Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 22, 2019

O. Garcia-Morchon
Philips
T. Dahm
Google

October 19, 2018

Automated IoT Security
draft-garciamorchon-t2trg-automated-iot-security-01

Abstract

The Internet of Things (IoT) concept refers to the usage of standard Internet protocols to allow for human-to-thing and thing-to-thing communication. The security needs are well-recognized but the design space of IoT applications and systems is complex and exposed to multiple types of threats. In particular, threats keep evolving at a fast pace while many IoT systems are rarely updated and still remain operational for decades.

This document describes a comprehensive agile security framework to integrate existing security processes such as risk assessment or vulnerability assessment in the lifecycle of a smart object in an IoT application. The core of our agile security approach relies on two protocols: the Protocol for Automatic Security Configuration (PASC) and the Protocol for Automatic Vulnerability Assessment (PAVA). PASC is executed during the onboarding phase of a smart object in an IoT system and is in charge of automatically performing a risk assessment and assigning a security configuration - applicable to the device or the system - to defeat the identified risks. The assigned security configuration fits the specific environment and threat model of the application in which the device has been deployed. PAVA is executed during the operation of the IoT object and ensures that vulnerabilities in the smart object and IoT system are discovered in a proactive way.

These two protocols can benefit users, manufactures and operators by automating IoT security.

Copyright Notice

Table of Contents

1. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

2. Integrating automated security processes in the IoT lifecycle

The lifecycle of many smart objects in IoT applications such as building automation follows the design and manufacturing processes of traditional hardware components. This means that devices go through a number of phases in their lifecycles that are predefined and rigid, namely design, manufacturing, installation, commissioning, or operation, to name a few of them [IOTSec]. This implies that security is often pre-configured, and this pre-configuration leads to a number of security problems for manufacturers, users, and system operators.

To deal with these problems, we propose the definition of two protocols, PASC and PAVA. PASC aims at automating the security configuration based on information provided by devices, users, manufactures, and system operators. PAVA aims at automating the discovery of new bugs, potential vulnerabilities, and security misconfigurations by gathering information from the actual system, analyzing it, and updating security settings.

2.1. Automated Security Processes for Manufacturers

A manufacturer cannot be aware at design place about the security risks that might appear in the future. Also, often a manufacturer cannot be absolutely certain how his product will be used later on and in what function. A famous example is the newspaper which can also be used as fly swat. Thus, it is very hard for the manufacturer to foresee and implement all security mechanisms and policies that would be applicable to its devices in a wide variety of use cases.

This document introduces security automation into the IoT ecosystem by pursuing a Test Driven Development (TDD) approach as explained in [TDD]. The benefit of TDD for the manufacturer is that products, which pass all the tests, are ready to be shipped. Additionally, manufacturers benefit from this automation approach since they do not need to decide which security mitigations they require on a product. Instead of it, they just need to describe the expected usage of the product, e.g., via MUD files, the PASC and PAVA protocols will then automatically configure the security settings in the system.

2.2. Automated Security Processes for Users

A user is often interested in buying, combining, and running devices from multiple manufacturers. Uses might also have different security and privacy needs. From this point of view, users might have issues making sure that the security settings of his purchased devices and subsystems work together.

Users benefit from integrating security into the full IoT lifecycle since security configuration is transparently done in an automatic way by means of the PASC and PAVA protocols - they need to do nothing. Security settings are automatically configured according to the specific deployment environment that a user only needs to confirm.

2.3. Automated Security Processes for System Integrators

System integrators and operators have to make sure that the overall system - including multiple devices from different manufactures and interacting with many users - is deployed and executed in a secure way. Sometimes, it is also necessary or desired to use products not according to their original purpose, but to repurpose them for a more beneficial use case. Fixed configurations hinder those tasks and make it also difficult to rapidly act in the event of security vulnerabilities.

System operators benefit of PASC and PAVA since they minimize operational cost while ensuring that the system remains secure at any moment: PASC allows them to configure security automatically; PAVA allows for automated vulnerability detection A potential instantiation of part of these protocols follows a Software Defined Network methodology in which network interactions are enabled/ disabled by the network controller depending on the information available in the collected MUD files from the devices. Operators can also adopt the TDD approach and proof compliance with existing security policies for any IoT device by running continuous PAVA tests against the existing IoT installation. If events like software updates introduce an unexpected behavior, the SDN infrastructure will immediately catch and report it.

