Feedback on Public Consultation on Security Requirements for Residential Gateways

Author:

Eric Seow

Principal Technical Marketing

IFAP DSS SMD AP ESS TM

Contents

Summary of Major Points

With the increase in the cybersecurity attacks, it is no longer sufficient to look at security purely from a software perspective. This is because of the ever-increasing methods of cyber-attacks being discovered and openly shared in hacking conferences and open publications.

While majority of the cyber-attacks are being established on software vulnerabilities (e.g. HeartBleed, StageFright etc) and communication protocols (e.g. KRACK, BlueBourne, SweynTooth etc), there is an emerging trend of cyber-attacks that uses software to perform physical hardware attack without being in possession of the physical device (e.g. RowHammer, RAMBleed, ClckScrew etc). This is in addition to the occasional unrealities on hardware platforms such as SPECTRE and MELTDOWN that very often has huge devastating impact on the IoT applications.

The term "Defense-in-Depth" is very often used (but very often not implement enough) by Cybersecurity experts. "Defense-in-Depth" very well interpreted and understood in the software domain with various layering of software design and applying multiple layers of communication security such as WEP and TLS.

Unfortunately, "Defense-in-Depth" is usually, or conveniently, ignored or omitted in the hardware context. The MCUs used in majority of the IoT devices (and in this case the residential gateway) are often "off-the-shelf" MCUs, which means that an adversary is also able to acquire these MCUs to study and perform various forms of reverse-engineering. This can be seen from the various publications of hardware reverse-engineering attacks published in Blackhat and DEFCON conferences.

This feedback paper aims to address the "Defense-in-Depth" from a hardware perspective and how to harden various features of the residential gateway with a tamper-resistant hardware trust anchor.

Statement of Interest

To adopt for a hardware "Defense-in-Depth" approach, it is necessary to incorporate a hardware tamper-resistant trust anchor for the storage of cryptographic keys, certificates and wifi Login credentials that is protected against extraction and theft.

A hardware tamper-resistant trust anchor can added to the residential gateway together with the MCU. The MCU may also be implemented some forms of software sandboxing strategies such as Trusted Execution Environment (TEE). The hardware trust anchor is able to complement such a design as the TEE typically provides only logical protection with software separation.

As most of the hardware tamper-resistant trust anchors in the market are designed to meet the stringent security requirements of banking standards and/or ePassports applications, they are normally certified to Common Criteria EAL 5+ with the protection profile of "PP0084".

This will achieve the principal of "hardware separation" in the event that the MCU is compromised via malware or malicious software, the keys and credentials are still well protected in the hardware trust anchors.

A good example will be the example drawn from "TR64 [1]" and "Singapore cybersecurity guide [2]" which advocates the use of secure element (such as TPM or its equivalent) as a hardware trust anchors.

Comments on the Specifications

Comments on Section 4.1 – Login Credentials Management

i. Login credentials that are stored on the MCU can be extracted and stolen when the device is compromised by malware or malicious software.
ii. Recommendation:
    a. Login credentials (such as username, passwords etc) should be stored securely in hardware tamper-resistant trust anchors.
    b. The Trust anchors should enforce security policies before allowing the MCU to access these credentials.

Comments on Section 4.2.3 – Authentication Handling (reference to point b)

i. As with all cryptography operations, encryption of authentication credentials require cryptographic keys (we can call it key encryption keys as an example).
ii. The paramount question here will be how can we secure the key encryption key against theft and extraction?
iii. One recommendation will be to securely store these credentials in hardware trust anchors.
iv. Bind such credentials with security policies enforced by the hardware trust anchors that govern their access only when certain security conditions are fulfilled.

Comments on Section 4.2.4 – Credentials Handling

i. Refer to comments on section 4.2.3.
ii. Credentials need to be securely stored in hardware trust anchors governed by security policies for access.

