Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

O. Garcia-Morchon
Philips
T. Dahm
Google

July 02, 2018

Automated IoT Security
draft-garciamorchon-t2trg-automated-iot-security-00

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 and and many standardization steps for providing security have been taken, for example, the specification of Constrained Application Protocol (CoAP) over Datagram Transport Layer Security (DTLS). However, 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 has three main parts: First, it summarizes exemplary security threats and suitable mitigation strategies to protect against multiple types of threats. Second, it describes a comprehensive agile security framework to integrate existing security processes such as risk asssement or vulnerability assessment in the lifecycle of a smart object in an IoT application. Thus, instead of having a security configuration that is fixed at manufacturing time, our approach allows us to apply a - security profile - on the device tailored for a specific environment at any point of time. Third, we discuss the concept of security profiles and give examples of them.

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). PACS 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 profile to defeat the identified risks. The assigned security profile 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. We describe a few examplary security profiles that could be applicable in different application areas and automatically configured by means of PASC and PAVA.

Status of This Memo

This Internet-Draft will expire on January 3, 2019.

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. Introduction

The Internet of Things (IoT) denotes the interconnection of highly heterogeneous networked entities and networks following a number of communication patterns such as: human-to-human (H2H), human-to-thing (H2T), thing-to-thing (T2T), or thing-to-things (T2Ts). The term IoT was first coined by the Auto-ID center [AUTO-ID] in 1999. Since then, the development of the underlying concepts has ever increased its pace. Nowadays, the IoT presents a strong focus of research with various initiatives working on the (re)design, application, and usage of standard Internet technology in the IoT.

The IoT is exposed to a high number of attack vectors, that if sucessfully exploited by an attacker can have severe consequences. Thus, this document firstly provides an overview of general threats.

Which mitigation strategies are most suitable to and required in an IoT system depends on several factors, including, the operational features of the IoT system or the threats that are applicable to that system. Thus, this document further discusses processes that facilitate the proper design and operation of secure IoT systems, namely business impact analysis, risk assessment, privacy impact analysis, vulnerability analyis and incident reporting. We further argue that even if these processes help IoT system designers to make secure products, a better approach would be to fully integrate these processes in the lifecycle of a smart object in an IoT application. The reason is that IoT products are designed assuming a given environment and threat model that determines the require mitigation strategies. However, in practice, a IoT product can be deployed in very different environments and very different threat models. Furthermore, while threats keep appearing at a very fast pace, IoT systems remain operational - with limited amount of updates - for a very long period of time.

Thus, in order to integrate security processes in the IoT lifecycle, we describe two protocols, the Protocol for Automatic Security Configuration (PACS) and the Protocol for Automatic Vulnerability Assessment (PAVA). These two protocols allow us to integrate risk analysis, privacy impact analysis, and vulnerability assessments in the actual lifecycle of the smart objects so that smart objects can be configured - continuously - with security profiles tailored to the very specific environment in which they are deployed.

Finally, this document describes diffent four exemplary security profiles, each comprising a set of threats, mitigation strategies, and configuration parameters, that would be automatically applied to smart objects when joining different environments.

The rest of the Internet-Draft is organized as follows. Section Section 3 summarizes the design space of secure IoT systems, including lifecycle, device capabilities, and operational features. Section Section 4 discusses general threats that should be considered when designing and operating an IoT system. In Section Section 5, general mitigation strategies to the identified threats are listed. Choosing which mitigation strategies apply to which use cases is not trivial since it is required to find a proper balance between security, cost and usuability. Thus, Section Section 6 details methodologies for managing risks when designing a secure IoT system and dealing with vulnerabilities when operating the system. This section further describes how these methodologies can be integrated in the lifecycle of a smart object. Section Section 7 proposes the Protocol for Automatic Security Configuration (PASC) that allows moving methodologies for risk assessment and privacy impact analysis from the implementation to the onboarding phase of a device. This is enforced since each device discloses its operational requirements when joining an IoT system, and at this specific point of time, a security profile is applied to the device. Section Section 8 describes the Protocol for Automatic Vulnerability Assessment (PAVA) that allows gathering information on potential vulnerabilities as detected by different devices so that vulnerabilities are detected and action can be taken, including the creation of incident reports delivered to the user and manufacturers. Section Section 9 describes how manufactures and users will benefit from PASC and PAVA when creating or using IoT systems. Finally, Section 10 proposes a number of illustrative security profiles applicable to different illustrative clases of IoT systems. Each security profile comprises a set of mitigation strategies can provide a suitable security level and can be automatically deployed using PASC. Section Section 11 includes final remarks and conclusions.

3. The design space of secure IoT systems

This section describes the design space of IoT systems regarding two aspects: a) the lifecycle of a device and b) how an IoT system is architectured.

3.1. The Thing Lifecycle

The lifecycle of a thing refers to the operational phases of a thing in the context of a given application or use case. Figure 1 shows the generic phases of the lifecycle of a thing. This generic lifecycle is applicable to very different IoT applications and scenarios.