3. Integrating security workflows in the IoT lifecycle

This section first discusses existing security workflows and how they are usually applied and then it explains how to integrate those security workflows in the IoT lifecycle.

3.1. Security workflows: which ones and how they are traditionally applied.

Dealing with security threats and finding suitable security mitigations is challenging: there are very sophisticated threats that a very powerful attacker could use; also, new threats and exploits appear in a daily basis. Therefore, the existence of proper secure product creation processes that allow managing and minimizing risks during the lifecycle of the IoT devices is at least as important as

being aware of the threats. A non-exhaustive list of relevant processes include:

1. A Business Impact Analysis (BIA) assesses the consequences of loss of basic security attributes, namely, confidentiality, integrity and availability in an IoT system. These consequences might include impact on data lost, sales lost, increased expenses, regulatory fines, customer dissatisfaction, etc. Performing a business impact analysis allow determining the business relevance of having a proper security design placing security in the focus.

2. A Risk Assessment (RA) analyzes security threats to the IoT system, considering their likelihood and impact, and deriving for each of them a risk level. Risks classified as moderate or high must be mitigated, i.e., security architecture should be able to deal with that threat bringing the risk to a low level. Note that threats are usually classified according to their goal: confidentiality, integrity, and availability. For instance, a specific threat to recover a symmetric-key used in the system relates to confidentiality.

3. A privacy impact assessment (PIA) aims at assessing Personal Identifiable Information (PII) that is collected, processed, or used in the IoT system. By doing so, the goals is to fulfill applicable legal requirements, determine risks and effects of the manipulation of PII, and evaluate proposed protections.

4. Procedures for vulnerability assessment (VA) aim at assessing whether the IoT system is secure or any vulnerabilities are present. This can be due to changes in the context information such as people involved in the IoT system or new software vulnerabilities discovered.

5. Procedures for incident reporting (IR) and mitigation refer to the methodologies that allow becoming aware of any security issues that affect an IoT system.

Traditionally, BIA, RA, PIA or VA are to be realized during the creation of a new IoT system, introduction of new technologies in the IoT system, or deployment of significant system upgrades. In general, it is recommended to re-assess them on a regular basis taking into account new use cases or threats. VA is also often realized before deployment, e.g., by performing a penetration test before the new product release is deployed. Incident reporting is done during operation of the IoT system, when a vulnerability is discovered.

All these processes, namely BIA, RA, PIA, VA, and IR, are a must in the design of any IoT system. If they are not performed, the risk of not having a secure enough system is very high. However, even if these procedures are in place, the IoT systems can still have an unsatisfactory security level because of two main reasons: fixed design decisions do not necessarily apply to all deployments due to specific requirements of users and operators or the nature of the final system. new vulnerabilities might appear.

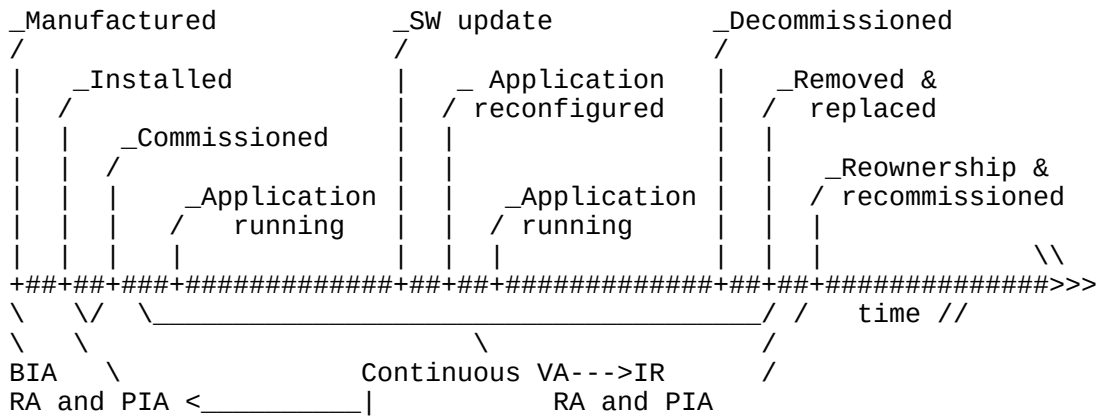

Figure 1: Security workflows integrated in the lifecycle of a thing in the Internet of Things.

