IoT Security: Threats and Possible Solutions

Andrei-Robert CAZACU
Bucharest University of Economic Studies
cazacuandrei17@stud.ase.ro

Despite IoT being in the spotlight for several years now, Gartner still predicts a five time increase in number of devices from 2018 to 2028 [1], up to 1.9 billion units, which further intensifies the need for security among IoT nodes. Through the years, a significant amount of effort has been spent standardizing the communication and improving inter-operability among IoT devices. Even so, payload which is not properly secured, or impersonation of the node is still a reality, therefore, great attention still must be paid to the security model used. The main objectives of IoT security are to maintain confidentiality and integrity of the data, availability of services and authentication of the node.
Keywords: *Internet of Things, Security, IoT, Fog computing, Cloud computing, MQTT, CoAP*
DOI: 10.24818/issn14531305/26.2.2022.06

1 Introduction

There is an industry-wide trend of having remote monitoring capabilities and easy automation which has opened a lot of questions regarding the security of such solutions. Some of these might be as how the origin of node is verified, how to maintain confidentiality and integrity of the secrets stored inside, or as simple as how to secure the data transmitted. While these might be easily assured in a controlled environment, such as in the testing facility of the manufacturer, the real world poses greater risks. A user with technical knowledge and physical access can inspect the contents the memory, be it non-volatile, or volatile during execution, or sniff the payload for any useful information.

Seeing as IoT is an evolution of already existing technologies, deriving from Machine-to-Machine(M2M) services, some of the techniques used before can still be applied, albeit with the technical limitations imposed. M2M cover a wide range of applications such as smart homes, smart cities, smart grids, etc.

These limitations include limited processing power and small amount of memory, which greatly reduce the possibilities of using complex systems involving multiple digital signatures of asymmetric encryption. Also, some of these devices use battery for power, providing benefits to using a streamlined security model, even at the cost of reduced security. With an ever-growing volume of data and the desire to achieve a real time effect, securing the payload might take a backstep in the priorities of the implementer, choosing not to incur the performance cost.

With these limitations in mind, there is no wonder that many of these devices have little to no security in place besides a username and password combination. Malware like Mirai, which preys on the users not changing default credentials particularly [2], are particularly effective, having been used in some of the largest and most disruptive distributed denial of service attacks. Although, changing the password is not enough of a security measure, as shown by the Prowli malware, which is a traffic manipulation and crypto currency mining malware [3], that use brute force to crack the passwords.

This stressed the need for efficient means of securing IoT applications at a more thorough level.

Such efforts have spawned several lightweight application layer protocols such as CoAP [4], or means to encode data that further reduces the size of the payload such as CBOR [5]. Reducing the payload size helps by decreasing the amount of data that needs to be encrypted, effectively reducing the encryption cost.

With an ever-growing list of devices supported ranging from smart home appliances to

IoT nodes transmitting mission critical data, Cisco Inc. having forecasted a total of 14.7 billion devices by 2023 [6], implementing effective security can incur significant costs. As such, this paper aims to use industry standards.

Huge amounts of money are lost due to security breaches in IoT devices, be it directly or indirectly by making the service unavailable. This issue is also exacerbated by improper security measures of the end-user, such as not changing the default credentials in a home usage example, or by not enforcing proper physical access policies or insufficient network security measures in an enterprise usage.

The purpose of this study is to outline several pitfalls when securing an IoT system, along with comparisons of security measures needed based on the network topology. Also, multiple solutions are explored, weighing in the pros and cons of each. The hardware limitations are also factored in, outlining solutions which require less processing power or have a smaller memory footprint.

In the second chapter, literature review, an analysis of the existent publications will be offered, along with the contribution that this paper offers. In the third chapter, research methodology, the two most common IoT architectures will be defined, along with the security threats that pose significant risk of compromising the system. In the fourth chapter, findings, mitigations are provided, while also explaining the differences between the required measures depending on the IoT architecture used. Last of all, the fifth chapter, conclusions, a summary is provided along with possible enhancements to this paper.

2 Literature review

The large number of studies performed about IoT security shows that this issue is as pressing as ever, providing incrementally more secure solutions based on previous knowledge.

Authors of [7] split the security of the system based on three defined layers, perception, network and application, analyzing them independently. The study outlines the threats at the time of emergence for IoT systems, offering corresponding secure strategies for problems existing at that time.

This provided a baseline for future studies, such as [8] which enumerates security critical applications in IoT and outlines sources of security threats along with possible solutions emphasizing the benefits of using blockchain, Fog Computing, Edge Computing and Machine Learning. The proceeding [9] also uses the same three-layered architecture, with security principles which should be enforced at each layer.