We consider an example, a Building Automation and Control (BAC) system, to illustrate the lifecycle and the meaning of these different phases. A BAC system consists of a network of interconnected nodes that performs various functions in the domains of HVAC (Heating, Ventilating, and Air Conditioning), lighting, safety etc. The nodes vary in functionality and a majority of them represent resource constrained devices such as sensors and luminaries. Some devices may also be battery operated or battery-less nodes, demanding for a focus on low energy consumption and on sleeping devices. In our example, the life of a thing starts when it is manufactured. Due to the different application areas (i.e., HVAC, lighting, safety) nodes are tailored to a specific task. It is therefore unlikely that one single manufacturer will create all nodes in a building. Hence, interoperability as well as trust bootstrapping between nodes of different vendors is important. The thing is later installed and commissioned within a network by an installer during the bootstrapping phase. Specifically, the device identity and the secret keys used during normal operation are provided to the device during this phase. Different subcontractors may install different IoT devices for different purposes. Furthermore, the installation and bootstrapping procedures may not be a defined event but may stretch over an extended period of time. After being bootstrapped, the device and the system of things are in operational mode and execute the functions of the BAC system. During this operational phase, the device is under the control of the system owner. For devices with lifetimes spanning several years, occasional maintenance cycles may be required. During each maintenance phase, the software on the device can be upgraded or applications running on the device can be reconfigured. The maintenance tasks can thereby be performed either locally or from a backend system by means of an end-to-end connection. Depending on the operational changes of the device, it may be required to re-bootstrap at the end of a maintenance cycle. The device continues to loop through the operational phase and the eventual maintenance phase until the device is decommissioned at the end of its lifecycle. However, the end-of-life of a device does not necessarily mean that it is defective but rather denotes a need to replace and upgrade the network to next-generation devices in order to provide additional functionality. Therefore the device can be removed and re-commissioned to be used in a different system under a different owner by starting the lifecycle all over again.

```
    _Manufactured           _SW update          _Decommissioned
   /                       /                   /
   |   _Installed          |   _ Application   |   _Removed &
   |  /                    |  / reconfigured   |  /  replaced
   |  |   _Commissioned    |  |                |  |
   |  |  /                 |  |                |  |   _Reownership &
   |  |  |    _Application |  |   _Application |  |  / recommissioned
   |  |  |   /   running   |  |  / running     |  |  |
   |  |  |   |             |  |  |             |  |  |             \\
   +##+##+###+#############+##+##+#############+##+##+##############>>>
       \/  \______________/ \/  \_____________/ \___/         time //
       /           /         \          \          \
   Bootstrapping  /      Maintenance &   \     Maintenance &
                 /      re-bootstrapping  \   re-bootstrapping
           Operational                Operational
```

Figure 1: The lifecycle of a thing in the Internet of Things.

3.2. Classifying IoT Use Cases

An IoT system is architectured according to four main aspects below.

1. Device: what is the role of the devices, what their capabilities are, and which assumptions are posed on them.

2. Network: how the communication happens either in the local network or going towards remote systems.

3. Application and user: requirements and assumptions of the application running on multiple devices on required input information or interactions with the users.

4. System: interacions between multiple devices and users.

3.3. Examplary use cases and security challenges

One of the challenges for IoT security is the diversity in IoT systems and use cases. Examples of use cases with different needs are as follows:

1. A lighting system that runs in a fully isolated manner and only requires some initial interaction by to user to associate a light bulb to a switch.

2. A personal healthcare system in which a user carries medical sensors that monitor the user’s health status in real time and allows the user to share this information with his family doctor.

3. A heating, ventilation and air conditioning system used in a office building that allows controlling settings.

4. A nation-wide smart grid that allows controlling the electrical grid including tasks such as demand-response.

5. A smart home environment in which multiple devices targeted for different applications (e.g., smart lighting, smart lock, smart scale,) can be integrated.

4. Security Threats

Different use cases have different types of threats.

In the following, we describe specific threats. This list is not exhaustive and can be further extended in the future.

1. Cloning of things

2. Counterfeiting

3. Malicious substitution of thing

4. Eavesdropping attack

5. Message injection

6. Message modification

7. Man-in-the-middle attack

8. Firmware Replacement attack

9. Extraction of private information

10. Routing attack
11. Timing attacks
12. Privacy threat - identification
13. Privacy threat - localization
14. Privacy threat - profiling
15. Privacy threat - interaction
16. Privacy threat - lifecycle transitions
17. Privacy threat - inventory attacks
18. Privacy threat - linkage
19. Data leakage - cryptographic keys
20. Data leakage - source code
21. Data leakage - propietary algorithms
22. Denial-of-Service attack on device
23. Denial-of-Service attack on network:
24. Store and decrypt attack (Quantum-resistance)
25. Software vulnerabilities

Tables Figure 5 and Figure 6 in Section Section 13 summarize how these threats apply to different parts of an IoT system at different phases in the device lifecycle.

5. Security Mitigations

Deal with the security threats detailed in Section 4 requires a number of security mitigations as the ones detailed in Internet Draft [ID-Moore]. In this section, we further detail some of them that will be used later to compose security profiles:

1. Capability to perform an authenticated software update.
2. Capability to perform server authentication.
3. Capability to perform client authentication.

4. Capability to encrypt communications.

5. Capability to encrypt communications.

6. Application isololation.

7. Management gateway.

8. Two factor authentication of application requests.

9. Physical security of the device.

10. Usage of application layer proxy.

11. Regular update of authentication credentials.

6. Integrating security processess in the IoT lifecycle

Dealing with above 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 systeoT

Traditionally, BIA, RA, and PIA are usually 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 due to multiple reasons:

1. First example: a risk assessment is performed, but the product is deployed in an environment in which the threats and boundaries are different. This leads to the situation in which an IoT system was properly designed, but it is being used in an environment with different security needs.

2. Second example: a risk assessment is performed during the design phase, then also a vulnerability assessment is executed including a penetration test and the product is released to the customers. Some time later, new vulnerabilities appear in a new devices that was installed in the same IoT network. This leads to the situation in which an IoT system was properly designed and tested for vulnerabilities, but it becomes later unsecured due to changes in the environment.

