Poster: "*Belt and suspenders*" or "*just red tape*"?: Investigating Early Artifacts and User Perceptions of IoT App Security Certification

Prianka Mandal, Amit Seal Ami, Victor Olaiya, Sayyed Hadi Razmjo, Adwait Nadkarni
William & Mary
{pmandal, aami, voolaiya, srazmjo, apnadkarni}@wm.edu

I. PUBLICATION INFORMATION

This work will be published in the 33rd USENIX Security Symposium 2024 [6].

II. EXTENDED ABSTRACT

As IoT security regulations and standards emerge, the industry has begun adopting the traditional enforcement model for software compliance to the IoT domain, wherein Commercially Licensed Evaluation Facilities (CLEFs) certify vendor products on behalf of regulators (and in turn consumers). Since IoT standards are in their formative stages, we investigate a simple but timely question: *does the traditional model work for IoT security, and more importantly, does it work as well as consumers expect it to?* This paper investigates the initial artifacts resultant from IoT compliance certification, and user perceptions of compliance, in the context of certified mobile-IoT apps, i.e., critical companion and automation apps that expose an important IoT attack surface, with a focus on three key research questions: (1) are certified IoT products vulnerable?, (2) are vulnerable-but-certified products non-compliant?, and finally, (3) how do consumers perceive compliance enforcement?.

Mobile-IoT App Analysis: To answer our first research question, we focused on analyzing *mobile-IoT apps*, i.e., companion apps and third-party automation services help users control devices and manifest an important attack surface of the IoT system. We analyzed 11 certified mobile-IoT apps from IOXT [5], focusing on vulnerabilities resulting from *cryptographic API misuse* that critically impact on the secrecy/integrity of IoT data and are highly relevant to mobile apps. We focused on a systematic manual analysis rather than using automated tools (e.g.,CogniCrypt, MobSF, and CryptoGuard) to find common vulnerabilities as our goal is to uncover gaps in compliance enforcement, and not to measure the state of crypto-misuse in general. Our analysis of crypto-API misuse finds 35 serious vulnerabilities in 9/11 apps, including attempts to *evade compliance and security checks* (See 1 for example).

```
//The string operations below result in: "AES/" + "E" +
     "C" + "B" + "/NoPadding" = "AES/ECB/NoPadding".
this.ALGO= "AES/" + ((char) ("AES/GCM/NoPadding".charAt(4)
     - 2)) + "AES/GCM/NoPadding".charAt(5) + ((char)
     ("AES/GCM/NoPadding".charAt(6) - 11)) + "/NoPadding";
Cipher cipher = Cipher.getInstance(this.ALGO);
```

Listing 1: A complex instantiation of `AES` in `ECB` mode in a mobile-IoT app, made to look like the `GCM` mode instead.

After finding those vulnerabilities, we performed additional analysis to understand their implications, examining the surrounding code semantics to understand how vulnerable code is used. Moreover, We compare the state of certified vs non-certified mobile-IoT apps to answer a simple question: *are certified mobile-IoT apps more or less vulnerable than a comparable set of non-certified apps?*. In addition, we target mobile security issues explicitly outlined in standards, namely requesting more permissions than necessary, and leaking private data. The findings are highlighted as follows:

- Mobile-IoT apps can evade compliance checks by disguising vulnerable code as compliant, which indicates a serious challenge for CLEFs, and a pressing need to perform a *hostile review* of products ($\mathcal{F}_1$).
- Some certified mobile-IoT apps use vulnerable encryption when transmitting/receiving sensitive audio/video data to/from devices such as cameras ($\mathcal{F}_2$).
- Some certified mobile-IoT apps override `TrustManagers` and `HostnameVerifiers` in ways that make critical communication for user authentication and account management vulnerable to MiTM attacks ($\mathcal{F}_3$)
- The latest versions of vulnerable certified mobile-IoT apps are generally similarly or more vulnerable than the (older) certified versions ($\mathcal{F}_4$).
- Three automated tools, CogniCrypt, MobSF, and CryptoGuard, do not detect several of the 35 critical vulnerabilities discovered using manual reverse engineering, i.e., 33/35, 28/35, and 15/35 respectively, despite generating 89, 137, and 546 alarms, respectively ($\mathcal{F}_5$).
- Our equivalent set of certified and non-certified mobile-IoT apps are similarly vulnerable in terms of the crypto-API misuse cases we analyzed for ($\mathcal{F}_6$).
- All the certified mobile-IoT apps (11/11) request at least one dangerous permission that is not justified in the app

description or privacy policy ($\mathcal{F}_7$).
- Both certified and non-certified mobile-IoT apps from our set leak privacy-sensitive data such as location, the device ID, and sometimes the user-provided password, to the logs, but not to external storage ($\mathcal{F}_8$).