In [10], the author outlines the security risks of using IoT devices and the tight coupling between our non-virtual life and smart devices which can directly influence our physical security.

The authors of [11] emphasize the impact that IoT failures can have, and how the security of such devices gained traction during the years, along with a state report at the time of writing.

In [12], the researchers outline a defense-in-depth approach by using blockchain and taking a deeper dive into how it can be used for securing IoT nodes. Authors of [13] also evaluate blockchain as a possible security piece to solver many IoT security solutions.

The proceeding [14] emphasizes the need for securing the sensitive information transmitted by the devices, composing a summary of information security related issues, and pointing out future research directions.

The call for a new paradigm of security is done by the authors of [15], claiming that the usual approach to security issues, typical of more classical systems and networks, does not grab all the aspects related to the way IoT works, being split into actuators and sensors that share their data.

Also, GlobalPlatform's blog post [16] defines device attestation and outlines its importance, and the security needs which it solves, such as node origin, impersonation, tampering, or bad software.

The whitepaper [17] also outlines the need for a way of identifying, characterizing, and authenticating IoT nodes, while also stating that the usual Internet protocols and services for security and authentication are not device oriented and are not suitable for the job.

The main contributions this paper aims to offer are to expose security threats at the level of the device, transport, or application, and to provide solutions while leveraging advancements made in the field. It also shows how different network topologies affect the security requirements, and outlines pro and cons of different solutions.

3 Research methodology

Since network topology is a defining factor when assessing security needs in IoT systems, the research will be split in two parts, a cloud connected sensor that represents a typical setup which connects directly to the internet, and a fog computing example, where a gateway is placed at the edge of the network which then relays only necessary information further in the cloud, placing the sensors in a private network.

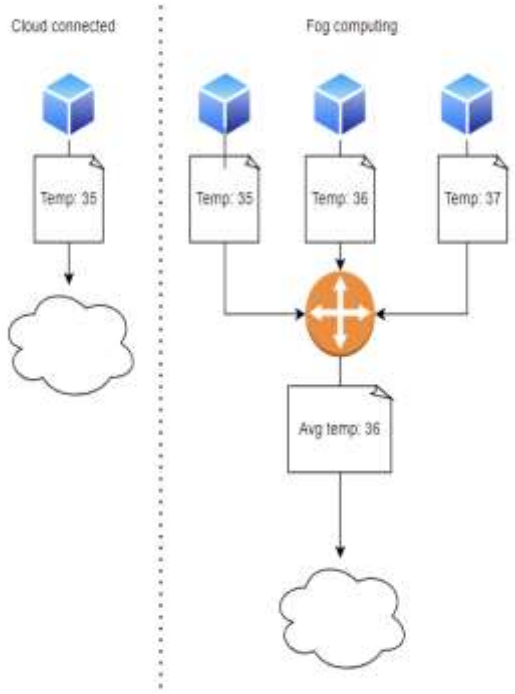

Fig. 1. Possible architectures

For clarity, the arguments will be structured as perception level security, network, and application, and the possible security threats will be first outlined and then treated based on the specific architecture.

At the perception level, threats can either originate from physical attacks, or via software. Major threats are:

- **Node Capturing:** IoT systems are comprised of several IoT nodes; with physical access, these nodes can be replaced with malicious ones, offering the attacker access to the system, compromising the security of the entire application
- **Code Injection Attack:** To be able to provide firmware upgradeability without having to return the equipment to the manufacturer or possess special tools, memory can be flashed with an over-the-air firmware update; if the image is compromised, the node can perform unintended functions and provide system access to the attacker
- **False Data Injection Attack:** Once a node is compromised, the attacker can use it to send bad data into the system which may lead to skewed results or system malfunction; the attacker can also flood the network, resulting in a DDoS attack
- **Side-Channel Attack:** An attack based on information gained from the implementation of the system rather than exploiting a vulnerability; can be used to gain knowledge about cryptographic algorithms used, of even access to sensitive data if exploiting the cache
- **Booting Attacks:** It is a particular example of a side-channel attack; the edge devices are more vulnerable during the booting process since not all security systems are still activated; this can be exploited by triggering a restart of the node
- **Sleep Deprivation Attack:** This affects the availability of the service, attackers attempting to drain the battery of the IoT nodes

At the network level, several threats have been identified:

- **Eavesdropping:** Most often IoT system are deployed in open environments, making them susceptible to eavesdropping; if the payload is not protected, access to sensitive data can be gained by the attacker