Thus, the authors believe that the above procedures should be fully integrated in the lifecycle of a smart object as showed in Figure 2. BIA still takes place during the design phase of the new IoT device.

However, RA and PIA are moved now 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. The VA keeps running during the operation of the IoT system. Information gathered during VA can feed 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.

```
    _Manufactured           _SW update          _Decommissioned
   /                       /                   /
   |   _Installed          |   _ Application   |   _Removed &
   |  /                    |  / reconfigured   |  /  replaced
   |  |   _Commissioned    |  |                |  |
   |  |  /                 |  |                |  |   _Reownership &
   |  |  |    _Application |  |   _Application |  |  / recommissioned
   |  |  |   /   running   |  |  / running     |  |  |
   |  |  |   |             |  |  |             |  |  |             \\
   +##+##+###+#############+##+##+#############+##+##+##############>>>
    \   \/  \______________________________________/ /   time //
     \   \                        \                 /
    BIA   \               Continuous VA--->IR      /
           RA and PIA <__________|           RA and PIA
```

Figure 2: Security processes integrated in the lifecycle of a thing in the Internet of Things.

In Section Section 7 we describe the Protocol for Automatic Security Configuration (PACS) that addresses how to solve the integration of the RA and PIA processes in the installation and commissioning phase. Then, in Section Section 8 we describe the Protocol for Automatic Vulnerability Assessment that addresses how to perform continuous vulnerability assessment.

7. Protocol for Automatic Security Configuration (PASC)

Traditional IoT systems are created from scratch and require a suitable security design following the phases descrbed in Section Section 6. Many generic IoT platforms are emerging that can be instantiated in different products that can be deployed in many different environments. Thus, we describe the Protocol for Automatic Security Configuration (PASC) that enables automatic security configuration by shifting methodologies for risk management from the tailored product design and implementation phases to the onboarding phase.

```
  __________________________________________________
|                                                    |
| Thing1       Thing2         GW         Router      |   M1     Platform
|                                                    |
|    ++++++++++ m1 +++++++++>
|                                                    |
|                          +++++++++++++ m2 ++++++++++++>
|                                                    |
|                          <++++++++++++ m3 ++++++++++++
|                                                    |
|                      RA and PIA
|                                                    |
|                          ++++++++++++++++++++ m4 +++++++++++++++++++>
|                                                    |
|                          <+++++++++++++++++++ m5 +++++++++++++++++++
|                                                    |
|      <+++++++++ m6 +++++++++
|                                                    |
|              <++++ m7 ++++++
|                                                    |
|                          ++++ m8 ++++>
|                                                    |
|              IoT Security Domain                   |
  __________________________________________________
```

Figure 3: Protocol for Automatic Security Configuration.

Figure 3 depicts the main parties involved in the protocol: two smart objects denoted as ’Thing1’ and ’Thing2’, a device controlling the IoT domain called ’GW’, a router towards the IoT domain, the manufacturer server of ’Thing1’ denoted as ’M1’ and the server of the platform denoted as ’platform’.

The main protocol steps of PASC are as follows: When ’Thing1’ is introduced in the IoT domain, ’Thing1’ first publishes its profile to the available ’GW’ in message ’m1’. ’GW’ then gathers information from ’m1’ regarding ’Thing1’ in messages ’m2’ and ’m3’. At this stage, ’GW’ has information about the available smart objects in the IoT domain and also can gather input from the user on the usage and expected interactions of the smart object with other devices in the deployment environment. Thus, ’GW’ can perform an automated risk assessment of the IoT device in the security domain determining potential threats on the device and on the system, and assigning a security profile containing security mitigations to the identified threats. In messages ’m4’ and ’m5’ the GW can gather security updates from ’platform’ that might be required for the new situation after the introduction of ’Thing1’ in the IoT security domain. Finally, messages ’m6’, ’m7’ and ’m8’ are used to deploy updated security profiles to the new smart object ’Thing1’ and potentially also to other devices already present in the deployment environment, namely, the ’router’ and other smart objects (e.g., ’Thing2’).

In practice, PACS can be created by extending and combining a number of protocols. Messages ’m1’, ’m2’, and ’m3’ resemble steps of the Manufacturer Usage Descriptor (MUD) protocol. After these messages, RA and PIA can be executed given available information on the expected usage of the devices and input from the user. Messages ’m4’ and ’m5’ require standardization since they resemble the access for various software updates that might be required to fullfil security needs. Configuration messages ’m6’ and ’m7’ might be instantiated by a combination and extension of ACE and MUD protocol. Message ’m8’ requires standarization to automatically configure router and firewall rules.

8. Protocol for Automatic Vulnerability Assessment (PAVA)

Today vulnerability assessment is either not performed at all or it is only performed when products are designed. The Protocol for Automatic Vulnerability Assessment (PAVA) overcomes this. PAVA relies on each smart object (e.g., Thing1) sending standarized reports of potential vulnerabilities to ’GW’, the device managing the IoT security domain. Such reports would build on RFC 5424, RFC 5425 and RFC 5426. Reports and methodology can also benefit from RFC6872. The ’GW’ then analyzes the logs and takes a decision regarding the existence of a vulnerability, its origin and its impact. Output of this decision is threefold:

1. incident report towards the user

2. update of security profiles in smart objects of the IoT security domain.

