IOTOPS
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

B. Moran
Arm Limited

July 12, 2021

A summary of security-enabling technologies for IoT devices
draft-moran-iot-nets-00

Abstract

The IETF regularly develops new technologies. Sometimes there are several standards that can be combined to become vastly more than the sum of their parts. Right now, there are six technologies either recently adopted or poised for adoption that create such a cluster. Combining secure onboarding, remote attestation, secure update, software bill-of-materials/expected attestation, automated network policy enforcement, and trusted execution environment provisioning, devices can be defended from many threats. This is an opportunity for an inflection point for more secure and trustworthy devices. Simultaneous adoption of two or more of these six standards could create the foundation of computing devices that are worth trusting.

Table of Contents

1. Introduction

IoT devices (unattended devices with network connections) are often considered a weak point in networks and have often been used by malicious parties to extract information, serve as relays, or mount attacks. Appropriate use of security technologies can mitigate this trend and enable users allow for security polices that do not have to be overly protective of IoT systems and enable them to add the full potential of value they were designed to add.

This draft addresses six trustworthiness problems in IoT devices and proposes solutions to them with six technologies. The problems are:

1. What software is my device running?

2. How should my device connect to a network?

3. With which systems should my device communicate?

4. What is the provenance of my device's software?

5. Who is authorised to initiate a software update and under what circumstances?

6. How should my device update its trusted software?

Each of these questions is answered by recently developed or developing standards.

2. Barriers to IoT Adoption

IoT adoption is generally presented as a platform problem or a data acquisition and analysis problem. The result is a proliferation of communication formats, radio standards, network technologies, operating systems, data gathering schemes, etc. Despite this effort, IoT is not growing at the projected rates.

IoT is not simply a combination of a device platform and a data-gathering platform. In the life-cycle of devices, they must be commissioned and onboarded. When a flaw is discovered, they must be updated to restore trustworthiness and there must be evidence that they are effectively trustworthy (e.g. running the intended/expected software). Acknowledging the chance of security breaches, network infrastructure must be configured to allow access to necessary services and restrict access to everything else.

Commissioning, onboarding, attestation, update, and access control are complex core technologies that are difficult to implement well. This can be seen with the plethora of poorly implemented IoT devices that have been reported in the news whenever a defect is found.

IoT adoption is hampered by a lack of core technologies surrounding the development of trustworthy devices and device trustworthiness. These core technologies do not present obvious revenue streams and they require cooperation between many vendors for them to succeed, which may explain the low rate of innovation in this space.

To reduce this barrier to entry, the IETF has been investing in these core technologies.

3. Foundations of Trustworthy IoT

IoT devices can bring a lot of value to businesses and individuals, but they are also difficult to manage because of their diversity, difficulty in auditing, maintenance, onboarding practices, and lack of visibility about device security posture and device software.

Initiatives such as PSA Certified focus on device level security principles and encourage the use of a hardware Root of Trust (RoT) that provides a source of confidentiality and integrity for IoT systems. The security principles led security requirements of PSA Certified Level 1 cover topics such as trusted boot, validating updates, attestation and secure communications. Complementary to this, IETF provides standards that can be used to create secure

networks; this memo focuses on six standards that can beneficially be used together at the network level.

Building trustworthy IoT is about more than just building devices conforming to best-practice security. Users, Owners, Operators, and Vendors must be able to respond when a compromise occurs. Responding to compromised IoT consists of three key points:

1. Detecting a compromise

2. Halting malicious activity

3. Remedying vulnerabilities and flawed software.

Once a compromise has been detected, the affected device needs to be quarantined from the network, then security patches must be applied.

3.1. Detecting a Compromise

There are two broadly applicable ways to remotely detect a compromised IoT device:

1. Detect anomalous software on the device.

2. Detect anomalous network traffic from the device.

Detecting anomalous software on the device requires remote attestation of software measurements; the report of what software the device is running must be trustworthy even if the software is not. However, attestation is an incomplete solution if the recipient of attestation evidence does not know what to expect. Hence, trustworthy and authortative sources with understanding of what is to be expected are required. Furthermore, automated systems for delivering these expected values must be very secure or else they will become targets for threat actors as well.

