# A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS

SANTIAGO FIGUEROA-LORENZO, JAVIER AÑORGA, and SAIOA ARRIZABALAGA,
Ceit and Universidad de Navarra, San Sebastián, Spain

Industrial Internet of Things (IIoT) is present in many participants from the energy, health, manufacturing, transport, and public sectors. Many factors catalyze IIoT, such as robotics, artificial intelligence, and intelligent decentralized manufacturing. However, the convergence between IT, OT, and IoT environments involves the integration of heterogeneous technologies through protocols, standards, and buses. However, this integration brings with it security risks. To avoid the security risks, especially when systems in different environments interact, it is important and urgent to create an early consensus among the stakeholders on the IIoT security. The default Common Vulnerability Scoring System (CVSS) offers a mechanism to measure the severity of an asset's vulnerability and therefore a way to characterize the risk. However, CVSS by default has two drawbacks. On the one hand, to carry out a risk analysis, it is necessary to have additional metrics to the one established by CVSSv3.1. On the other hand, this index has been used mostly in IT environments and although there are numerous efforts to develop a model that suits industrial environments, there is no established proposal. Therefore, we first propose a survey of the main 33 protocols, standards, and buses used in an IIoT environment. This survey will focus on the security of each one. The second part of our study consists of the creation of a framework to characterize risk in industrial environments, i.e., to solve both problems of the CVSS index. To this end, we created the Vulnerability Analysis Framework (VAF), which is a methodology that allows the analysis of 1,363 vulnerabilities to establish a measure to describe the risk in IIoT environments.

CCS Concepts: • **Security and privacy** → **Vulnerability management**; **Security protocols**; • **Networks** → **Network architectures**;

Additional Key Words and Phrases: Industrial Internet of Things, IIoT, Risk analysis, Industrial Security, Operational Technologies, Information Technologies, CVSS, Attack Patterns, Cybersecurity pillars

**ACM Reference format:**
Santiago Figueroa-Lorenzo, Javier Añorga, and Saioa Arrizabalaga. 2020. A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS. *ACM Comput. Surv.* 53, 2, Article 44 (April 2020), 53 pages.
https://doi.org/10.1145/3381038

This research was supported in part by the Basque Government (Elkartek program) through both projects "CYBERPREST-Cybersegurtasunerako gaitasun osoa" with Project No. KK-2018-00076 and "SENDAI- SEgurtasun integrala iNDustria AdImentsurako" with Project No. KK-2019/00072.
Authors' addresses: S. Figueroa-Lorenzo, J. Anorga, and S. Arrizabalaga, CEIT-Basque Research and Technology Alliance (BRTA), Manuel Lardizabal 15, 20018 Donostia / San Sebastián, Spain and Universidad de Navarra, Tecnun, Manuel Lardizabal 13, 20018 Donostia / San Sebastián, Spain; emails: {sfigueroa, jabenito, sarrizabalaga}@ceit.es.

0360-0300/2020/04-ART44 $15.00
https://doi.org/10.1145/3381038

## 1 INTRODUCTION

Industrial Internet of Things, also known as IIoT, transforms industrial and business operations by adding smart connectivity to machines, peoples, and processes. IIoT integrates technical areas such as network connectivity (e.g., low energy wireless protocols, edge computing, and cloud computing), low cost to apply machine learning in both sensors and computing, big data produced by sensors, m2m (machine to machine) communications, and the traditional automation technologies, i.e., traditional industrial control systems (ICS). Although the term Industry 4.0 is associated with how the manufacturing sector is transformed by technologies such as big data, data analytics, and IoT, this concept can be also associated with the interconnection of cyber physical systems, which enables us to use the term IIoT and Industry 4.0 indiscriminately. For this reason, several factors are considered as catalysts for both IIoT and Industry 4.0, such as the convergence of information technologies (IT) and operational technologies (OT), robotics, data, artificial intelligence, smart decentralized manufacturing, auto-optimizing systems, and the digital supply chain information management [1]. Although IIoT architectures have many use case-specific variations, in Figure 1 we consider a basic IIoT architecture model, which includes four subsystems: a sensor network (e.g., communication over Wi-Fi and BLE), a controller or aggregator, an edge gateway, and the business application (through Cloud IIoT Platform). To contextualize this architecture, in Figure 2, the SCADA network is connected to the cloud via an edge gateway. From a higher range connectivity system based, for example, on LoRa, it is possible that many windmills are controlled by the ICS/Scada system. The data received from the turbines are sent to the data center for cloud analysis. The turbine data flows through an edge device, which could be a gateway, central concentrator or edge controller. This edge device will have to support many protocols. Therefore, a distinctive feature of IIoT environments is the convergence of protocols, standards, and buses of different technologies, i.e., the integration. Although, this integration is not the only area of an IIoT system, given the convergence of protocols, standards, and buses is a representative feature. For that reason, the process of convergence can be described around the protocols, standards, and buses that form the following three paradigms: OT (e.g., Modbus, EtherNet/IP, and OPC-UA), IT (e.g., WebSocket, HTTP, and XMPP) and IoT (e.g., Wi-Fi, RFID, and Bluetooth). Therefore, this is the first focus of attention of our work. The digital connectivity (promoted by the integration of protocols, standards, and buses) of industrial machinery and industrial equipment with any physical asset with an IT platform is a unique advance that sets a social precedent of business opportunities. This convergence of the physical world and cyber on an industrial scale allows operations to be handled in thousands of ways. An example of a use case is to prevent critical machine failures through both predictive and proactive detection and maintenance. Another example of a use case is to provide a digital tracking capability in supply chain assets, to name a few use cases. Many other uses cases could be mentioned.

However, these benefits bring associated risks, because cyberthreats are a break point in ubiquitous connectivity and are now a critical constraint on the adoption of IIoT technology. These cyber threats exploit the weaknesses of IIoT environments, causing vulnerabilities. According to IETF RFC 4949 [2] a system can have three types of vulnerabilities: vulnerabilities in design or specification; vulnerabilities in implementation; and vulnerabilities in operation and management. The design vulnerabilities are inherent to a protocol specification, present even in perfect implementations (e.g., a specification using weak cryptography has a design vulnerability) [3]. In addition, implementation vulnerabilities are inherent to assets and correspond to how a particular protocol, standard, or bus is implemented in an asset [3]. The operation vulnerabilities are any event that alter the correct functioning of the asset despite the correct design and implementation of the system.

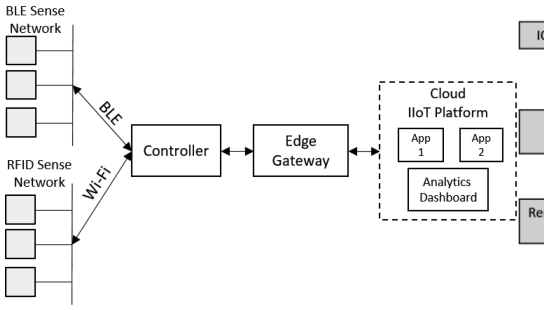


Fig. 1. General IIoT Architecture.

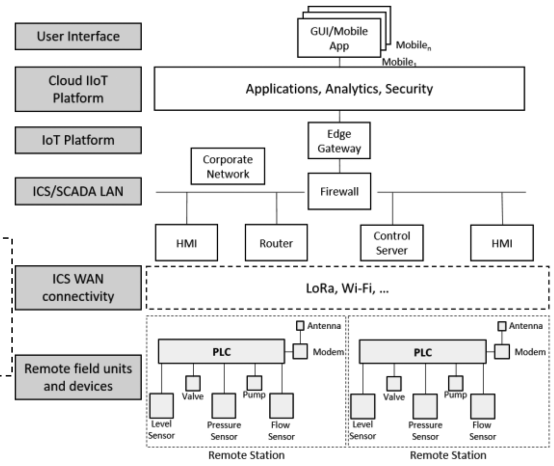


Fig. 2. IIoT Architecture model contextualized.

According to NIST 800-82r2 Reference [4], there are a wide range of threats and vulnerabilities in IIoT environments. These can be classified in general way into policy and procedural vulnerabilities (e.g., there is no formal security training for ICS or OT), architecture and design vulnerabilities (e.g., no security perimeter defined), configuration and maintenance vulnerabilities (e.g., poor remote access controls), physical vulnerabilities (e.g., lack of backup power), software development vulnerabilities (e.g., improper data validation) and communication and network configuration (e.g., data flow controls not employed). For this reason, it is important that IIoT environments include risk analysis to identify and detect malfunctions, as well as to prevent incidents and accidents, i.e., be prepared. The risk is the result of uncertainty about objectives, which considers the probability of an event occurring along with the impact, i.e., the effects, of that event if it happens. Precise product and system design, including design reviews and tests, should to prevent malfunctions and improve the system's robustness to possible or potential events identified in the risk evaluation [5]. As we have mentioned, the starting point (phase 1) of our work is based on an analysis of the integration of IIoT protocols, standards, and buses from the perspective of the OT, IT and IoT paradigms. Therefore, our second line of research focuses on providing a risk characterization mechanism for an IIoT environment (phase 2) based on the analysis of vulnerabilities that may affect the protocols, standards, and buses that are considered during phase 1 of our work. Summarizing, the major contributions of this work are (1) a comprehensive survey of the 33 protocols, standards, and buses most commonly used in IIoT environment, (2) a comprehensive review of the tools available to characterize CVSS in industrial environments, (3) an exhaustive collection of OSINT, data sources, and integration tools that allow carrying out vulnerability studies, (4) a presentation and deployment of a Vulnerability Analysis Framework (VAF) to analyze 1,363 vulnerabilities and from which the two major gaps of CVSS are solved, i.e., (1) CVSS alone is not enough to determine risk, i.e., it is not a risk measure; (2) CVSS introduces different complexities when calculating the severity of a vulnerability in industrial environments. The remainder of the manuscript is organized as follows: Section 2 carries out a comprehensive survey of the most-used 33 protocols, standards, and buses in IIoT environments. Section 3 analyzes CVSS as an evaluation tool for ICS system. Section 4 provides the proposed methodological framework and the information data sources. Section 5 summarizes the results obtained after VAF application. Section 6 provides conclusions and future research lines.

Table 1. IIoT Protocols Classification

| Category | Protocol/Standard/Bus | FAP | PAP | ICSP | BAP | PSAP | AMRP | SHAP | VAP |
|---|---|---|---|---|---|---|---|---|---|
| IT | REST/HTTP/TLS | | | | | | | [6] | |
| | WebSocket | | | | | | | [6] | |
| | JMS | | [7] | | | | | | |
| | DDS | | [8] | | | [9] | | | [10] |
| | XMPP | | | | | [11] | | [12] | |
| IoT | Wi-Fi | | [13] | | [14] | | | [15] | |
| | 4G/LTE | [16] | | [17] | [18] | | | [19] | |
| | 5G | [16] | | | | | | [20] | [20] |
| | MQTT | | [21] | | | | | [22] | |
| | CoAP | | [21] | | [23] | [24] | | [25] | |
| | AMQP | | | | | | | | |
| | Bluetooth | | [26] | | [27] | | | [28] | [29] |
| | ZigBee | | [13] | | [30] | [31] | [32] | [33] | |
| | RFID/NFC | [34] | | | | | | [35] | |
| | LoRa | | | | [36] | | [36] | | |
| | NB-IoT | [37] | | | | | [38] | [39] | |
| | WirelessHart | [39] | [40] | | | | | | |
| | Z-Wave | | | | [41] | | [42] | [41] | |
| OT | PROFINET | [43] | [44] | | | | | | |
| | Niagara AX | | | | [45] | | | | |
| | CAN | | | | [46] | | | | [47] |
| | Siemens S7 | [48] | [49] | | | | | | |
| | EtherNet/IP | [50] | [51] | | | | | | |
| | Hart/IP | | [52] | | | | | | |
| | BACnet | | | | [53] | | | | |
| | Modbus | | [49] | [54] | [55] | [56] | [57] | | |
| | OPC-DA | | | [58] | | | | | |
| | OPC-UA | | | [59] | | | | | |
| | DNP3/IEC-62351 | [60] | | | | [56] | | | |
| | ANSI C12.22 | | | | | | [61] | | |
| | IEC-61850/IEC-62351 | | | | | [44] | | | |
| | IE-60870-5-104/IEC-62351 | | | | | [62] | | | |
| | IEC-60870-6-503/IEC-62351 | | | | | [63] | | | |

**FAP**: Factory Automation Protocol, **PAP**: Process Automation Protocol, **ICSP**: Industrial Control System Protocol, **BAP**: Building Automation Protocol, **PSAP**: Power System Automation Protocol, **AMRP**: Automatic Meter Reading Protocol,
**SHAP**: Smart Home Automation Protocol, **VAP**: Vehicular Automation Protocol.

## 2 SURVEY OF IIOT PROTOCOLS, STANDARDS, AND BUSES

Until now, we have focused on the overall architecture of an IIoT environment. However, the essential feature of this environment is that it has a high level of integration in terms of communication protocols, standards, and buses. For this reason, first, the study of the 33 protocols, standards, and buses most used in IIoT environments will be introduced. To this end, we will divide our section into three subsections according to the categories established in Table 1: IT, IoT, and OT, and from this point on, the IIoT, protocols associated with each category will be analyzed. Although common

criteria are analyzed for each protocol, which will be listed in the following paragraph, the reference is security. Due to the importance of the security of the protocols, standards, and buses in the survey, the need to carry out a risk analysis for these protocols, standards, and buses is detected.

Since we have mentioned at this first stage, we will focus our attention on the IIoT connectivity, which is related by the integration in the same environment (IIoT) of OT, IT, and IoT protocols. For this reason, Table 1 summarizes the protocols, standards, and buses, according to categories to which they belong, i.e., IT, OT, and IoT, as well as, the classification according to the industrial area in which they have been applied. To classify the industrial context, the following categories are established: Process Automation, Industrial Control System, Building Automation, Power System Automation, Automatic Meter Reading and Home Automation. Certain protocols must be excepted as they are on the border between the main categories: IT, IoT, and OT. For instance, XMPP is an instant messaging protocol, i.e., recently having a huge adoption in IoT environments, however, since its application was originally in the IT environment, we have added it in this category. Similar cases are found in the Wi-Fi and 4G/LTE standards that although their initial adoption was in the IT and IoT environments, both have a strong presence in industrial environments. Another consideration has been a protocol such as AMQP, designed to support a broad range of both messaging applications as well as communication patterns from banking organizations efficiently; however, its current use is in IoT environments. The last case to consider is DDS, which by default addresses the needs of applications like aerospace, defense, and air-traffic control. At present, DDS participates in others application context such as autonomous vehicles, smart grid management, power or energy generation, simulation and testing environments, health devices, transport systems, and others real-time applications. Therefore, DDS could perfectly be included in an IoT or OT environment, but given the closeness to the user it has been added within the IT paradigm. Next, the 33 main IIoT protocols, standards, and buses are analyzed in this section. A summary of the protocol, as well as its topology, architecture, types of messages/frames exchanged, and security are used as common criteria in the survey. However, the TCP or UDP port associated with the protocols, standards, and buses is defined by the Internet Assigned Numbers Authority (IANA) recommendations [64].

## 2.1 IT Protocols and Standards

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the IT category according to Table 1.

*2.1.1 REST/HTTP/TLS (TCP: 80).* REST (REpresentational State Transfer) is the dominant model for Web application design, which aims to reduce latency and network communication, maximizing the scalability of independent component deployments. Reference [65] invokes an in-depth study of the use of API REST in IoT environments. The three defined aspects: URI, MIME, and the set of operations relate RESTful (web services based on HTTP and REST fundamentals) with IoT, since the sensors send the information in a MIME format, RESTful sets up a URL for each resource. As Reference [66] mentions, it is simple to retrieve sensor information as a web resource. The security, in HTTP, has typically been delegated to the lower layer via SSL (Secure Sockets Layer) or Transport Layer Security (TLS), and both cases are referred such as HTTPS. These two protocols provide both clients and servers HTTP, security based on asymmetric cryptography for authentication enabling the exchange of keys, and symmetric cryptography for confidentiality. Currently, because SSL is considered unsecure, TLS is mostly used in its latest version 1.3. In addition to SSL/TLS, Reference [67] is a review of mechanisms used by REST to provide security to IoT environments through HTTP authentication schemes. These are divided into basic authentication schemes, token-based authentication, OAuth and OpenID. However,

since all the mentioned applications use SSL/TLS, which guarantees the security of the transport and not of the applications, vulnerabilities and attack patterns must be considered, mainly due to implementation failures. For instance, Reference [68] demonstrates a proof of concept of JSON injection, Reference [69] reported serious logic flaws in OpenID systems, Reference [70] reported cross-site request forgery for OAuth Clients and injection attacks in Web Services are analyzed in Reference [71]. Given that the Open Web Application Security Project (OWASP) introduces in Reference [72] some different kinds of REST attack and the how-to prevents, this resource must be used for both penetration test and design phases.

*2.1.2 WebSocket (TCP: 80).* WebSocket protocol is a low latency (real-time), full duplex (bidirectional), long running (persistent), single connection (TCP) between a client and server. WebSocket provides benefit for real-time, live text chat, video conferencing, VoIP, IoT control and monitoring. The full-duplex aspect of WebSocket allows a server to initiate communication with a client whenever it would like, which is contrary to other protocols, such as REST/HTTP, in which the client must initiate communication. Bidirectional connection allows the server to update the client application without an initiating request from the client. WebSocket not only allows for low-latency communication between a server and client but also reduces network traffic by eliminating the need for a client to send a packet just to request data from the server (the server can send data as soon as they are available or when states have changed without the need to issue a request). For this reason, WebSocket is commonly used in applications like traffic reports, browser-based multiplayer games, and IoT application, e.g., to control servomotors [73]. In addition, WebSocket is used like transport layer of messaging IoT protocols, standards, and buses like XMPP, AMQP, MQTT, and JMS [74]. Since WebSocket is a very young technology, the security best practices around WebSocket are still evolving. The common security levels used in WebSockets not only include WSS (WebSockets over TLS) but also include avoid tunneling, validate client input, validate server data, authentication/authorization and origin header [75]. However, Reference [76] demonstrates attacks such as fingerprinting and fuzzing, JavaScript overload and denial of service (DoS) to both client and server.

*2.1.3 JMS (TCP).* Java Message Service (JMS) is a Java Message Oriented Middleware (MOM) API used to create, sending, receiving and reading messages between two or more clients. JMS is a component of the enterprise edition of the Java EE Platform and it was defined according to the JSR 914 specification, supported by the community. Therefore, JMS enables communication among the different distributed application components to be coupled, asynchronous and reliable as well as, supports publish-and-subscribe and point-to-point routing [77]. The main JMS constraints are because it is a standardized API for JAVA only and it does not define a wire protocol. Therefore, the JMS deployments from several vendors are not interoperable, for instance, with Microsoft NMS (.Net Messaging Service). In that sense, two programs written in two different programming languages cannot communicate with each other over asynchronous messaging. The JMS message objects can be configured to include more information in the message. This information is useful when more complex business logic is required in enterprise messaging applications [78]. JMS does not provide an API to control messages' integrity and privacy, and neither specifies how distributed digital signatures or keys are. Each JMS provider manages security specifically. For instance, the JMS provider TIBCO EMS supports Java clients that can use either the Java Secure Sockets Extension (JSSE) Java package, or an SSL implementation [79]. JMS's leading vendors provide several levels of security. Typically, this means supporting facilities for client authentication and access control. As well as HTTP, JMS supports TLS. Java Authentication and Authorization Service (JAAS) supports several of the major JMS implementations to provide authentication and authorization [80]. In spite of that, McAfee describes in

Reference [81] penetration-testing techniques to assess the security of ActiveMQ (based on Enterprise Messaging Systems (EMS) written using JMS API), as well as demonstrates vulnerabilities as insecure communication, insecure password storage, and weak encryption password. However, deserialization vulnerability to JMS is presented by Reference [80].

*2.1.4 DDS (UDP: 7400, TCP: 7400).* Data Distribution Service (DDS) is a data-focused publishing and subscription protocol that grew out of the aerospace and defense environments and was developed by the Object Management Group (OMG). DDS has been designed to handle applications critical to the enterprise such as air traffic control, financial trading and intelligent network management. The current application environments range from autonomous vehicles through smart grid management to power generation (Table 1). DDS provides to both publishers and subscribers with a scalable, real-time, reliable, high-performance and interoperable data exchange. In addition, DDS specifications set important protocols as part of the DDS suite: Data Centric Publish Subscribe (DCPS); DDS Interoperability Wire Protocol (DDSI) and Real-Time Publish-Subscribe (RTPS). DCPS provides a set of tools that targeting real-time information-availability [82], while DDSI ensures portability and interoperability application [83]. Finally, RTPS manages the discovery process [84]. By default, DDS uses UDP but supports other options like IP multicast [85] and TCP [86]. Since DDS is language and operating system independent, it is suitable for running on both embedded devices and large-scale enterprise systems [87]. The DDS security specification determines both the Security Model (SM) as well as the Service Plug-in Interface (SPI) architecture [88]. DDS-SM is implemented through the invocation of SPIs. The SPI enables to DDS users to customize the behavior for authentication, access control, encryption, data tagging and digital signing logging [88]. Some vulnerabilities are tested in Reference [89], e.g., listing the devices that are communicating using DDSI-RTPS is enabled via an unauthenticated client in both passive and active mode. DoS attack is achieved in the same reference. In addition, Reference [90] shows how DDS can be manipulated to support malicious activity through client-side attacks.

*2.1.5 XMPP (TCP: 5222, 5269, 5280, 5281).* The Extensible Messaging and Presence Protocol (XMPP) is a semi-real-time message exchange protocol that transmits XML elements. By default, XMPP is used to deploy instant messaging applications, multimedia, lightweight middleware, social networking services, and IoT applications. The fast and asynchronous exchange of small payloads of structured information between entities is possible by XML streams and XML stanzas [91]. An XML stream is a container that exchanges XML elements between two entities through the network [91]. The XML stanza is a discrete semantic unit of structured information that is sent between entities through an XML sequence [91]. The XMPP client device nodes establish asynchronous communication with the XMPP server, which is an intermediary component that provides routes among both type of clients, senders and receivers, i.e., entities [92]. The XMPP core specification includes security features. By default, the client must establish an XML stream with a server authenticating itself via the credentials of an account registered through SASL's negotiation with Public-Key Infrastructure X.509 (PKIX) certificates to provide strong authentication [91]. In addition, the XMPP specification introduces the concept of a security label, or confidentiality label, which allows a structured representation of sensitive information [93]. The security labels are used in combination with an entity authorized to access and a security policy to control access to each piece of information. For instance, a message could be tagged as "SECRET" and therefore require that both, the sender and recipient are authorized to access the "SECRET" information. In addition, the specifications establish requirements such as that all communication must be done via properly encrypted links and the data must be encrypted using industry standard encryption on all links and end-to-end. In that sense, given the existing limitation of end-to-end

encryption techniques due to the lack of full stanza encryption, they promote the use of Double ROT-13 as a transparent encryption that provides excellent interoperability benefits [94]. However, vulnerabilities and backdoors appear due to poor quality code implementations (e.g., buffer overflow attacks), poisoning blacklists, attacking the Domain Name System (DNS) infrastructure, amplifying network traffic and hijacking the TCP communication for both clients and servers are referred in Reference [95].

### 2.2 IoT Protocols and Standards

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the IoT category according to Table 1.

*2.2.1 Wi-Fi.* The IEEE 802.11 standard was designed to be the comparable, i.e., equivalent, to the physical and MAC layers of the Ethernet standard (IEEE 802.3), so the distinction between a Wi-Fi and an Ethernet network is the way the frames are transmitted. The IEEE 802.11 standard contains other standards as IEEE 802.11b, IEEE 802.11g, and IEEE 802.11n over 2.4 GHz with speeds of up to 11 Mbit/s, 54 Mbit/s, and 300 Mbit/s, respectively. IEEE 802.11ac is another included standard that operates in the 5GHz band and although it has a smaller range (approximately 10%) with respect to the other standards, there is less interference, because there are no other technologies as established as Bluetooth and ZigBee in this frequency. To ensure security in a Wi-Fi network there are several alternatives ranging from using encryption protocols such as WEP, WPA and WPA2 (IEEE 802.11i) to MAC filtering and IP tunnels (IPSEC). In addition, there are WPA2 Enterprise solutions that use RADIUS back-end servers to store user credentials and even relay the authentication process response to the network access server (NAS), granting access to network resources [96]. However, it is also known that vulnerabilities exist at several levels of the security algorithms. For instance, in WEP most vulnerabilities are introduced because of a weak encryption scheme was implemented while in WPA (2)-PSK it is the authentication mechanism that introduces one significant vulnerability in this algorithm. However, there are different types of attacks on Wi-Fi networks to the extent that there are numerous tools to carry out penetration tests such as the one detailed in Reference [97]. The following two references show examples of some of the most common attacks on Wi-Fi networks. Since, WPA2 still uses a password as the key to authenticate the user; it is susceptible to password-based attacks such as brute force, dictionary, rainbow and the more prominently: the phishing [98]. However, Reference [99] mentions other security issues and commons attacks over Wi-Fi networks such as non-authorized access to target information, replay, DoS, and Pseudo-AP interference. In addition, Wi-Fi can also be used as vector attack, since that Wi-Fi offloading contributes to mitigate the gap between cellular network capacity and mobile data traffic. Therefore, Reference [100] demonstrates a method to build a low-rate DoS attacks via the offloading architecture. Finally, other possible attack in Wi-Fi is the deauthentication attack, which addresses the communication between a router and the device for the effective deactivation of the Wi-Fi connection on the device. The deauthentication attacks use a deauthentication frame, which is sent from a router to a device, forcing the device to disconnect, therefore this attack does not require credentials.