3. automatic incident reporting towards the manufacturer

4. automatic incident reporting towards the platform provider

9. Benefits of integrating security processes in the IoT lifecycle through PASC and PAVA

Section Section 8 describes how manufacturers, system operators and end users benefit from PASC and PAVA when creating, making or using IoT systems.

Users benefit since security configuration is done in an automatic way - they need to do nothing. Security settings are automatically configured according to the specific deployment environment that a user only needs to confirm.

Manufacturers benefit 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 that is then confirmed by the user. Security profiles are then automatically deployed on the smart object.

System operators use these protocols to minimize operational cost while ensuring that the system remains secure at any moment.

10. Security Profiles

We expect the various types of IoT deployments to be widespread and to penetrate almost all areas of our personal and professional life including building automation systems, healthcare, smart cities, logistics, etc. For each of these environments, properties such as device capabilities, network infrastructure, or available security services can be completely different. That makes it difficult to define and deploy complete security configurations for each generic use case. Furthermore, each of those applications is featured by a different number of actors deployed in very different environments and with very different purposes. Consequently, when a Business Impact Analysis or Risk Assessment is performed, not only the types of threats will be different, but also their likelihood and potential impact. This determines that different applications tend to require different or complementary types of security mechanisms mitigating the identified risks.

This section describes some exemplary Security Profiles that can be automatically created by means of PASC fitting the security needs of applications with the same characteristics and requirements. These security profiles are beneficial since they make the underlying threats transparent, allow for interoperability while preserving security and prevent possible security misconfiguration. It is expected that the security profiles defined in this section need to be extended and adapted based on the individual risk profiles of each environment as described in Section 6 of this document.

Each security profile includes:

1. a short descriptive name,

2. an exemplary application that might use the security profile,

3. the main security threats applicable to the profile,

4. the security mitigations required by the profile,

5. specific configuration parameters for the protocols and actors involved in the application.

10.1. Classes of IoT Systems

Based on the PASC the IoT devices can be grouped by function, by required access and by deployment scope into individual IoT device classes. While grouping things into individual device classes based on function and required access is a universal part of each PASC independent of the desired deployment environment, the deployment scope MUST be considered as well based on the different threats in various deployment environments. For example, the same thing deployed in smart homes or in smart cities will have the same PASC entries for function and required access, however, the deployment scope and the inherited security threats from the different environments will require different PASC and PAVA for the two deployment scenarios.

Each one of these IoT device classes will represent an isolated segment in itself and will receive an individual and continuous PAVA during the lifetime of the things in the device class. In order to connect with things in different segments, the management gateway MUST be used.

The goal of creating device classes for IoT devices is to enable the near-automatic management of a clear separation of security threats and risk assessments by enforcing device segmentation for each class of devices. This segmentation process SHOULD therefore be automated, but the automation part itself is out of scope for this document. The segments must be pre-defined before the PASC is created. If the PASC requires a new segment to introduce a thing into a certain environment, the segment MUST be defined first. Protocols like MUD SHOULD be used as a valuable source of information during the classification and provisioning process in PASC.

We consider four generic security profiles applicable to four exemplary application areas as summarized in the table below:

	Exemplary IoT Application	Description
SecProf_1	Home usage	Enables operation between home things without interaction with central device
SecProf_2	Managed Home usage	Enables operation between home things. Interaction with a central and local device is possible
SecProf_3	Industrial usage	Enables operation between things. Relies on central (local or backend) device for security
SecProf_4	Advanced Industrial usage	Enables ad-hoc operation between things and relies on central device or on a collection of control devices

Figure 4: Security profiles and application areas.

The currently existing IoT products can be loosely categorized in 4 different profiles, where SecProf_1 would be the lowest category of security profiles and SecProf_4 would be the highest category of security profiles. It is considered best practice in the security world to allow higher security profiles to connect to lower security profiles, but to never let lower security profiles connect to higher security profiles. The same precautions SHOULD be used for the IoT Security Profiles defined below. The separation between the Security Profiles described in Figure 4 is not a strict physical separation, but a logical one. A home IoT device and its management software may include components that fall into the SecProf_1 as well as SecProf_2 category. Within every security profile exists a graduation of different security levels. The exact category within a security profile will be determined with a risk analysis of the thing and its functionality and MUST be reviewed on a regular basis. This is because each security profile will contain devices with a high lifecycle variation. Certain IoT devices are meant to be used for a few hours only, while others are expected to last decades. Given the technological progress, the security of a thing may degenerate over time within the same security profile.

The best mitigation strategy against unknown future threats are software updates, for example, to replace a broken hash algorithm with a more secure one as long as the thing can handle the computational load of the new hash algorithm.

10.2. Security Profile 1: Home usage

SecProf_1 categorizes unmanaged IoT devices mostly found in private homes. The things in this Security Profile are single-purpose devices, used either on a daily or less frequent basis. The types of threats those things will face are usually minimal risk. The likelihood of misuse entirely depends on physical proximity to the thing.

Given the example of an internet-connected button for the delivery of fresh bananas, it would require physical interaction ("button press") and SHOULD make use of technologies like fingerprint sensors to limit the order ability to a small set of authorized individuals. A misuse would at maximum lead to an unwanted delivery of fruits, and a supermarket can easily enforce a maximum amount of fruits an individual household would order before assuming malicious intent.

This Security Profile requires unidirectional communication from the thing to a specific service. Additional services like order confirmation will be handled via separate channels. Mitigations for security threats identified in the PASC MUST contain encryption on the transport layer of the application, a strict isolation from other nodes in a shared network and a proper physical placement of the thing. Additionally, a strong identification mechanism, like X.509 Certificates, MUST be used to identify the exact thing that talks to the specific service.