Comments on Section 4.3 – Firmware Updates

i. Firmware updates are key to patching security loopholes and managing the risk and impact of software vulnerabilities.
ii. With greater emphasis (and greater occurrences of firmware update), adversaries will make use of this opportunity to proliferate malicious firmware versions and even downgrade to older version that contains exploitable vulnerabilities.
iii. Therefore, it is of great importance to provide guidelines for a secure firmware update.
iv. Recommendation:
    a. Verify digitally signed firmware and this has been clearly spelt out in the specs.
    b. Prevent installation of older versions of firmware that contain vulnerabilities.
    c. If an attack can replace the key used for verifying the firmware signature, then any malicious firmware can be installed because the verification will always pass.
    d. Therefore, the FW verification key must be secure in a hardware trust anchor.
    e. Verification of the FW digital signature can be done in the hardware trust anchor with the verification key stored inside.

Comments on Section 4.5 – Data Protection

i. It is generally a good practice to protect data-at-rest with standardized encryption algorithms.
ii. However, cryptographic encryption/decryption requires cryptographic keys, which themselves cannot encrypt themselves.
iii. So a good complementary recommendation will be to securely store then encryption keys in hardware tamper-resistant trust anchors so that these keys are protected against extraction and theft.
iv. The encryption keys is only exposed to the Host MCU once the required security policies are fulfilled.

Comment to Queston1:

i. Randomized and unique factory pre-loaded login credentials is currently adopted by most commercial residential gateways.
ii. Most residential gateways have the ID usually “admin” and a random password of a specific strength printed on a label.
iii. These labels may fade way over time or printed with font that is too small to read which result in device not useable after factory reset.

iv. Alternative approach is to have the RG self-generate a unique and randomize picture code every time after a factory reset is perform and require the user to perform a specific task to be authenticated
v. Upon successful authentication, require user to enter new password word specify in 4.1.2 before enabling only basic features enough of common usage.
vi. All other usage should be done manually.
vii. Factory reset process should require physical access to the Residential Gateway. E.g holding the reset button for 10 sec.
viii. Regarding, minimum key strength, In section 4.1.2 (a) why at least 3 out of the following 4 complexity rules and not all the complexity rules stated.

Comment to Question 2:

i. Telnet should not even be in the system.
ii. Before user change the Residential Gateway admin password WAN connection should be turn off too.

Comment to Question 3:

i. For most common device management usage SSH is not needed.
ii. Unless in-depth of highly customize configuration is needed.
iii. This could then be enable through advance setting.

Comment to Question 4:

i. Option should be provided for user during setup for auto firmware updates to be enable or disable.

Comment to Question 5:

i. If auto firmware updates features is enable the Residential Gateway could be set to poll site for updates at a specific time settable by the user.
ii. During registration, phase of the RG good to give user an option to register their e-mail for updates notification.
iii. It is not practical to impost a fix timeline for patches to be applied

Comment to Question 6:

i. Displaying warning on usage of weaker encryption algorithm is a good approach and should not be allow disabling it.

Comment to Question 7:

i. Agree

Comment to Question 8:

i. Good to use a standard that is adopted/recognize by our Singapore Gov.
ii. As this standard is periodically reviewed and updated.
iii. Using international standard alone could at time may not be practical for Singapore usage.

Comment to Question 9:

i. Have from the very start to perform device switching. E.g upgrading from an older model to a newer model.
ii. There may not be only one Residential Gateway in a residential home.
iii. Need also specification for Mesh/multiple Residential Gateway with a single WAN output.

Comment to Question 10:

i. No comment

Comment to Question 11:

i. No comment

Conclusion

With the feedback from the above sections, we hope to stress again on the importance of adopting a hardware tamper-resistant trust anchor to complement the off-the-shelf host MCU used in the residential gateway.

Cryptographic keys, certificates, user credentials should be securely stored on the trust anchors with access controls governed by security policies and enforce by the hardware trust anchor.

Having a hardware trust anchor also provides a hardware separation and provides for a more robust and longer life-span of the residential gateway against cyber-attacks and theft of credentials.

References

[1] TR64 : 2018 - "Guidelines for IoT Security for Smart Nation"

[2] 'IMDA IoT Cyber Security Guide", Version 1, Jan 2019.