*2.2.2 4G/LTE.* To address the growing use of mobile data through video, image, audio and text enabled by applications such as social networks, 3GPP specified both Long Term Evolution (LTE), and LTE-Advanced (LTE-A) standards that became mobile broadband wireless technologies [101]. LTE divides the protocol stack into user plane protocols, which support routing of users' data between UEs and S-GWs and control plane protocols that are used for exchanging signaling messages between various devices within the network [102]. On the air interface (Uu), the user equipment (UE) functionalities are controlled by the Mobility Management Entity (MME).

However, the communication between UE and MME is established via the evolved node base stations (eNodeB) [102]. The eNodeB supports both user plane and control plane protocols. The user plane protocols include the packet data convergence protocol (PDCP), the Radio link control (RLC), the medium access control (MAC), and the physical (PHY) layer protocols. In addition, the protocols of the control plane include the radio resource control (RRC) protocols [102]. The Uu interface is further divided into two levels of protocols: the access stratum (AS) and the non-access stratum (NAS). The MME signaling lies in the NAS level but is transported within the network using AS protocols. The LTE-A system specified by 3GPP LTE Release 10 was created to improve LTE systems, i.e., to handle significantly superior data usage, even smaller latencies and higher spectral efficiency [103]. Additionally, both systems were designed to handle features such as IP compatibility, complete interoperability with other wireless networks and different kinds of base-stations (BS), as well as nodes retransmission in a large cellular, i.e., macro cellular network [101]. LTE/LTE-A improved security over its predecessor Universal Mobile Telecommunication System (UMTS). For example, to ensure mutual authentication between the UE and the MME, LTE improved both the authentication process and key handle with respect to UMTS. The MME is the termination point in the network for ciphering/integrity protection for NAS signaling and handles the security key management. Additionally, the key hierarchy and handover key management mechanisms were introduced to improve the access security and the mobility process in the LTE architecture [104]. Additionally, the "key hierarchy" and "handover key management" mechanisms were introduced to improve the access security and the mobility process in the LTE architecture [104]. The introduction of new entities such as Machine-type Communication (MTC) [105], Home eNodeB (HeNB) [106] and Relay nodes [107] have been another mechanism used by LTE-A to improve security with respect to LTE. Despite this, several vulnerabilities have appeared in LTE/LTE-A technologies. However, Reference [101] summarizes vulnerabilities of the LTE system architecture, the LTE Access Procedure, the LTE Delivery Procedure, the IMS Security Mechanism and the MTC Security Architecture. For each case, the same reference ([101]) includes attack patterns. Although the same reference ([101]) proposes solutions for the issues listed, the experts of this reference argue that many security issues remain in LTE and LTE-A networks.

*2.2.3 5G.* The fifth-generation mobile network (5G) aims to address the limitations of previous cellular standards, which makes it a key factor for IIoT environments. In general terms, the architecture to be deployed must be between the implementation of a complete autonomous 5G (SA) network that provides experience with end-to-end (E2E 5G) or implementation of a non-autonomous 5G (NSA) network will be complemented and supported by the LTE network [108]. The stand-alone 5G network introduces both, a new 5G-air interface, the "new Radio Access Network" (RAN) that focuses on defining the new radio access, which is flexible enough to support two frequency bands: lower than 6 GHz and higher than 24 GHz and a 5G Core (5GC). Due to the broad range of carrier frequencies supported, orthogonal frequency division multiplexing (OFDM) is the base of the 5G new radio (NR) air interface. This model requires interaction with the LTE network to cover areas outside 5G and to integrate 5G users with non-5G users [108]. The non-standalone (NSA) 5G network contains 5G NR cells connected to the Evolved Packet Core (EPC), which provides LTE as the core. Therefore, 5G cells are fully dependent on the LTE network for control functions and additional services. The NSA architecture functions as a master-slave where the 4G node is the master and the 5G access node is the slave [108]. However, the 5G efficiency includes enhancements to the self-organized networks 5G (SON) and Big Data capabilities, MIMO enhancements, improved power consumption, support for exchange of device capabilities, and a support study for non-orthogonal multiple access (MOMA) [109]. Considering the depth of 5G, for example, in IIoT environments, as analyzed in reference [110], security becomes an imperative

factor, becoming a threat surface for 5G. For this reason, 5G has considered security in its design phase through the SA3 Working Group (WG), which is responsible for security and privacy in systems, determining both security and privacy requirements, and the specification of security architectures and protocols as well as the availability of cryptographic algorithms that must form part of the specifications. Reference [111] sets out the main security features in 5G such as increased home control, the unified authentication framework, the security anchor function (SEAF), the subscriber identifier privacy (SUPI), the subscription concealed identifier (SUCI), the subscription identification security, the subscription identifier de-concealing function (SIDF), the permanent equipment identifier, the globally unique temporary identifier, the procedure for using the temporary subscriber identifier, the subscriber's privacy, the secure steering of roaming, the security edge protection proxy (SEPP), the UE assisted network-based detection of false base station, the network redundancy at the 5G core, as well as the network slicing. In addition, 3GPP has included for 5G an entire security architecture based on elements such as network access security, network domain security, application domain security, SBA domain security, and security visibility and configurability. Despite the high number of security features introduced in 5G, Reference [111] sets out the landscape of the 5G threats, divided into threats in terminals, network or access nodes, core network and external and internal services and applications. Reference [111] illustrates a scenario where adversaries exploit zero-day vulnerabilities in devices or terminals belonging to the massive IoT (MIoT) from which they perform a DDoS attack on a 5G RAN. In addition, Reference [111] mentions the rogue base station (RBS) threat, where an attacker can use the RBS directive to launch different attacks on mobile users and networks. The RBS disguises itself like a real, i.e., "legitimate," base station to address a MITM ("Man-In-The-Middle") among the mobile UE and the mobile network. Finally, in Reference [111], threat scenarios, both the privacy of the subscriber and the heart of the network, are carefully analyzed.

*2.2.4 MQTT (TCP: 1883, 8883).* MQTT is a message transport protocol for client-server environments, which includes as main features to be lightweight, designed to be easy to use, and intended for use in very large environments, including limited environments, e.g., m2m, IoT and IIoT, i.e., scenarios that require a tiny code and/or high bandwidth [112]. The MQTT protocol is covered by "ISO/IEC 20922:2016, which represents the MQTTv3.1.1 according to [112]. In addition, MQTT version 5.0 was released in 2019 and includes new features like enhanced authentication, flow control, maximum packet size, user properties, server keep alive, assigned client ID, and others [113]. MQTT works over a transport protocol, which offers bidirectional, ordered and loss-free connections, e.g., TCP. MQTT also uses a publication/subscription messaging pattern that provides one to many distributions of messages and application decoupling. MQTT, is agnostic to the payload content, which presents three QoS attributes for message transmission: (QoS 0) "At most once," (QoS 1) "At least once," and (QoS 2) "Exactly once" [114]. The QoS level zero guarantees a best-effort delivery, the QoS level 1 guarantees that a message is delivered at least one time to the receiver and the QoS level 2 guarantees that each message is received only once by the intended recipients. In addition, an important feature is that the protocol provides a small transport overhead and minimized packet exchanges to decrease network traffic [112]. MQTT provides their own authentication mechanism, using CONNECT Packet that supports username and password. In addition to the username and password, MQTT clients provide other information that can be used for authentication such as client identifier and X.509 client certificate. The authorization is further another security mechanism used by the protocol via access control list (ACL) [115] and role base access control (RBAC) [116]. External authentication schemes such as OAuth 2.0 or LDAP are also supported by MQTT [117]. Other important security feature supported by the protocol is data encryption, which can be independently managed via SSL or TLS. The port 8883 is exclusively

reserved for MQTT over TLS [118]. Despite the numerous security mechanisms supported by MQTT some issues such as lack of: authentication, authorization, confidentiality, integrity are analyzed in [119], as well as, different attack scenarios are examined in Reference [120].

*2.2.5 CoAP (UDP: 5683).* The Constrained Application Protocol, CoAP is a protocol focused on web traffic for the use of nodes and networks with restricted resources. These nodes, for example, have eight-bit microcontrollers with low quantities of memory, whereas a restricted network are, e.g., 6LoWPANs, which provides a transition for the IPv6 over WPAN, and address both high-rate packet error and high-throughput. Since CoAP is built for m2m environments, e.g., "smart energy" and "building automation," it is oriented on the client-server paradigm with an architecture grounded in RESTful. It therefore adopts RESTful concepts, where the resources are "abstractions" controlled by the server, which put it to disposition by a process that uses a Universal Resource Identifiers, also known as URI, for identification [23]. In addition, the protocol CoAP was designed over UDP, it uses Datagram Transport Layer Security (DTLS) to provide the same properties that SSL/TLS provides to TCP connections, i.e., authentication and end-to-end security [121]. Since, some networks do not forward UDP packets, recently, CoAP has been introduced over TCP, WebSocket and even incorporates TLS [122]. In addition, the protocol defines four security modes, which are detailed in Reference [123]: "NoSec," "PreshardKey," "RawPublicKey," and "Certificates." Reference [124] proposes CoAP security implementations based on DTLS and IPSec. However, even though CoAP supports multicast connections, DTLS only secure unicast message [123]. The biggest challenge to CoAP is to maintain high levels of performance while keeping security standards and ensuring protection [123]. By default, DTLS is the application layer security protocol that CoAP can deploy, however it is not exempt from constraints and additional problems, including compression due to message length and handshake [125], as well as not adapting to "CoAP proxy modes" [126]. Reference [127] demonstrates a series of attacks against CoAP and DTLS such as MITM attack, Sniffing, Spoofing, DoS, Hijacking, Cross-Protocol attacks, Replay attacks, and Relay attacks.

*2.2.6 AMQP (TCP: 5673).* Advanced Message Queuing Protocol (AMQP) is an application layer protocol of open-standard, "message-oriented," which is used commonly in middleware. AMQP grew out from the financial market with the purpose of releasing users from non-interoperable and proprietary messaging systems. It is an enterprise oriented messaging protocol designed to ensure security, reliability and interaction with other systems [128]. AMQP was designed to support both operational models "request/response" and "publish/subscribe" [129]. In addition, as detailed in Reference [128], the protocol allows a wide variety of messaging related utilities, such as "reliable queuing," "topic-based publish-and-subscribe messaging," "flexible routing" and "transactions." Additionally, its communication system involves the publisher/consumer setting up an "exchange" with identification through a name and broadcasting it. Publishers and consumers then use the name to discover each other [130]. AMQP will exchange messages in a variety of forms: directly, in fan-out form, by topic, or based on headers [130]. The AMQP frame normally requires a fixed 8-byte header with small and customizable payloads with a size that based on the broker/server or the programming language used. Reference [131] describes how AMQP enables a security based on different models of TLS negotiation: Single-port TLS Model, Pure TLS and WebSockets Tunnel TLS Model. Therefore, AMQP explicitly allows the integration of both TLS (e.g., TLS virtual server extensions, also known as, SNI) and Simple Authentication and Security Layer, also known as SASL [131]. TLS is mostly associated with encrypting the connection and SASL with authenticating the connection. Access control based on authentication and authorization are other security mechanisms supported by AMQP [132]. However, Reference [133] illustrates how AMQP's security is often affected because of the code is "poorly written" by developers. In

addition, Reference [134] analyzes known AMQP vulnerabilities and the threats/attacks associated to these vulnerabilities such as replay attack, masquerade, messages modification and DoS.

*2.2.7 Bluetooth.* At 2018, around 4 billion Bluetooth (BTL) devices were deployed as Reference [135] indicates. BTL technology opens up more markets, including "automotive," "smart buildings," "smart cities," and "smart homes," highlighted by the recently launched BTL mesh. The topology and architecture of BTL are described by [136], from which, two forms of BTL are mentioned: "Basic Rate/Enhanced Data Rate" (BR/EDR) and "Bluetooth Low Energy" (BLE). The first topology involves a star network topology in Piconet [24]. In addition, BR/EDR includes Scatternet topology, with every Piconet holding a unique master and several slaves, where others Piconets participate of a time-division multiplexing base. The second topology enables a BLE device to perform central and peripheral role. The device with the role of central initiates the connection establishment; if another device accepts, then this connection will have the role of peripheral. BTL Core system includes a host, a primary controller and none or some secondary controllers [136]. Furthermore, BTL specification provides interoperability among independent BTL systems, because it defines protocol messages exchanged between corresponding layers and defining a common and specific interface between both controllers and hosts [136]. The Host layer contains the Security Manager (SM) module. It defines the mechanisms, i.e., the methods and protocols to pair the devices and to distribute the keys. The Security Manager Protocol (SMP) sets the format of the frame and type of the pairing command, as well as the frame structure and the time-out restriction. Key distribution methods used by SM provides both identity as well as encryption [137]. Pairing is carried out to exchange the keys that are used to encrypt the channel. The use of the keys can be extended to encrypt the link at reconnect the devices; verification of information, specifically, signed data; and carry out identification of address. Reference [137], defines three phases for the pairing: (1) "Pairing Feature Exchange," (2) "LE Legacy Pairing" or "LE Secure Connections," and (3) "Transport Specific Key Distribution." Through an exchange of messages, the intervening devices, i.e., central and peripheral, examine the input and output (I/O) capabilities of the other, making possible to determine the appropriate pairing mechanism: "LE Legacy pairing" or "LE Secure Connection." "LE Legacy pairing" includes the methods "Just Works," "Passkey," and "Out-of-Band" (OOB). "LE Secure Connection" includes the three previous methods and introduces "Numerical Comparison" [138]. However, the BTL protocol for both forms described has security limitations. For BLE in particular, tools such as Ubertooth One, Kismet, Wireshark and Crackle [139] allow to exploit design and implementation vulnerabilities based on eavesdropping attack, packet decoding and packet injection [140]. A complete revision of BTL weakness is presented by the NIST reference [141].

*2.2.8 ZigBee.* The ZigBee (ZB) standard was designed for wireless short-range communications (e.g., in the order of 10 to 100 m). As BTL, the physical layer is built on the IEEE 802.15.4 standard. ZB focuses on "low-power" environments and handles different profiles (e.g., "ZB Home Automation (ZBHA)," "ZB Health Care (ZBHC)," "ZB Building Automation (ZBBA)„" etc.), each device is designed in accordance with the needs of the environment [142, 143]. Specific profiles can be found in Table 1. A ZB network consists of three logical devices: the coordinator, the router, and the end device. The coordinator, manages the overall network; the routers, manages the whole Personal Area Network (PAN) and serves as intermediate nodes among the coordinator and the end devices; and terminal equipment or end device, is the simplest type of device on a ZB network, and it is often low power or battery-power. ZB protocol stack contains four layers [143]. Like other wireless protocols, ZB architecture comprises four layers: physical (PHY), media access control (MAC); network (NWK) and application (APL). NWK performs important functionalities such as network topology construction, network topology maintenance, as well as, binding and

naming. APL contains Application Support Sublayer (APS), ZB Device Object (ZDO) and software application. ZDO is in charge of managing the entire device, while APS supports both ZDO and ZB applications [142, 143]. By default, ZB provides security mechanisms in three protocol stack layers: the MAC, NWK, and APL [143]. ZB supports distributed and centralized network architectures, as well as associated security models. A distributed network is made up of two devices: a router and an end device. The router will be in charge of distributing the network key, which will be used by all network devices to protect the messages. This centralized model includes a third device, usually the coordinator, which represents the trust center (TC). The TC enables the other two devices, i.e., the router and the terminal equipment to connect to the network with the appropriated credentials. The TC can only issue encryption keys and it also sets unique TC link keys to each device on the network [143]. A tradeoff that the ZB Alliance chose to make between security and simplicity is the distribution of a unique network key when a device joins the network first time. There is a security issue in ZB HA 1.2 specification if an attacker captures ZB network traffic at the same time that a new device is being joined to the network [142]. This method has been removed in the ZB 3.0 specification and replaced with a process that requires a per-device installation code (like OOB in Bluetooth), i.e., used to generate a unique joining key, which is then used to acquire the ZB network key. The security of a ZB network is built on the ability to manage encryption keys properly [144]. However, the security of the ZB protocol is likely to be limited by physical capabilities, so lower-performance devices are also lower secure [145]. Significant security risks such as: key management issues and secure routing issues (e.g., defaults keys) are mentioned in Reference [144] and sleep deprivation are mentioned in Reference [146].

*2.2.9 RFID/NFC.* The RFID allows objects automatic identification by radio waves, supporting contactless between devices. RFID tags include a micro-chip and a transceiver or antenna, which is only activated by a RFID reader that receives a signal from the tag. The RFID technologies are categorized according to three criteria: (1) operating frequency, (2) power supply, and (3) memory type. The Operating frequencies supported are in the following four frequencies bands: low (LF), high (HF), ultra-high (UHF), and microwave (MWF). The power source classification includes type of tags active, passive and battery-assisted passive. The memory type includes "read only," "write once read many," "read and write." There are passives RFID tags in all frequency's bands mentioned. The Near Field Communication (NFC) is a standard associated with passive high frequency tags. Reference [147] provides a representation of the NFC architecture, including all protocols that comprise the standard. In addition, NFC includes the following standards: ISO/IEC 14443, FeliCa, and ISO/IEC 18092. Beyond RFID/NFC technologies, an RFID/NFC system goes through the integration of these technologies in IIoT environments. In addition to the already mentioned RFID/NFC tags and readers, the more general architecture involves RFID/NFC middleware and the business layer [148]. The RFID/NFC middleware performs functions such as reader management, information collection and integration, data filtering, while the business layer integrates systems such as enterprise resource planning (ERP), customer relationship management (CRM) and track and trace applications. For this reason, the security threats contained in RFID systems are different from traditional wireless security threats, and can be grouped into: (1) physical components of RFID (e.g., clone tags, reverse engineering, tag modification), (2) the communication channel (e.g., eavesdropping, skimming, repeat attack), and (3) threats to the global system (e.g., spoofing, Denial of Service (DoS), and tracking and tracing) [149]. A detailed analysis of the vulnerabilities affecting RFID systems for each category mentioned can be found in reference [138], while in particular for NFC some vulnerabilities due to a Cryptographically Weak Pseudo-Random Number Generator (PRNG), relay attack, and eavesdropping are analyzed, respectively, in References [150–152].

*2.2.10 LoRa.* The LoRa is a proprietary technology of the Low Power Wide Area (LPWA) family, as well as a patented wireless technology of Semtech Corporation. The specification included in Reference [153], indicates the frequency ranges for LoRa, where regulators control the proper use of assigned frequency ranges [154]. Since the wireless architecture and hierarchical organization used by LoRa is quite simple, it is suitable for IoT environments [36]. Since Lora defines only the PHY layer, LoRaWAN is the protocol developed to define the top layers of the network. LoRaWAN is a cloud-based MAC protocol, which acts primarily as a NWK protocol for managing communication, i.e., the network built between LPWAN gateways and end node devices [36]. The network topology is star, enabling the interconnection between terminal nodes and base stations/gateways. The gateways allow messages to be exchanged over the internet with the LoRa network servers. These servers ensure the information exchanged by the terminal devices. LoRa defines several kinds of devices, keys, and encryption capabilities to ensure the network. The cipher suite relies on AES-128 working in CTR mode. Throughout the LoRaWAN network architecture multiple layers are encrypted with the same scheme [153]. LoRaWAN uses different encryption keys to protect devices, the network layer and the application layer. In this way, the technology ensures the security of the lower layers, i.e., intermediate nodes such as gateways and cloud routers, to perform network routing-and maintenance at the same time as ensuring application data confidential. Although LoRaWAN provides extreme-to-extreme (end-to-end) security, which include the use of keys in both layers APL and NWK, an attacker that get physical access will be able to compromise the LoRa terminal devices and therefore not only devices but also NWK keys. In addition, Reference [154] mentions that LoRa is susceptible to jamming, replay and wormhole attacks.

*2.2.11 NB-IoT.* NB-IoT ("Narrowband-Internet of Things"), such as LoRa, is an LPWA technology. It was designed to support a broad variety of different IoT devices as well as services. As Table 1 shows, NB-IoT is used in services such as AMR (e.g., smart metering) and SHA (e.g., smart parking). Since the LTE standard is the base of NB-IoT, we can say that it is a light-weight version of LTE [155]. However, NB-IoT is as simple as possible to minimize costs of devices and reduce the battery consumption, and thus it removes many capabilities/features of LTE, such as handover, monitoring the channel quality, "carrier aggregation," and "dual connectivity" [156]. The NB-IoT core network is based on the evolved packet system (EPS) and is defined for the cellular Internet of Things (CIoT) both the optimization of the user plane CIoT EPS and the optimization of the control plane CIoT EPS [156]. The cellular access procedure of an NB-IoT user is like that of LTE. To optimize the control plane CIoT EPS, the "evolved terrestrial radio access network" UMTS (E-UTRAN) is responsible for radio communications among the EU and the MME, and includes the eNodeB [157]. The data was then transmitted to the packet data network gateway (PGW) using the service gateway (SGW) [157]. If the data is not IP, then the service capability exposure Function (SCEF) is transferred to the node, which provides machine-type data through the control plane and supplies an abstract interface for the services. From EPS optimization of the CIoT user plane, IP and non-IP data can be transmitted by radio carriers through SGW and PGW to the application server. Therefore, for NB-IoT both the existing/current E-UTRAN network architecture for LTE and the backbone are re-used [157]. NB-IoT also adopts security features from LTE such as authentication and encryption. A security feature applied is Data over NAS (DoNAS), which enables the network to transmit user data via the MME into NAS signaling messages. DoNAS is used to carry IP as well as non-IP traffic. Therefore the entity's data is encrypted, ensuring the integrity through the same mechanism/process reserved for network signaling [158]. Another security feature is transporting data via Non-IP Data Delivery (NIDD) using SCEF, which provides a means to securely display service and network capabilities/features through network "application programming interfaces" (APIs) [158]. The use of VPN together with private, secure access point (APN) names dedicated to

specific entities to keep their data communications isolated from the rest of the traffic is another security feature [158]. However, like Reference [155] mentions, perceptron layer is susceptible to both active and passive attacks. In passive attack, the attacker will simply monitor the network traffic while an active attack will compromise the integrity of message, as well as, the forgery of encrypted data. At the perceptron layer, each node is able to communicate with the base station directly so routing security issues are prevented during networking [155]. In addition, Reference [159] establishes proof of concept of an attack based on scan using malicious User Equipment (UE).

*2.2.12 WirelessHart.* WirelessHart is a specification to communicate Hart protocol (Hart-IP is detailed in Section 2.21), which is a digital transmission method in process instrumentation through a wireless connection. It was issued by HART Communication Foundation (HCF) as HART Version 7 in September 2007 [40]. From the OSI model WirelessHart, communication stack contains five layers: PHY, data link layer (DLL), NWK, transport (TRT) and APL. The PHY is built on "IEEE 802.15.4-2006 2.4 GHz DSSS," i.e., direct sequence spread spectrum. This layer includes two important features: channel hopping and transmit power. DLL is based on "time division multiple access" (TDMA), which applies a distinct feature of this layer: a rigid collection of time-slots with a time duration of 10 ms, which are merged to generate one or more superframes [160]. A superframe groups a sequence of consecutive time slots. All superframes starts with Absolute Slot Number (ASN) 0. All nodes must synchronize with their neighbors to determine the exact time of transmitting or receiving, according to the ASN. A time interval (timeslot) ensures the communication of the data and its acknowledgement. Before the transmitter begins to transmit, the receptor node activates the radio in reception mode for a period, also known as, guard time. If the receptor node is not receiving data during this period, then the node returns to standby (sleep) mode and expects the next time interval (timeslot). On the opposite, it waits for the end of the transmission, then it validates it and transmits the acknowledgement to the transmitter node. As we mention, WirelessHart includes other layers such as network and transport layers. The key function of the network layer is to transfer information between a source and a destination promptly and reliably. This process is called routing, which includes some ways to route data. For instance, there are four approach established to route: "source routing," "graph routing," "superframe routing," and "proxy routing." Reference [161] defines in details each of routing approach. Finally, APL defines several device commands, responses, data types, and reports of the network and devices status [161]. We consider important to mention when the sensor network is deployed, the devices are organized to build a mesh topology. In addition, on top of the network are both a central gateway and central network manager. WirelessHart is a protocol with security by design. The protocol provides important security features, such as confidentiality, authentication, and integrity in both communication "hop-to-hop" as well as "end-to-end," through the cipher suite AES 128-bit key combined with CCM mode [162]. Since it is a mesh network, the nodes depend on their neighbors to send the data to the network administrator. Although at the "hop-to-hop" level, the data link layer does not encrypt the information, this layer guarantees both the authenticity and integrity of the information, protecting the network from attackers [163]. Additionally, the network layer provides "end-to-end" security, where only the frame header is not encrypted [164]. Despite WirelessHart standard provides several security mechanisms that make the network more robust to attacks, Reference [160] performs attacks against the network such as Jamming and Advertisement-based Attacks.