Threats	Mitigations
T4	M4
T5	M2, M3, M8
T6	M2, M3, M8

10.3. Security Profile 2: Managed Home usage

SecProf_2 categorizes managed IoT devices mostly found in private homes. The things in this Security Profile are more complex, often multi-purpose devices, and meant to be used on a daily basis. The types of threats those things will face are usually in the medium to high risk category. Misuse of the thing depends on the security of the managed service bundled to the thing.

Given the example of an smart door lock, the PASC contains physical and logical security risks. The physical security of the lock MUST be on the same standard that non-smart door locks provide. For the logical security of the door lock, physical presence close to the smart door lock MUST be enforced for the unlocking functionality, while the locking functionality might also be used remotely. Key escrow must be possible via a secure procedure for emergency services like Police or the Fire Brigade.

This Security Profile requires bidirectional communication from the thing to a specific management gateway. All communication with specific services as well as other smart objects MUST go through the management gateway. The management gateway may act as an application layer proxy when it is used as a relay to enable communication between smart objects and nodes within a single domain or local network. Mitigations for security threats identified in the PASC MUST contain encryption on the transport layer of the application and a strict isolation from other nodes except the management gateway in a shared network. Additionally, a strong identification and authentication mechanism, like X.509 Certificates, MUST be used to identify and authenticate the thing when it talks to the management gateway. The credentials used for authentication and authorization MUST be refreshed on a regular basis.

10.4. Security Profile 3: Industrial usage

SecProf_3 categorizes unmanaged or partially managed IoT devices found in industrial or commercial environments. The things in this Security Profile are single-purpose devices, used by a number of unidentified people. The types of threats those things will face are in the minimal or medium risk category. Misuse could lead to a certain inconvenience, but would not put the operation of the industrial or commercial environment at risk.

Given the example of a HVAC system in a commercial office building, the components of such a system would include a central HVAC management service for the building, temperature sensors spread across the whole building and heating and cooling devices at certain places across the building. Communication from the smart objects spread across the building would be unidirectional depending on their functionality. The temperature sensors would unidirectional communicate frequently with the HVAC central management service. The HVAC central management service would unidirectional communicate as needed with the heating and cooling devices to regulate the temperature across the building.

This Security Profile requires a mix of unidirectional and bidirectional communication between the things and a specific service. Mitigations for security threats identified in the PASC MUST contain encryption on the transport layer of the application, a strict isolation from other nodes in a shared network for the smart things and a strong identification mechanism, like X.509 Certificates, MUST be used to identify the exact thing that talks to the central management service. Mitigations for security threats identified in the PASC for central management service which requires bidirectional communication with multiple things MUST contain encryption on the transport layer of the application and MUST use a strong identification and authorization mechanism, like X.509 Certificates, to identify and authenticate the central management service when it talks to the individual smart objects. The central management service may act as an application layer proxy when it is used as a relay to enable communication between smart objects and nodes within a single domain or local network. The credentials used for authentication and authorization MUST be refreshed on a regular basis.

10.5. Security Profile 4: Managed Industrial usage

SecProf_4 categorizes fully managed IoT devices found in industrial or commercial environments. The things in this Security Profile are multi-purpose devices, used by a number of authenticated and authorized people. The types of threats those things will face are in the high risk category. Misuse could lead to a partial or full compromise of the industrial or commercial environment.

Given the example of a physical security system with managed access in a commercial datacenter, the components of such a system would include components like cameras, infrared sensors, access control systems and fire safety. All components have either unidirectional or bidirectional connectivity to a local or remote management gateway. All communication with specific services as well as other smart objects MUST go through the management gateway. The management gateway controls the functionality of each smart component within the integrated physical security system. The management gateway may act as an application layer proxy when it is used as a relay to enable communication between the individual components of the integrated physical security system and external nodes within a single domain or local and remote networks.

Mitigations for security threats identified in the PASC MUST contain encryption on the transport layer of the application and a strict isolation from other nodes except the management gateway in a shared network. Additionally, a strong identification and authentication mechanism, like X.509 Certificates, MUST be used to identify and authenticate all IoT components for the communication with the management gateway. The credentials used for authentication and authorization MUST be refreshed on a regular basis.

11. Conclusions

The main contribution of this document is to describe and propose protocols to automate IoT security. 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 that join a given network and system. Second, the PAVA protocol allows to automatically monitor the operation of the network and system in order to defeat any attack. A key contribution of this document is the definition of exemplary security profiles that can be deploy to the devices.

12. Security Considerations

Security is a key factor in the acceptance and long-term success of IoT systems. When comparing established Things that already exists as non-smart versions in the real word for a long time, for example light switches or door locks, and the typical modern approach to software engineering, we can often see a culture clash. This culture clash is not surprising. The reasons for this are simple, the building and manufacturing industry for example are some of the slowest changing industry sectors in the world, often also due to high demands and regulations on safety and security of the physical products they produce, e. g. bridges or houses. On the other side, we have the IT and Web industry, one of the most dynamic industry sectors currently existing. While the formula on how to mix concrete or unlocking a door with a physical key has not changed much in the last 100 years, we went to a huge number of fundamental changes in the software industry in a relatively short period of time.

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 difficult to trust IoT devices. There are simply too many of them, and due to their constrained nature there are often compromises that weaken security overall. 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.

13. Summary of threats

We can classify threats presented in Section Section 4 according to two criteria: a) what is the target of the threat? and b) when does the threat take place?