Detecting anomalous traffic from the device requires a baseline of expected traffic; otherwise, network infrastructure cannot know what traffic is legitimate and what is not. This expected traffic information needs to be closely associated with each individual device, since network traffic patterns may shift from device to device or version to version. These trustworthy and authoritative sources of patterns must also be protected: a compromised device could report an incorrect expected network traffic pattern, or a threat actor could modify an expected network traffic pattern.

3.2. Halting Malicious Activity

Halting malicious activity is done by network infrastructure. A Network Access Control (NAC) system, such as a router, gateway, firewall, or L3 managed switch, can apply a network access policy on a per-device basis. The NAC system uses a policy that is provided to it in advance in order to determine the access requirements of each connected autonomous device. Assuming that these policies are constructed according to the principle of least privilege, This allows the NAC system to drop any communication that does not match the defined policies, effectively eliminating the use of IoT devices as relays, proxies, or mechanism to pivot in a network. It may even prevent compromises before they occur because inbound traffic to IoT devices that does not conform to policy can be discarded.

For shared media, such as radio protocols, intra-LAN policies cannot be preemptively effectively enforced, but they can be monitored and enforced after violation, for example by removing network access rights. Per-device Internet-to-LAN policies and LAN-to-Internet policies can still be applied as normal.

3.3. Remedying Vulnerabilities

Remedying vulnerabilities requires a remote update system. Where there are secure components that are independently updatable, additional considerations are required. In both cases, the new software must be signed, but that alone is insufficient: new software must be authenticated against a known, authorized party. It must also come with a statement of provenance: a software bill of materials or SBoM. This statement must describe all the components of the software along with defining the authorship of the software, which may be separate from the authority to install that software on a given device.

4. Baseline Requirements for Secure Networks

To establish a trustworthy IoT network, devices MUST be able to prove:

1. What software they are running and, by extension:
    1. The provenance of the software.
    2. (Optionally) that it has been checked for common malware, backdoors, etc.
2. Who they will connect to or exchange data with so that anomalies can be registered.

To install and maintain IoT devices, authorized entities MUST be able to:

1. Connect a device to a secure network.

2. Initiate an update of a device.

3. (Optionally) Add or remove authorized entities from the device.

4. (Optionally) Deploy and remove protected assets to and from the device.

Each of these requirements stops a particular avenue of attack against device, networks, or data collection systems.

5. IoT Technologies for Secure Networks

Assembling the foundations of trustworthy IoT and the baseline requirements for secure networks, the result is a set of requirements, described here with enabling standards:

1. To deploy new keys into a device and connect it to a network, devices SHOULD support a secure onboarding protocol such as FIDO Device Onboarding [FDO] or LwM2M Bootstrap ([LwM2M]).

2. To enable devices to report their current software version and related data securely, devices SHOULD support a support a mechanism of performing attestation measurements in a trustworthy way and a Remote Attestation protocol, such as [I-D.ietf-rats-eat].

3. To enable devices to be updated securely in the field, they SHOULD support a remote update protocol such as [I-D.ietf-suit-manifest].

4. To prove the provenance of a firmware update, update manifests SHOULD include (directly, or by secure reference) a Software Identifier or Software Bill of Materials, such as [I-D.ietf-sacm-coswid].

5. To enable a Relying Party of the Remote Attestation to correctly evaluate the Attestation Report, the SBoM (such as [I-D.ietf-sacm-coswid]) SHOULD contain expected values for the Attestation Report.

6. To ensure that network infrastructure is configured discern the difference between authentic traffic and anomalous traffic, network infrastructure SHOULD contain a [RFC8520] Manufacturer

Usage Description (MUD) Controller which accepts MUD files in order to automatically program rules into the network infrastructure.

7. In order for network infrastructure to be configured in advance of any changes to devices, MUD files SHOULD be transported (directly or by secure reference) within update manifests.

