Security issues of the Web of Things: challenges and solutions

Abdelaziz Laaychi[1,*], *Mariam* Tanana[2] and *Saiida* Lazaar[3]

[1,2] Innovative Technologies Laboratory (LTI). National School of Applied Sciences – Tangier, Morocco
[3] Mathematics, Computer Science and Application (ERMIA Team). National School of Applied Sciences – Tangier, Morocco

Abstract. Our world is hugely impacted by revolutionary innovations in telecommunication technologies where connected objects increase rapidly. The Web of Things (WoT) represents the most promising application models of the Internet of Things (IoT). The ultimate goal of the WoT is to create a search engine where the user or devices can find other devices anywhere and anytime in order to take advantage of the resources of other devices. Nowaday, IoT systems imperatively need scalability and flexible coupling, which are easily offered by WoT. However, researchers have raised some concerns about the security of WoT. This paper provides a recent review of WoT literature. A special attention is accorded to the most relevant WoT security issues. Our paper also deals with the most important attacks in the domain and with some selected security solutions.

1 Introduction

According to the statistics connected devices will reach 16.4 billion by 2022 [1]. These devices form the IoT [2] which is a "*system of physical objects that can be discovered, monitored, controlled, or interacted with, by electronic devices that communicate through numerous networking interfaces and eventually can be connected to the wider internet*" [3]. The limitations of the IoT become visible as soon as we want to integrate devices from differents manufacturers into a single application or system. Indeed, each device or object has their own specifications and protocols supported [3].

The WoT is a subset of the IoT, where web standards are used to seamlessly integrate and connect physical objects [3]. The main goal of the WoT is to extend the different techniques and tools that exist on the Web to the construction of IoT networks. In this way, the WoT will be able to reuse the existing Web protocols (http, https, WebSocket, ...) to communicate, without the creation of new complex protocols that might not facilitate interoperability between IoT. However, to meet the challenges of WoT, privacy and security requirements play an important role. These requirements include data integrity, privacy, reliability, access control and authentication within the WoT network [4].

The rest of this paper is organized as follows. In section 2, we provide an overview of the origin, architectures and utility of WoT. Section 3 provide a survey about security challenges that encountered by WoT, and the proposed solutions to fix security issues. Section 4 conclude the paper.

2 Overview of WoT

2.1 Definition and origin of WoT

The first version of the Electronic Product Code (EPC) network was launched by the Auto-ID Center to identify and recognize the delivery of goods in supply chains. This was the first time WoT caught the eye. WoT is treated as a further development of the IoT architecture. IoT was first mentioned in a document from the Auto-ID Center dealing with Electronic Product Code and written by David Brock in 2001 [5]. As a result, WoT has been seen as a future value that is essential for the Internet.

You can't find a universal definition that can really define what WoT is. Therefore, it is better to define the core concept of WoT. The core idea of WoT defines it in such a way that not only can everyday objects be equipped with sensors, networks and processing options, which in turn allow objects to communicate with one another, but these objects of the real world can also be included in the architecture. Although, the technology behind this core concept is not new, these technologies include the use of RFID (Radio Frequency Identification) and sensors in the industrial and manufacturing context to identify valuable items. WoT also represents the interconnection of devices, as well as their interconnection over the network [5].

Existing IoT approaches have some limitations and

* Corresponding author: abdelaziz.laaychi@etu.uae.ac.ma
[2] mtanana@uae.ac.ma
[3] slazaar@uae.ac.ma

issues because they do not use a standard application layer protocol for connected devices. We will describe in the following the advantage of using a WoT approach instead of the IoT.

Easier to program: In the WoT approach, we can interact with connected objects using web APIs. Developers can therefore use these same APIs to effortlessly develop simple web applications that can communicate quickly and easily with new devices.

Open and Extensible Standards: WoT is based on open web standards, which are stable. The choice of HTTP and REST protocols seems obvious when you want to allow public access to certain data on the devices.

Quick and Easy Deployment, Maintenance, and Integration: In the IoT, devices use various protocols depending on the network they belong to, and every time one of those protocols changes, the others all need to be updated. On the contrary, WoT devices integrate naturally into the web like any other device.

2.2 WoT architectures

WoT has more than one architecture, such as the user-adapted WoT architecture [6], the WoT Servient architecture [7] and the architecture of the WoT Open Platform [8]. But the four-layer WoT architecture [9] (see figure 1) remains the most well known. Now let's look at the different layers of this WoT architecture and describe their purpose:

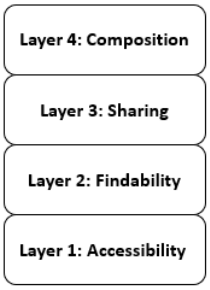

Fig. 1. The WoT Architecture.

Accessibility layer: The goal of this layer is simple, things can be integrated into the web by exposing their services through the use of a RESTful API, and using HTTP requests to communicate with each other, just like other Web resources [8][10].