The target of the threat can be - as described in Section 3.2 - the IoT architecture (T-arch), the device (T-dev), the network (T-nwk), and the application (T-app). The lifecycle moment in which the threat takes place can be - as described in Section 3.1 - during manufacturing (L-make), commissioning process (L-conf), operation (L-oper), software updates (L-update), and decommissioning (L-deconf).

	T-arch	T-dev	T-nwk	T-app
1	y	y		
2	y	y		
3	y	y		
4	y		y	y
5	y		y	y
6	y		y	y
7	y		y	y
8	y	y		y
9	y	y		
10	y		y	
11	y	y	y	y
12	y	y	y	y

13	y	y	y	y
14		y		y
15		y		y
16	y	y	y	y
17	y			y
18	y			y
19	y	y	y	y
20	y	y		
21	y	y		y
22		y		
23	y	y	y	
24	y			y
25	y	y		y

Figure 5: This tables illustrates which parts of the IoT system are affected by different theats.

	L-make	L-conf	L-oper	L-upd	L-dec	L-after
1	y		y	y		
2	y		y	y		
3	y		y			
4		y	y	y		
5		y	y	y		
6		y	y	y		
7		y	y	y		

8	y			y		
9	y		y	y		
10		y	y	y	y	
11	y	y	y	y	y	
12	y	y	y	y	y	
13		y	y	y	y	
14			y			
15			y			
16		y	y	y	y	y
17		y	y	y	y	
18	y			y		
19	y	y	y	y	y	
20	y		y	y		
21	y		y	y		
22		y	y	y	y	
23		y	y			
24			y			y
25		y	y	y	y	

Figure 6: This tables illustrates in which moment of a thing’s lifecycle a threat can take place.

14. IANA Considerations

This document contains no request to IANA.

15. Acknowledgments

16. Informative References

[Article29] "Opinion 8/2014 on the on Recent Developments on the Internet of Things", Web http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp223_en.pdf, n.d..

[AUTO-ID] "AUTO-ID LABS", Web http://www.autoidlabs.org/, September 2010.

[BACNET] "BACnet", Web http://www.bacnet.org/, February 2011.

[BITAG] "Internet of Things (IoT) Security and Privacy Recommendations", Web http://www.bitag.org/report-internet-of-things-security-privacy-recommendations.php, n.d..

[cctv] "Backdoor In MVPower DVR Firmware Sends CCTV Stills To an Email Address In China", Web https://hardware.slashdot.org/story/16/02/17/0422259/backdoor-in-mvpower-dvr-firmware-sends-cctv-stills-to-an-email-address-in-china, n.d..

[CSA] "Security Guidance for Early Adopters of the Internet of Things (IoT)", Web https://downloads.cloudsecurityalliance.org/whitepapers/Security_Guidance_for_Early_Adopters_of_the_Internet_of_Things.pdf, n.d..

[d2dsecurity] Haus, M., Waqas, M., Ding, A., Li, Y., Tarkoma, S., and J. Ott, "Security and Privacy in Device-to-Device (D2D) Communication: A Review", Paper IEEE Communications Surveys and Tutorials, 2016.

[DALI] "DALI", Web http://www.dalibydesign.us/dali.html, February 2011.

[DHS] "Strategic Principles For Securing the Internet of Things (IoT)", Web https://www.dhs.gov/sites/default/files/publications/Strategic_Principles_for_Securing_the_Internet_of_Things-2016-1115-FINAL....pdf, n.d..

[ENISA_ICS]
"Communication network dependencies for ICS/SCADA Systems", European Union Agency For Network And Information Security , February 2017.

[ETSI_GR_QSC_001]
"Quantum-Safe Cryptography (QSC);Quantum-safe algorithmic framework", European Telecommunications Standards Institute (ETSI) , June 2016.

[Fairhair]
"Fairhair Alliance", Web https://www.fairhair-alliance.org/, n.d..

[FCC] "Federal Communications Comssion Response 12-05-2016", FCC , February 2016.

[FTCreport]
"FTC Report on Internet of Things Urges Companies to Adopt Best Practices to Address Consumer Privacy and Security Risks", Web https://www.ftc.gov/news-events/press-releases/2015/01/ftc-report-internet-things-urges-companies-adopt-best-practices, n.d..

[GSMAsecurity]
"GSMA IoT Security Guidelines", Web http://www.gsma.com/connectedliving/future-iot-networks/iot-security-guidelines/, n.d..

[ID-6lodect]
Mariager, P., Petersen, J., Shelby, Z., Logt, M., and D. Barthel, "Transmission of IPv6 Packets over DECT Ultra Low Energy", draft-ietf-6lo-dect-ule-09 , December 2016.

[ID-6lonfc]
Choi, Y., Hong, Y., Youn, J., Kim, D., and J. Choi, "Transmission of IPv6 Packets over Near Field Communication", draft-ietf-6lo-nfc-05 , October 2016.

[ID-6tisch]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-11 , January 2017.

[ID-aceoauth]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-oauth-authz-05 , March 2011.

[ID-bootstrap]
Sarikaya, B. and M. Sethi, "Secure IoT Bootstrapping : A Survey", draft-sarikaya-t2trg-sbootstrapping-01 , July 2016.

[ID-cose] Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-24 , November 2016.

[ID-Daniel]
Park, S., Kim, K., Haddad, W., Chakrabarti, S., and J. Laganier, "IPv6 over Low Power WPAN Security Analysis", draft-daniel-6lowpan-security-analysis-05 , March 2011.

