Politecnico di Torino
Dipartimento di Automatica e Informatica

PhD in Computer and Control Engineering
XXXVI cycle

Fondazione CRT

Supervisors
Fulvio Corno
Antonio Lioy

Supporting Developers in the Cybersecurity of IoT Systems

PhD Candidate: *Luca Mannella*

1. Context

Internet of Things (IoT) systems are very widespread but often not properly protected. If compromised, they can create serious issues and even cyber-physical attacks.

2. Goal

To simplify the development of secure and reliable IoT solutions. Primarily, the focus is on developers with limited experience in *IoT* and/or *security* fields.

3. Security guidelines for cloud-IoT systems

Paper 1

Through a survey, we discovered that *novice IoT developers* do not pay too much attention to security (at least in early development stages). Hence, I analyzed two major cloud-IoT platforms (AWS and Azure) focusing on the security features of their IoT components. We observed that they are quite able to compensate the developers' shortcoming (if correctly configured). Thus, we proposed a set of guidelines to enhance security during the creation of a cloud-IoT solution [1].

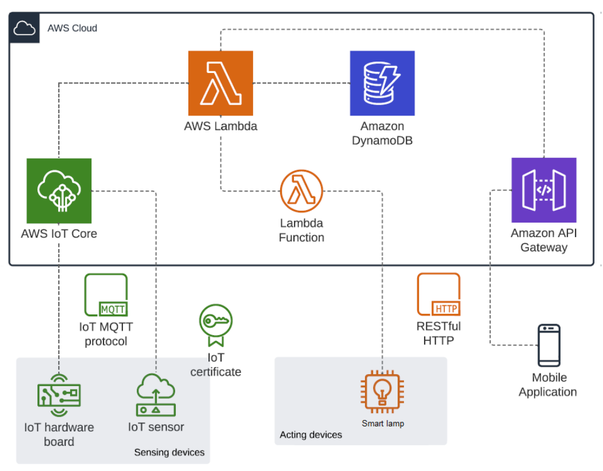

Table 1 Summary of the proposed guidelines.

ID	Guideline	Description
GL1	Use a threat model	Design new applications starting from security models
GL2	Protect data in transit	Protect every data flow with encryption mechanisms.
GL3	Configure encryption	Ensure all encryption mechanisms are properly configured; using TLS, configure TLS $\geq$ 1.2, and select a *recommended* cipher suite.
GL4	Use platforms' SDKs	Always use the platforms' SDKs to connect IoT devices to the cloud.
GL5	Use support services	Always use platform's support services to manage the devices.
GL6	Verify code security	Always use compiler features and code checkers to avoid insecure library functions or language constructs.
GL7	Improve authentication	Always use 2FA and/or complex passwords (e.g., at least two words, including at least three special characters).
GL8	Secondary accounts	Always create and use secondary accounts
GL9	Use POLP	Assign to each secondary account the fewest possible privileges.
GL10	Do not share accounts	Developers must have their own account.
GL11	Protect data at rest	Always ensure that data protection at rest is enabled.
GL12	Use DiD strategies	Add more protection's layer using Defense in Depth strategies.
GL13	Hash passwords	Salt and hash passwords during user registration or authentication.
GL14	Have backups	Always enable periodic database backups (even in different regions).

4. A threat model for extensible smart home gateways

Paper 2

Smart home devices are often affected by vulnerabilities (that can impact even other house's devices). When IoT objects are managed by a smart home gateway (SHG), there is a single point of failure. Moreover, the risk of a bugged gateway raises when it can be extended by third-parties plug-ins. Hence, we proposed a threat model [2] to help developers during the creation (or the evaluation) of plug-ins. We demonstrated the presented threats developing a set of proofs of concept for a widespread python-based open-source SHG: Home Assistant.

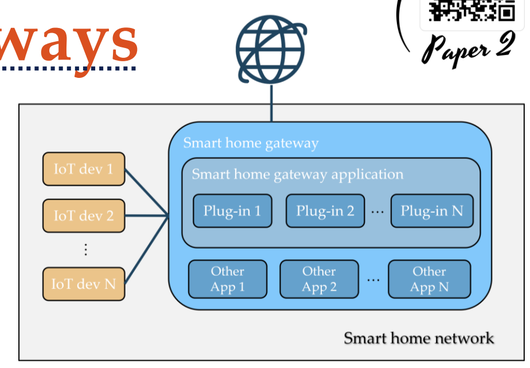

Table 2 Proposed threat model.

Category	ID	Description
Confidentiality	T1	a plug-in could *access and use* private data of other attack targets (i.e., data outside its scope)
	T2	a plug-in could *access and spread* private data of other attack targets (i.e., data outside its scope)
Integrity	T3	a plug-in could *alter the state* of smart home *devices* outside its scope
	T4	a plug-in could *alter private data* of other *attack targets* outside its scope
Availability	T5	a plug-in could *delay* the regular functionality of an attack target
	T6	a plug-in could *alter* one of the regular functionalities of an attack target
	T7	a plug-in could *alter* the regular functionality of an attack target, preventing the smart home users from using it
	T8	a plug-in could physically *damage* an attack target
Authentication	T9	a plug-in could interact with an attack target, pretending to be a different entity
Authorization	T10	a plug-in could access an authorization level higher than expected
Non-Repudiation	T11	a plug-in could anonymously communicate with an attack target

5. Current Developments

How to adopt the Manufacturer Usage Description (MUD) standard (RFC-8520) to support developers in plug-in creation.

6. References

1. "Helping Novice Developers Harness Security Issues in Cloud-IoT Systems" — Corno, F.; De Russis, L.; Mannella, L. — Journal of Reliable Intelligent Environments (Springer); Issue 3/2022 — pp. 261-283
2. "A Threat Model for Extensible Smart Home Gateways" — Corno, F.; Mannella, L. — SpliTech 2022: 7th International Conference on Smart and Sustainable Technologies (IEEE) — Split / Bol, Croatia, July 5-8, 2022 — pp. 1-6.