Findability Layer: The main functionality of this layer is to explore and locate objects on the web. Objects can therefore be easily found and used by other WoT applications or http clients [11]. The strategy adopted here is to use the semantic standards of the web to define objects and their functions. This allows search engines and web indexes to find objects, and also to automatically generate user interfaces or tools to communicate with objects. In addition, it allows one machine to communicate with another machine based on a small set of standards defined formats [11].

Sharing Layer: This layer describes how to effectively and securely share data generated by objects across the Web. For this, other web protocols are useful. First, the TLS protocol which secures transactions on the web. Then there are other techniques such as web-based delegated authentication mechanisms, such as OAuth, which can be integrated into our object APIs [8]. The most important criterion for a WoT sharing platform is to be secure, so that access to smart objects is only possible for authorized persons. A WoT sharing platform should be simple and easy to use.

Composition Layer: The objective of this last layer is to incorporate the services and data offered by the objects into the high-level web, further simplifying the creation of applications that include virtual objects and web services [8].

The objective of this last layer is to incorporate the services and data offered by the objects into the high-level web.

Unlike OSI (Open System Interconnection) the WoT architecture is composed of layers that add functionality, as shown in Figure 1. Every layer helps to further merge objects with the Web and thus makes them more accessible by others applications. The WoT architecture stack starts where the OSI and Internet protocol stack ends. It looks at all the protocols and tools that are at the application layer and above (layer 7 and above). This means you don't have to worry about the underlying layers (1-6), because the WoT is only interested in the application layer protocols and what you do with them, not the underlying protocols used [9].

However, WoT not only has specific IoT properties, but also many security problems. In the next section we will present a survey about security concerns in WoT.

3 Security in WoT

3.1 WoT security challenges

Sharing objects and making them accessible on the web only to authorized persons (or other objects) with appropriate privileges poses many problems in terms of identity management (authentication and identification), privacy and security. As a result, these issues negatively impact trust in WoT as a standard platform that connects all types of objects.

We focus our attention to the various challenges, especially in the sharing layer of the WoT architecture.

Unauthorized Access: Unauthorized users can access a WoT devices of another person's account through bypassing the different authentication and authorization methods, and then get the possibility to modify any private data stored at the WoT device [12].

SSL Striping attack: Attackers try to convert encrypted and secure traffic to an unencrypted and insecure traffic. They can configure incorrectly unencrypted protocols making a request to TLS, in such a way as to not use Secure Socket Layer/Transport Layer Security (SSL/TLS) [13].

Eavesdropping: It is a type of man-in-the-middle attack in which confidentiality can be compromised as traffic flows between different objects. By which an attacker can have access to the credentials of the victim by stealing information such as login, and password [12].

Denial Of Service attack: The DoS attack affects the

entire network, prevents the accessibility of the server or a component of the IoT/WoT and thus violates one of the essential components of cybersecurity: Availability. One of the goals of hackers is to compromise the availability, as no administrator rights are required. Compared to compromising the confidentiality and integrity components to obtain and change confidential information. DoS has a more harmful impact on high profile organizations such as banks and governments, resulting in significant financial and time losses [14].

Tampering attack: It is mainly based on altering the parameters exchanged between a client and a server in order to change the data of an application, such as user credentials, permissions, etc. Usually, this information is saved in cookies, which is hidden in URL query strings used to enhance the functionality and control of the application. A tampering attack can be made by a malicious person who wants to modify the application to his advantage or by an attacker who wants to harm a third party by using a man-in-a-middle attack [12].

Virtual host confusion: This attack can occur if there is poor session cache management on the server side. By exploiting session cache vulnerabilities instead of an authenticated connection to the server, an attacker could establish a malicious connection to a virtual host. An environment such as WoT can be heavily affected by these types of attacks, as session cache sharing is very common [13].

To summarize, the issues of security, trust and privacy in the WoT require further research and improvement [4]. "*Privacy involves addressing the issues raised by sharing a web accessible object with others on the web, allowed just to authorized persons, hiding personal information as well as being able to control what happens with that information. Security concerns issues related to who will have access to the object and what they can do while they have access. Finally, trust concerns issues related to the interactions between different WoT entities on the web*" [4].

3.2 WoT security solutions

In order to face the threats already mentioned, the WoT must offer certain security and confidentiality guarantees, taking into consideration the properties of the network. The protection of the information and resources of the objects is important to keep them from being accessed by unauthorized parties.

The rest of the section gives an overview of various proposed security mechanisms, such as:

Identity management: Is based on authentication and authorization verification of users in order to access the desired information. Depending on the circumstances of the individuals in a system, identity management can define the identification rules of these individuals. Indeed, by assigning restrictions, the system can control the access of users to its resources. In identity management the determination of the appropriate policy is necessary to decide whether the entity is allowed on the network or not [4].

Confidentiality and integrity: This is an important property in order to guarantee the security of communications between the different elements of the WoT environment. Also, to prevent the interception of data exchanged between parts of the system. Smart devices often suffer from low performance which makes encryption and cryptographic computation very difficult because of its huge consumption of device resources like memory. To overcome this problem CoAP is invented which is an end-to-end encryption solution, which can secure the communication at application layer and transport layer [4].