[ID-dietesp]
Migault, D., Guggemos, T., and C. Bormann, "Diet-ESP: a flexible and compressed format for IPsec/ESP", draft-mglt-6lo-diet-esp-02 , August 2016.

[ID-Hartke]
Hartke, K. and O. Bergmann, "Datagram Transport Layer Security in Constrained Environments", draft-hartke-core-codtls-02 , July 2012.

[ID-HIP] Moskowitz, R., "HIP Diet EXchange (DEX)", draft-moskowitz-hip-rg-dex-06 , May 2012.

[ID-Moore]
Moore, K., Barnes, R., and H. Tschofenig, "Best Current Practices for Securing Internet of Things (IoT) Devices", draft-moore-iot-security-bcp-00 , October 2016.

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

[ID-Nikander]
Nikander, P. and J. Melen, "A Bound End-to-End Tunnel(BEET) mode for ESP", draft-nikander-esp-beet-mode-09 , August 2008.

[ID-OFlynn]
O’Flynn, C., Sarikaya, B., Ohba, Y., Cao, Z., and R. Cragie, "Security Bootstrapping of Resource-Constrained Devices", draft-oflynn-core-bootstrapping-03 , November 2010.

[ID-OSCOAP]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security of CoAP (OSCOAP)", draft-selander-ace-object-security-05 , July 2016.

[ID-proHTTPCoAP]
Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Best practices for HTTP-CoAP mapping implementation", draft-castellani-core-http-mapping-07 , February 2013.

[ID-rd] Shelby, Z., Koster, M., Bormann, C., and P. Stok, "CoRE Resource Directory", draft-ietf-core-resource-directory-09 , October 2016.

[ID-senml]
Jennings, C., Shelby, Z., Arkko, J., Keranen, A., and C. Bormann, "Media Types for Sensor Measurement Lists (SenML)", draft-ietf-core-resource-directory-09 , October 2016.

[ID-Tsao] Tsao, T., Alexander, R., Dohler, M., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-ietf-roll-security-framework-07 , January 2012.

[ID-Williams]
Williams, M. and J. Barrett, "Mobile DTLS", draft-barrett-mobile-dtls-00 , March 2009.

[IEEE802ah]
"Status of Project IEEE 802.11ah, IEEE P802.11- Task Group AH-Meeting Update.",
Web http://www.ieee802.org/11/Reports/tgah_update.htm, n.d..

[IIoT] "Industrial Internet Consortium",
Web http://www.iiconsortium.org/, n.d..

[IoTSecFoundation]
"Establishing Principles for Internet of Things Security", Web https://iotsecurityfoundation.org/establishing-principles-for-internet-of-things-security/, n.d..

[iotsu] "Patching the Internet of Things: IoT Software Update Workshop 2016", Web https://www.ietf.org/blog/2016/07/patching-the-internet-of-things-iot-software-update-workshop-2016/, n.d..

[IPSO] "IPSO Alliance", Web http://www.ipso-alliance.org, n.d..

[JOURNAL-Perrig]
Perrig, A., Szewczyk, R., Wen, V., Culler, D., and J. Tygar, "SPINS: Security protocols for Sensor Networks", Journal Wireless Networks, September 2002.

[lora] "LoRa - Wide Area Networks for IoT", Web https://www.lora-alliance.org/, n.d..

[nbiot] "NarrowBand IoT", Web http://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_69/Docs/RP-151621.zip, n.d..

[NHTSA] "Cybersecurity Best Practices for Modern Vehicles", Web https://www.nhtsa.gov/staticfiles/nvs/pdf/812333_CybersecurityForModernVehicles.pdf, n.d..

[NIST] Dworkin, M., "NIST Specification Publication 800-38B", 2005.

[NIST-Guide]
Ross, R., McEVILLEY, M., and J. Oren, "Systems Security Engineering", Web http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160.pdf, n.d..

[nist_lightweight_project]
"NIST lightweight Project", Web www.nist.gov/programs-projects/lightweight-cryptography, www.nist.gov/sites/default/files/documents/2016/10/17/sonmez-turan-presentation-lwc2016.pdf, n.d..

[OCF] "Open Connectivity Foundation", Web https://openconnectivity.org/, n.d..

[OneM2M] "OneM2M", Web http://www.onem2m.org/, n.d..

[OWASP] "IoT Security Guidance", Web https://www.owasp.org/index.php/IoT_Security_Guidance, n.d..

[PROC-Chan] Chan, H., Perrig, A., and D. Song, "Random Key Predistribution Schemes for Sensor Networks", Proceedings IEEE Symposium on Security and Privacy, 2003.

[PROC-Gupta] Gupta, V., Wurm, M., Zhu, Y., Millard, M., Fung, S., Gura, N., Eberle, H., and S. Shantz, "Sizzle: A Standards-based End-to-End Security Architecture for the Embedded Internet", Proceedings Pervasive Computing and Communications (PerCom), 2005.

[PROC-Smetters-02] Balfanz, D., Smetters, D., Steward, P., and H. Chi Wong,, "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks", Paper NDSS, 2002.

[PROC-Smetters-04] Balfanz, D., Durfee, G., Grinter, R., Smetters, D., and P. Steward, "Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute", Paper USENIX, 2004.

[PROC-Stajano-99] Stajano, F. and R. Anderson, "Resurrecting Duckling - Security Issues for Adhoc Wireless Networks", 7th International Workshop Proceedings, November 1999.

[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>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <https://www.rfc-editor.org/info/rfc3756>.