- **Interferences:** When placed in an open environment, the payload can be affected by interferences, which can be exploited by an attacker to affect the availability of the service and the integrity of the data
- **Access Attack:** This attack is similar to eavesdropping, but it is also working when using encrypted transmission protocols by gaining access to the network which nullifies the aforementioned encryption
- **Denial of Service Attack:** Be it distributed or from a single node, service availability can be affected by an attacker when issuing multiple rapid requests; can be either a transport or application-level attack
- **Routing attacks:** In such attacks, compromised IoT nodes might try to redirect the routing paths during transit

At the application level, several attacks can be performed:

- **Man-in-the-Middle Attack:** This attack has several forms depending on the protocol used, but essentially places an intermediate object between the transmitter and receiver in order to intercept the transmitted data;
- **SQL Injection Attack:** This involves the attacker embedding malicious SQL statements in the input in order to retrieve more data than otherwise would have been allowed to.

4 Findings

First of all, the devices used differ based on the network topology chosen.

For the cloud connected architecture, an IoT node was devised using an Espressif ESP32, a low-powered system on a chip microcontroller featuring Wi-Fi and Bluetooth connectivity, among a 32bit architecture and a dual core CPU with 520KB of RAM and 4MB of flash storage, and a DHT11 temperature and humidity sensor.

For the fog computing example, an IoT node and an IoT gateway was used. As a node, and ESP8266, a single core system on a chip microcontroller featuring Wi-Fi and manufactured by Espressif, was used, along a DHT11 temperature and humidity sensor. The sensor data is then transmitted via Wi-Fi to the gateway, a Raspberry Pi 4, a single board computer featuring 1GB of RAM and quad core Cortex-A72 CPU at 1.5GHz along with a 64bit architecture. The node is placed inside the protected network and only the gateway is web-facing, being able to transmit data the cloud.

In order to mitigate node capturing, proper physical access policies must be implemented. If the environment doesn't allow, e.g. open air, the device must be tamper-resistant. Also, the device must undergo an attestation process where the identity and its origin are verified.

In the cloud connected sensor example, the chosen node must be powerful enough to run complex cryptographic functions, while also offering a mean to securely store sensitive data such as secret keys or certificates. This can be achieved either by a dedicated secure element or some sort of runtime transparent encryption, with the ESP32 featuring the latter, storing the AES key in eFUSE.

At the core level, the attestation process must at least verify the origin of the node but can also establish secrets to be used for encrypting the payload. Depending on the security requirements, an attestation process that also establishes a secret key while also ensuring perfect forward secrecy by changing the keys each 24 hours can be implemented by using elliptic curve cryptography.