*2.2.13 Z-Wave.* Z-Wave is a proprietary, low-energy wireless protocol that builds a mesh network to communicate from one device to another. Z-Wave reliably transmits short messages from the control device to one or more network devices with minimal noise. Each device or node of the network must distinguish from the Network ID or Home ID the nodes of its network from the

nodes of the neighboring networks while from the Node ID distinguishes the nodes of its network. Since all nodes participate in the mesh, network sends and receives control commands allowing Z-Wave to cover a larger area. Additionally, the mesh network uses the intermediate nodes to find a route where the nodes connect to each other to reach the destination. However, the smart home networks, which implement Z-Wave, contain up to 232 devices divided into controllers and slaves. The slaves receive and execute commands from the controller and do not contain a routing table, although they can have a network map with routes to the devices. Although there may be more than one central controller that handles both routing and security of the network, a single controller provides reliable information about the network topology. Z-Wave is composed of an architecture based on four layers: APL, Routing, MAC, and Transfer [165]. Although security for Z-Wave is in the development phase, for example, from the design of devices that allow the capture of radio packets and software, some steps have been taken in the detection of vulnerabilities. Reference [166] provides the presentation of the open-source tool "EZ-Wave," which allows penetration tests on Z-Wave networks and shows "a rapid process for destroying florescent lights." Since the insecure mode of Z-Wave is based on a unique identifier and does not introduce encryption, Reference [167] (using the "Waving-Z" tool, which encodes and decodes Z-Wave frames) allows the Home ID/Network ID to be modified. Although secure mode supports encrypted communications, these are not supported by all devices, are not enabled by default, require additional action for activation, and present poor information for clients. For this reason, Reference [168] has detected vulnerabilities from a manufacturer's implementation error. A vulnerability was discovered in "AES-encrypted" Z-Wave door locks, which would be exploited remotely to unlock doors despite unknown "encryption keys" and because of the changed keys, succeeded network messages/commands will be bypassed by the established controller of the network. However, as of 2016, Z-Wave Alliance announced strengthened security mechanisms, which include encryption standards for transmissions between nodes, and involves new pairing mechanism for each device, based on unique PIN or QR codes on each device, i.e., an OOB mechanism.

### 2.3 OT Protocols, Standards, and Buses

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the OT category according to Table 1.

*2.3.1 PROFINET.* PROFINET (acronym for "Process Field Net") is an open Industrial standard Ethernet-based, i.e., built and maintained by PROFIBUS & PROFINET International (PI). This is contained in the standards: "IEC 61158" and "IEC 61784." PROFINET satisfies the requirements for automation (i.e., "plant and machine manufacturers") and fully compatible with Ethernet according to IEEE 802.3 and multi-protocol parallel Ethernet operation. PROFINET offers two approaches: PROFINET CBA ("Component-based Automation") and PROFINET IO (IO acronyms mean: "discrete input," "discrete output," "analog input" and "analog output"). PROFINET CBA is appropriate for communication m2m via TCP/IP and it is also used for real-time (RT) communication [169]. PROFINET IO contains both RT communication and real-time isochronous communication (IRT) [170]. PROFINET uses TCP/IP (or UDP/IP) communications for certain non-time critical tasks, such as configuration, parameterization, and diagnostics [171]. This occurs because when a packet is transmitted from one PROFINET node to another via TCP/IP, the delay occurs in packing and unpacking across layers, this is a non-deterministic process, which produces jitter. Therefore, the TCP/IP communication is unsuitable for time-critical environments [171]. PROFINET RT handles time-critical data exchange. An arriving PROFINET RT Ethernet frame has the PROFINET EtherType: $0 \times 8892$. Upon arrival, the frame is directed to the PROFINET application directly from Layer 2 to Layer 7 [171]. Unlike TCP/IP, this process is deterministic. For the most demanding

applications, PROFINET can use additional techniques for even faster performance with the PROFINET IRT channel. PROFINET IRT is a step beyond PROFINET RT. PROFINET is based on a phased security concept [172]. The protocol specifies an optimized security model according to the application environment, security zones defined. However, different attacks modalities and how to take a control of a PROFINET IO node is described in Reference [173].

*2.3.2 Niagara AX (TCP: 1911).* The "guide specification to include smart buildings" describes the capabilities and functions of the Niagara Framework, which is scaled on any systems connected network, locally or remotely, and accessible via the internet via web browsers over OT and IIoT networks [174]. The Niagara Framework was designed to allow integrators and developers to connect, manage and control any device, regardless of manufacturer, using any protocol. The Niagara Framework is described by Reference [174] like a Framework Architecture for Edge-to-Cloud technology. Niagara designed the Tridium Fox protocol to drive tunnels to SCADA networks [175]. It is applied in building automation environments (Table 1). As Reference [175] also indicates; in contrast to other protocols, Tridium Fox has not a straight communication with an industrial assets, however, it facilitates interaction among workstations and devices (e.g., Modbus, BACnet, and DNP3). Tridium includes four key parts: "Niagara AX Architecture," "Niagara Structures," "Niagara Protocols," and "Niagara Platforms" [176]. Niagara AX is an open framework based on Java, which could connect both to practically any device (for example, embedded systems) and independent communication protocols, i.e., is able to integrate several manufacturers. We can therefore conclude that it is an agnostic communication framework. It contains a full set of graphical tools that allow users to create sophisticated applications in a drag-and-drop environment to easily handle and manage assets via a web browser. The latest release of Niagara AX was version 3.8, which support TLS 1.2 [177]. However, some attacks due to improper input validations, improper access control and clear text passwords are described by Reference [178].

*2.3.3 CAN.* Currently, all vehicles use the CAN bus, Controller Area Network, as the main serial communication protocol between electronic control units (ECU) for reasons such as ease of adding and removing nodes, failure of a node does not bring down the bus, is implemented with a relative low amount of wires and allows independence between nodes [179]. Another functional strength of the CAN bus is to be a standard that allows, for example, microcontrollers and devices/equipment to communicate with other applications without a central host. Its use extends to applications in autonomous vehicles. For instance, within the implementation proposed in Reference [179], the CAN bus allows the transmission and reception of data to the main card, as well as, the communication between two different ECU through only two CANL (Low CAN) and CANH (High CAN) wires, thus reducing the number of cables in the vehicle. In general, we should mention that the CAN bus is a multi-master serial bus that connects ECU, also known as nodes, therefore, a CAN network will need at least two nodes. Each node/ECU requires a central processing unit (CPU), a controller and a transceiver. In this way, the node can send and receive messages, although in a half-duplex way. Each frame contains an identifier (ID), which signifies the message priority, eight bytes of data, cyclic redundancy check (CRC), an acknowledgment field (ACK), and overload. There is an extended version of 64 bytes of data per frame (CAN FD). The devices that typically interconnect CAN are sensors, actuators and control devices. CAN has no built-in default security mechanisms, such as encryption, therefore organizations are expected to implement their own security mechanisms such as authentication of network commands or devices. If an inadequate implementation is done or a security mechanism is not implemented, then the protocol is susceptible to, for example, package interception as well as MITM, because CAN messages are broadcast. Additionally, Reference [180] mentions other possible attacks like DoS or replay attacks and concludes that the opponent will be able to send valid messages and thereby control

the physical components of the vehicle or system. Reference [180] establishes an authentication model based on SHA-1 due to the efficiency and speed for the HMAC, also implemented the timestamp for the verification of the message. This implementation did not imply a severe impact on the transmission speed and latency and the prevention of both the DoS attack and the replay attack was demonstrated. However, the implementation was carried out on a test network with only two nodes.

*2.3.4 Siemens S7 (TCP: 102).* PLCs are responsible for controlling critical processes and they are considered essential to field automation [181]. Siemens Simatic S7 or Simatic STEP 7 is a PLC product line that succeeded to Simatic S5. Siemens PLCs use two protocols to communicate via Ethernet through communication processors (CP): Open TCP/IP and S7 Protocol. For instance, CP 1543-1 is a communication module to support TCP/IP and multicast over UDP connection [182]. When the PLC S7 (i.e., S7-1500) uses the CP module (i.e., CP1543-1) is always the server (passive connection establishment) [182]. "Open TCP/IP" is a TCP/IP protocol implementation, dedicated to connect PLCs with non-Siemens hardware [183]. S7 protocol is known as the backbone of Siemens communications. This Ethernet deployment is based on "ISO TCP (RFC1006)" that, for design, is a block-oriented system. S7 Protocol, ISO TCP and TCP/IP use the encapsulation model defined by OSI in which the protocol data is the "payload" of the subsequent protocol [183]. S7 is a command/reply-based protocol, in which each interaction is a command or a reply. S7 uses PROFINET, which is based on Ethernet, and is currently regarded as the most reliable "fieldbus." The PROFINET protocol is used for communication between PLCs and IO modules. Basically, PROFINET infrastructure are "industrial Ethernet cables," known as industrial "Cat5" or "two-pair Cat5," for connecting industrial fieldbus systems via TCP/IP. They are desirable for fixed or flexible and dynamic industrial automation applications, as they offer outstanding resistance to both active and passive electrical interference, as demanded by PROFINET and Cat5e specifications. Siemens recommends security measures based on defense in depth (DiD) strategies to minimizing risk of products like S7-300/400/WinAC/1200/1500 [184]. However, S7 is sensitive to attacks such as "spoofing," "session hijacking," and "denial of service" [185]. Other scenarios use Frameworks like Metasploit to carry out attacks as replay [181].

*2.3.5 EtherNet/IP (TCP: 44818, UDP: 2222).* Rockwell Automation designed EtherNet/IP or "Ethernet Industrial Protocol" in the decades of the 1990s. It combines Ethernet with "Common Industrial Protocol" (CIP). CIP covers a complete set of messages and services for a wide range of manufacturing and process automation applications (Table 1). Open DeviceNet Vendor Association, Inc. (OVDA) maintains both EtherNET/IP and CIP. EtherNet/IP provides a certified standard for the development of automation devices with the below properties: (1) It employs Ethernet, (2) it is based on a broadly accepted CIP protocol layer, and (3) it is a verifiable standard. CIP is a protocol used in the transfer of automation data between two devices. The CIP protocol represents each device in the network as a set of objects, where each object is just a gathering of data values related to a device [50]. CIP defines three types of objects: application objects, required objects, and vendor-specific objects. The required objects contain three other types of objects: identity object, message router object, and network object [50]. The protocol lacks built-in security protections and the cybersecurity for EtherNet/IP and CIP is built on a defense-in-depth approach based on external mechanism [154]. ODVA defines some best practices for different types of industrial network installations for EtherNet/IP[186], where it includes isolating the control network with a single and multiple controllers, VLAN, Firewalls, and DMZ. However, some vulnerabilities such as improper input validation have been confirmed and exploited by Reference [187].

*2.3.6 Hart-IP (TCP: 5094).* The HART protocol "Highway Addressable Remote Transducer" is a universal standard to send and receive digital information via analog 4–20 mA cables between smart devices and control systems [188]. More recently, HART has been extended to include communication across IP networks (HART-IP) [188] as well as wireless mesh network (Wireless HART) [189]. HART-IP was built to enable HART devices over Ethernet. This implementation includes a conventional client/server architecture [190]. The client can be either a host system or a host application while servers can be Wireless HART gateways, HART multiplexers, HART Remote I/O or individual HART devices [191]. Client/server communication utilizes either/both UDP and TCP transport. In addition, servers support a minimum of two simultaneous client sessions [192]. Since HART-IP is predominantly used within the plant perimeter, security measures should be employed to protect the data during transport: firewalls, Virtual Private Network (VPN) tunneling, SSL, and remote authentication [193]. In addition, it is recommended to restrict HART-IP to internal networks. Traffic from HART-IP (usually UDP or TCP port 5094) should be restricted to a management VLAN segment with strong network controls. Some vulnerabilities, including DoS, in HART-IP networks are demonstrated by Reference [194].

*2.3.7 BACnet (UDP: 47808).* The "American Society of Heating, Refrigerating and Air-Conditioning Engineers," also known as ASHRAE, designed BACnet, which is the acronym for "Building Automation and Control Network." In general, BACnet is a communication protocol and a standard designed for both building automation and control systems (see Table 1). The BACnet protocol determines the way and which messages (data frames) may be transmitted from a device/system to other. BACnet's architecture is formed by four layers (OSI model oriented): "Physical Layer" (PL), "Data Link Layer" (DLL), "Network Layer" (NL), and "Application Layer" (AL), although only NL and AL are purely BACnet. At a data link and physical layers level, BACnet uses several protocols, including "Ethernet," "BACnet/IP," "ARCNET," "MS/TP," "Lon Talk," or "PTP" (point-to-point) as detailed in Reference [195]. BACnet/Ethernet is used directly with Ethernet IEEE 8802.3 networks. It may run on different physical supports, such as cable or optic fiber. It is limited to physical infrastructure that only uses MAC addresses to establish communications. BACnet MS/TP is based in the master-slave model or the token passing model in the data link layer [196]. This BACnet variant uses a serial channel, typically RS-485, for communications [196]. The BACnet PTP type of media access control is only used over telephone networks [197]. The EIA-232 type of direct connection is less and less used, and Ethernet is usually preferred in its place. BACnet over ARCNET is variant allows using BACnet over a coaxial cable or a RS-484 serial cable. It improves slightly the BACnet MS/TP features, but few manufacturers support it. BACnet/IP is a standard that uses IP addresses and ports, which allows it to run over the available Ethernet architecture, as well as to use of VLAN networks [198]. The messages are transported using UDP. It is different from BACnet/Ethernet in the fact that it uses IP addresses instead of MAC addresses. BACnet uses broadcast messages. For instance, to identify connected devices, BBMDs ("BACnet/IP broadcast management devices") are needed [199]. BACnet protocol security includes security keys, encryption models or authentication systems [200]. BACnet defines six different types of security keys to be used depending on the task to be carried out. Keys are distributed to all devices from the network security key server, some of which are even distributed jointly. It is not necessary to assign the security key server function to a specific device, but there must be a server containing all keys and a list of all devices to be managed [200]. BACnet applies message security at the network layer. BACnet non-encrypted messages are placed in the data section of a new secure message. This encryption is subject to four network security policies depending on whether hardware security, protocol security or no security is applied [200]. In all cases either where a message is deprived of security measures, in the target device or because the target device has a different security policy,

messages must be authenticated. Such authentication consists on validation of source MAC address, an unique message ID, a time stamp, and the message signature [200]. Reference [201] mentions BACnet devices implementations could be vulnerable to malformed packets and other types of attacks, so they can be considered un-robust and unreliable for handling irregular traffic. Techniques of attacks against BACnet as attacks on BACnet routing, network mapping, DoS, and spoofing are included by the tool BAF (BACnet Attack Framework) [202]. BACnet Anomaly Detection Framework (BADF) offers a convenient approximation of BACnet's current attack surface [203].

*2.3.8 Modbus (TCP: 502, TLS: 802).* Modbus was established in 1979 by Modicon as a serial communication protocol, open-standard to be used by programmable logic controllers (PLCs). Modbus actually is open protocol for industrial networks, which recently includes several environments such as "building automation" or "energy management systems" (see Table 1). The Modbus functions are to control: PLC, HMI, and I/O devices or sensors. Modbus can be used over Ethernet as well as serial cable. There are three established variations of the Modbus protocol: "Modbus ASCII," "Modbus RTU," and "Modbus TCP/IP." Modbus was originally developed using ASCII character to encode messages and this version of the protocol is still in use today. Modbus RTU is by far the most common implementation, based on the use of binary code and CRC error validation. Modbus RTU devices typically use one of three electrical interfaces: RS232, RS485, RS422, and a master-slave architecture. SCADA/HMI systems typically would be the master, communicating with a series the Modbus slaves' device (e.g., PLCs). A Modbus serial network has a master device, which issues commands to the slave devices. The slaves will not transmit information unless they receive a command to do so. The big difference with Modbus TCP/IP is that a Modbus Application Header (MBAP) is inserted at the beginning of each message. Modbus TCP/IP uses the terms client and server instead of master and slave. Modbus clients send the commands (e.g., SCADA/HMI). [204]. Modbus has two strong constraints: (1) it has a limit to 240 devices per network, (2) although the protocol is handled and defined by the organization itself, numerous vendor extensions are owned but without documentation, leading in interoperability problems. The Modbus organization released security specifications, which provide robust protection by combining TLS with the traditional Modbus protocol. TLS will encapsulate Modbus packets to enable both authentication and ensure message-integrity [205]. The new protocol/mechanism takes advantage of the X.509v3 digital certificates for server and client authentication [205]. Reference [206], developed a proof of concept by implementing TLS over a Modbus channel for smart grids. The results determined that the solution accomplished request/response times significantly below the 16.67 ms period of the 60 Hz grid cycle, demonstrating a minor effect on smart grid applications. In addition, Reference [207], addresses the security problems of the Modbus protocol, through a new secure version of a RBAC model, which takes advantage of the authentication provided by TLS, as well as granting authorization of the client on the server, as well as of the Modbus frame. However, Modbus is sensitive to classic information security threats as described in Reference below [172]. A summary of the attacks according to their threat categories, targets and impact on the control system assets is presented by Reference [208]. Other attacks such as MITM and DoS attacks are demonstrated by Reference [209].

*2.3.9 OPC-DA (TCP: 135).* One of the greatest attempts for automation software standardization in the last years has been the access to device automation data, with several protocols, different bus systems and interfaces are available. The result was OPC-DA (Open Platform Communications-Data Access). After, other two important OPC specification was developed: Alarm and Event, abbreviated A&E [210], and Historical Data Access, abbreviated HDA [211]. The client/server architecture in the Microsoft components was the disruptive aspect of OPC. An OPC server encloses the information generated by the industrial process and makes it

available through its interface. An OPC client connects to the OPC server to access the available information (see Reference [212]). Object mapping performed by OPC Classic is implemented via Microsoft technologies such as COM ("Component Object Model") and DCOM ("Distributed Component Object Model"). OPC Classic does not define security as part of the specifications, so it is delegated to DCOM/COM protocols [213]. The OPC Foundation establishes guidelines to configure the DCOM/COM layer to provide security mechanism [214]. In general, OPC Classic supports signing of the data flow and encryption among systems, identification and securing applications and user access rights. However, OPC tunneling solution provides security robustness [215]. OPC tunneling aims to eliminate DCOM, which is usually performed by exchanging the DCOM network protocol with TCP. The connection is set up between the OPC client and the OPC tunneling application that acts as an OPC server [215]. Other security recommendations to reduce the attack surface, layering defenses and defense in depth are analyzed through a practical example [216]. OPC Classic can be set up to provide security, but these security features are provided by the functionalities in Windows and DCOM/COM. However, it requires a lot of configuration knowledge, besides some environments do not support security configuration and some applications could not support some security configuration [213]. For example, OPC Classic uses DCOM, which employs RPC internally. By default, RPC uses dynamic port mapping. This means that it is very difficult to set up a firewall, since a large number of ports must be opened [217].

*2.3.10 OPC-UA (TCP: 4840).* Aimed at the SOA (Service Oriented Architecture) paradigm, the OPC Foundation has created the new standard OPC-UA (Open Platform Communications-Unified Architecture) that unifies the entire specification of OPC Classic (OPC-DA): "Data Access," "Alarms & Events," "Historical DA," and "Complex Data and Commands." OPC-UA (Unified Architecture) can be implemented in a wide range of device types such as 16-bit up to 64-bit architectures x86, ARM, and PowerPC. It is also agnostic to the Operating System (OS), so it can be deployed in Windows, Linux, VxWorks and different Real Time (RTOS) [218]. The UA workgroup set up a binary protocol built on TCP, which supplies the communication stack in three default-programming languages: "NET," "ANSI C," and "JAVA." OPC UA separates services from the implementation language protocol, which is the basis for flexibility and usability in domains even outside the classical communication model, SCADA-PLC, HMI-PLC, e.g., integration of the OPC-UA server with field devices can be with much reduced footprint [219]. Smart Grids is a field of action of OPC-UA with application in a wide range of devices ranging from controllers of wind or photovoltaic power plants to systems like SCADAs or Energy Management Systems (EMS) [220]. OPC UA is secure-by-design addressing security next concerns: (1) authentication of Users, application instances (Software), (2) confidentiality and integrity by signing and encrypting messages, (3) availability by minimum processing before authentication, and (4) auditability by defined audit events for OPC UA operations [221]. However, a full test of the reference implementation of the OPC UA communications stack, reported through Reference [222], revealed that some bugs detected in the dynamic code analysis have a significant negative impact on server availability due to memory utilization or failures. In addition, sequenceNumber is not proven in UA Secure Conversation, i.e., a security vulnerability. The proof of concept of the vulnerability exploitation is referenced in [222]. However, it is recommended to carry out specific proof of concept (PoC) of: (1) "Replay attacks exploiting missing tests of the sequenceNumber," (2) "Exploitation of compromised certificates whose validity is not detected due to missing tests of CRLs," and (3) "DoS attacks exploiting memory leaks" [222].

*2.3.11 DNP3 (TCP: 20000).* DNP3 is made up of a suite of protocols employed in electrical grid automation environments. The Distributed Network Protocol was developed by GE-Harris Canada in 1990, has been extensively implemented in utilities such as electrical, water, sewage, oil and gas.

It was created for SCADAs interaction environments; the protocol enhances the data acquisition information and control commands transmission from master (control centers) to remote stations (remote computers) via event-based data reports. In addition, DNP3 was designed so that data acquisition and control equipment can interact. In addition, it is widely employed for communications from "master stations" to "remote terminal units" (RTUs) or "intelligent electronic devices" (IEDs) [223]. The DNP3 stack is divided in three layers: link, transport and application. The ability to transport generic data is given by the independence with respect to the channel, for example: serial or TCP/IP. Application Data Service Units (ASDUs) are entities that transmit a combination of functions code and objects through the application layer in a standardized data format, so that it can be used by the lower layers [224]. ASDU messages are divided into fragments. The maximum fragment size is associated with the implementation as it is defined by the buffer size of the receiver device, e.g., a normal range is 2 to 4 KB. A message larger than a fragment will require some fragments. Message fragmentation is delegated to the application layer. The DNP3 protocol provides a secure authentication (DNP3-SA), which is a one-side authentication (TLS with certificate authentication) procedure employed to protect the DNP3 messages transferred among interconnected stations are secure from unauthorized applications. The system works in two modes: "Non-aggressive" and "Aggressive" modes. DNP3 is an early "standardized SCADA protocol" that aims to enable a cryptographic security embedded in the operations [225]. Another way to provide a TLS-based connection to DNP3 is through IEC 62351-5, with which DNP3 is compliant. DNP3 has integrated the IEC-62351 security requirements/capabilities in the "IEEE 1815 DNP3" standard [226]. However, DNP3 vulnerabilities was demonstrated using attack patterns like fuzzing [227] and crafted malformed frames [228]. Since 2013 to 2014, over 33 CVEs related to input validation with DNP3 implementations [229].

*2.3.12 ANSI C12.22 (TCP: 1153, UDP: 1153).* Since the datasets, data structures, and communications protocols for electricity meters were all exclusively proprietary, ANSI (American National Standards Institute) standards were established to describe the datasets and data structures of the meters (C12.19), and to enable an optical point-to-point optical communications protocol (C12.18) that would enable them to interface with ANSI standard meters. To allow the transmission and reception of this information, C12.18 was adapted to set up C12.21 for telephone modems, using point-to-point communication. Later, C12.22 was structured to include other communication networks such as TCP/IP or UDP/IP and SMS on GSM [230]. The ANSI C12.22/IEEE 1703 protocol defines the scope of "ANSI C12.22/IEEE 1703 Advanced Metering Infrastructure" (AMI) Application Layer message transport over a TCP/IP network in the smart grid environment [230]. The C12.22 standard specifies both a transportation-independent application level protocol for information exchange among nodes and a physical and data link protocol for meter connection and communications technology. [231]. The "C12.22 IP" communications system include several elements, as Reference [232] detailed "C12.22 IP Node," "C12.22 IP Network Segment," "C12.22 IP Relay," and "C12.22 Gateway." The protocol units are ANSI C12.22 / IEEE STD 1703, IETF RFC 6142, IGMP with UDP multicast, TCP or UDP transport, Abstract Syntax Notation One (ASN.1) [233], and Object identifiers (OIDs) [234]. In the ANSI C12.22 frame, it is possible to verify the authentication mechanism used, as well as the user information exchanged through an EPSEM message. "Extended Protocol Specification for Electronic Measurement" (EPSEM) provides the ability to link commands to manage communications across a multi-node medium. A full deployment of the ANSI C12.22 security and the authentication process, coupled with the ANSI C12.19 event logger, enables a utility to achieve all the required features. As Reference [231] details, within the security mechanisms that can be included are: "encryption," "authentication," "credential management," "intrusion detection," "logging," and "auditing of all changes in data and configuration." However, Reference [235] describes the next attack scenarios against EAX-prime, a "standard security

function" used by ANSI C12.22 as authenticated encryption: "attacks exploit the wrong tweaking method of CMAC (Cipher-based Message Authentication Code) in EAX," "plaintext Recovery Attacks," "distinguishing attack," and "forgery attack."