[RFC3833] Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)", RFC 3833, DOI 10.17487/RFC3833, August 2004, <https://www.rfc-editor.org/info/rfc3833>.

[RFC4016] Parthasarathy, M., "Protocol for Carrying Authentication and Network Access (PANA) Threat Analysis and Security Requirements", RFC 4016, DOI 10.17487/RFC4016, March 2005, <https://www.rfc-editor.org/info/rfc4016>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <https://www.rfc-editor.org/info/rfc4555>.

[RFC4621] Kivinen, T. and H. Tschofenig, "Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol", RFC 4621, DOI 10.17487/RFC4621, August 2006, <https://www.rfc-editor.org/info/rfc4621>.

[RFC4738] Ignjatic, D., Dondeti, L., Audet, F., and P. Lin, "MIKEY-RSA-R: An Additional Mode of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 4738, DOI 10.17487/RFC4738, November 2006, <https://www.rfc-editor.org/info/rfc4738>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC5191] Forsberg, D., Ohba, Y., Ed., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, DOI 10.17487/RFC5191, May 2008, <https://www.rfc-editor.org/info/rfc5191>.

[RFC5206] Nikander, P., Henderson, T., Ed., Vogt, C., and J. Arkko, "End-Host Mobility and Multihoming with the Host Identity Protocol", RFC 5206, DOI 10.17487/RFC5206, April 2008, <https://www.rfc-editor.org/info/rfc5206>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5713] Moustafa, H., Tschofenig, H., and S. De Cnodder, "Security Threats and Security Requirements for the Access Node Control Protocol (ANCP)", RFC 5713, DOI 10.17487/RFC5713, January 2010, <https://www.rfc-editor.org/info/rfc5713>.

[RFC5903] Fu, D. and J. Solinas, "Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2", RFC 5903, DOI 10.17487/RFC5903, June 2010, <https://www.rfc-editor.org/info/rfc5903>.

[RFC6345] Duffy, P., Chakrabarti, S., Cragie, R., Ohba, Y., Ed., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Relay Element", RFC 6345, DOI 10.17487/RFC6345, August 2011, <https://www.rfc-editor.org/info/rfc6345>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <https://www.rfc-editor.org/info/rfc6551>.

[RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, DOI 10.17487/RFC6568, April 2012, <https://www.rfc-editor.org/info/rfc6568>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/info/rfc6690>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC7158] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7158, DOI 10.17487/RFC7158, March 2014, <https://www.rfc-editor.org/info/rfc7158>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <https://www.rfc-editor.org/info/rfc7390>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/info/rfc7401>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/info/rfc7517>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <https://www.rfc-editor.org/info/rfc7668>.

[RFC7696] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

[RFC7815] Kivinen, T., "Minimal Internet Key Exchange Version 2 (IKEv2) Initiator Implementation", RFC 7815, DOI 10.17487/RFC7815, March 2016, <https://www.rfc-editor.org/info/rfc7815>.

[RFC7925] Tschofenig, H., Ed. and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things", RFC 7925, DOI 10.17487/RFC7925, July 2016, <https://www.rfc-editor.org/info/rfc7925>.

[RG-T2TRG] "IRTF Thing-to-Thing (T2TRG) Research Group", Web https://datatracker.ietf.org/rg/t2trg/charter/, December 2015.

[SchneierSecurity] "The Internet of Things Is Wildly Insecure--And Often Unpatchable", Web https://www.schneier.com/essays/archives/2014/01/the_internet_of_thin.html, n.d..

[sigfox] "Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)", Web https://www.sigfox.com/, n.d..

[SPEKE] "IEEE P1363.2: Password-based Cryptography", 2008.

[THESIS-Langheinrich]
Langheinrich, M., "Personal Privacy in Ubiquitous Computing", PhD Thesis ETH Zurich, 2005.

[Thread] "Thread Group", Web http://threadgroup.org/, n.d..

[TinyDTLS]
"TinyDTLS", Web http://tinydtls.sourceforge.net/, February 2012.

[TR69] "Too Many Cooks - Exploiting the Internet-of-TR-069-Things", Web https://media.ccc.de/v/31c3_-_6166_-_en_-_saal_6_-_201412282145_-_too_many_cooks_-_exploiting_the_internet-of-tr-069-things_-_lior_oppenheim_-_shahar_tal, n.d..

[WG-6LoWPAN]
"IETF 6LoWPAN Working Group", Web http://tools.ietf.org/wg/6lowpan/, February 2011.

[WG-ACE] "IETF Authentication and Authorization for Constrained Environments (ACE) Working Group", Web https://datatracker.ietf.org/wg/ace/charter/, June 2014.

[WG-CoRE] "IETF Constrained RESTful Environment (CoRE) Working Group", Web https://datatracker.ietf.org/wg/core/charter/, February 2011.

[WG-LWIG] "IETF Light-Weight Implementation Guidance (LWIG) Working Group", Web https://datatracker.ietf.org/wg/lwig/charter/, March 2011.

[WG-MSEC] "MSEC Working Group", Web http://datatracker.ietf.org/wg/msec/, n.d..

[wink] "Wink’s Outage Shows Us How Frustrating Smart Homes Could Be", Web http://www.wired.com/2015/04/smart-home-headaches/, n.d..

[ZB] "ZigBee Alliance", Web http://www.zigbee.org/, February 2011.

[Ziegeldorf]
Ziegeldorf, J., Garcia-Morchon, O., and K. Wehrle,, "Privacy in the Internet of Things: Threats and Challenges", Paper Security and Communication Networks - Special Issue on Security in a Completely Interconnected World, 2013.

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