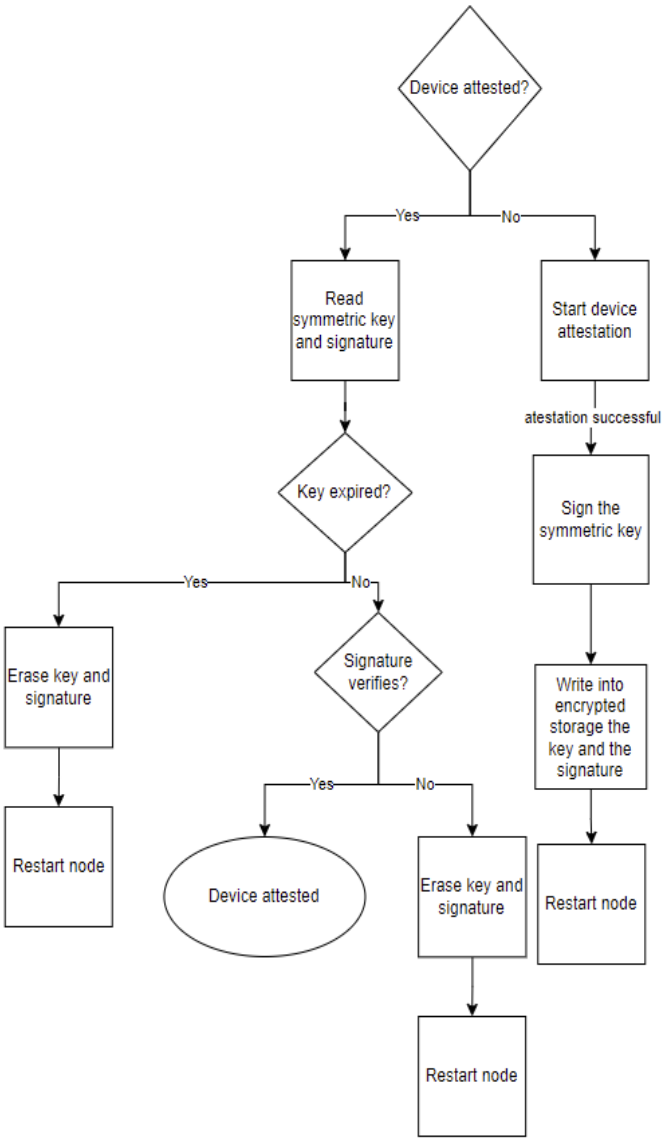

Fig. 2. Attestation flow

The device is provisioned with the certificate of the CA but also with its own key-pair. The device initiates the attestation process by sending its own certificate, and the server responds with his certificate. Each party must check the authenticity of the certificate using the CA certificate. If it succeeds, the device will then send the ECDH parameters in plain along with the signature. The server responds with his own parameters, his signature and test bytes to be used in order to check the success of the key exchange process. Ultimately, the client sends the encrypted bytes along with the used IV for verification.

In the fog computing topology, attestation can have several nuances: attestation of the node to the gateway and attestation of the gateway to the cloud. Depending on the processing power available, the same approach can be taken as in the case of the cloud connected sensor. If the node's hardware is a limiting factor, a master nodes approach can be taken, where a majority of such nodes must accept the newly introduced IoT device into network before granting access to the resources.

By guaranteeing node integrity and origin of the node, false data injection attacks are also prevented.

For code injection attacks, several approaches can be taken depending on the processing power available. Both at gateway and node level, when receiving an over-the-air(OTA) update, the image must be signed using the CA certificate in order to guarantee

authenticity. If the device doesn't support asymmetric cryptography, a message authentication code can also be used, with the drawback of having to store the secret key used to compute the message both on the node and the server. If none are viable solutions, OTA updates must be disabled.
For side-channel type attacks, one step would be to initialize all services at boot time in order not to offer the attacker any information as to which service is being used. These are also defense mechanisms for this type of attack built into the chips. If security of cryptographic functions warrants the cost, a tamper resistant device can be used to offload such operations, such as a Secure Element.
In the case of sleep deprivation attacks, a mechanism to detect to limit the amount of request can be put in place, and to also block requests with suspicious patterns. This will greatly limit the effect of such attacks on the IoT node.
The transport level security is greatly influenced by the network topology used.
While in the fog computing example, eavesdropping between the sensor and gateway is practically nullified by the network-level encryption, in the secure sensors demo the payload must be encrypted in order to ensure confidentiality.
Another notable aspect is the fact that only the cloud connected sensor topology can ensure end-to-end encryption, while the fog computing one still can't. Even if the data transmitted between the node and the gateway is encrypted, the gateway must perform decryption and encryption in order to send it in the cloud.
Depending on the architecture used, several application protocols can be used, some offering both UDP and TCP variants, either in plaintext or encrypted version.

Table 1. Application protocols suitable for IoT

Protocol	Plaintext	Encrypted	Network protocol
HTTP	Yes(TCP)	Yes(TLS)	IP, Bluetooth*
MQTT	Yes(TCP)	Yes(TLS/GATT)	IP, Bluetooth
CoAP	Yes(TCP/UDP)	Yes(TLS, DTLS, GATT)	IP, Bluetooth

As it can be seen in the table above, there are several choices for transport level protocol, be it in plaintext such as TCP/IP, or UDP/IP, or encrypted variants such as TLS/IP, DTLS/IP or GATT/Bluetooth, specifying that getting HTTP to work over Bluetooth requires a proxy with Bluetooth and IP stack in order to mediate the communication.
In the fog computing architecture, the level of security required dictates the protocols used. It has been demonstrated that CoAP over UDP greatly outperforms both HTTP and MQTT [18], which should be plenty of reason in order to achieve decreased power consumption and reduced computational needs but leaves the system vulnerable to the related access attack. This kind of attack is based on the adversary's ability to break into the protected sensor network.
This can be mitigated by having transport level encryption if computational power of the node permits. A lower cost can be incurred by using pre-shared key cipher suites which can be obtained in the attestation process as depicted above.
No matter the network topology used, interferences can occur, be it produced by an attacker or from natural means. Mitigation can be put in place by bundling in the payload a digital signature, message authentication code or a digest. The use of a digest to verify the integrity of the payload must be avoided in the

case of unencrypted communication since the message can be changed and the digest re-calculated.