Authorization: Flexible access of authorized entities is an important objective of WoT to be a system that is open, the most used authorization architecture consisting of a server takes care of complex tasks requiring processing resources. The server is placed between the requester and the smart device. However, the smart object must distinguish between requests from various parties and execute the correct authorization decision. LWAES [15] and OAuth [4] are lightweight algorithms that have emerged specifically to achieve this goal. CoAP Delegated Authentication and Authorization Framework (DCAF) [4] is another solution that allows delegating complicated cryptographic computations to the external entities and providing a secure DTLS channel between resource-constrained nodes. This protocol can be used to delegate the management of authentication and authorization of communicating peers to a trusted third party [16].

Access control: The main goal of the access control is to secure the data based on the attributes and the identity of the user. It protects the data against modification, viewing and copying by unauthorized users. Usually, Access control is used "*to protect system resources, by using limits on what users can do, who can access the data, what resources they have, and what activities are allowed on the data*" [4]. Many standard authorization models have been proposed for access control. As for example, attribute-based access control, role-based access control, dynamic authorization, and multi-level security using information flow [4].

Table 1 summarizes this section, giving the benefits and proposed solutions for each security mechanism.

Table 1. Solutions for the various WoT security mechanisms.

Security mechanisms	Benefits	Proposed solutions
The management of the identity	Describes individual's identity, authentication, and authorization	• User-centric model [12]. • Centralized federation model [12]. • Decentralized federation model [12].
Confidentiality and integrity	Prevent third parties from snooping on the information exchange between the different WoT modules.	• Lightweight Advanced Encryption Standard (LWAES) algorithm [15]. • Datagram Transport Layer

		Security (DTLS) protocol [17].
Authorization	Enable flexible access to only authorized parties.	• OAuth protocol [4]. • Delegated CoAP Authentication and Authorization Frame work (DCAF) [4].
Access control	Ensures that access to information or resources is given to ap proved parties only.	• Role Based Access Control (RBAC) [18]. • Attribute Based Access Control (ABA) [18].

In the above section, we examined how identity is managed in WoT architectures and the different identity management models. We then looked at how to ensure data confidentiality and integrity by securing the channels between communication devices. Finally, we presented authorization and access control and how they provide security in the WoT.

4 Conclusion

In this article, we discussed the WoT as an evolution of the IoT. We also presented the origin and utility of WoT, as well as a detailed explanation of its architecture. Security is an important feature of all objects. Here, we discussed several security challenges, especially in the sharing layer, like unauthorized access, SSL Striping attack, eavesdropping, Tampering attack, denial of service attack and virtual host confusion. We also focused on some security solutions such as identity management, authorization and access control, data confidentiality and integrity. We conclude that the research community needs to do more effort to ensure the safety of WoT.

References

1. K. Lueth (2020), State of the IoT 2020: 12 billion IoT connections, IoT Analytics, Last accessed 19 June 2021, https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/
2. E. Barka, S. Mathew, Y. Atif," Securing *the Web of Things with Role-Based Access Control*", Springer, C2SI, (2015).
3. N. Jabeur, H. Haddad, "*From intelligent web of things to social web of things",* (January 2016).
4. S. El Jaouhari, A. Bouabdallah, J. Bonnin. *"Security Issues Of The Web Of Things*", (2017).
5. A. Jain, R. Gonzalez Crespo and M. Khari. "*Smart Innovation of Web of Things*", (2020).
6. I. Torre and I. Celik. "*User-Adapted Web of Things for Accessibility*", (August 2015).
7. M. McCool and E. Reshetova. "*Distributed Security Risks and Opportunities in the W3C Web of Things*", (2018).
8. F. Parwej, N. Akhtar and Y. Perwej. "*An Empirical Analysis of Web of Things (WoT)*", (June 2019).
9. D. Guinard Vlad M. Trifa. "*Building the web of things*", (2017).
10. Y. Perwej et al. "*The Future of Internet of Things (IoT) and Its Empowering Technology*". IJESC, (March 2019).
11. Y. Zhou, S. Wei Wang, and K. Moessner. "*Search techniques for the web of things : A taxonomy and survey*", (2016).
12. R. Sardar and T. Anees. "*Web of Things:Security Challenges and Mechanisms*", (2017).
13. N. Kumar and S. Ahmad. "*Security Threats in Layered Architecture of Web of Things*", (2020).
14. N. Abughazaleh, R. bin Jabal and M. Btish. "*DoS Attacks in IoT Systems and Proposed Solutions*", (June 2020).
15. A. H.Mohammed and M. Mosa Jafer. "*Secure web of things based on a light-weight Algorithm*", (December 2019).
16. S. Gerdes et al. "*Delegated coap authentication and authorization framework (dcaf)*", IETF, (April 2016).
17. T. Kothmayr et al. "*DTLS based security and two-way authentication for the Internet of Things*", (2013).
18. K. Mishra and K. Yadav. "*Access Control In IoT Networks: Analysis And Open Challenges*", ICICC (2020).