Security Compliance Analysis: We systematically evaluate 5 popular IoT security standards, namely: (1) IOXT, (2) OWASP Mobile Application Security Verification Standard (MASVS) [4], (3) IoT Security Foundation (IOTSF) standard [2], (4) IoT Alliance Australia (IOTAA) security guidelines [1], and finally, (5) NIST's Core IoT Cybersecurity Capabilities Baseline (NISTIR 8259A) [3]. To analyze compliance with respect to each standard, we systematically transformed the relevant parts of the standard into specific *criteria*, and then compared these criteria with the vulnerabilities we found in certified mobile-IoT. This analysis is composed of two aspects: determining *what* criteria apply to the vulnerabilities, and using additional information (e.g., test cases) to determine *how* they apply. The findings are as follows:

- Certain standards criteria are overly broad, making them appear comprehensive. However, a literal interpretation of the same may help developers claim vulnerable code as compliant ($\mathcal{F}_9$).
- Test cases accompanying criteria contain ambiguous phrases (e.g., "excessive permissions"), allowing significant discretion to the tester, preventing an unequivocal determination of compliance ($\mathcal{F}_{10}$).
- IOXT makes certain precise criteria discretionary for developers to comply with, leaving developers with the flexibility of choosing what communication or data to protect, which may result in vulnerable apps that developers may contest are compliant with the standard ($\mathcal{F}_{11}$).

User Perceptions and Expectations: We conducted a survey with 173 IoT users to gauge their awareness of IoT security compliance regulations, expectations from certified products and stakeholders, as well as perceptions regarding vulnerabilities, compromises, and accountability. Our survey consists of 39 questions with a mix of 15 open-ended and 24 close-ended questions, where we asked participants about their experience with mobile-IoT apps, their familiarity with oT security compliance standards, their xpectations from certified apps and who do they find responsible for enforcing correct compliance. The findings from our survey response are:

- Users are generally not informed of IoT compliance standards, and often unaware of the certified (status of the) mobile-IoT apps they use ($\mathcal{F}_{12}$).
- While a significant number of users are likely to check the certification status of the mobile-IoT apps they use, mainly for additional assurance, an equal proportion believe brand reputation and popularity to be more valuable ($\mathcal{F}_{13}$).
- Users overwhelmingly put their trust in certification, assuming that (1) certified apps are more secure (i.e., less prone to vulnerabilities), (2) their developers spend more effort on security, and (3) they can be trusted to handle security/privacy sensitive information ($\mathcal{F}_{14}$).
- Users hold CLEFs, developers, and standard organizations as almost equally responsible, and are able to clearly define what role each party plays in correctly enforcing security compliance standards ($\mathcal{F}_{15}$).
- Users rate a set of vulnerabilities found in our analysis to be both severe and likely ($\mathcal{F}_{16}$).
- Users generally hold developers as the most liable in the event of vulnerabilities and security breaches in certified mobile-IoT apps, since they develop the vulnerable code. CLEFs are considered the second most liable as vulnerable-but-certified apps represent a dereliction of duty, followed by standards bodies who are blamed for weak standards or enforcement ($\mathcal{F}_{17}$).

Takeaways: Given our findings ($\mathcal{F}_1 \rightarrow \mathcal{F}_{17}$), it seems that the traditional security certification model does not work for IoT, and as consumers expect it to. That is, while consumers have high expectations and place a significant degree of trust in the certification process, we find that certified mobile-IoT apps are generally vulnerable, with vulnerabilities that seriously impact IoT security, and no better than non-certified apps regardless of what consumers want to believe. Therefore, the traditional security certification model for IoT needs reforming through effective checks and balances, e.g., developing tools that evaluate the CLEFs' effectiveness at detecting vulnerabilities. Moreover, given the high degree of trust put by users in certified apps and the certification process, mechanisms must be built into the certification model to deter, prevent, and detect evasive behavior of developers. Further, effective, robust vulnerability discovery tools are needed as there is a lack of tools for the security-critical task of compliance enforcement. Finally, there should be dedicated consumer education programs so that users are informed about their rights and IoT product security certifications if things break.

To avoid a future where IoT compliance enforcement is treated as a liability shield, we seek to start a conversation in the security community, between researchers, practitioners, and policymakers, on how to *transition from the "just red tape" status quo to a practical "belt and suspenders" future.*

III. Acknowledgement

The authors have been supported in part by the NSF-2237012 grant, and a COVA CCI Dissertation Fellowship.

References

[1] IoTAA Security Guideline V1.2 November 2017. https://www.iot.org.au/wp/wp-content/uploads/2016/12/IoTAA-Security-Guideline-V1.2.pdf.

[2] IoTSF IoT Security Assurance Framework Release 3.0 Nov-2021. https://www.iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf.

[3] NISTIR 8259A IoT Device Cybersecurity Capability Core Baseline. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259A.pdf.

[4] OWASP Mobile Application Security Verification Standard v1.4.2 January 2022. https://mas.owasp.org/MASVS/.

[5] ioXt Alliance Members. ioxt: The global standard for iot security. https://www.ioxtalliance.org/, 2021.

[6] Prianka Mandal, Amit Seal Ami, Victor Olaiya, Sayyed Hadi Razmjo, and Adwait Nadkarni. "Belt and suspenders" or "just red tape"?: Investigating Early Artifacts and User Perceptions of IoT App Security Certification. In *33rd USENIX Security Symposium (USENIX Security)*, 2024.