*2.3.13 IEC-61850/IEC-62351 (TCP: 102).* IEC-61850 is a popular protocol in the smart grid sector. The IEC-61850 standard was initially designed for substation automation but has expanded into other domains such as wind power plants, hydropower plants, microgrids and distribution automation domains [236]. The protocol was built to enable interoperability between vendors, allowing devices to define its intrinsic functionality and simplifying its communication. IEC-61850 unifies the different functions such as measurement, control, protection and monitoring. End devices to IEC-61850 are intelligent electronic devices (IEDs), which are classified based on their function such as relay devices, voltage regulators, circuit breakers, and so on [237]. A typical IEC-61850 substation architecture includes two kinds of communication bus: Substation Bus (SB) and Process Bus (PB) in the substation, which connects all the IEDs. Both SB and PB mapped over Ethernet medium, however they may have different bandwidths, e.g., SB (10/100/1000 Mbps) and PB (0.1/1/10 Gbps). SB handles the requests/responses and general event substation messages. Generally, there is only one global SB. However, PB interfaces IEDs to traditional devices such as merge units. There can be more than one PB inside the substation [238]. In a very simple way, an IED is a physical device (PD) that hosts all logical devices (LD). PD connects to the network via network address. LD is a collection of LN (Logical Nodes) (e.g., a breaker). LN is the core that constitutes a single functional unit to power automation environments. LNs are stand-alone devices and can be set up flexibly on any IED. LN contains data (e.g., position (pos) and operation count (opcnt)) and objects [239]. IEC-61850 substation interactions can be grouped into following three categories: data monitoring/reporting, data gathering/setting, and event logging. To realize the above mentioned interactions IEC-61850 standard has defined a fairly complicated communication structure that defined five types of communication profiles: Generic Object-Oriented Substation Events (GOOSE), Sample Value (SMV), Simple Network Time Protocol (SNTP), Abstract Communication Services Interface (ACSI), and Generic Substation State Events (GSSE) [238]. IEC-61850 includes several basic security features, although these differ according to the part of the protocol being inspected. Thus, no security options exist in this IEC-61850 version 1 and 2 [240]. The use of a secured tunneling protocol such as TLS (with client certificates) or VPN guidelines can be found in the IEC-62351 standards. The IEC-62351-4 enables security over MMS and the IEC-62351-6 defines over GOOSE and SNTP. IEC-62351-6 recommends that the use of VLANs is required for GOOSE [240]. IEC-62351-7 emphasizes on Network and System Management (NSM) of the "information infrastructure," which defines data objects for the power system operational environment, which contains the information needed to manage the "information infrastructure" as reliably as the power system infrastructure is managed. The NSM data objects can be assigned to IEC-61850 [226]. However, Reference [241] identified some weaknesses in the IEC-62351standard, which were documented as penetration tests to perform the protocol. Within these weaknesses with an attack pattern associated are found: Replay After "stNum" Reset in the GOOSE Protocol, Cross Receiver Replay in the Sampled Values Protocol and Attacks on the Simple Network Time Protocol [241].

*2.3.14 IEC-60870-5-104/IEC-62351 (TCP: 2404).* IEC-60870-5 is one between the six parts of IEC-60870 standard, which sets mechanisms applied to tele-control systems especially SCADA systems in power system automation and electrical engineering. Part 5 describes the communication module used to send tele-control messages between two directly linked systems. Tele-control refers to the sending of monitoring data and requests for data collection to control power transmission grids. This part contains seven documents defining the standard tele-control, tele-protection, and related telecommunications for electrical power systems. IEC-60870-5-101

(IEC-101) and IEC-60870-5-104 (IEC-104) are the protocols that meet these standards. The IEC-104 protocol is an analogy to the IEC-101 protocol that adapts the functions defined by IEC 101 to a TCP/IP network [242]. The IEC-104 telegram structure is composed of three sub-layers (1) Application Protocol Control Information (APCI), (2) Application Service Data Unit (ASDU), and (3) Application Protocol Data Unit (APDU). Since the IEC-104 protocol transmits clear text messages without any authentication mechanism, it is the target of different attack patterns [243]. Clear text data transmission is a potential risk for "eavesdropping," "sniffing," and "tampering" for substation [243]. The lack of authentication of the commands questioning, remote control and remote tuning, allows potential exploiters to gain non-authorized access to SCADA systems, breaking the integrity and the availability, as well as releasing spoofing attacks, replay attacks and MITM attacks [243]. However, one way to provide a TLS-based connection to IEC-104 is through IEC 62351-5 (as with DNP3) [226]. In addition, IEC-104 is adopting IEC-62351 (as is the case with IEC-61850). A sample of this is that NSM data objects can be assigned to IEC-104 [226].

*2.3.15 IEC-60870-6-503/IEC-62351 (TCP: 102).* The IEC-60870-6-503 (also known as TASE.2 or Inter Control Center Protocol (ICCP)), connects control centers (e.g., "Independent System Operators" (ISO), "Regional Transmission Operators" (RTO), and some generators [244]). ICCP is a complete modern client/server protocol, i.e., relies on TCP/IP [244]. ICCP defines a mechanism for critical data sharing among locations. ICCP enables both "real-time commands" and "historical monitoring" by incorporating an "object-oriented" layout in which devices are objects with related behaviors. Objects could be specific devices (e.g., "transformers" and "relays") or abstract data structures (e.g., transfer sets). By default, IEC-60870-6 ICCP/TASE.2 is not secure, since it is a protocol that transmits in plain text without any mechanisms to ensure confidentiality or integrity. However, it promotes IEC-62351-4 like secure TASE.2. Then, some strategies to securing ICCP like VPN are analyzed in Reference [244]. However, ICCP can include mechanisms to ensure communication such as 1,024-bit asymmetric key length implementations and multi-certified by link. Asymmetric key length implementations of 1,024 bits are broadly supported now and are usable without the obsolete hashes and cyphers. Certificates by link allow certificate expirations to overlap to permit certificate updates with minimized effect on data transmissions. Like DNP3 and IEC-60850, ICCP complies with IEC-62351-3, protecting against eavesdropping and replay attacks through TLS encryption, against human security risk in the environment through message authentication and against spoofing through security certificates (node authentication); while like IEC-60850, it complies with IEC-62351-4 for using MMS [226]. However, from Reference [227], it is established that ICCP vulnerabilities were demonstrated using attack patterns such as fuzzing.

We can conclude that all protocols, standards, and buses analyzed during the survey have the security issues as a common factor, i.e., vulnerabilities associated with either design, implementation, or operation. At this point, we are unable to recommend which protocol, standard, or bus is better or worse for a category (e.g., IT or OT) or IIoT classification (e.g., FAP, PAP, ISCP, BAP, PSAP, AMRP, SHAP, or VAP). For this reason, an in-depth study of the vulnerabilities, which affect these IIoT protocols, standard and buses is required. This study is carried out via methodological framework. The results returned once the framework is applied can be used as a parameter to evaluate the risk in assets. In that sense, the Common Vulnerability Scoring System (CVSS), is an essential component of our framework as an evaluation tool for industrial environments. Since the CVSS categorizes the vulnerabilities detected in a specific asset, based on the implementation and operation of the protocol, and not from the perspective of the design of the protocol itself, our framework focuses on the analysis of implementation and operational vulnerabilities of the 33 protocols analyzed during the survey.

## 3 CVSS AS EVALUATION TOOL FOR ICS SYSTEMS

CVSS is a scoring system that offers a standard and opened method to estimate the severity of a vulnerability. Its use is widespread, especially for IT environments, but when this framework is used to determine the severity of vulnerabilities that affect the industrial devices, different problems arise. To illustrate the problem mentioned next an example. If it is considered, on the one hand, a pacemaker programmer Medtronic 2090 Carelink, to which was associated the vulnerability CVE-2018-10596 and, on the other hand, a digital backbone (Schneider Electric ExoStruxure), on which the vulnerability CVE-2018-7797 has been found. Although the first vulnerability can be used to kill someone and the second used to execute a phishing attack, the CVSSv3 base categorizes the first with a severity of 7.1 and the second with a severity of 7.4. For this reason, the second stage of our manuscript establishes a methodology to describe a vulnerability for an industrial environment, additionally allows us to establish a comparison of the impact of suffering the same vulnerability in an industrial environment (OT) regarding an IT environment. To achieve this, we will first analyze the pillars of cybersecurity due to their high value for the CVSS, then we will briefly analyze the CVSSv3 with respect to CVSSv2, then introduce the improvements to CVSSv3 proposed by CVSSv3.1, and finally, we will review other proposals of methodologies that include strategies to improve CVSS for an industrial environment.

### 3.1 Cybersecurity Pillars

As mentioned in Section 1, the benefits of convergence between IT and OT bring associated risks. The first point at which risk manifests is the importance that each environment assigns to the three-cybersecurity pillars. Table 2 illustrates the degree of importance of each pillar according to the environment: IT or OT. Therefore, for an IT environment, it is more important that information is not made accessible or distributed to non-authorized individuals, entities, or processes, whereas in OT environments, it is a requirement on-demand, timely, and reliable access to and use of information by an authorized user, i.e., available. However, beyond the fact that the pillars contribute to accentuate the differences between environments (IT and OT), these and other metrics are the base of the CVSS score, which is briefly analyzed in the following section.

### 3.2 CVSS in Industrial Environment

The CVSS scoring system allows us to standardize a vulnerability scoring methodology that is agnostic to any platform, so it is an open framework, which provides visibility to individual characteristics and the methodology used to obtain a score. The CVSS is composed of three metric groups: Base (BM), Temporary (TM), and Environmental (EM).

According to CVSSv3.1 specification (released on June, 2019), the BM reflects the severity of a vulnerability according to its intrinsic characteristics, which are constant over time and have a reasonable impact in the worst case in the different environments deployed. The BM includes the set of metrics the exploitability metrics (EM) and the impact metrics (IM), where the EM measure the technical means and facility with which vulnerability is exploited and the IM involves the three-cybersecurity pillars (Table 2). The TM adjusts the BM severity of a vulnerability based on factors that change over time, such as the availability of exploitation code. The EM adjusts BM and TM severity to a specific environment. Figure 3 contains all the metrics that make up BM, TM, and EM, which by default are represented by a string to which a numerical value is associated and then, allow to compute each metric score from of the equations included in the specifications. Since it is typical that only the BM are published, since these do not change over time and are common to all environments, it is therefore common to face two challenges when working with the CVSS. On the one hand, the calculation of TM and EM, and on the other hand, the fact that

Table 2. IT-OT Cybersecurity Pillars Comparison

| Order | IT | OT |
|---|---|---|
| 1 | Confidentiality | Availability |
| 2 | Integrity | Integrity |
| 3 | Availability | Confidentiality |

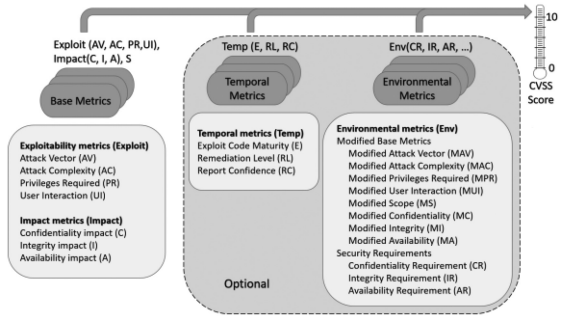


Fig. 3. CVSS metrics and equations.

Table 3. CVSSv2, CVSSv3, and CVSSv3.1 Comparison

| CVSSv2 | CVSSv3 | CVSSv3.1 |
|---|---|---|
| Vulnerabilities are rated according to the overall impact on the platform. | Vulnerabilities score according to the impact on the impacted component. | Vulnerabilities score according to the impact on the impacted component. |
| There is no knowledge of instances where a vulnerability in one application impacted other applications on the same system. | The Scope metric allows to indicate if the vulnerability affected other components of the system. | The metric Scope, Vulnerable Component and Impacted Component concepts are reformulated to clarify them. |
| Access Vector can combine local system access and physical hardware attacks. | Local and Physical values are separated in the Attack Vector metric. | The descriptions of the values (Network, Adjacent, Local and Physical) of the Attack Vector (AV) metric are reformulated. |
| Access Complexity combined system configuration and user interaction. | Access Complexity has been separated in Attack Complexity and User Interaction. | The access complexity metric of the attack eliminates an ambiguity in its description. |
| Authentication metric leaves out many aspects of vulnerability. | The required privileges reflect the privileges necessary to affect the attack. | When scoring impact, to consider the privileges the attacker has prior to exploiting a vulnerability and compare those to the privileges they have after exploitation. |
| Impact metrics reflected percentage of impact caused to a vulnerable application. | Impact metric values reflect the degree of impact, and are renamed to None, Low and High. | It is specified that only the increase in access, privileges gained, or other negative outcome as a result of successful exploitation are considered to score the impact metrics of a vulnerability. |
| The environmental metrics are not considered | The environmental metric allows to understand how vulnerability is reflected in an application environment. | Change to Modified Impact Sub-formula in Environmental Metric Group. |

many vulnerabilities have been determined only for CVSSv2 and not for CVSSv3 or for CVSSv3.1. Table 3 highlights the main improvements that CVSSv3 introduces over CVSSv2. In addition, it should be noted that changes between CVSSv3.0 and CVSSv3.1 clarify and improve the standard without introducing new metrics or metric values, and without making major changes to existing equations, hence the need to adopt it.

### 3.3 Efforts to Adapt CVSS to Industrial Environment: Related Work

To overcome the aforementioned problems presented by CVSS (Section 3 introduction) to describe vulnerabilities in industrial environments, several experts in the sector are looking

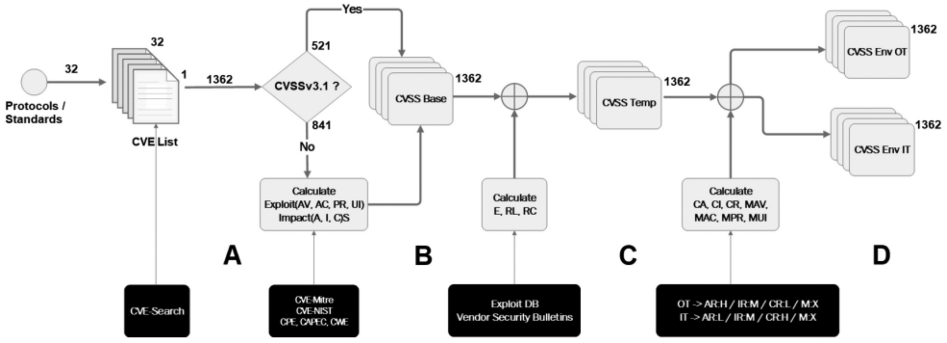


Fig. 4. Methodology and information data sources.

for alternatives among which are RSS [245], TEMSL [246], and IVSS [247]. RSS (Risk Scoring System) is an exclusive alternative for vulnerabilities in the health, aviation and weapon sectors, which proposes to incorporate new factors, such as the duration of the attack or the chain of exploitation. IVSS (Industrial Vulnerability Scoring System) is a calculator and a description of the metrics and factors evaluated in it such as Base Severity Level (BS); Base Exploitation Level (BEX); accessibility, impact, and consequences; base score and IVSS final score, made up of all the previous values. However, IVSS is a project still in development. TEMSL (Threat, Exposure, Mission, Safety, and Loss) performs the calculation associated with the severity of vulnerability with decision trees. These trees qualify threats as evidence of exploitation of vulnerability; exposure as the extent of access (external or local network); mission as operation, service delivery, and data protection; physical security as injuries or deaths and losses as costs associated with exploitation of vulnerability beyond mission or physical security. However, as the associated footer shows, there is no formal (e.g., a white paper) or scientific reference associated with the project yet. As a summary of this section, we can conclude that there is a need to establish an index that can characterize the severity of a vulnerability in industrial environments, for that reason, numerous projects such as RSS, IVSS and TEMSL are making efforts to develop a model that meets the requirements. However, since there is no established methodology yet, in the next section, we will present our methodology through the VAF framework.

## 4 METHODOLOGY AND THE INFORMATION DATA SOURCE

Before proceeding to make our proposal, the advantage provided by working with CVSSv3.1, which is the basis of our methodology, should be emphasized, because as we have mentioned it leads to a risk analysis (Section 3.2). Therefore, through CVSSv3.1 our methodology solves two concrete problems: (1) the framework allows for risk analysis; (2) the framework is coupled to an industrial environment. To perform a risk analysis from CVSSv3.1, both temporal and environmental metrics must be calculated and then complemented by external factors such as exposure and threat.

To link the analysis to an industrial environment, the environmental metrics are contextualized for both an IT and an OT environment, and the behaviors that would have the same vulnerability if exploited in one or the other environment are analyzed. A methodology known as the Vulnerability Analysis Framework (VAF) was designed to enable the vulnerability analysis (Figure 4). VAF is powered by open source data sources, which are described throughout this section. These data sources are classified into dictionaries, Open Source Intelligence (OSINT) and integration tools.

1. CVE-Mitre (OSINT) is the acronym for "Common Vulnerabilities and Exposures" and is a set of items that includes an ID number (e.g., CVE-2017-2681), a descriptive statement, and a minimum of one published reference for well-known cybersecurity vulnerabilities [248]. Mitre Corporation supports CVE-Mitre.
2. CAPEC (dictionary) is the acronym for "Common Attack Pattern Enumeration and Classification." It is a complete dictionary and categorization of known attacks used by cybersecurity researchers, programmers, auditors (or cybersecurity testers) and academics to further enhance community awareness and improve defenses [249].
3. CWE (dictionary) is the acronym for "Common Weakness Enumeration," which is a set of common software security weaknesses created by the community. CWE is a standard used to describe the security weaknesses of software in architecture, design or code. It is also a benchmark for software security tools and a guideline for identifying, mitigating and preventing weaknesses [250].
4. CPE (dictionary) is the acronym for "Enumeration of Common Platforms." It is a structured naming framework and a uniform approach to describe and to identify several types of applications, OS, and hardware devices contained in an organization's data-processing assets [251].
5. EDB (OSINT) is the acronym for "Exploitation Database." It is a public exploits file compatible with CVE and the associated vulnerable software. EDB is supported by Offensive Security [252]. EDB has been developed for penetration testers and cybersecurity researchers [253]. The aim is to store the complete collection of exploits and POC collected via direct post, mailing lists. This collection is released as part of an open source repository [253].
6. MB (Integration tool) is the acronym for "Microsoft Bulletin," which is a Microsoft Security Response Center (MSRC) initiative [254]. MSRC publishes monthly security bulletins, which cover security vulnerabilities in Microsoft software. These bulletins describe both the issue and the fix to the vulnerabilities as well as provide links to updates relevant to the affected software.
7. In addition to CVE-Mitre, CVE-NVD is classified as integration tool. The NVD is the U.S. "National Vulnerability Database." NVD is a standards-based vulnerability database managed by the Security Content Automation Protocol (SCAP) [255]. NVD includes a comprehensive report, consisting of the following elements: a brief description of the existing vulnerability; impact metrics, references to alerts, solutions and tools; technical details including the type of weakness and known affected software configurations.
8. CVE-Search (integration tool) is a tool that incorporates lists of assets, dictionaries and vulnerabilities, including those: CVE, CWE and CPE. These elements are inserted into a MongoDB database to simplify the searching and processing of CVE [256]. The objective of the tools is to search for vulnerabilities in a local database. In addition to the backend where the data at rest is, CVE-Search includes a user-friendly web interface for vulnerability searching and managing, a set of tools to access the system, and a web API interface.
9. CVE-Details (integration tool) provides a compliant to use web interface to CVE vulnerability data [257]. Allows access to vendor information, associated assets, firmware releases, OS and CVE inputs, and associated vulnerabilities. It enables us to see vendor, products, and version statistics. CVE-Details data is taken from CVE-NVD and various sources such as EDB exploits, vendor declarations, and supplemental data provided by vendors. Metasploit modules are issued as well as NVD-CVE data.
10. Vulnerability Analysis Framework (VAF) methodology starts with CVE List that contains all CVE numbers from origin to mid-2018 for each protocol, standard and bus (Figure 4).

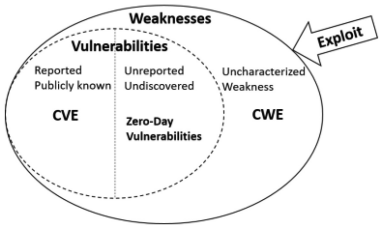


Fig. 5. Vulnerability, weakness, and exploit.

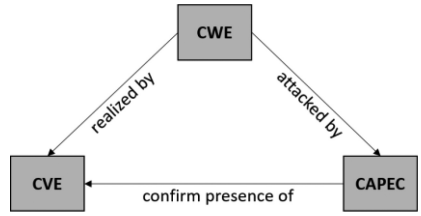


Fig. 6. Relationship between CVE, CWE, and CAPEC.

CVE List is an output of CVE-Search tool. Because of this first step, 1,363 vulnerabilities are collected, with quantities distributed very heterogeneously between protocols, standards, and buses (Figure 7). VAF then determines whether the vulnerability has the CVSSv3.1 calculated. CVSSv3.1 was calculated for 841 of the 1,363 vulnerabilities analyzed, assigning base metrics values, following analyst-scoring strategies based on both recovering and analyzing information from data sources such as CVE-Mitre, CVE-NIST, CPE, CAPEC, and CWE (Figure 4). With the base metrics standardized to CVSSv3.1, again scoring analyst strategies are followed to determine the submetrics E, RC, RL (Figure 3). Both EDB and vendors security bulletins (e.g., Microsoft) were the data sources (Figure 4). As a result, temporal metrics are normalized as well as base metrics. VAF uses the customizable nature of environmental metrics. Keeping the rest of the metrics constant, from Table 2, values have been assigned according to the importance of the cybersecurity pillar for the environment. In this way, it is determined whether the severity of a given vulnerability affects an IT environment more than an OT or vice versa. To customize the OT environment, a high value to AR, a medium value to IR and a low value to CR are assigned, while to customize the IT environment, a high value to CR, a medium value to IR and a low value to AR are assigned (Figure 4). As a result, the data analyzed in the following section expands to 1,363 custom vulnerabilities to OT environment and 1,363 custom vulnerabilities to IT environment (Figure 4).

### 4.1 Relationship Between CVE, CWE, and CAPEC

It should be noted that there is interaction between some of the information sources, for instance, CVE, CWE, and CAPEC. In this section, we will mention the interaction between CVE, CWE, and CAPEC, which will be used in our research within the results analysis section. To understand the relationship between CVE, CWE, and CAPEC, first, we must understand the relationship between vulnerability, weakness, and exploit.

As shown in Figure 5, there are weaknesses discovered, characterized, exploitable and possibly with mitigations, which are grouped within the CWE. However, there are also weaknesses in assets and protocols that have not been characterized. If a weakness is exploited from an exploit, then it becomes in a vulnerability. The vulnerabilities can be reported, publicly known and exposed through the CVE. However, unreported or undiscovered vulnerabilities may also exist. If previously unmitigated weaknesses are exploited with little or no warning, then they become in zero-day vulnerabilities. Since CAPEC describes the common attributes and techniques used by adversaries to exploit known weaknesses (e.g., SQL Injection, XSS, Session Fixation, Clickjacking), as Figure 6 illustrates, the relationship between CWE, CVE, and CAPEC is given, because a vulnerability (CVE) is the materialization of a weakness (CWE) through a known attack pattern (CAPEC).

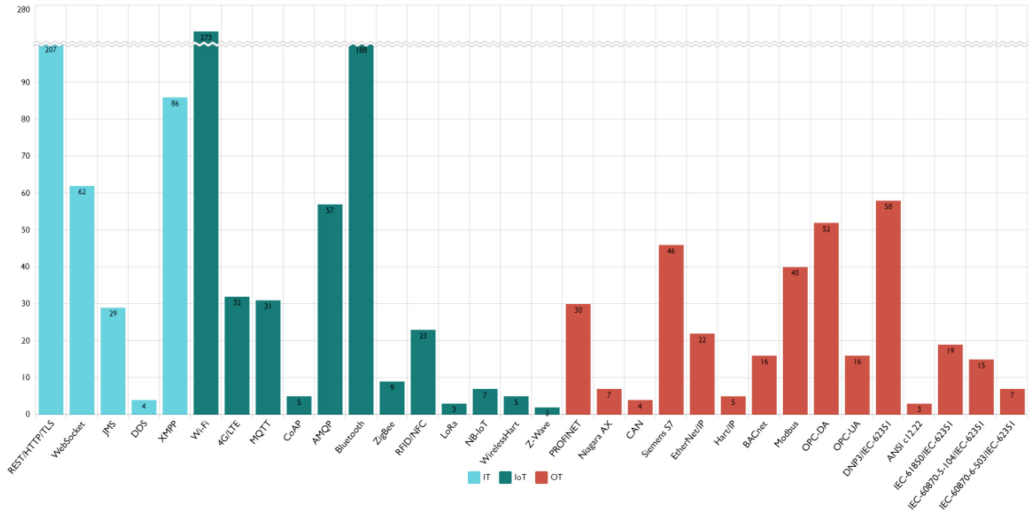


Fig. 7. Distribution of vulnerabilities by protocol, standard, and bus.