Denial of service attacks can be mitigated by baking in mechanism for attack detection and automatic blocking of suspicious activities. One such mechanism might use Machine Learning to detect abnormal traffic patterns and flag them for blocking.

In order to mitigate routing attacks, strict control policies over the devices attached to the network have to be put in place. This not only applies to the IoT nodes, but also other devices connected to the same private network. In the case of the cloud connected sensors, the data is pushed directly into the cloud, generating no additional Intranet activity, while the fog computing example relies heavily on Intranet activity and seldom pushes data in the Internet.

At the application level, security is influenced by the protocol used.

Man-in-the-middle attacks can be prevented by using a protocol which supports transport level encryption, be it HTTPS, MQTTS, CoAP over TSL or DTLS. Besides that, standard security practices can be applied such as SQL injection mitigations, XSS prevention, etc.

Additional protection can be provided by also encrypting the application level data with either a conventional symmetric encryption algorithm such as AES, or using one of NIST's short listed lightweight cryptography cipher [19].

5 Conclusions

While this paper strived to offer a comprehensive overview of the industry-approved technologies available, it merely showed that in a segment heavily defined by fragmentation there is no right or wrong tool to use.

Firstly, the network topologies used in this paper were defined, along with a clear separation among device level, transport level and application level security threats while also describing them.

After this, the two-physical representation of the topologies were described, using an ESP32 system on a chip along with an DTH11 temperature sensor to represent the cloud connected sensor, and an ESP8266 system on a chip with the same temperature sensor that transmits data to the gateway, and Raspberry Pi 4 single board computer. The threats that were discussed previously were assessed and mitigation options were provided. Where multiple options were present, the pros and cons of each were discussed.

With all of these in mind, the theoretical basis needed to build up a reliable IoT security model has been offered. This paper can also be further enhanced by introducing Machine Learning derived mitigation measures or blockchain solutions to existing security threats.

References

[1] Modern Materials Handling, "Gartner's Hype Cycle: IoT in "trough" but transformational stage on the way," 9 September 2020. [Online]. Available: https://www.mmh.com/article/gartners_hype_cycle_iot_in_trough_but_transformational_stage_on_the_way.

[2] Cloudflare, "What is the Mirai Botnet," [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/.

[3] Logpoint, "Prowli malware detection," 29 June 2018. [Online]. Available: https://www.logpoint.com/en/blog/prowli-malware-detection/.

[4] IETF, "RFC 7252 - The Constrained Application Protocol (CoAP)," [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7252.

[5] IETF, "RFC 7049 - Concise Binary Object Representation (CBOR)," [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7049.

[6] Cisco Inc., "Cisco Annual Internet Report (2018–2023) White Paper," 9 March 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[7] Q. Gou, L. Yan, Y. Liu and Y. Li, "System, Construction and Strategies in IoT Security," Beijing, China, 2013.
[8] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," 2019.
[9] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," London, UK, 2015.
[10] Y. H. Hwang, "IoT Security & Privacy: Threats and Challenges," in IoTPTS '15: Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security, 2015.
[11] M. b. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," Computer Networks, pp. 283-294, 15 January 2019.
[12] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," Internet of Things, pp. 1-13, September 2018.
[13] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, pp. 395-411, May 2018.
[14] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," Matsue, Japan, 2014.
[15] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou and A. Bouabdallah, "A Systemic Approach for IoT Security," Cambridge, MA, USA, 2013.
[16] J. O'Donoghue, "What is device attestation?," [Online]. Available: https://globalplatform.org/insight-series-what-is-device-attestation/.
[17] ARM, "Entity Attestation Token White Paper," 2020. [Online]. Available: https://www.psacertified.org/app/uploads/2020/02/PSA_Certified_Entity_Attestation_Overview_Whitepaper.pdf.
[18] R. M. Z. L. Jaime Jiménez, "Evaluating the Performance of CoAP, MQTT, and HTTP in Vehicular Scenarios," [Online]. Available: https://datatracker.ietf.org/meeting/103/materials/slides-103-maprg-evaluating-the-performance-of-coap-mqtt-and-http-in-vehicular-scenarios-jaime-jimenez-00.
[19] NIST, "Lightweight Cryptography," [Online]. Available: https://csrc.nist.gov/Projects/lightweight-cryptography.
[20] Connectivity Standards Alliance, "matter," [Online]. Available: https://csa-iot.org/all-solutions/matter/.

Andrei CAZACU has graduated the Bucharest University of Economic Studies in 2020 and is currently activating in the professional field as a software developer.