webopedia, “PALM.”

[122] G. Xiaopeng, W. Sumei, and C. Xianqin, “VNSS: A network security sandbox for virtual computing environment,” *Proceedings - 2010 IEEE Youth Conference on Information, Computing and Telecommunications, YC-ICT 2010*, pp. 395–398, 2010.

[123] SYBASE, “Dynamic credentia.”

[124] Tenable, “Tenable.io Web Application Scanning — Tenable.”

[125] Z. Wang and X. Jiang, “HyperSafe: A lightweight approach to provide lifetime hypervisor control-flow integrity,” *Proceedings - IEEE Symposium on Security and Privacy*, pp. 380–395, 2010.

[126] N. P. Smart and F. Vercauteren, “Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes,” *springer*, pp. 420–443, 2010.

[127] Eric Z Goodnight, “What Is SHAttered? SHA-1 Collision Attacks, Explained.”

[128] ALIEN VAULT, “Brute Force Attack Mitigation: Methods & Best Practices — AlienVault,” 2016.

[129] N. Kaaniche and M. Laurent, “Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms,” *Computer Communications*, vol. 111, pp. 120–141, 2017.

[130] Claudia Chandra, “Data Loss vs. Data Leakage Prevention: What’s the Difference?” 2017.

[131] Cloud Security Alliance, “Top Threats to Cloud Computing V1.0,” 2010.

[132] W. Dawoud, I. Takouna, and C. Meinel, “Infrastructure as a service security: Challenges and solutions,” *Informatics and Systems (INFOS), 2010 The 7th International Conference on*, pp. 1–8, 2010.

[133] W. A. Jansen, “Cloud hooks: Security and privacy issues in cloud computing,” *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 1–10, 2012.

[134] B. Grobauer, T. Walloschek, and E. Stöcker, “Understanding cloud computing vulnerabilities,” *IEEE Security and Privacy*, vol. 9, no. 2, pp. 50–57, 2011.

[135] B. A. Sullivan, “Securing the cloud: Cloud computer security techniques and tactics,” *Security Journal*, vol. 27, no. 3, pp. 338–340, jul 2014. [Online]. Available: http://link.springer.com/10.1057/sj.2012.16

[136] J. Rittinghouse and J. Ransome, *Cloud computing\nImplementation, Management, and Security*, 2010.

[137] Kelly Jackson, “Hacker’s Choice: Top Six Database Attacks.”

[138] M. Stevens, A. Lenstra, and B. de Weger, “Chosen-prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities,” nov 2007.

[139] R. Singh, J. Singh, and R. Singh, “ATTACKS IN WIRELESS SENSOR NETWORKS : A SURVEY,” vol. 5, no. 5, pp. 10–16, 2016.

[140] U. Sabeel and N. Chandra, “Categorized Security Threats in the Wireless Sensor Networks : Countermeasures and Security Management Schemes,” vol. 64, no. 16, pp. 19–28, 2013.

[141] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, p. 39, apr 2004.

[142] A. Tsow, “Phishing with Consumer Electronics: Malicious Home Routers,” *2007*.

[143] SecurityWeek News, “IoT Devices Easily Hacked to be Backdoors: Experiment — SecurityWeek.Com,” 2016.

[144] The OWASP Foundation, “Owasp top 10 - 2013 the ten most critical web applications security risks,” 2013.

[145] J. S. SARA BODDY, “The Hunt for IoT: The Rise of Thingbots,” 2017.

[146] D. Papp, Z. Ma, and L. Buttyan, “Embedded Systems Security : Threats , Vulnerabilities , and Attack Taxonomy,” *ieee*, pp. 145–152, 2015.

[147] M. Msgna, K. Markantonakis, D. Naccache, and K. Mayes, “Verifying Software Integrity in Embedded Systems: A Side Channel Approach.” Springer, Cham, 2014, pp. 261–280.

[148] M. Msgna, K. Markantonakis, and K. Mayes, “The B-Side of Side Channel Leakage: Control Flow Security in Embedded Systems,” *springer*, pp. 288–304, 2013.

[149] W. L. W. Michael K. Bugenhagen, “Pin-hole firewall for communicating data packets on a packet network,” 2007.

[150] The OWASP Foundation, “Owasp enterprise security api.”

[151] OWASP, “Top 10 2013-A5-Security Misconfiguration - OWASP.”

[152] M. Ongtang, S. Mclaughlin, W. Enck, and P. Mcdaniel, “Semantically Rich Application-Centric Security in Android,” 2009.

[153] Kaspersky, “The Kaspersky IoT Scanner app helps you secure your smart home Kaspersky Lab official blog.”

[154] S. V. Mahavidyalaya, “Wireless Sensor Networks: Security, Attacks and Challenges,” 2010.

[155] M. Backes and C. Hricu, “Practical Aspects of Security Control Hijacking Attacks,” 2009.

[156] G. Hoglund, G. Mcgraw, and A. Wesley, “Exploiting Software How to Break Code,” 2004.

[157] D. Miessler, “Securing the Internet of Things: Mapping Attack Surface Areas Using the OWASP IoT Top 10.”

[158] K. Angrishi, “Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets,” pp. 1–17, 2017.