## 5 ANALYSIS RESULTS

This section summarizes the results obtained through the application of VAF on each of the 32 protocols, standards, and buses commonly used in IIoT environments. Figure 7 represents a distribution of documented vulnerabilities until the first half of 2018 (Q2, 2018) for 32 of the 33 protocols analyzed. In addition, it includes a division into categories based on the distribution (IT, IoT, and OT) of Table 1. Since 5G is a technology in the process of implementation and deployment, the following section presents the results found in the security field, because there is not any CVE yet. As Figure 7 illustrates, the distribution is highly heterogeneous.

Therefore, we can find a set of protocols, standards, and buses such as Bluetooth, REST/HTTP/TLS, and Wi-Fi where a large number of vulnerabilities are documented; others such as DDS, LoRa, and ANSI c12.22 with few identified vulnerabilities and another set, where WebSocket, AMQP, and MQTT are found, which maintains an intermediate number of documented vulnerabilities. We conclude by saying that the susceptibility of the protocol, standard, or bus depends to a high degree on its usability, i.e., if DDS were so popular and were on mobile devices, wearables, routers, among others, such as Bluetooth, then definitely the number of documented vulnerabilities would be greater. Therefore, the increased use of protocols through the integration proposed by IIoT will increase the interest of security researchers to discover new vulnerabilities in devices that implement these protocols, standards, and buses. However, Figure 8 provides a distribution by year of the 1,363 vulnerabilities studied. The contribution of each protocol per year allowed us to identify in the first place the age of protocols, standards, and buses, which entails security challenges, as well as, in the second place, to confirm the increase in the detection of vulnerabilities at present, highlighting the need to consider security from the design phase. First, OPC-DA reported vulnerabilities detected since 1999. Although in 2006 there was an increase in detection, the behavior was relatively stable until 2010. The disruptive year 2011 marks the beginning of a consistent growth in vulnerability detection. It should be noted that our analysis was performed until the middle of 2018, so the value reflected in the graph is lower. To analyze each of the 1,363 vulnerabilities, the methodology proposed by the VAF framework has been applied. The main purpose of the VAF application on 32 protocols, standards, and buses is to perform a risk analysis. To this end, we have divided our analysis into four steps that are derived throughout the application of VAF (5.2–5.6) and a

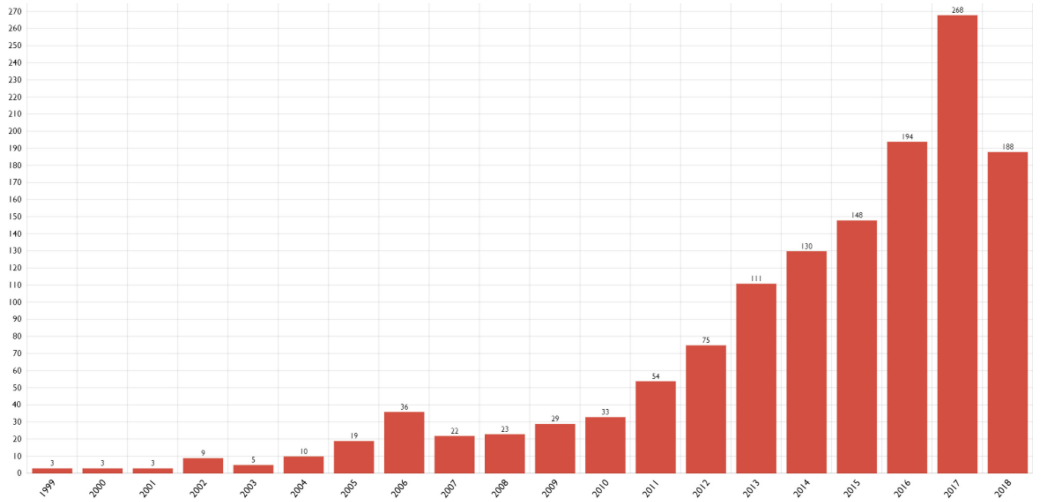


Fig. 8. Distribution of vulnerabilities by year.

classification proposed by NIST (5.6). Additionally, we have focused the first point of our analysis on the 5G technology, because currently there is no vulnerability of implementation documented with CVE, so in this case it has been impossible to apply VAF on this technology. Therefore, the analysis of results is divided into (1) recent attacks in 5G; (2) comparison between base and temporal metrics; (3) comparison between contextualized environmental metrics for both OT and IT environments; (4) impact of vulnerabilities on the pillars of cybersecurity: availability, confidentiality, and integrity; (5) impact of the cybersecurity pillars on IIoT categories; (6) attack patterns and weakness associated with vulnerabilities; (7) classification of vulnerabilities according to NIST Reference [4].

### 5.1 Recent Attacks in 5G

Currently, many countries have released or are close to releasing commercially 5G services, because the 3GPP group has developed the 5G standards where security procedures are included, as was mentioned previously in Section 2.2.3 [258]. As mentioned in Section 2.2.3, air security between mobile phones and mobile phone towers has been improved to overcome several threats, for example, through certain security measures fake base-station-type attacks (also known as IMSI or Stingray sensors) can be mitigated [258]. However, some of 4G's wireless features are reused in 5G. For instance, the 3GPP standards group has designed various capabilities in 4G and 5G specifications to support a wide range of applications such as smart homes, critical infrastructure, industrial processes, autonomous vehicles, and so on. This type of mechanism indicates to the network the type of device, i.e., a mobile device, a vehicle, an IoT device, so that it can receive specialized services and connectivity. A group of researchers through the works referenced in Reference [259] have found the following vulnerabilities:

1. A "protocol vulnerability" in specifications of 4G and 5G TS 33.410 [260] and TS 33.501 [261] allow a false base station to steal information about the device and mount identification attacks.
2. An "implementation vulnerability" in equipment of the cellular network operators that can be exploited during the "registration phase" of a device.

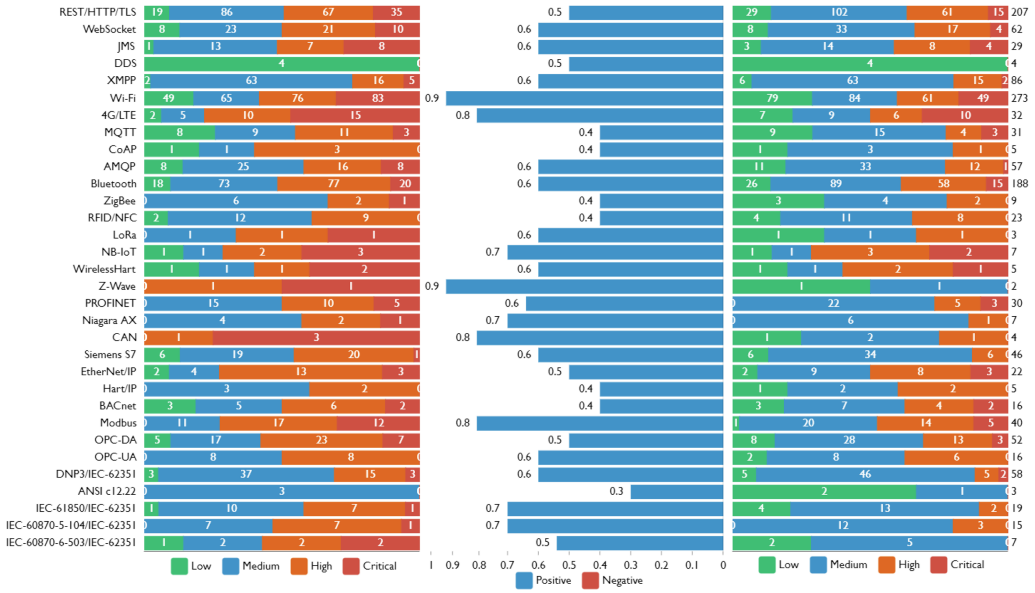


Fig. 9. From left to right: (a) BM, (b) Avg (BM) - Avg (TM), and (c) TM.

3. A "protocol vulnerability" in the first version of LTE NB-IoT that affects the "battery life" of low-power devices.

To each one of the vulnerabilities cited, the researchers associated the following attacks, respectively: mobile network mapping attack (MNmap) (active or passive), "bidding down" via MITM, and "battery drain" via MITM. The attacks demonstrations were announced at a prestigious security conference [259]. The vulnerabilities and attack patterns mentioned received responsible disclosure and were notified to GSMA through its Coordinated Vulnerability Disclosure Program (CVD) [262]. In addition, the researchers notified 3GPP, responsible for the design of the 4G/5G security specifications and the mobile network operators concerned via CVD-2019-0018 [258].

## 5.2 Comparison Between Base Metrics and Temporal Metrics

Based on the methodology proposed by VAF, following the risk analysis criteria of CVSSv3.1, the first step is to determine the base and temporal metrics. These steps correspond to stages B and C of Figure 4. From this point, an analysis is established between the behaviors of each protocol, standard and bus. Figure 9 summarizes the results obtained, which are divided into mapping of the vulnerabilities in Figure 7, distribution of the severity of the vulnerabilities for both the base metric (BM) and the temporal metric (TM), and determination of the difference between the average severity for the base metric and the temporal metric, which corresponds to the average BM and average TM index.

As shown in Figure 9 for each protocol, standard, or bus, the sum of the vulnerabilities distributed according to their severity coincides with Figure 7. For example, REST/HTTP/TLS contains 207 vulnerabilities analyzed. In addition, for each protocol, standard, or bus the vulnerabilities are divided according to their severity into critical, high, medium, and low for both base and temporal metrics. For example, of the 207 vulnerabilities analyzed for REST/HTTP/TLS for the base metrics, 35 are critical, 67 are high, 86 medium, and 19 low; while for the temporal metric 15 are

critical, 61 are high, 102 medium, and 29 low. Finally, to characterize the behavior of the protocol, standard, or bus in a general way, the average value of the severity obtained is determined for both the base and the temporal metrics. This result is shown in the middle of Figure 9. For example, for REST/HTTP/TLS the average severity of the vulnerabilities for the base metric is 6.7, while the average severity for the temporal metric is 6.2, resulting in 0.5 as the difference between them. For each of the protocols, standards, and buses, Figure 9 shows that the severity of the base metric is greater than the severity of the temporal metrics. The number of critical vulnerabilities decreases practically for each one of the protocols independently of the category IT, IoT, and OT. For instance, for REST/HTTP/TLS (IT) it decreases from 35 to 15, AMQP (IoT) decreases from 8 to 1, and Modbus (OT) decreases from 12 to 5. The result is expected, because the temporal metric has the "vulnerability remediation level factor," which measures the vulnerability patching level and indicates if the vulnerability has not been patched, presents a temporary solution or hotfixes that offers a temporary solution until a patch or official update is issued. If the temporal metric is closer to the base metric, then it will show a lower level of patching. The ANSI C12.22 protocol is a case where severity does not decrease between base and time metrics. Although we could argue that the severity of the detected vulnerabilities is medium and that it presents a small number of detected vulnerabilities (only three vulnerabilities), the truth is that at the time of the study, there was a high risk that some of the vulnerabilities could be exploited in some of the devices where it was detected. To establish which of the protocols, standards, or buses analyzed have the best results in this comparison, we have decided to compare the mean severities for the base and temporal metrics. Therefore, the protocols, standards, or buses that present a higher index, i.e., the difference of the average base severity is much greater than the average temporal severity, would theoretically be more secure. It should be noted that, although the result provides us with a classification it is not determinant, because although a protocol has a high index it is the result of the average of severities, which does not mean that one of the severities that has been averaged is, for example, critical and permanently affects the availability of the asset. Therefore, it is recommended not to take the following result out of context. From our index, we can affirm that the least susceptible protocols would be Wi-Fi (0.9), Z-Wave (0.9), CAN (0.8), and Modbus (0.8); while the most susceptible would be ANSI C12.22 (0.3), CoAP (0.4), MQTT (0.4), ZigBee (0.4), RFID/NFC (0.4), Hart/IP (0.4), and BACnet (0.4). Two paragraphs above mention that the remediation level (RL) of vulnerability is the main factor that determines the behavior that the temporal metric will have.

For this reason, Figure 10 is a representation of that factor, illustrating what percentage of the vulnerabilities of a specific protocols, standards, or buses has been patched or updated, presents a provisional solution, i.e., "workaround" or "hotfixes," against the percentage that does not present any type of solution. Each of these corresponding stages adjusts the temporal metric down, revealing the reduction in vulnerability level as the remediation is made final. Figure 10 indicates the patched, updated, workaround, or hotfixes vulnerability percentages in blue color. The protocols ANSI C12.22, CoAP, ZigBee, RFID/NFC, EtherNet/IP, Hart/IP, and BACnet, which according to Figure 9 present the worst values of our index, also present the worst remediation levels (RL) in Figure 10. In addition, Wi-Fi, 4G/LTE, Z-Wave, CAN, and Modbus have high remediation level (RL). For instance, the Modbus protocol shows that 80% of the detected vulnerabilities have patch, update, workaround, or hotfixes. The DDS protocol is a particular case with four vulnerabilities with low base metric severity (Figure 9), therefore once applied the remediation level (RL), which is 100%, as shown in Figure 10, the severity of the vulnerabilities are kept at a low level; hence, the severity index is higher than that of the previously mentioned protocols as less secure. All the examples analyzed above confirm the robustness of the proposed analysis.

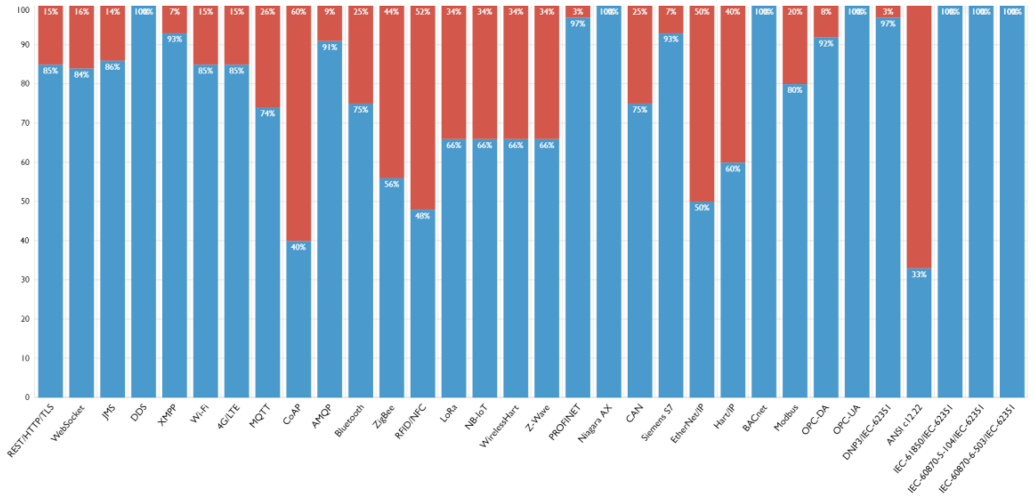


Fig. 10. Remediation level.

### 5.3 Comparison Between Environmental Metrics Contextualized to OT and IT Environment

Once the second stage of the risk analysis has been completed, where emphasis has been placed on base metrics, temporal metrics, and comparisons between them, this new phase, which corresponds to stage D in Figure 4, VAF uses environmental metrics as a risk analysis mechanism. As mentioned above, organizations must specify environmental metrics, because they are the most appropriate to measure the potential impact of a vulnerability within their own computational environment. Therefore, Environmental Metrics adjust the Base (BM) and Temporal (TM) severities to a specific environment. Figure 3 establishes the set of submetrics that takes environmental metrics into account. The strategy used in this research will be to keep the values of the submetrics MAV, MAC, MPR, MUI, MS, MC, MI, and MA unchanged and to customize the values of the "security requirements" (CR, IR, and AR) according to the order of importance given in Table 2 to the cybersecurity pillars. This strategy aims to customize the CVSS score according to the relevance of the environment. From our results, we will be able to determine if an asset presents a certain risk of being compromised and this vulnerability is exploited in which environment (IT or OT) the impact will be greater. Figure 11 shows the results of applying stage D of Figure 4 to VAF. These results are divided into distribution of the severity of environmental metrics in both OT and IT environments and determination of the difference between the average severity for operational environmental metric ($EM_{OT}$) and information environmental metric ($EM_{IT}$), which corresponds to the average $EM_{OT}$ index and average $EM_{IT}$. It should be noted that as with the results of Figure 9 for each protocol, standard, or bus, the sum of the vulnerabilities distributed according to their severity coincides with Figure 7. Figure 11 shows that the impact of applying compensation for OT environments is greater than applying compensation for IT environments in most protocols, standards, and buses. This behavior indicates that if an asset has a vulnerability and it is exploited, then the impact will be greater in an OT environment than in an IT environment. For example, 4G/LTE has 19 vulnerabilities with critical severity for $EM_{OT}$ and 7 vulnerabilities with critical severity for $EM_{IT}$. Additionally, Bluetooth has 38 vulnerabilities with critical severity for $EM_{OT}$ and 15 vulnerabilities with critical severity for $EM_{IT}$. The most representative case is that of Wi-Fi, which has 90 critical vulnerabilities for $EM_{OT}$, while 33 vulnerabilities with critical severity for

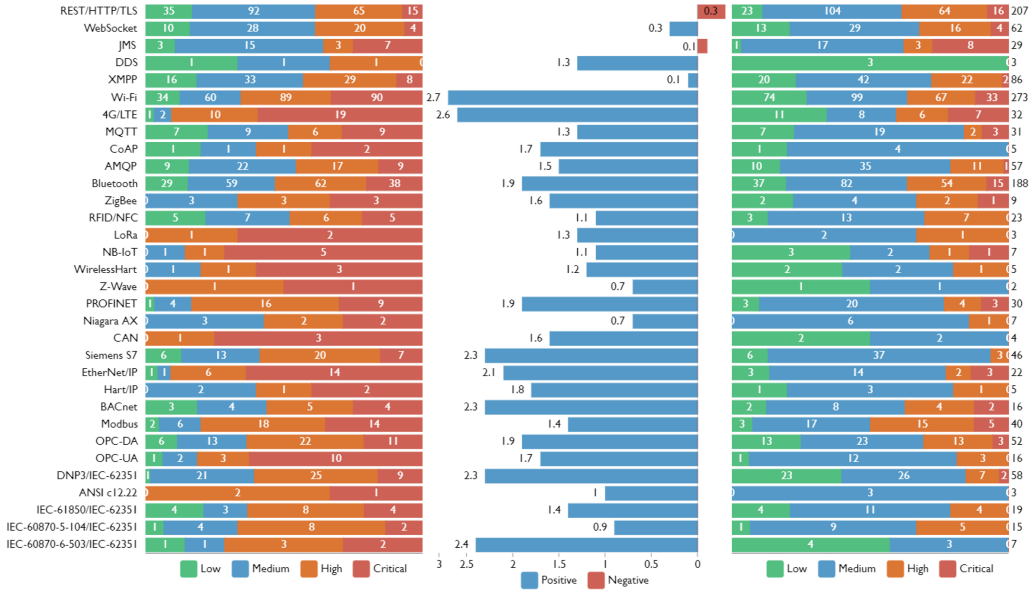


Fig. 11. From left to right: (a) $EM_{OT}$; (b) Avg ($EM_{OT}$) - Avg ($EM_{IT}$), and (c) $EM_{IT}$.

$EM_{IT}$, which represents the most significant difference between all the protocols, standards, and buses analyzed. Similar results are achieved for XMPP, CoAP, AMQP, MQTT, ZigBee, RFID/NFC, LoRa, NB-IoT, WirelessHart, Z-Wave, PROFINET, Niagara AX, CAN, Siemens S7, EtherNet/IP, Hart/IP, BACnet, Modbus, OPC-DA, OPC-UA, DNP3, ANSI c12.22, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503. The DDS protocol is a case to consider, since it has no vulnerabilities with critical severity, the reason why the analysis of the behaviors once applied the requirements of OT and IT, must extend to vulnerabilities of high and medium severity. Despite this, it is observed that severity is greater when OT requirements are applied than when IT is applied. In addition to DDS, the WebSocket protocol has a different behavior. As it can be seen in Figure 11, the number of vulnerabilities with critical severity is the same (four) for both $EM_{OT}$ and $EM_{IT}$. Therefore, the analysis should be extended to vulnerabilities with high severity, where if the number for $EM_{OT}$ is greater than for $EM_{IT}$. However, there are two protocols, which do not follow this pattern: REST/HTTP/TLS and JMS. As shown in Figure 11, the number of vulnerabilities with critical severity is greater for $EM_{IT}$ than for $EM_{OT}$. The explanation for this result comes from the fact that these protocols are mostly used in IT environments and that they are recently finding applicability in IoT environments. Like the analysis applied in the previous section to establish which of the protocols, standards, or buses analyzed have the best results, we have decided to compare the average severities for $EM_{OT}$ and $EM_{IT}$. Therefore, the protocols, standards, or buses that have a higher index, i.e., the difference in average $EM_{OT}$ severity is much greater than the average $EM_{IT}$ severity, as a vulnerability is exploited it will have a greater impact on an OT environment than on an IT environment. As a conclusion of this phase, we can say that the implication of this study is that if the same vulnerability were detected in both an industrial and an information environment, the consequence of the exploitation of this vulnerability would be greater in the industrial environment. This result guarantees the adaptation of our framework to an industrial environment.

Table 4. Vulnerability Impact on Security Pillars

| Protocol | NoV | A | I | C | A (%) | I (%) | C (%) |
|---|---|---|---|---|---|---|---|
| REST/HTTP/TLS | 207 | 77 | 76 | 91 | 37 | 37 | 44 |
| WebSocket | 62 | 23 | 16 | 31 | 47 | 26 | 50 |
| JMS | 29 | 15 | 13 | 16 | 52 | 45 | 55 |
| DDS | 4 | 0 | 1 | 3 | 0 | 25 | 75 |
| XMPP | 86 | 19 | 31 | 27 | 22 | 36 | 31 |
| Wi-Fi | 273 | 135 | 75 | 87 | 49 | 27 | 32 |
| 4G/LTE | 32 | 16 | 9 | 10 | 50 | 28 | 31 |
| MQTT | 31 | 11 | 8 | 4 | 35 | 26 | 13 |
| CoAP | 5 | 3 | 0 | 0 | 60 | 0 | 0 |
| AMQP | 57 | 22 | 13 | 13 | 38 | 23 | 23 |
| Bluetooth | 188 | 99 | 95 | 93 | 53 | 51 | 44 |
| ZigBee | 9 | 4 | 3 | 1 | 44 | 33 | 11 |
| RFID/NFC | 23 | 9 | 8 | 9 | 39 | 35 | 39 |
| LoRa | 3 | 2 | 1 | 0 | 67 | 33 | 0 |
| NB-IoT | 7 | 4 | 2 | 1 | 57 | 29 | 14 |
| WirelessHart | 5 | 4 | 2 | 0 | 80 | 40 | 0 |
| Z-Wave | 2 | 1 | 0 | 1 | 50 | 0 | 50 |
| PROFINET | 30 | 24 | 7 | 6 | 80 | 23 | 20 |
| Niagara AX | 7 | 4 | 3 | 4 | 43 | 43 | 57 |
| CAN | 4 | 3 | 2 | 3 | 75 | 50 | 75 |
| Siemens S7 | 46 | 29 | 3 | 4 | 63 | 7 | 9 |
| EtherNet/IP | 22 | 17 | 5 | 4 | 77 | 23 | 18 |
| Hart/IP | 5 | 2 | 1 | 1 | 40 | 20 | 20 |
| BACnet | 16 | 5 | 3 | 5 | 31 | 19 | 31 |
| Modbus | 40 | 27 | 20 | 21 | 68 | 50 | 53 |
| OPC-DA | 52 | 33 | 17 | 19 | 63 | 33 | 37 |
| OPC-UA | 16 | 10 | 3 | 5 | 63 | 19 | 31 |
| DNP3 | 58 | 42 | 9 | 8 | 72 | 16 | 14 |
| ANSI C12.22 | 3 | 3 | 0 | 0 | 100 | 0 | 0 |
| IEC-61850 | 19 | 9 | 4 | 4 | 47 | 21 | 21 |
| IEC-60870-5-104 | 15 | 9 | 5 | 4 | 60 | 33 | 27 |
| IEC-60870-6-503 | 7 | 4 | 0 | 0 | 57 | 0 | 0 |

Table 5. Summary of the Vulnerability Impact on Security Pillars

| Category | NoV | A | I | C |
|---|---|---|---|---|
| IT | 388 | 134 | 137 | 168 |
| IoT | 635 | 310 | 216 | 219 |
| OT | 340 | 221 | 82 | 88 |

Table 6. Impact of Cybersecurity Pillars on IIoT Categories

| IIoT Classification | NoV | A | I | C |
|---|---|---|---|---|
| FAP | 223 | 145 | 45 | 42 |
| PAP | 739 | 389 | 246 | 253 |
| ICSP | 140 | 86 | 49 | 55 |
| BAP | 579 | 299 | 211 | 225 |
| PSAP | 243 | 117 | 73 | 68 |
| AMRP | 64 | 41 | 26 | 24 |
| SHAP | 925 | 401 | 323 | 355 |
| VAP | 196 | 102 | 98 | 99 |

### 5.4 Impact of the Vulnerabilities on Security Pillars

In the two previous sections, we have given a measure that describes how secure a protocol is based on the patches and updates it presents, as well as a measure of the impact of a vulnerability on an OT or IT environment. To this end, we have adopted the methodology of the VAF framework. Since our framework is based on CVSSv3.1 and this indicates that risk analysis goes beyond the measurement of base, temporal and environmental metrics, the following analysis focuses on how the cybersecurity pillars of Table 2 have been impacted according to each protocol, standard, or bus. Exploring the impact of vulnerabilities on the three pillars of cybersecurity is one of VAF's aims. Table 4 provides the number of times availability (A), integrity (I), and confidentiality (C) have been fully affected with respect to the number of vulnerabilities (NoV) of the protocol, standard, or bus. This means, for example, that for REST/HTTP/TLS, of the 207 vulnerabilities examined, 77 of them have fully affected the availability, 76 vulnerabilities have totally affected the integrity, and

85 vulnerabilities have totally affected the confidentiality. It should be noted that a vulnerability could affect more than one pillar at a time. Analyzing the results provided by Table 4 shows that traditional industrial protocols (e.g., DNP3, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503), IoT protocols (e.g., Wi-Fi, 4G/LTE, Bluetooth, ZigBee, and RFID/NFC), including IoT messaging protocols (e.g., MQTTT, CoAP, and AMQP) have availability as the most affected parameter. From column "A (%)," it is also possible to see how other OT protocols such as PROFINET, OPC-DA, OPC-UA, and DNP3 have the worst impact on availability. However, other protocols and standards, such as REST/HTTP/TLS, WebSocket, JMS, and DDS, present confidentiality as the most affected parameter. These protocols are of IT nature, i.e., recently have a significant utility in IIoT environments. Finally, the XMPP protocol should be mentioned, which, although it has a similar origin to the protocols, the most affected pillar is integrity. Table 5 provides an overview of the behavior of each of the categories associated with our protocols, standards, and buses in Table 1, i.e., IT, IoT, and OT.