8. To enable rapid response to evolving threats, the MUD controller SHOULD also support dynamic update of MUD files.

9. Network infrastructure SHOULD apply risk management policy to devices that attest non-compliant configuration. For example, a device with out-of-date firmware may only be permitted to access the update system.

5.1. Trust Relationships in Secure IoT Networks

[FDO] and [LwM2M] enable the installation of trust anchors in IoT devices. These enable the services to ascertain that the devices are not counterfeit. They also enable the devices to trust that the services are not on-path attackers.

The combination of SUIT, CoSWID and RATS Attestation secures these trust relationships further. A device operator receives a SUIT manifest, that contains a CoSWID. They apply the SUIT manifest to a device. The newly updated device then attests its software version (one or more digests) to the device operator's infrastructure. The device operator can then automatically compare the attestation evidence to the CoSWID.

The device operator can trust that expected values are correct because they are signed by the software author. The device operator can trust that the attestation report is correct because it is signed by the verifier and, finally, the device operator can trust the device because its attestation evidence content matches its CoSWID.

To extend this relationship to Trusted Applications (TAs) as well, devices that support TAs can also implement [I-D.ietf-teep-architecture].

Adding MUD to the combination above cements the established trust with enforcement. The network operator also receives the SUIT manifest for the device. The manifest contains a MUD file in addition to the above. The device does not need to report a MUD URI as described in [RFC8520]-which stops the device from falsifying it. Instead, the network operator also receives an attestation report for the device. If the attestation report matches the CoSWID in the

manifest, then the network operator automatically applies the MUD file that is also contained in the manifest. This allows a secure link to be established between a particular MUD file and a particular software version.

The trust relationships are somewhat more complex with MUD: the network operator may not trust the software author to produce vulnerability-free software. This means that the network operator may choose to override the MUD file in the manifest. Because the MUD file is not even reported by the device, the network operator is free to do this. The network operator can trust the attestation report because it is signed by the verifier. They trust that the values reported in the CoSWID are accurate because it is signed by the software author who also signs the software. They trust that the device is running the software described in the CoSWID because it matches the attestation report. They trust the MUD file because it is signed by the software author - or because they have supplied that MUD file themselves. MUD files may also be obtained from third-party providers, such as Global Platform Iotopia (https://globalplatform.org/iotopia/mud-file-service/).

6. Normative References

[FDO] FIDO Alliance, ., "FIDO Device Onboarding", n.d., <https://fidoalliance.org/specs/FDO/FIDO-Device-Onboard-RD-v1.0-20201202.html>.

[I-D.ietf-rats-eat] Mandyam, G., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", draft-ietf-rats-eat-09 (work in progress), March 2021.

[I-D.ietf-sacm-coswid] Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D. Waltermire, "Concise Software Identification Tags", draft-ietf-sacm-coswid-17 (work in progress), February 2021.

[I-D.ietf-suit-manifest] Moran, B., Tschofenig, H., Birkholz, H., and K. Zandberg, "A Concise Binary Object Representation (CBOR)-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest", draft-ietf-suit-manifest-12 (work in progress), February 2021.

[I-D.ietf-teep-architecture]
Pei, M., Tschofenig, H., Thaler, D., and D. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", draft-ietf-teep-architecture-14 (work in progress), February 2021.

[IoTopia] "Global Platform Iotopia", n.d., <https://globalplatform.org/iotopia/mud-file-service/>.

[LwM2M] "LwM2M Core Specification", n.d., <http://openmobilealliance.org/release/LightweightM2M/V1_2-20201110-A/OMA-TS-LightweightM2M_Core-V1_2-20201110-A.pdf>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/info/rfc8520>.

[SWID] NIST, ., "Software Identification (SWID) Tagging", n.d., <https://csrc.nist.gov/Projects/Software-Identification-SWID/guidelines>.

Author's Address

Brendan Moran
Arm Limited

EMail: Brendan.Moran@arm.com