3.2. Automating security workflows

Automating IoT security means integrating IoT security workflows in the IoT lifecycle. Figure 1 depicts this concept: on the top part of that figure, we see the traditional steps in the lifecycle of a device: manufacturing, installation, commissioning, application running, etc. Usually, the security workflows discussed in Section 2.1 would only happen at the beginning. The goal is to move integrate them during the lifecycle - as shown on the bottom part of the figure. With this we aim at:

1. making sure that the security settings, methods and policies applied to a given IoT deployment fit the requirements and threats in that specific deployment.

2. ensuring fast reaction in case of new vulnerabilities or changes in the security requirements.

In the figure, we observe that RA and PIA are moved from the design phase to the installation and commissioning phases of the devices

since it is then when the actual environment in which smart objects are deployed is really known. At this point of time, it is possible to gather information about the security requirements of the users, other devices in the system that my pose a threat to the new devices or even new vulnerabilities that might have appeared since the manufacturing of the device till the installation phase.

The VA is executed not only during implementation, but it keeps running during the operation of the IoT system. Information gathered during VA is fed into the RA and PIA processes to update security settings. Similarly, security incidents found out during continuous VA lead to IR. When smart objects are sold or the system updated, this triggers again RA and PIA.

4. Automated IoT security protocols: PASC and PAVA

This section introduces the two protocols for automated IoT security that this document proposes: Protocol for Automatic Security Configuration (PASC) and Protocol for Automated Vulnerability Assessment (PAVA).

The underlying idea of the protocols is shown at a very high level in Figure 2. PASC is used initially when a device first joins the IoT system to adjust the system and device security settings. Then PAVA starts its operation monitoring potential vulnerabilities. If changes in security settings are required, those are then applied by means of PASC messages.

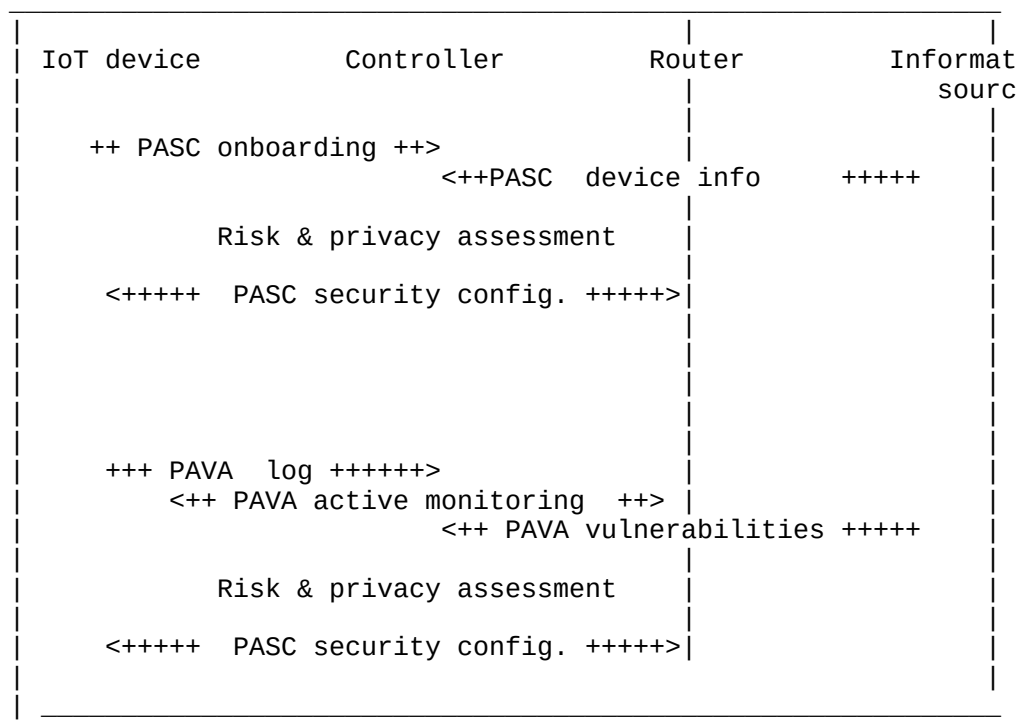

Figure 2: PASC and PAVA interactions.