Table 5 summarizes the behavior of the cybersecurity pillars for each of these paradigms (IT, IoT, and OT), considering that it has only been represented when the pillar has been totally compromised. The results allow to distinguish as for the protocols, standards, or buses associated to the IT environment the order of impact is confidentiality, integrity and availability, while for both IoT and OT paradigms the order of impact is availability, confidentiality, and integrity.

### 5.5 Impact of the Security Pillars on IIoT Categories

The previous section provided an overview of the behavior of the pillars of cybersecurity (confidentiality, availability, and integrity) on each of the protocols, allowing the protocols to be grouped according to the paradigm IT, IoT, and OT. In this section, we would like to move one step further in the risk analysis evaluation provided by our research. To this end, we will analyze the relationship that exists between the IIoT categories defined in Table 1 and the cybersecurity pillars. Contextualizing, again only the values of confidentiality, integrity and availability that have been totally compromised will be represented. Table 6 defines the number of vulnerabilities (NoV), as well as the pillars of cybersecurity compromised for each of the IIoT categories defined. The number of vulnerabilities that affect each category is a function of the number of protocols, standards, or buses that are associated with it as well as the vulnerabilities that have been analyzed for each of the protocols, standards, or buses. Due to the wide application context, as well as the growing number of protocols that implement this category, Smart Home Automation Protocol (SHAP) contains the largest number of associated vulnerabilities. For SHAP it should be noted that although availability is the most affected pillar, confidentiality is the second most affected, which is manifested, for example, through the high risk of leakage of personal information by users who use the Smart Home. Although all IIoT classification are relevant, particularly Factory Automation Protocols (FAP) and Process Automation Protocols (PAP) present availability as the essential pillar, which for both categories in Table 6, is the most impacted pillar. Finally, it is important to highlight that the IIoT categories where the least difference between the numbers of compromised pillars exists is in "Vehicle Automation Protocols" (VAP), where confidentiality, integrity and availability are around 100 times compromised.

### 5.6 Attack Patterns and Weakness Associated to the Vulnerabilities

As mentioned in Section 4.1, there is a close relationship between weakness (CWE), vulnerability (CVE), and attack pattern (CAPEC). In this section, we take advantage of that relationship to add a new layer to the risk analysis. A criterion mentioned as important within the risk analysis as a complement to the CVSSv3.1 analysis are the weaknesses and patterns of attacks, so the VAF allows a relationship between vulnerability, weakness, and pattern of attack. The methodology

Table 7. Weakness and Attack Pattern Associated to Each Protocol

| Protocol | Weakness | Attack Pattern |
|---|---|---|
| REST/HTTP/TLS | Information exposure | API manipulation |
| WebSocket | Improper restriction[1] | TCP Flood |
| JMS | Information exposure | Object injection |
| DDS | Resource management errors | Flooding |
| XMPP | Improper input validation | Social Engineering |
| Wi-Fi | Improper input validation | Traffic injection |
| 4G/LTE | Improper input validation | Cellular Jamming |
| MQTT | Improper Input Validation | TCP Flood |
| CoAP | NULL point dereference | Traffic injection |
| AMQP | Improper input validation | TCP Flood |
| Bluetooth | Information exposure | Flooding |
| ZigBee | Improper access control | Flooding |
| RFID/NFC | Information exposure | Flooding |
| LoRa | Improper input validation | Jamming |
| NB-IoT | Improper input validation | Cellular Jamming |
| WirelessHart | Improper input validation | Jamming |
| Z-Wave | Improper Certificate Validation | Malicious Root Certificate |
| PROFINET | Improper input validation | Flooding |
| Niagara AX | Improper authentication | Authentication Abuse |
| CAN | Improper restriction[1] | Code Injection |
| Siemens S7 | Resource management errors | Flooding |
| EtherNet/IP | Improper restriction[1] | TCP Flood |
| Hart/IP | Improper restriction[1] | Overflow Buffers |
| BACnet | Improper restriction[1] | Cross site request forgery |
| Modbus | Information exposure | TCP Flood |
| OPC-DA | Improper restriction[1] | Flooding |
| OPC-UA | Improper input validation | TCP Flood |
| DNP3 | Improper input validation | Input Data Manipulation |
| ANSI C12.22 | Improper input validation | Target Programs[2] |
| IEC-61850 | Improper access control | Flooding |
| IEC-60870-5-104 | Improper input validation | Input Data Manipulation |
| IEC-60870-6-503 | Improper restriction[1] | TCP Flood |

[1]Improper Restriction of Operations within the Bounds of a Memory Buffer.
[2]Target Programs with Elevated Privileges.

applied through VAF has been extracted for each of the 1,363 vulnerabilities, analyzing the weakness or weaknesses exploited and associating them with the corresponding attack pattern. Table 7 associates to each protocol, standard, or bus, the weakness exploited more times, as well as the attack pattern used more times. However, it is not necessary to have a match between the weaknesses and the attack patterns shown in Table 7, because as mentioned, a weakness can have several attack patterns associated with it and a pattern of attack with several vulnerabilities.

For instance, the CAPEC-125: Flooding has two associated weaknesses; CWE-404: Improper Resource Shutdown or Release; and CWE-770: Allocation of Resources without Limits or Throttling. Since the flooding attack is the most used, it is detailed below. The flooding attack allows an adversary to consume the resources of a target due to the high amount of targeted interactions.

This attack commonly exposes a weakness in rate or flow limitation. If performed correctly, then this attack impedes the access of legitimate users to the service and may result in the target being locked out. The number of requests the attacker makes in a certain period is the main factor in a flood attack. The use of the flood attack pattern is very common. For instance, EtherNET/IP allows that an attacker can send malformed packet CIP to Port 44818/TCP, Modbus allows an attack pattern by sending specifically designed (crafted) packages to port 502/TCP, and MQTT allows to send crafted CONNECT packets. For all three protocols, the flooding category used is TCP flood. An illustration of the connection between vulnerability, weakness, and attack pattern is demonstrated through AMQP, which it presents input validation errors (e.g., does not properly restrict incoming client connections). This weakness allows flooding attack (TCP Flood) and the possibility of DoS (see CVE-2012-2145). Another point to consider is that for wireless IoT technologies such as 4G/LTE, LoRa, NB-IoT, and WirelessHart, the most common attack pattern is the jamming. In this type of attack, an attacker would use noise or radio signals to disrupt communications. By intentionally overwhelming system resources with illegitimate traffic, legitimate traffic from authorized users is denied service. Two exceptions need to be mentioned: BACnet and XMPP. BACnet is a protocol used in building automation, and nevertheless it presents an attack pattern that is mostly used in web technologies. This is because authentication vulnerabilities and cross-site request forgery (CSRF) were identified on KMC Controls' BACnet Conquest routers via their web interface. XMPP has a social-engineering-like attack pattern. Although XMPP is now linked to IIoT environments, it is a technology associated also with instant messaging. Tools like Pidgin are based on XMPP protocol. In this case, a Pidgin plugin called "Message Carbon" located in several XMPP clients enables a remote attacker to masquerade as either user, even contacts. Another exception occurs with Z-Wave where the weakness and attack pattern are not of a wireless nature and this is because the vulnerabilities, for example, are found in a device that implements the protocol and does not implement HTTP Public Key Pinning (HPKP). This weakness, allows to obtain the commands that are transmitted to the controller using a fake certificate in a proxy server, enabling to control each node of the HUB, reaching to get the Z-Wave network key.

### 5.7 Vulnerabilities Classification

As we mentioned in Section 1, the NIST 800-82r2 Reference [4] proposes a general vulnerabilities classification into: policy and procedural vulnerabilities, architecture and design vulnerabilities, configuration and maintenance vulnerabilities, physical vulnerabilities, software development vulnerabilities, and communication and network configuration vulnerabilities. In addition, each category includes a group of vulnerabilities associated, i.e., a specific vulnerabilities classification. Table 8 summarizes both the most exploited general vulnerability and the most exploited specific vulnerability for each of the 32 protocols, standards, and buses analyzed. The first conclusion reached after analyzing Table 8 is that there are a set of vulnerabilities that present similar behaviors as those associated with wireless IoT protocols, standards, and buses of the IoT category such as Lora and WirelessHart. In general, the most common vulnerability they face is the physical type. From a more specific point of view, these technologies are mostly affected by radio frequency vulnerabilities, since the hardware used for control systems is vulnerable to radio frequency. The impact can vary from temporal interruption of command and control to continuous failure to circuit boards. However, outside that group are protocols such as Bluetooth, RFID/NFC, Wi-Fi and ZigBee. The first three (Bluetooth, RFID/NFC, and Wi-Fi) present vulnerabilities of the type inappropriate information protection among "wireless clients" and "access points," which are not protected. The fourth (ZigBee) presents mostly vulnerabilities of the type inadequate access controls applied, which belongs to the category of configuration and maintenance vulnerabilities. Another group of protocols, standards, and buses with vulnerabilities of a similar nature are those

Table 8. Vulnerabilities Classification According to the Classification Proposed by Reference [4]

| Protocol | Category | General Vulnerability Classification | Specific Vulnerability Classification |
|---|---|---|---|
| REST/HTTP/TLS | IT | Software Development Vulnerabilities | Improper Data Validation |
| WebSocket | | Software Development Vulnerabilities | Improper Data Validation |
| JMS | | Software Development Vulnerabilities | Improper Data Validation |
| DDS | | Software Development Vulnerabilities | Improper Data Validation |
| XMPP | | Software Development Vulnerabilities | Improper Data Validation |
| Wi-Fi | IoT | Communication and Network Configuration | Inadequate data protection between wireless clients and access points |
| 4G/LTE | | Communication and Network Configuration | Inadequate data protection between wireless clients and access points |
| MQTT | | Software Development Vulnerabilities | Improper Data Validation |
| CoAP | | Software Development Vulnerabilities | Improper Data Validation |
| AMQP | | Software Development Vulnerabilities | Improper Data Validation |
| Bluetooth | | Communication and Network Configuration | Inadequate data protection between wireless clients and access points |
| ZigBee | | Configuration and Maintenance Vulnerabilities | Inadequate access controls applied |
| RFID/NFC | | Communication and Network Configuration | Inadequate data protection between wireless clients and access points |
| LoRa | | Physical Vulnerabilities | Radio frequency, electromagnetic pulse (EMP), static discharge, brownouts and voltage spikes |
| NB-IoT | | Communication and Network Configuration | Inadequate data protection between wireless clients and access points |
| WirelessHart | | Physical Vulnerabilities | Radio frequency, electromagnetic pulse (EMP), static discharge, brownouts and voltage spikes |
| Z-Wave | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| PROFINET | OT | Software Development Vulnerabilities | Improper Data Validation |
| Niagara AX | | Software Development Vulnerabilities | Improper Data Validation |
| CAN | | Software Development Vulnerabilities | Improper Data Validation |
| Siemens S7 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| EtherNet/IP | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| Hart/IP | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| BACnet | | Software Development Vulnerabilities | Improper Data Validation |
| Modbus | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| OPC-DA | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| OPC-UA | | Software Development Vulnerabilities | Improper Data Validation |
| DNP3 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| ANSI C12.22 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| IEC-61850 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| IEC-60870-5-104 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |
| IEC-60870-6-503 | | Architecture and Design Vulnerabilities | Inadequate incorporation of security into architecture and design |

belonging to the OT category, i.e., the classic ICS. In protocols and standards such as Siemens S7, EtherNet/IP, Hart/IP, Modbus, OPC-DA, DNP3, ANSI C12.22, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503, the most common type of vulnerabilities reported are architecture and design. There is also similarity in the type of specific vulnerabilities for most of these protocols and standards, with an inadequate incorporation of security inappropriate into architecture and design, which is justified by the fact that most have migrated to TCP/IP environments and therefore lack default security mechanisms, which is the reason they lack identification, authorization, and authentication mechanisms, i.e., access control, as well as mechanisms that guarantee the configuration and integrity of the system. Another group is made up of PROFINET, Niagara AX, CAN, and OPC-UA, with a general classification of software-development-type vulnerabilities and a specific classification of the type of inadequate data validation. In addition, there is an important group formed by protocols and standards IT nature, such as REST/HTTP/TLS, WebSocket, JMS, and so on, that present as major vulnerabilities the improper input validation, which is characterized because of the IIoT software does not validate user entries or received data correctly to guarantee their authenticity. Therefore, invalid data can lead to several vulnerabilities, among which are cross-site scripting, path traversals, command injections, and buffer overflows.

## 6 CONCLUSION

The IIoT architecture is complex, largely due to the convergence of protocols, standards, and buses. For this reason, a comprehensive survey of the 33 most useful protocols, standards, and buses in the IIoT environment is performed based on characteristics such as architecture, topology, messages/data, and security. From this analysis, we detect the existence of security problems in each of the protocols, standards, and buses. Therefore, an extensive assessment is necessary to measure the risk of existing documented vulnerabilities. Since the CVSS offers a way to collect the main properties of a vulnerability and generate a numbered score that indicates its severity, it is presented as the fundamental bedrock of our approach. At this point, we find two situations, on the one hand, to carry out the risk analysis it is necessary to complement the CVSSv3.1 with other elements and, on the other hand, CVSS presents problems to characterize the industrial environments. For this reason, we present the VAF methodological framework. For this purpose, an exhaustive compilation of OSINT tools such as CVE-Mitre, dictionaries such as CPE, and integration tools such as CVE-Search is performed. These data sources, our experience as analysts, and the practical context provided by CVSSv3.1 are the core of the Vulnerability Analysis Framework (VAF). This methodological framework enabled the analysis of 1,363 vulnerabilities from 33 protocols, standards, and buses. From the VAF, we divide our analysis into seven steps: (1) Vulnerability search (CVE) for the IT/OT/IoT standards, protocols and buses; (2) comparison between base and temporal metrics; (3) comparison between contextualized environmental metrics for both OT and IT environments; (4) impact of vulnerabilities on the pillars of cybersecurity: availability, confidentiality and integrity; (5) impact of the cybersecurity pillars on IIoT categories; (6) attack patterns and weakness associated with vulnerabilities; (7) classification of vulnerabilities according to NIST reference. It should be noted that was added a step only for 5G due to the novelty of the technology does not present active with CVE, but only a CVD confirmed by 3GPP: recent attacks in 5G. Each of these steps constitutes a phase of our risk analysis. The comparison between base and temporal metrics enabled to establish how the level vulnerability severity could be reduced through the application of security updates and at the same time, it grows through the deployment of exploits. In this sense, the impact of the amount of fixed vulnerabilities over Temporal Metrics were examined considering the existing relationship with the remediation level. The comparison between environmental metrics revealed that if an asset uses one or a combination of the 33 protocols, standards, and buses analyzed, when suffering one of the 1,363 vulnerabilities analyzed, the severity will be higher if the

asset is located as part of an OT environment (e.g., Control and Operations Domain for IIoT functional domains) than of an IT environment (e.g., Information and Applications Domain for IIoT functional domains). The impact on three security pillars (availability, integrity and confidentiality) was determined that for the protocols, standards, or buses associated to the IT environment the order of impact is confidentiality, integrity, and availability, while for both IoT and OT paradigms the order of impact is availability, confidentiality, and integrity. We have also been able to evaluate the impact of the pillars of cybersecurity in the established IIoT classification, where although availability is the most affected parameter for each of them, which is critical for Fabric and Process automation Protocols, for various environments such as Smart Home Automation Protocols and Vehicular Automation Protocols confidentiality is the second most affected parameter, being very close to availability in both cases. An accurate integration between attack pattern, vulnerability, and weakness was obtained, which demonstrated that "flooding" is the most used attack pattern and "inadequate entry validation" is a more exploited weakness. In addition, it is also confirmed that the most common attack used in IoT wireless technologies, such as 4G/LTE, LoRa, and NB-IoT, is jamming. Based on the NIST 800-82r2 recommendation, which establishes a classification of vulnerabilities in industrial environments, we classify the vulnerabilities that most affect our IIoT protocols, standards, and buses. Although there are numerous exceptions, we can detail three major groups associated with the classification established for IIoT in IT, IoT, and OT. In the case of IT, vulnerabilities related to improper data validation predominate. In the case of IoT, physical and network configuration and communication, vulnerabilities predominate. In the case of OT, inadequate incorporation of security into architecture and design vulnerabilities predominate.

The future research focuses on fully automating the VAF methodology, incorporating searches from the Common Platform Enumeration (CPE), so that we can associate to a given asset all the results that VAF offers.

## REFERENCES


[1] S. Bhattacharjee. 2018. Practical industrial internet of things security. *Packt Publishing Ltd.* Retrieved from https://www.packtpub.com/eu/business/practical-industrial-internet-things-security.

[2] R. Shirey. 2007. Internet security glossary, version 2. Retrieved from https://tools.ietf.org/html/rfc4949.

[3] S. Whalen, M. Bishop, and S. Engle. 2005. Protocol vulnerability analysis. *Citeseer* 14 (2005). Retrieved from https://pdfs.semanticscholar.org/cb46/7b25e76e309b15fef603882c8b9892a2ddc7.pdf 7.

[4] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn. 2015. NIST special publication 800-82: Guide to industrial control systems (ICS) security. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf.

[5] R. Martin et al. 2016. Industrial Internet Security Framework Technical Report, Second. Highland Avenue Needham, MA. Industrial Internet Consortium. Retrieved from https://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf.

[6] A. Hasibuan, M. Mustadi, I. E. Y. Syamsuddin, and I. M. A. Rosidi. 2015. Design and implementation of modular home automation based on wireless network, REST API, and websocket. In *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS'15)*. 362–367. DOI: https://doi.org/10.1109/ISPACS.2015.7432797

[7] V. M. Trifa, D. Guinard, and M. Koehler. 2008. Messaging methods in a service-oriented architecture for industrial automation systems. In *Proceedings of the 5th International Conference on Networked Sensing Systems*. 35–38. DOI: https://doi.org/10.1109/INSS.2008.4610893

[8] J. Yang, K. Sandström, T. Nolte, and M. Behnam. 2012. Data distribution service for industrial automation. In *Proceedings of IEEE 17th International Conference on Emerging Technologies and Factory Automation (ETFA'12)*. 1–8. DOI: https://doi.org/10.1109/ETFA.2012.6489544

[9] A. Alaerjan and D. Kim. 2017. Configuring DDS features for communicating components in smart grids. In *Proceedings of the IEEE International Conference on Smart Energy Grid Engineering (SEGE'17)*. 162–169. DOI: https://doi.org/10.1109/SEGE.2017.8052793

[10] J. Rodríguez-Molina, S. Bilbao, B. Martínez, M. Frasheri, and B. Cürüklü. 2017. An optimized, data distribution service-based solution for reliable data exchange among autonomous underwater vehicles. *Sensors (Basel)* 17, 8 (2017), 1802. DOI: https://doi.org/10.3390/s17081802

[11] A. A. Khan and H. T. Mouftah. 2012. Secured web services for home automation in smart grid environment. In *Proceedings of the 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'12)*. 1–4. DOI: https://doi.org/10.1109/CCECE.2012.6335018

[12] Y. Wenbo, W. Quanyu, and G. Zhenwei. 2015. Smart home implementation based on Internet and WiFi technology. In *Proceedings of the 34th Chinese Control Conference (CCC'15)*. 9072–9077. DOI: https://doi.org/10.1109/ChiCC.2015.7261075

[13] G. Afifi, H. H. Halawa, R. M. Daoud, and H. H. Amer. 2016. Dual protocol performance using WiFi and zigbee for industrial WLAN. In *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED'16)*. 749–754. DOI: https://doi.org/10.1109/MED.2016.7535854

[14] K. Khanchuea and R. Siripokarpirom. 2019. A multi-protocol iot gateway and WiFi/BLE sensor nodes for smart home and building automation: Design and implementation. In *Proceedings of the 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES'19)*. 1–6. DOI: https://doi.org/10.1109/ICTEmSys.2019.8695968

[15] G. V. Vivek and M. P. Sunil. 2015. Enabling IOT services using WIFI-Zigbee gateway for a home automation system. In *Proceedings of the IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN'15)*. 77–80. DOI: https://doi.org/10.1109/ICRCICN.2015.7434213

[16] S. A. Ashraf, I. Aktas, E. Eriksson, K. W. Helmersson, and J. Ansari. 2016. Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G. In *Proceedings of the IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA'16)*. 1–8. DOI: https://doi.org/10.1109/ETFA.2016.7733543

[17] P. Orosz, P. Varga, G. Soós, and C. Hegedűs. 2019. QoS guarantees for industrial iot applications over LTE—A feasibility study. In *Proceedings of the IEEE International Conference on Industrial Cyber Physical Systems (ICPS'19)*. 667–672. DOI: https://doi.org/10.1109/ICPHYS.2019.8780308

[18] A. Ahmed, M. M. Khan, and W. Ahmed. 2016. Cloud based network management and control for building automation. In *Proceedings of the 19th International Multi-Topic Conference (INMIC'16)*. 1–6. DOI: https://doi.org/10.1109/INMIC.2016.7840123

[19] X. Feng, S. Zhao, Y. Wang, and T. Qi. 2018. The application on 4G-LTE system with evaporation duct. In *Proceedings of the 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC'18)*. 1970–1975. DOI: https://doi.org/10.1109/IMCEC.2018.8469369

[20] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz. 2018. A survey on 5g networks for the internet of things: Communication technologies and challenges. *IEEE Access* 6, (2018), 3619–3647. DOI: https://doi.org/10.1109/ACCESS.2017.2779844

[21] D. Bezerra, R. R. Aschoff, G. Szabo, and D. F. H. Sadok. 2018. An IoT protocol evaluation in a smart factory environment. In *Proceedings of the Latin American Robotic Symposium, Brazilian Symposium on Robotics (SBR'18) and Workshop on Robotics in Education (WRE'18)*. 118–123. DOI: https://doi.org/10.1109/LARS/SBR/WRE.2018.00030

[22] R. K. Kodali and S. Soratkal. 2016. MQTT based home automation system using ESP8266. In *Proceedings of the IEEE Region 10 Humanitarian Technology Conference (R10-HTC'16)*. 1–5. DOI: https://doi.org/10.1109/R10-HTC.2016.7906845

[23] C. Bormann, A. P. Castellani, and Z. Shelby. 2012. CoAP: An application protocol for billions of tiny internet nodes. *IEEE Internet Comput.* 16, 2 (2012), 62–67, 2012. DOI: https://doi.org/10.1109/MIC.2012.29

[24] I. Shin, D. Eom, and B. Song. 2015. The CoAP-based M2M gateway for distribution automation system using DNP3.0 in smart grid environment. In *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm'15)*. 713–718. DOI: https://doi.org/10.1109/SmartGridComm.2015.7436385

[25] D. Halabi, S. Hamdan, and S. Almajali. 2018. Enhance the security in smart home applications based on IOT-CoAP protocol. In *Proceedings of the 6th International Conference on Digital Information, Networking, and Wireless Communications (DINWC'18)*. 81–85. DOI: https://doi.org/10.1109/DINWC.2018.8357000

[26] M. Grover, M. E. I. I. Year, S. K. Pardeshi, N. Singh, and S. Kumar. 2015. Bluetooth low energy for industrial automation. In *Proceedings of the International Conference on Electronics, Circuits, and Systems (ICECS'15)*. 512–515. DOI: https://doi.org/10.1109/ECS.2015.7124960

[27] E. J. Sebastian, A. Yushev, A. Sikora, M. Schappacher, and J. A. Prasetyo. 2016. Performance investigation of 6Lo with RPL mesh networking for home and building automation. In *Proceedings of the 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS'16)*. 127–133. DOI: https://doi.org/10.1109/IDAACS-SWS.2016.7805801

[28] P. H. B. Shinde, A. Chaudhari, P. Chaure, M. Chandgude, and P. Waghmare. 2017. Smart home automation system using android application. Retrieved from https://www.irjet.net/archives/V4/i4/IRJET-V4I4604.pdf.

[29] X. Shen, X. Wang, and M. Jia. 2017. Design and implementation of traffic information detection equipment based on bluetooth communication. In *Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC'17)*. 1595–1601. DOI: https://doi.org/10.1109/ITNEC.2017.8285063

[30] H. P. Lin, S. C. Cheng, D. B. Lin, C. H. Chung, and R. S. Hsiao. 2012. Integrating zigbee lighting control into existing building automation systems. In *IET International Conference Information Science and Control Engineering (ICISCE'12)*. 3.48–3.48. DOI: https://doi.org/10.1049/cp.2012.2445

[31] W. Yang, H. Jiang, J. Wu, and C. Zhang. 2010. Research on the hybrid network technology of industrial ethernet and Zigbee for monitoring the ship power system. In *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR'10)*. 460–463. DOI: https://doi.org/10.1109/CAR.2010.5456798

[32] S. Rana, H. Zhu, C. W. Lee, D. M. Nicol, and I. Shin. 2012. The not-so-smart grid: Preliminary work on identifying vulnerabilities in ANSI C12.22. In *Proceedings of the IEEE Globecom Workshops (GC'12)*. 1514–1519. DOI: https://doi.org/10.1109/GLOCOMW.2012.6477810

[33] F. M. Schaefer, T. Groß, and R. Kays. 2013. Energy consumption of 6LoWPAN and Zigbee in home automation networks. In *Proceedings of the IFIP Wireless Days Conference (WD'13)*. 13–15. DOI: https://doi.org/10.1109/WD.2013.6686463

[34] C. Swedberg. 2014. General motors factory installs smart bolts in engine blocks, cylinder heads. *RFID J.* Retrieved from https://www.rfidjournal.com/articles/view?11329.

[35] O. Bindroo, K. Saxena, and S. K. Khatri. 2017. A wearable NFC wristband for remote home automation system. In *Proceedings of the 2nd International Conference on Telecommunication and Networks (TEL-NET'17)*. 1–6. DOI: https://doi.org/10.1109/TEL-NET.2017.8343563

[36] A. Haidine, A. Aqqal, and A. Dahbi. 2018. Performance evaluation of low-power wide area based on lora technology for smart metering. In *Proceedings of the 6th International Conference on Wireless Networks and Mobile Communications (WINCOM'18)*. 1–6. DOI: https://doi.org/10.1109/WINCOM.2018.8629693

[37] K. A. Nsiah, Z. Amjad, A. Sikora, and B. Hilt. 2019. Performance evaluation of latency for NB-LTE networks in industrial automation. In *Proceedings of the IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'19)*. 1–7. DOI: https://doi.org/10.1109/PIMRC.2019.8904407

[38] S. Chen, Y. Li, M. H. Memon, and F. Lin. 2018. Design and implementation of cell search in NB-IoT downlink receiver. In *Proceedings of the IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA'18)*. 20–21. DOI: https://doi.org/10.1109/CICTA.2018.8705949

[39] J. Du and X. Liu. 2019. Design and implementation of smart socket based on NB-IoT. In *Proceedings of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC'19)*. 1033–1037. DOI: https://doi.org/10.1109/ITNEC.2019.8729284

[40] H. Hayashi, T. Hasegawa, and K. Demachi. 2009. Wireless technology for process automation. In *Proceedings of the International ICCAS-SICE Joint Conference*. 4591–4594. Retrieved from https://ieeexplore.ieee.org/document/5333003.

[41] M. Kuzlu, M. Pipattanasomporn, and S. Rahman. 2015. Review of communication technologies for smart homes/building applications. In *Proceedings of the IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA'15)*. 1–6. DOI: https://doi.org/10.1109/ISGT-Asia.2015.7437036

[42] S. Ahmad. 2011. Smart metering and home automation solutions for the next decade. In *Proceedings of the International Conference on Emerging Trends in Networks and Computer Communications (ETNCC'11)*. 200–204. DOI: https://doi.org/10.1109/ETNCC.2011.5958516

[43] P. Ferrari, A. Flammini, F. Venturini, and A. Augelli. 2011. Large PROFINET IO RT networks for factory automation: A case study. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA'11)*. 1–4. DOI: https://doi.org/10.1109/ETFA.2011.6059160

[44] J. Vasel. 2012. One plant, one system: Benefits of integrating process and power automation. In *Proceedings of the 65th Annual Conference for Protective Relay Engineers*. 215–250. DOI: https://doi.org/10.1109/CPRE.2012.6201235

[45] K. T. Smith and C. Architect. 2017. Cybersecurity and the IoT—Threats. *Best Practices and Lessons Learned*. Retrieved from https://bit.ly/2Ua6tMq.

[46] T. C. Hooi, M. Singh, Y. K. Siah, and A. R. bin Ahmad. 2001. Building low-cost intelligent building components with controller area network (CAN) bus. In *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology (TENCON'01)*, vol. 1, 466–468. DOI: https://doi.org/10.1109/TENCON.2001.949636

[47] D. Liaw, C. Yu, and K. Wu. 2014. A CAN-based design for the control of electric vehicle. In *Proceedings of the 14th International Conference on Control, Automation and Systems (ICCAS'14)*. 1233–1237. DOI: https://doi.org/10.1109/ICCAS.2014.6987745

[48] A. I. Abashar, M. A. Mohammedeltoum, and O. D. Abaker. 2017. Automated and monitored liquid filling system using PLC technology. In *Proceedings of the International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE'17)*. 1–5. DOI: https://doi.org/10.1109/ICCCCEE.2017.7866699

[49] S. S. Khuzyatov and R. A. Valiev. 2017. Organization of data exchange through the modbus network between the SIMATIC S7 PLC and field devices. *Proceedings of the International Conference Industrial Engineering Applications and Manufacturing (ICIEAM'17)*. 15–17. DOI: https://doi.org/10.1109/ICIEAM.2017.8076369

[50] J. Rinaldi. 2003. An overview of ethernet/IP. *An Application Layer Protocol for Industrial Automation*. Retrieved from https://www.rtautomation.com/technologies/ethernetip/.

[51] S. Rao, G. V Chatrapathi, and T. Yashashwini. 2017. EtherNet/IP + FDI: Value in process automation. *Proceedings of the 2nd International Conference on Emerging Computer Information Technology*. 1–5. DOI: https://doi.org/10.1109/ICECIT.2017.8453324

[52] Thomas Hilz. 2015. HART at the speed of Ethernet. Retrieved from http://www.controlengeurope.com/article/106724/HART-at-the-speed-of-Ethernet.aspx.

[53] S. H. Hong. 2013. Development of a BACnet-ZigBee gateway for demand response in buildings. In *Proceedings of the Pan African International Conference on Information Science and Computer Telecommunications (PACT'13)*. 19–23. DOI: https://doi.org/10.1109/SCAT.2013.7055082

[54] H. Dachao, H. Yu, and C. Shaokuan. 2007. Research and application of sinec L2 and modbus plus networks on industrial automation. In *2007 International Conference on Mechatronics and Automation*. 3424–3428. DOI: https://doi.org/10.1109/ICMA.2007.4304113

[55] T. Tenkanen and T. Hamalainen. 2018. Security assessment of a distributed, modbus-based building automation system. *Proceedings of the 17th IEEE International Conference on Computer Information Technology (CIT'17)*. 332–337. DOI: https://doi.org/10.1109/CIT.2017.38

[56] Triangle Microworks. 2018. Scada Data Gateway. Retrieved from http://www.trianglemicroworks.com/products/scada-data-gateway/iccp-tase-2.

[57] A. C. D. Bonganay, J. C. Magno, A. G. Marcellana, J. M. E. Morante, and N. G. Perez. 2014. Automated electric meter reading and monitoring system using zigbee-integrated raspberry pi single board computer via modbus. In *Proceedings of the IEEE Students' Conference on Electrical, Electronics, and Computer Science (SCEECS'14)*. DOI: https://doi.org/10.1109/SCEECS.2014.6804531

[58] Y. Shimanuki. 1999. OLE for process control (OPC) for new industrial automation systems. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, vol. 6, 1048–1050. DOI: https://doi.org/10.1109/ICSMC.1999.816721

[59] H. Haskamp, F. Orth, J. Wermann, and A. W. Colombo. 2018. Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system. In *Proceedings of the IEEE Conference on Interdisciplinary Cyber-Physical Systems (ICPS'18)*. 115–121. DOI: https://doi.org/10.1109/ICPHYS.2018.8387646

[60] K. S. Manoj. 2019. *Industrial Automation with SCADA : Concepts, Communications and Security*, 1st ed. Notion Press. Retrieved from https://books.google.es/books?id=FgCRDwAAQBAJ&lpg=PP1&pg=PP1#v=onepage&q&f=false.

[61] S. Kim, H. Chng, and T. Shon. 2015. Survey on security techniques for AMI metering system. In *Proceedings of the International SoC Design Conference (ISOCC'14)* 192–193. DOI: https://doi.org/10.1109/ISOCC.2014.7087691

[62] B. Chen, M. Chen, H. Tian, and L. Chen. 2017. Advanced application of IEC60870-5-101 protocol on feeder terminal unit. In *Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification (ASID'18)*, vol. 2017-Octob. 142–145. DOI: https://doi.org/10.1109/ICASID.2017.8285761

[63] N. V. Mago, J. D. Moseley, and N. Sarma. 2013. A methodology for modeling telemetry in power systems models using IEC-61968/61970. In *Proceedings of the IEEE Innovation of Smart Grid Technology-Asia (ISGT Asia'13)*. 1–6. DOI: https://doi.org/10.1109/ISGT-Asia.2013.6698713

[64] I.A.N.A. 2019. Service name and transport protocol port number registry. Retrieved from https://bit.ly/346PQCN.

[65] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel. 2019. Program analysis of commodity iot applications for security and privacy: Challenges and opportunities. *ACM Comput. Surv* 52, 4 (2019), 74:1–74:30 DOI: https://doi.org/10.1145/3333501

[66] X. Zhang, Z. Wen, Y. Wu, and J. Zou. 2011. The implementation and application of the internet of things platform based on the REST architecture. In *Proceedings of the International Conference on Business Management and Electronic Information*, vol. 2, 43–45. DOI: https://doi.org/10.1109/ICBMEI.2011.5917838

[67] B. N. Nakhuva and T. A. Champaneria. 2017. Security provisioning for RESTful web services in Internet of things. In *Proceedings of the 4th International Conference on Advanced Computing and Communication Systems (ICACCS'17)* 2017, 1–6. DOI: https://doi.org/10.1109/ICACCS.2017.8014642

[68] H. Lee and M. R. Mehta. 2013. Defense against REST-based web service attacks for enterprise systems. *Commun. IIMA* 13, 1 (2013), 57–68.

[69] R. Wang, S. Chen, and X. Wang. 2012. Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services. In *Proceedings of the IEEE Symposium on Security and Privacy*. 365–379. DOI: https://doi.org/10.1109/SP.2012.30

[70] Egor Homakov. 2012. The most common oauth2 vulnerability. Retrieved from http://homakov.blogspot.com/2012/07/saferweb-most-common-oauth2.html.

[71] V. Clincy and H. Shahriar. 2017. Web service injection attack detection. In *Proceedings of the 12th International Conference for Internet Technology and Secured Transactions (ICITST'17)*. 173–178. DOI: https://doi.org/10.23919/ICITST.2017.8356371

[72] Open Web Application Security Project. 2018. REST Security Cheat Sheet. Retrieved from https://www.owasp.org/index.php/REST_Security_Cheat_Sheet.

[73] G. C. Fernandez, E. S. Ruiz, M. C. Gil, and F. M. Perez. 2015. From RGB led laboratory to servomotor control with websockets and IoT as educational tool. In *Proceedings of the 12th International Conference on Remote Engineering and Virtual Instrumentation (REV'15)*. 32–36. DOI: https://doi.org/10.1109/REV.2015.7087259

[74] F. Greco. 2014. API design and websocket. Retrieved from https://www.slideshare.net/grecof/api-designandwebsocket?from_action=save.

[75] H. D. Center. 2019. WebSocket Security. Retrieved from https://devcenter.heroku.com/articles/websocket-security.

[76] M. Shema, S. Shekyan, and V. Toukharian. 2012. Hacking with webSockets. Retrieved from https://bit.ly/38SmPxt.

[77] G. Chen, Y. Du, P. Qin, and L. Zhang. 2013. Research of JMS based message oriented middleware for cluster. In *Proceedings of the International Conference on Computational and Information Sciences*. 1628–1631. DOI: https://doi.org/10.1109/ICCIS.2013.426

[78] IBM. 2019. Developing client applications. Retrieved from https://www.ibm.com/support/knowledgecenter/en/SSWMAJ_5.0.0/com.ibm.ism.doc/Developing/develop_guide.html.

[79] IBM Integration Bus. 2019. Configuring the integration node to use SSL with JMS nodes. Retrieved from https://ibm.co/36ENEDH.

[80] O. Corporation. 2010. Using JAAS-Based Authentication. Retrieved from https://docs.oracle.com/cd/E19879-01/820-6740/gepfq/index.html.

[81] G. S. Kalra. 2014. A pentesters guide to hacking activeMQ-based JMS applications. Retrieved from https://bit.ly/2P43Vg6.

[82] Object Management Group (OMG). 2019. The Real-Time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol (DDSI-RTPS) Version 2.3. Retrieved from https://www.omg.org/spec/DDSI-RTPS/2.3/PDF.

[83] OMG. 2012. Extensible and dynamic topic types for dds. Retrieved from https://www.omg.org/spec/DDS-XTypes/About-DDS-XTypes/.

[84] L. Bertaux, A. Hakiri, S. Medjiah, P. Berthou, and S. Abdellatif. 2014. A DDS/SDN based communication system for efficient support of dynamic distributed real-time applications. In *Proceedings of the IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications*. 77–84. DOI: https://doi.org/10.1109/DS-RT.2014.18

[85] G. Yoon, J. Choi, H. Park, and H. Choi. 2016. Topic naming service for DDS. In *Proceedings of the International Conference on Information Networking (ICOIN'16)*. 378–381. DOI: https://doi.org/10.1109/ICOIN.2016.7427138

[86] A. Alaerjan, D. Kim, H. Ming, and K. Malik. 2018. Using DDS based on unified data model to improve interoperability of smart grids. In *Proceedings of the IEEE International Conference on Smart Energy Grid Engineering (SEGE'18)*. 110–114. DOI: https://doi.org/10.1109/SEGE.2018.8499513

[87] P. Peniak and M. Franekova. 2015. Open communication protocols for integration of embedded systems within industry 4.0. In *Proceedings of the International Conference on Applied Electronics (AE'15)*. 181–184.

[88] OMG. 2018. DDS Security. Retrieved from https://www.omg.org/spec/DDS-SECURITY/About-DDS-SECURITY/.

[89] T. White and M. N. Johnstone. 2017. An investigation into some security issues in the DDS messaging protocol. In *Proceedings of the Australian Information Security Management Conference*. 132–139. DOI: https://doi.org/10.4225/75/5a84fcff95b52

[90] M. J. Michaud, T. Dean, and S. P. Leblanc. 2018. Attacking OMG data distribution service (DDS) based real-time mission critical distributed systems. In *Proceedings of the 13th International Conference on Malicious and Unwanted Software (MALWARE'18)*. 68–77. DOI: https://doi.org/10.1109/MALWARE.2018.8659368

[91] P. Saint-andre. 2011. Extensible messaging and presence protocol (XMPP): Core. Retrieved from https://xmpp.org/rfcs/rfc6120.html.

[92] Peter Saint-Andre. 2014. Extensible messaging and presence protocol (XMPP): Core. Retrieved from https://xmpp.org/rfcs/rfc6120.html.

[93] K. Zeilenga. 2019. XEP-0258: Security Labels in XMPP. Retrieved from https://xmpp.org/extensions/xep-0258.html.

[94] C. Davidland and L. George. 2019. XEP-0419: Improving baseline security in XMPP. Retrieved from https://xmpp.org/extensions/xep-0419.html.

[95] P. Saint-andre. 2018. XEP-0205: Best practices to discourage denial of service attacks. Retrieved from https://xmpp.org/extensions/xep-0205.html.

[96] B. Chifor, S. Teican, M. Togan, and G. Gugulea. 2018. A flexible authorization mechanism for enterprise networks using smart-phone devices. In *Proceedings of the International Conference on Communications (COMM'18)*. 437–440. DOI: https://doi.org/10.1109/ICComm.2018.8484268

[97] A. Esser and C. Serrao. 2018. Wi-Fi network testing using an integrated evil-twin framework. In *Proceedings of the 5th International Conference on Internet of Things: Systems, Management, and Security*. 216–221. DOI: https://doi.org/10.1109/IoTSMS.2018.8554388

[98] C. Sudar, S. K. Arjun, and L. R. Deepthi. 2017. Time-based one-time password for Wi-Fi authentication and security. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI'17)*. 1212–1216. DOI: https://doi.org/10.1109/ICACCI.2017.8126007

[99] B. Li, H. Yu, and F. Tan. 2018. Wireless network security detection system design based on client. In *Proceedings of the International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS'18)*. 227–230. DOI: https://doi.org/10.1109/ICITBS.2018.00066

[100] Z. Liu and J. Zhang. 2018. Launching low-rate dos attacks with cache-enabled wifi offloading. In *Proceedings of the 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN'18)*. 171–176. DOI: https://doi.org/10.1109/MSN.2018.00028

[101] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo. 2014. A survey on security aspects for LTE and LTE-a networks. *IEEE Commun. Surv. Tutorials* 16, 1 (2014), 283–302. DOI: https://doi.org/10.1109/SURV.2013.041513.00174

[102] Y. Mehmood, C. Görg, M. Muehleisen, and A. Timm-Giel. 2015. Mobile M2M communication architectures, upcoming challenges, applications, and future directions. *EURASIP J. Wirel. Commun. Netw.* 250 (2015). DOI: https://doi.org/10.1186/s13638-015-0479-y

[103] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas. 2010. LTE-advanced: Next-generation wireless broadband technology. *IEEE Wirel. Commun.* 17, 3 (2010), 10–22. DOI: https://doi.org/10.1109/MWC.2010.5490974

[104] T. Specification. 2015. Security architecture (3GPP TS 33.401 version 12.13.0 Release 12). Retrieved from https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296.

[105] H. Shariatmadari et al. 2015. Machine-type communications: Current status and future perspectives toward 5G systems. *IEEE Commun. Mag* 53, 9 (2015), 10–17. DOI: https://doi.org/10.1109/MCOM.2015.7263367

[106] 3GPP. 2012. Service requirements for Home Node B (HNB) and Home eNode B (HeNB). Retrieved from https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=626.

[107] 3GPP. 2017. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN). Retrieved from https://bit.ly/2tfmBBr.

[108] S. Teral. 2019. 5G best choice architecture. *IHS Markit Technol.* Retrieved from https://cdn.ihs.com/www/prot/pdf/0519/IHSMarkit5GBestChoiceArchitecture.pdf.

[109] 5G Americas. 2018. The evolution of security in 5G. Retrieved from https://www.5gamericas.org/wp-content/uploads/2019/08/5G-Security-White-Paper-07-26-19-FINAL.pdf.

[110] M. Suryanegara, A. S. Arifin, and M. Asvial. 2017. The IoT-based transition strategy towards 5G. In *Proceedings of the International Conference on Big Data and Internet of Thing*. 186–190. DOI: https://doi.org/10.1145/3175684.3175728

[111] 5G Americas. 2018. The evolution of security in 5G. Retrieved from https://www.5gamericas.org/wp-content/uploads/2019/08/5G-Security-White-Paper-07-26-19-FINAL.pdf.

[112] Raphael J. Cohn and Richard J. Coppen. 2015. MQTT Version 3.1.1. Retrieved from http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html.

[113] M. Version. 2019. MQTT Version 5.0. Retrieved from https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.

[114] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi. 2017. Internet of things: Survey and open issues of MQTT protocol. In *Proceedings of the International Conference on Engineering & MIS (ICEMIS'17)*. 1–6. DOI: https://doi.org/10.1109/ICEMIS.2017.8273112

[115] M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai. 2018. Analysis of vulnerabilities in MQTT security using shodan API and implementation of its countermeasures via authentication and ACLs. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI'18)*. 2244–2250. DOI: https://doi.org/10.1109/ICACCI.2018.8554472

[116] M. Erber. 2019. Role based access control to secure an MQTT broker. Retrieved from https://www.hivemq.com/blog/rbac-for-the-control-center-with-ese/.

[117] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya. 2016. Authorization mechanism for MQTT-based internet of things. In *Proceedings of the IEEE International Conference on Communications Workshops (ICC'16)*. 290–295. DOI: https://doi.org/10.1109/ICCW.2016.7503802

[118] T. H. Team. 2015. TLS/SSL-MQTT security fundamentals. Retrieved from https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl/.

[119] S. H. Ramos, M. T. Villalba, and R. Lacuesta. 2018. MQTT security: A novel fuzzing approach. *Wirel. Commun. Mob. Comput.* 11 (2018).

[120] S. Andy, B. Rahardjo, and B. Hanindhito. 2017. Attack scenarios and security analysis of MQTT communication protocol in IoT system. In *Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI'17)*. 1–6. DOI: https://doi.org/10.1109/EECSI.2017.8239179

[121] V. Lakkundi and K. Singh. 2014. Lightweight DTLS implementation in CoAP-based internet of things. In *Proceedings of the 20th Annual International Conference on Advanced Computing and Communications (ADCOM'14)*. 7–11. DOI: https://doi.org/10.1109/ADCOM.2014.7103240

[122] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor. 2018. CoAP (constrained application protocol) over TCP, TLS, and WebSockets. Retrieved from https://tools.ietf.org/html/rfc8323.

[123] R. A. Rahman and B. Shah. 2016. Security analysis of IoT protocols: A focus in CoAP. In *Proceedings of the 3rd MEC International Conference on Big Data and Smart City (ICBDSC'16)*. 1–7. DOI: https://doi.org/10.1109/ICBDSC.2016.7460363

[124] T. A. Alghamdi, A. Lasebae, and M. Aiash. 2013. Security analysis of the constrained application protocol in the internet of things. In *Proceedings of the 2nd International Conference on Future Generation Communication Technologies (FGCT'13)*. 163–168. DOI: https://doi.org/10.1109/FGCT.2013.6767217

[125] A. Capossele, V. Cervo, G. De Cicco, and C. Petrioli. 2015. Security as a CoAP resource: An optimized DTLS implementation for the IoT. In *Proceedings of the IEEE International Conference on Communications (ICC'15)*. 549–554. DOI: https://doi.org/10.1109/ICC.2015.7248379

[126] J. Granjal, E. Monteiro, and J. S. Silva. 2015. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Commun. Surv. Tutorials* 17, 3 (2015), 1294–1312. DOI: https://doi.org/10.1109/COMST.2015.2388550

[127] S. Arvind and V. A. Narayanan. 2019. An overview of security in CoAP: Attack and analysis. In *Proceedings of the 5th International Conference on Advanced Computing & Communication Systems (ICACCS'19)*. 655–660. DOI: https://doi.org/10.1109/ICACCS.2019.8728533

[128] N. Naik. 2017. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP, and HTTP. In *Proceedings of the IEEE International Systems Engineering Symposium (ISSE'17)*. 1–7. DOI: https://doi.org/10.1109/SysEng.2017.8088251

[129] N. S. Han. 2015. Semantic service provisioning for 6LoWPAN: Powering internet of things applications on web. Institut National des Télécommunications.

[130] M. Phillips, P. Adams, D. Rokicki, and E. Johnson. 2011. URI Scheme for Java(tm) Message Service 1.0. Retrieved from https://tools.ietf.org/html/rfc6167.

[131] R. Cohn. 2012. A Comparison of AMQP and MQTT. Retrieved from https://lists.oasis-open.org/archives/amqp/201202/msg00086/StormMQ_WhitePaper_-_A_Comparison_of_AMQP_and_MQTT.pdf.

[132] RabittMQ. 2019. Authentication, Authorisation, *Access Control*. Retrieved from https://www.rabbitmq.com/access-control.html.

[133] D. Braue. 2019. Small, unsophisticated developers perpetuating IoT security lapses. Retrieved from https://www.cso.com.au/article/560521/small-unsophisticated-developers-perpetuating-iot-security-lapses-ibm/.