In the event of a PAVA_VULNERABILITY being received from an INFORMATION SOURCE which is not already patched in the IoT device, the CONTROLLER SHOULD aim to mitigate this PAVA_VULNERABILITY by blocking access to the vulnerable IoT device temporary until the device can be updated.

4.1. PASC: Protocol for Automatic Security Configuration

Figure 1 depicts the main parties involved in an IoT system: an IoT DEVICE, a device controlling the IoT domain called CONTROLLER, a ROUTER towards the IoT domain, and an INFORMATION SOURCE such as it might be a local user, the manufacturer of the IoT device or a cloud IoT management system.

The protocol flow is as follows:

- The IoT DEVICE performs a PASC ONBOARDING exchange in which the system CONTROLLER obtains information about the device from the IoT DEVICE itself.

- The CONTROLLER can also receive PASC DEVICE INFO from other INFORMATION SOURCES such as a local user, the manufacturer, vulnerability cloud,

- The CONTROLLER automatically performs a RISK ASSESSMENT and PRIVACY IMPACT ANALYSIS based on the information about the new IOT DEVICE, system, and information

- Finally, the CONTROLLER configures the system security by means of PASC SECURITY CONFIGURATION MESSAGE. Configuration can apply to the new IoT DEVICES, existing IoT devices, or networking infrastructure such as the ROUTER.

In certain IoT environments, a simple practical instantiation of PASC can be created by extending and combining a number of protocols. PASC ONBOARDING resemble steps of the Manufacturer Usage Descriptor (MUD) protocol by explicitly listing any internal and external accesses the device needs to make, and/or clearly specify if there’s an intentionally open server (e.g., HTTPS port exposed) and might be reused after potential enhancements. Additionally the PASC ONBOARDING needs to include the security policy of the environment the IoT devices are deployed within, for example by verifying the exposed HTTPS server includes a non-vulnerable TLS 1.2 implementation with the desired cipher suites. PASC SECURITY CONFIGURATION MESSAGE might be instantiated in a SDN fashion by means of influencing the routing flows . PASC SECURITY CONFIGURATION MESSAGES might also apply to end devices, and they might realized with extensions of ACE. Another alternative consists in changing actual software configurations in the end devices although this is a less realistic approach for IoT use cases.

The Test Specification must therefore be a description of the expected behavior of the IoT device that can be used to adjust tests accordingly. For example, the specification should explicitly list any internal and external accesses the device needs to make, and/or clearly specify if there’s an intentionally open server (e.g., HTTPS port exposed). This Thing description SHOULD come from Manufacturer Usage Description (MUD). Additionally the Test Specification needs to include the security policy of the environment the IoT devices are deployed within, for example additional tests to verify the exposed HTTPS server includes a non-vulnerable TLS 1.2 implementation with the desired cipher suites.

Network Services modules on the SDN Controller provide for core network services (such as DHCP, DNS, NTP) and mediated access to external resources (e.g., cloud services). A set of "foundational tests" (e.g., DHCP timeouts) SHOULD be part of any Test Specification. The system can capture a packet trace for the

individual device, which can be analyzed during the RISK ASSESSMENT as described in point 3 of section 3.1.

4.2. Protocol for Automatic Vulnerability Assessment (PAVA)

The Protocol for Automatic Vulnerability Assessment (PAVA) aims at assessing for vulnerability when the IoT DEVICES are operational. PAVA is designed to be a key factor for Test Driven Development (TDD) [TDD]. The main aspects of PAVA are as follows:

1. PAVA relies on each IoT DEVICE sending standardized reports PAVA_LOG of potential vulnerabilities to CONTROLLER, e.g., the SDN controller managing the IoT security domain. Such reports would build on RFC5424 (Syslog protocol), RFC5425 (TLS for Syslog) and RFC5426 (Syslog over UDP).

2. The CONTROLLER can also perform PAVA_ACTIVE_MONITORING that refers to messages aiming at verifying that the IoT DEVICE does not suffer known vulnerabilities.

3. The CONTROLLER can also receive PAVA_VULNERABILITIES messages from any INFORMATION SOURCE.

4. Based on the above information, the CONTROLLER can update RISK and PRIVACY ASSESSMENTS. The CONTROLLER reports and methodology can be based on related work such as RFC6872.

5. If needed, the controller can update security settings with a PASC_SECURITY_CONFIGURATION message. Output of this decision can result in 4 different actions:

   * incident report towards the user

   * update of security profiles in IoT DEVICES of the IoT security domain.

   * automatic incident reporting towards the manufacturer

   * automatic incident reporting towards the platform provider

5. Conclusions and security considerations

Security is a key factor in the acceptance and long-term success of IoT systems. Non-smart versions of physical objects in the real word, for example light switches or door locks, can benefit from the modern approach to software engineering. he building and manufacturing industry for example are relatively slowly changing industry sectors due to high demands and regulations on safety and

security of the physical products they produce, e. g. bridges or houses, however, the IT and Web industry are one of the most dynamic industry sectors currently existing and can bring capabilities to make products even safer.

Additionally, there is a fundamental difference of traditional connected and networked devices "for people" vs. IoT devices which are typically headless. E. g., many standard application layer authentication mechanisms like OAuth assume a person is there to "do something" in a challenge response sequence. Also, people have an identity, that typically links to authorization of resources, while an IoT device is more single-purpose and typically has no intrinsic sense of other resources it might/should communicate with. This distinction between devices lends itself to a number of considerations in terms of authentication, access control, manageability, and other challenges that will take time to properly normalize in a modern IoT enabled world.

From a security perspective, it is important to ensure that IoT devices can be trusted. There are simply too many of them, and due to their constrained nature there are often compromises that weaken security overall.

The main contribution of this document is to describe and propose protocols to automate IoT security to deal with the complex IoT security design space. This is done in two steps. First, the PASC protocol allows to automatically configure devices and deploying security profiles - sets of security configurations - to the devices and system infrastructure. Most IoT devices are typically focused on their physical task rather than on being general purpose computing platforms. Therefore, the security profiles described in this document aim to bridge the initial risk analysis gap between the involved industry sectors and put a higher emphasis on the minimizing risk and containing the blast radius factors. Second, the PAVA protocol allows to automatically monitor and audit the operation of the network and system. This ensures fast reaction to any potential vulnerabilities and attacks.

6. Next steps

This draft proposes to automate IoT security by means of PASC & PAVA protocols. IoT security automation would have clear benefits for manufactures, users, and system operators.

If this direction is attractive and supported, we envision the following IETF work:

1. Definition of IoT use cases, overall architecture for IoT security automation, and applicable techniques(e.g., MUD, SDN, ACE,...) to realize PASC & PAVA.

2. Define minimum viable PASC & PAVA protocols, i.e., protocols that allow realizing the concept of automated security with the smallest amount of work. This definition will target building automation use cases. This work requires the following:

   * specifying the information required during onboarding: (1) general provisioning information, for example QR codes containing information like MAC address of the IoT device for easy ingestion of those information into hardware databases; (2) a description of the expected behavior of the IoT device from Manufacturer Usage Description (MUD); (3) environment specific requirements, for example a security policy that is machine-readable; (4) network & application specific information including the definition of the supported protocols, e.g., IPv4, IPv6, application specific networking information, e.g., SSID, and authentication and authorization methodology, e.g., using WPA2 or 802.1X.

   * describing the required input for the automation part: (1) end-users should be allowed to enter security and privacy preferences that should be easily convertible into a machine readable policy; (2) manufacturers provide MUD files potentially with some extensions to support automated security uses cases; (3) system integrators provide the environment specific network and security specifications as listed above.

   * defining the output required or desired by users, routing infrastructure and end devices. This includes routing and firewalling policies for routing infrastructure; security policies and configurations for the end devices including blocked services, whitelist of services in other devices; security configurations and security reports for end users, system operators, and manufacturers (see Section 3.2 point #5).