[134] I. N. Mcateer, M. I. Malik, Z. Baig, and P. Hannay. 2017. Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things. In *Proceedings of the Australian Information Security Management Conference*. 70–80. DOI: https://doi.org/10.4225/75/5a84f4a695b4c

[135] I. Bluetooth SIG. 2016. Bluetooth core specification version 5.0. Retrieved from https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=421043.

[136] Bluetooth SIG Proprietary. 2019. Bluetooth Core Specification v5.1. Retrieved from https://bit.ly/2REkBwf.

[137] K. Ren. 2016. Bluetooth pairing part three low-energy legacy pairing passkey entry. Retrieved from https://blog.bluetooth.com/bluetooth-pairing-passkey-entry.

[138] S. Figueroa Lorenzo, J. Añorga Benito, P. García Cardarelli, J. Alberdi Garaia, and S. Arrizabalaga Juaristi. 2019. A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis. *Technologies* 7, 1 (2019) 15. DOI: https://doi.org/10.3390/technologies7010015

[139] M. Ryan. 2015. Crackle. Retrieved from https://lacklustre.net/projects/crackle/.

[140] P. Cope, J. Campbell, and T. Hayajneh. 2017. An investigation of Bluetooth security vulnerabilities. *Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC'17)*. 1–7. DOI: https://doi.org/10.1109/CCWC.2017.7868416

[141] J. Padgette et al. 2017. NIST Special Publication 800-121 Revision 2 Guide to Bluetooth Security. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf.

[142] Zigbee Alliance. 2019. ZigBee Specification v 1.0. Retrieved from https://www.zigbee.org/wp-content/uploads/2014/11/docs-05-3474-20-0csg-zigbee-specification.pdf.

[143] Zigbee Alliance. 2012. Zigbee Specification v2. Retrieved from https://zigbee.org/download/zigbee-3-0-base-device-behavior-specification/.

[144] T. Zillner and S. Strobl. 2015. Zigbee exploited. The good, the bad, and the ugly. In *Black Hat USA*. Retrieved from https://zigbee.org/download/new-white-paper-zigbee-securing-the-wireless-iot/.

[145] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, P. Toivanen, and K. Campus. 2014. Three practical attacks against zigbee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned. In *2014 14th International Conference on Hybrid Intelligent Systems*. 199–206. DOI: https://doi.org/10.1109/HIS.2014.7086198

[146] X. Cao, D. M. Shila, S. Member, Y. Cheng, and S. Member. 2016. Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks. *IEEE Internet Things J.* 3, 5 (2016), 816–829. DOI: https://doi.org/10.1109/JIOT.2016.2516102

[147] NFC Forum. 2017. Core protocol technical specifications. Retrieved from https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/protocol-technical-specifications/.

[148] S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde. 2019. An attribute-based access control using chaincode in RFID systems. In *Proceedings of the 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS'19)*. 1–5. DOI: https://doi.org/10.1109/NTMS.2019.8763824

[149] S. Figueroa, J. Añorga, and S. Arrizabalaga. 2019. An attribute-based access control model in RFID systems based on blockchain decentralized applications for healthcare environments. *Computers* 8, 3 (2019). DOI: https://doi.org/10.3390/computers8030057

[150] F. D. Garcia et al. 2008. Dismantling MIFARE classic. *Lect. Notes Comput. Sci.* 5283 (2008), 97–114. DOI: https://doi.org/10.1007/978-3-540-88313-5-7

[151] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. 2012. Practical relay attack on contactless transactions by using NFC mobile phones. *Cryptol. Info. Secur. Ser.* 8, 21–32, 2012. DOI: https://doi.org/10.3233/978-1-61499-143-4-21

[152] G. P. Hancke. 2008. Eavesdropping attacks on high-frequency RFID tokens. In *Proceedings of the 4th Workshop on RFID Security (RFIDsec'08)*. 1–36.

[153] LoRa Alliance Technical Commitee. 2017. LoRaWAN 1.1 Specification. Retrieved from https://lora-alliance.org/resource-hub/lorawanr-specification-v11.

[154] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes. 2017. Exploring the security vulnerabilities of lora. In *Proceedings of the 3rd IEEE International Conference on Cybernetics (CYBCONF'17)*. 1–6. DOI: https://doi.org/10.1109/CYBConf.2017.7985777

[155] S. Chacko and M. D. Job. 2018. Security mechanisms and vulnerabilities in LPWAN. *IOP Conf. Ser. Mater. Sci. Eng.* 396, 1 (2018). DOI: https://doi.org/10.1088/1757-899X/396/1/012027

[156] R. S. Sinha, Y. Wei, and S.-H. Hwang. 2017. A survey on LPWA technology: LoRa and NB-IoT. *ICT Express* 3, 1 (2017), 14–21. DOI: https://doi.org/.org/10.1016/j.icte.2017.03.004

[157] C. B. Mwakwata, H. Malik, M. M. Alam, Y. Le Moullec, S. Parand, and S. Mumtaz. 2019. Narrowband internet of things (NB-IoT): From physical (PHY) and media access control (MAC) layers perspectives. *Sensors (Switzerland)* 19, 11 (2019), 1–34. DOI: https://doi.org/10.3390/s19112613

[158] M. Iot and S. Report. 2019. Security features of LTE-M and NB-IoT networks. Retrieved from https://www.gsma.com/iot/resources/security-features-of-ltem-nbiot.

[159] F. L. Coman, K. M. Malarski, M. N. Petersen, and S. Ruepp. 2019. Security issues in internet of things: Vulnerability analysis of LoRaWAN, sigfox and NB-IoT. In *Proceedings of the Global IoT Summit (GIoTS'19)*. 1–6. DOI: https://doi.org/10.1109/GIOTS.2019.8766430

[160] D. Raposo, A. Rodrigues, S. Sinche, J. S. Silva, and F. Boavida. 2018. Securing wirelessHART: Monitoring, exploring and detecting new vulnerabilities. In *Proceedings of the IEEE 17th International Symposium on Network Computing and Applications (NCA'18)*. 1–9. DOI: https://doi.org/10.1109/NCA.2018.8548060

[161] R. Budampati and S. Kolavennu. 2016. *Industrial Wireless Sensor Networks. Monitoring, Control and Automation*. Elsevier.

[162] S. Raza, A. Slabbert, T. Voigt, and K. Landernäs. 2009. Security considerations for the wirelessHART protocol. In *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation*. 1–8. DOI: https://doi.org/10.1109/ETFA.2009.5347043

[163] C. Alcaraz and J. Lopez. 2010. A security analysis for wireless sensor mesh networks in highly critical systems. *IEEE Trans. Syst. Man Cybern. Part C* 40, 4 (2010), 419–428. DOI: https://doi.org/10.1109/TSMCC.2010.2045373

[164] L. Bayou, D. Espes, N. Cuppens-Boulahia, and F. Cuppens. 2017. Security analysis of wirelessHART communication scheme bt—Foundations and practice of security. (2017), 223–238. Retrieved from https://hal.archives-ouvertes.fr/hal-01411385/document.

[165] C. Gomez and J. Paradells. 2010. Wireless home automation networks: A survey of architectures and technologies. *IEEE Commun. Mag.* 92–101.

[166] M. Smith. 2018. EZ-Wave: A Z-Wave hacking tool capable of breaking bulbs abusing Z-Wave devices. Retrieved from https://www.csoonline.com/article/3024217/ez-wave-z-wave-hacking-tool-capable-of-breaking-bulbs-and-abusing-z-wave-devices.html.

[167] L. Rouch. 2019. A universal controller to take over a z-wave network. *Black Hat Present.* Retrieved from https://ubm.io/2U4WKau.

[168] B. Fouladi and S. Ghanoun. 2013. Security evaluation of the z-wave wireless protocol. *Black Hat* 6 (2013). Retrieved from https://bit.ly/2PurBZS.

[169] P. N. e. V. (PNO). 2014. PROFINET system description. *Technology and Application*. Retrieved from https://bit.ly/2vmu3ey.

[170] M. Yang and G. Li. 2014. Analysis of PROFINET IO communication protocol. In *Proceedings of the 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC'14)*. 945–949. DOI: https://doi.org/10.1109/IMCCC.2014.199

[171] C. Henning. 2014. PROFINET for network geeks (and those who want to be). Retrieved from https://us.profinet.com/profinet-network-geeks-want/.

[172] Z. Drias, A. Serhrouchni, and O. Vogel. 2015. Taxonomy of attacks on industrial control protocols. In *Proceedings of the International Conference onPerformance Engineering (ICPE'15) and International Conference on New Technologies and Distributed Systems (NTDS'15)*. DOI: https://doi.org/10.1109/NOTERE.2015.7293513

[173] J. Åkerberg and M. Björkman. 2009. Exploring security in PROFINET IO. In *Proceedings of the International Computer Software and Applications Conference*, vol. 1, 406–412. DOI: https://doi.org/10.1109/COMPSAC.2009.61

[174] Tridium Europe Limited. 2017. Tridium Niagara Framework Smart Buildings Guide Specification. West Sussex.

[175] A. Mirian et al. 2016. An internet-wide view of ICS devices. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust (PST'16)*. 96–103. DOI: https://doi.org/10.1109/PST.2016.7906943

[176] P. Zito 2017. What is tridium? Part 1. Retrieved from http://buildingautomationmonthly.com/what-is-tridium/.

[177] Tridium. 2017. Niagara AX 3. 8u3 Features overview. Retrieved from https://bit.ly/2rhOx6R.

[178] B. Rios. 2014. Owning a building. exploiting access control and facility management systems. In *Black Hat USA* (2014), 89. Retrieved from https://ubm.io/2YurdyO.

[179] A. M. Elshaer, M. M. Elrakaiby, and M. E. Harb. 2018. Autonomous car implementation based on CAN bus protocol for IoT applications. In *Proceedings of the 13th International Conference on Computer Engineering and Systems (ICCES'18)*. 275–278. DOI: https://doi.org/10.1109/ICCES.2018.8639206

[180] Z. King and S. Yu. 2017. Investigating and securing communications in the controller area network (CAN). In *Proceedings of the International Conference on Computing, Networking and Communications (ICNC'17)*. 814–818. DOI: https://doi.org/10.1109/ICCNC.2017.7876236

[181] D. Beresford. 2011. Exploiting siemens simatic S7 PLCs. In *Black Hat USA* (2011), 1–26. Retrieved from https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf.

[182] Siemens. 2013. S7-1500—Industrial Ethernet CP. Retrieved from https://support.industry.siemens.com/cs/attachments/76476576/GH_CP1543-1_76_en-US.pdf.

[183] D. Nardella. 2018. Snap 7. Retrieved from http://snap7.sourceforge.net/.

[184] Siemens AG. 2019. Security with SIMATIC control. Retrieved from https://support.industry.siemens.com/cs/attachments/90885010/77431846_Security_SIMATIC_DOKU_V20_en.pdf.

[185] A. Timorin. 2014. SCADA deep inside: protocols and security mechanisms. In *Hacktivity* (2014), 84. Retrieved from https://bit.ly/2YF0KPa.

[186] ODVA. 2019. Securing EtherNet/IP Networks. Retrieved from https://bit.ly/2Ywb1go.

[187] F. Tacliad, T. D. Nguyen, and M. Gondree. 2017. DoS exploitation of allen-bradley's legacy protocol through fuzz testing. In *Proceedings of the CEUR Workshop Proceedings*. 54–57. DOI: https://doi.org/10.1145/nnnnnnn.nnnnnnn

[188] Hart Communication Fundation. 2013. Hart communication. *application guide*. Retrieved from https://www.fieldcommgroup.org/sites/default/files/technologies/hart/ApplicationGuide_r7.1.pdf.

[189] M. Duijsens. 2019. WirelessHART: A security analysis. *Eindhoven University of Technology*. Retrieved from https://pdfs.semanticscholar.org/6d53/d09b602a6b315dc79bc746efa5bd4ba13e02.pdf.

[190] F. C. Group. 2019. Digital transformation in the age of IIoT. Retrieved from https://bit.ly/36jjUga.

[191] S. Shah and P. Bhargava. 2013. HART over IP for industrial automation networks. Retrieved from https://www.einfochips.com/images/in_sight/HART_Over_IP_EEIOL_2013JAN30_NET_TA_01.pdf.

[192] F. C. Group. 2019. Hart technology. *Leading the Digital Transformation*. Retrieved from https://www.fieldcommgroup.org/sites/default/files/technologies/hart/HART%20brochure%20web%20view.pdf.

[193] S. I. A. GmbH. 2015. HART-IP solution communicates at Ethernet speed. Retrieved from https://industrial.softing.com/fileadmin/sof-files/pdf/de/ia/Articles/HART-IP_IEB-1502.pdf].

[194] A. Bolshev. 2019. HART as an attack vector: From current loop to application layer. In *Proceedings of S4x14*. Retrieved from https://documents.pub/document/hart-as-an-attack-vector-from-current-loop-to-application-layer.html.

[195] BACnet. 1997. BACnet: Answers to frequently asked questions. *HPAC Heating/Piping/AirConditioning* (1997), 47–51.

[196] R. Automation. 2006. BACnet MS/TP Adapter. Retrieved from https://bit.ly/38PkxiL.

[197] L. K. Haakenstad. 1999. The open protocol standard for computerized building systems: BACnet. *Proceedings of the IEEE International Conference on Control Applications*. Vol. 2, 1585–1590. DOI: https://doi.org/10.1109/CCA.1999.801208

[198] H. Merz, T. Hansemann, and C. Hübner. 2018. *BACnet BT—Building automation: Communication Systems with EIB/KNX, LON and BACnet*, H. Merz, T. Hansemann, and C. Hübner (Eds). Springer International Publishing, Cham, 209–302.

[199] B. Isler. 2010. BACnet to leverage IT. A Whitepaper on BACnet/IT. Retrieved from http://www.bacnet.org/Bibliography/BACnet_IT_WhitePaper_2016121.pdf.
[200] D. Robin et al. 2010. Ashrae Standard BACnet—A data communication protocol for building automation and control networks. Retrieved from http://www.bacnet.org/Addenda/Add-135-2008t.pdf.
[201] J. Kaur, J. Tonejc, S. Wendzel, and M. Meier. 2015. Securing BACnet's Pitfalls BT - ICT Systems security and privacy protection. In *SEC 2015: ICT Systems Security and Privacy Protection*. 616–629. DOI: https://doi.org/10.1007/978-3-319-18467-8_41
[202] B. Bowers. 2013. How to own a building bacNET attack framework. In *ShmooCon*. Retrieved from https://archive.org/details/Shmoocon.
[203] Z. Pan, S. Hariri, and Y. Al-Nashif. 2014. Anomaly based intrusion detection for building automation and control networks. In *Proceedings of the IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA'14)*. 72–77.
[204] N. I. Corporation. 2017. The Modbus Protocol In-Depth. Retrieved from http://www.ni.com/white-paper/52134/en/.
[205] M. Organization. 2018. Modbus/TCP Security. Retrieved from http://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf.
[206] M. K. Ferst, H. F. M. de Figueiredo, G. Denardin, and J. Lopes. 2018. Implementation of secure communication with modbus and transport layer security protocols. In *Proceedings of the 13th IEEE International Conference on Industry Applications (INDUSCON'18)*. 155–162. DOI: https://doi.org/10.1109/INDUSCON.2018.8627306
[207] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga. 2019. A role-based access control model in modbus SCADA systems: A centralized model approach. *Sensors* 19, 20 (2019). DOI: https://doi.org/10.3390/s19204455
[208] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi. 2008. Attack taxonomies for the modbus protocols. *Int. J. Crit. Infrastruct. Prot* 1 (2008), 37–44. DOI: https://doi.org/10.1016/j.ijcip.2008.08.003
[209] B. Chen, N. Pattanaik, A. Goulart, K. L. Butler-purry, and D. Kundur. 2015. Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed. In *Proceedings of the IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR'15)*, 2015, 1–6. DOI: https://doi.org/10.1109/CQR.2015.7129084
[210] O. Foundation. 2002. OPC alarms and events custom interface. Retrieved from http://advosol.com/OpcSpecs/OPC%20AE%201.10%20Specification.pdf.
[211] OPC Foundation. 2003. OPC historical data access specification. Retrieved from http://advosol.com/OpcSpecs/OPC%20HDA%201.20%20Specification.pdf.
[212] O. Foundation. 1998. OPC Overview. Retrieved from https://invent.ge/2Pp0ek2.
[213] B. P. Hunkar, OPC UA vs. OPC Classic. Retrieved from http://www.dsinteroperability.com/OPCClassicVSUA.pdf.
[214] D. P. E. Byres. 2009. OPC security white paper #1 understanding OPC and how it is deployed. Retrieved from http://www.opcti.com/opc-security-white-paper-1.aspx.
[215] B. Mcilvride and A. Thomas. 2008. OPC tunnelling—Know your options. Retrieved from https://bit.ly/2RwlxCp.
[216] D. Kominek. 2011. Effective OPC security for control systems—Solutions you can bank on. Retrieved from https://bit.ly/37yZtgf.
[217] Microsoft. 2018. How to configure RPC dynamic port allocation to work with firewalls. Retrieved from https://bit.ly/2GHS8ix.
[218] U. Steinkrauss. 2010. Whitepaper—Overview: OPC unified architecture. Technical overview and short description. Retrieved from http://www.ascolab.com/images/stories/ascolab/doc/ua_whitepaper_technicaloverview_e.pdf.
[219] J. Imtiaz and J. Jasperneite. 2013. Scalability of OPC-UA down to the chip level enables "internet of things." In *Proceedings of the IEEE International Conference on Industrial Informatics (INDIN'13)*. 500–505. DOI: https://doi.org/10.1109/INDIN.2013.6622935
[220] M. D. Wolfgang Mahnke, Stefan-Helmut Leitner, *OPC Unified Architecture*, 1st ed. Springer-Verlag, Berlin.
[221] N. Pocock, D. Kominek, and P. Hunkar. 2014. OPC UA security—How it works. In *Proceedings of the Microsoft Conference Center*. 54.
[222] Bundesamt für Sicherheit in der Informationstechnik (BSI). 2019. OPC UA Security Analysis. [Online]. Retrieved from https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf.
[223] C. Wester, N. Engelman, T. Smith, K. Odetunde, B. Anderson, and J. Reilly. 2015. The role of the SCADA RTU in today's substation. In *Proceedings of the 68th Annual Conference for Protective Relay Engineers* 2015, 622–628. DOI: https://doi.org/10.1109/CPRE.2015.7102199
[224] Ken Curtis. 2005. A DNP3 protocol primer introduction. Retrieved from https://www.academia.edu/35049898/A_DNP3_Protocol_Primer.
[225] R. Amoah. 2016. Formal security analysis of the DNP3-secure authentication protocol. Queensland University of Technology Australia.

[226] F. Cleveland. 2016. IEC TC57 WG15: IEC 62351 Security standards for the power system information infrastructure. Retrieved from http://iectc57.ucaiug.org/wg15public/Public%20Documents/White%20Paper%20on%20Security%20Standards%20in%20IEC%20TC57.pdf.
[227] G. Devarajan. 2007. Unraveling SCADA protocols: Using sulley fuzzer. In *DefCon*. Retrieved from https://bit.ly/2PAhK54.
[228] C. S. Pe, A. Crain, C. Sistrunk, and A. Crain. 2014. Master serial killer. In *Proceedings of S4x14*. Retrieved from http://blog.cci-es.org/2014/01/review-of-master-serial-killer-project.html.
[229] S. Bratus et al. 2016. Implementing a vertically hardened DNP3 control stack for power applications. In *Proceedings of S4x16*. 9. DOI: https://doi.org/10.1145/3018981.3018985
[230] IEEE. 2012. *IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables*, 5th ed. IEEE, New York, NY.
[231] A. F. Snyder and M. T. G. Stuber. 2007. The ANSI C12 protocol suite—Updated and now with network capabilities. *Proceedings of the Power Systems Conference on Advanced Metering, Protection Control Communucation and Distributed Resources (PSC'07)*. 117–122. DOI: https://doi.org/10.1109/PSAMP.2007.4740906
[232] C. Greer et al. 2014. NIST framework and roadmap for smart grid interoperability standards, release 3.0. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1108r3.pdf.
[233] I. T. Union. 1999. X.237 bis. Retrieved from https://bit.ly/2rrr96O.
[234] F. D. R. Inc. 2017. Energy communications management exchange AMI/smartgrid engineering and consulting services. Retrieved from https://www.ecmx.org/public/.
[235] K. Minematsu, S. Lucks, H. Morita, and T. Iwata. 2014. Attacks and security proofs of EAX-prime. *Lect. Notes Comput. Sci.* 8424 (2014), 327–347. DOI: https://doi.org/10.1007/978-3-662-43933-3_17
[236] Y. Chen, Z. Zhu, B. Xu, K. Fan, and K. Wang. 2016. The use of IEC61850 for distribution automation. In *Proceedings of the China International Conference on Electricity Distribution (CICED'16)*. 1–4. DOI: https://doi.org/10.1109/CICED.2016.7576251
[237] Y. Liang and R. H. Campbell. 2007. Understanding and simulating the IEC 61850 standard. *IEEE Trans. Power Deliv.* 22 (2007), 1482–1489.
[238] International Standard. 2003. IEC 61850-7-1: Communication networks and systems in substations–Part 7-1: Basic communication structure for substation and feeder equipment–Principles and models. Retrieved from https://webstore.iec.ch/p-preview/info_iec61850-7-1%7Bed1.0%7Den.pdf.
[239] S. Patel. 2017. IEC-61850 protocol analysis and online intrusion detection system for SCADA networks using machine learning. University of Victoria. Retrieved from https://dspace.library.uvic.ca/handle/1828/9347.
[240] O. S. G. Platform. 2019. Introduction to the open smart grid platform. IEC-61850. Retrieved from http://documentation.opensmartgridplatform.org/Protocols/IEC61850/index.html.
[241] M. Strobel, N. Wiedermann, and C. Eckert. 2016. Novel weaknesses in IEC 62351 protected smart grid control systems. *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm'16)*. 266–270. DOI: https://doi.org/10.1109/SmartGridComm.2016.7778772
[242] IEC. 2006. International standard IEC 60870-5-104. DOI: https://doi.org/IEC 61672-1
[243] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang. 2013. Intrusion detection system for IEC 60870-5-104 based SCADA networks. In *Proceedings of the IEEE Power & Energy Society General Meeting*. 1–5. DOI: https://doi.org/10.1109/PESMG.2013.6672100
[244] J. T. Michalski, A. Lanzone, J. Trent, and S. Smith. 2007. Secure ICCP integration considerations and recommendations. Retrieved from https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/19-Secure_ICCP_Integration.pdf.
[245] QED Secure Solutions. 2019. Risk Scoring System. Retrieved from https://www.riskscoringsystem.com/.
[246] A. Manion. 2019. [Online]. TEMSL. Retrieved from https://www.youtube.com/watch?v=-6cThOCm9co&feature=youtu.be&t=1303.
[247] C. Bodungen. 2019. Industrial vulnerability scoring system (IVSS). Retrieved from https://securingics.com/IVSS/IVSS.html.
[248] MITRE Corporation. 2018. Common Vulnerabilities and Exposures (CVE). Retrieved from https://cve.mitre.org/.
[249] MITRE Corporation, 2019. Common attack pattern enumeration and classification (CAPEC). Retrieved from https://capec.mitre.org/index.html.
[250] MITRE Corporation. 2018. Common weakness enumeration: CWE. Retrieved from https://cwe.mitre.org/index.html.
[251] MITRE Corporation. 2018. Common platform enumeration: CPE dictionary. Retrieved from https://cpe.mitre.org/.
[252] Offensive Security. 2019. Offensive Security. Retrieved from https://www.offensive-security.com/.
[253] O. Security. 2019. The exploit database—Offensive security. Retrieved from https://www.exploit-db.com/.
[254] Micrososoft. 2019. Security Bulletins. Retrieved from https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/securitybulletins.

[255] National Institute of Standards and Technology (NIST). 2018. National Vulnerability Database. Retrieved from https://nvd.nist.gov/vuln.
[256] A. Dulaunoy. 2018. CVE-Search (Common Vulnerabilities and Exposures). Retrieved from https://www.cvedetails.com/.
[257] S. Özkan. 2018. CVE-details (Common Vulneravility Exposure). Retrieved from www.cvedetails.com.
[258] R. Borgaonkar. 2019. New vulnerabilities in 5G Security Architecture & Countermeasures. Retrieved from https://infosec.sintef.no/en/informasjonssikkerhet/2019/08/new-vulnerabilities-in-5g-security-architecture-countermeasures/.
[259] A. Shaik and R. Borgaonkar. 2019. Black Hat 2019: 5G Security Flaw Allows MiTM, Targeted Attacks. Retrieved from https://blacklakesecurity.com/black-hat-2019-5g-security-flaw-allows-mitm-targeted-attacks/.
[260] 3GPP System Architecture Evolution (SAE). 2018. Retrieved from https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296.
[261] 3GPP. 2018. Security architecture and procedures for 5G System. *Technical Specification (TS)* 33.501. Retrieved from http://www.3gpp.org/DynaReport/33501.htm.
[262] G. Association. 2019. GSMA mobile security hall of fame. 2019. Retrieved from https://www.gsma.com/security/gsma-mobile-security-hall-of-fame/.

Received September 2019; revised December 2019; accepted January 2020