   * standardizing the PASC Messages, message fields, and interactions between new device, controller, and routing infrastructure including transport protocol for PASC and PAVA messages as well as encoding of security configuration using YANG.

   * creating the RA and PIA logic to generate the (SDN) security configuration in controller and deploy to routers. This can include individual pre-computed flow tables per routing device

determining which end-devices can talk to each other and which services are available to each other. Non-allowed communication patterns are blocked.

* standardizing the PAVA policy and messages for vulnerability assessment as well as messages/Information required from services to perform PAVA. This involves the definition of a policy that determines the behaviour of PAVA regarding the monitoring capabilities (active vs passive), data collection capabilities, and reporting capabilities.

There are several groups within IETF and IRTF working on aspects related to the ideas presented in this group and for which this work can be interesting:

1. IRRF Thing to Thing Research Group (T2TRG) [T2TRG] investigates open research issues in turning a true "Internet of Things" into reality, an Internet where llow-resource nodes ("things", "constrained nodes") can communicate among themselves and with the wider Internet, in order to partake in permissionless innovation.

2. IETF Automated Networking Integrated Model and Approach (ANIMA) [ANIMA] develops a system of autonomic functions that carry out the intentions of the network operator without the need for detailed low-level management of individual devices.

3. IETF Operations and Management Area Working Group (OPSAWG)[OPSAWG] receives occasional proposals for the development and publication of RFCs dealing with operational and management topics that are not in scope of an existing working group and do not justify the formation of a new working group.

4. IETF Interface to the Routing System (I2RS) [I2RS] facilitates real-time or event driven interaction with the routing system through a collection of protocol-based control or management interfaces. These allow information, policies, and operational parameters to be injected into and retrieved (as read or by notification) from the routing system while retaining data consistency and coherency across the routers and routing infrastructure,

5. IETF Security Automation and Continuous Monitoring (SACM) [SACM]. In their charter, they write: "Securing information and the systems that store, process, and transmit that information is a challenging task for enterprises of all sizes, and many security practitioners spend much of their time on manual processes. Standardized protocols and models aiding collection and

evaluation of endpoint elements enable automation, thus freeing practitioners to focus on high priority tasks. Due to the breadth of this work, the working group will address enterprise use cases pertaining to the assessment of endpoint posture (using the definitions of Endpoint and Posture from RFC 5209)."

An open question for the authors is where this work could be done best.

7. Informative References

[ACE] "IETF Authentication and Authorization for Constrained Environments", Web https://datatracker.ietf.org/wg/ace/charter/, n.d..

[ANIMA] "IETF Automated Networking Integrated Model and Approach", Web https://datatracker.ietf.org/wg/anima/about/, n.d..

[I2RS] "IETF Interface to the Routing System", Web https://datatracker.ietf.org/wg/i2rs/about/, n.d..

[ID-MUD] Lear, E., Droms, R., and D. Domascanu, "Manufacturer Usage Description Specification", March 2017.

[IOTSec] Garcia-Morchon, O., Kumar, S., and M. Sethi, "State-of-the-Art and Challenges for the Internet of Things Security", draft-irtf-t2trg-iot-seccons-15 , May 2018.

[OPSAWG] "IETF Operations and Management Area Working Group", Web https://datatracker.ietf.org/wg/opsawg/about/, n.d..

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[SACM] "IETF Security Automation and Continuous Monitoring", Web https://datatracker.ietf.org/wg/sacm/about/, n.d..

[T2TRG] "IRTF Thing-to-Thing (T2TRG) Research Group", Web https://datatracker.ietf.org/rg/t2trg/charter/, n.d..

[TDD] Janzen, D. and H. Saiedian, "Test-driven development concepts, taxonomy, and future direction", Web https://ieeexplore.ieee.org/abstract/document/1510569, n.d..

Authors' Addresses

Oscar Garcia-Morchon
Philips
High Tech Campus 5
Eindhoven, 5656 AA
The Netherlands

Email: oscar.garcia-morchon@philips.com

Thorsten Dahm
Google
todo
Dublin
Ireland

Email: thorstendlux@google.com