Workshop on Smart Object Security, in conjunction with IETF83, Paris 23 March 2012

Security Considerations around End-to-End Security in the IP-based Internet of Things

Martina Brachmann, Oscar Garcia-Morchon, Sye-Loong Keoh, Sandeep S. Kumar
Philips Research Europe, High Tech Campus 34,
5656AE, Eindhoven, The Netherlands
{*martina.brachmann, oscar.garcia, sye.loong.keoh, sandeep.kumar*}@philips.com

Abstract

The IP-based Internet of Things refers to the interconnection of smart objects in a Low-power and Lossy Network (LLN) with the Internet by means of protocols such as 6LoWPAN or CoAP. The provisioning of an end-to-end security connection is the key to ensure basic functionalities such as software updates or network access. This is, however, very challenging due to the asymmetry of the devices in the system (smart objects are resource constrained when compared with traditional Internet devices), and the interaction of possibly different security protocols such as TLS and DTLS. This paper describes the security and threat model for this scenario to arrive at the following conclusion: end-to-end security in the IP-based Internet of Things is more than a simple end-to-end handshake and requires additional measures to protect the LLN while performing the handshake. We further describe some simple solutions and give hints for further work.

1. Introduction

The Internet of Things (IoT) denotes the interconnection of highly heterogeneous networked entities such as sensors, actuators, mobile devices, etc. The use of IPv6 and web services as fundamental building blocks for IoT applications has created a homogeneous protocol ecosystem that allows simple integration of IoT devices in a LLN with Internet hosts. This greatly simplifies the deployment of the envisioned scenarios ranging from building automation to production environments to personal area networks, in which heterogeneous *smart objects* in a LLN might interact with each other, and with a human carrying a smart phone, or with a back-end service in the Internet. The IETF CoRE working group aims at providing a framework for resource-oriented applications intended to run on constrained IP networks (6LoWPANs). A lightweight version of the HTTP protocol, the Constrained Application Protocol (CoAP) that runs over UDP has been defined to enable efficient application-level communication for IoT devices.

Security is a key aspect for the above application areas and it is crucial that the basic security services such as confidentiality, authentication, and freshness of keys between two communicating entities are provided. Information exchanged in the network must therefore be protected end-to-end (E2E). With this, CoAP has identified Datagram Transport Layer Security (DTLS) [1] as the mandatory approach to protect the exchange of a CoAP communication in a LLN. Due to the asymmetry of the system, there are other security considerations around E2E security such as protecting the LLN from flooding and replay attacks since devices in the LLN have significantly less computational resources and memory when compared to Internet devices. In this paper, we would like to highlight the security issues and the threat model, and subsequently provide an initial thought on how we can best approach them. The paper is organized as follows: In Section 2, we present the use cases and the architectures that we consider relevant. Section 3 describes the security challenges and threat models, while Section 4 discusses possible approaches to mitigate the security issues. Finally, we conclude the paper with future work.

2. Use Cases and Architectures

We consider the architectures as illustrated in Figure 1 and Figure 2. It consists of a back-end in the Internet, a 6LoWPAN Border Router (6LBR) and a group of nodes running CoAP located in a LLN (e.g. 6LoWPAN). The 6LBR interconnects the Internet with the LLN, thus allowing for the access to the CoAP/6LoWPAN devices from anywhere on the Internet.

As illustrated in Figure 1, when the two communicating endpoints understand CoAP, E2E security can be provided using DTLS [1]. However, there exist back-end legacy systems that only support

HTTP over TCP and rely on TLS for secure connections. This architecture is shown in Figure 2. It is thus important to ensure that a HTTP device on the Internet can use HTTP to access resources from a CoAP device directly in a secure manner. In this scenario, the protocol conversion is done by a HTTP/CoAP proxy that is present between them (e.g., in the 6LBR).

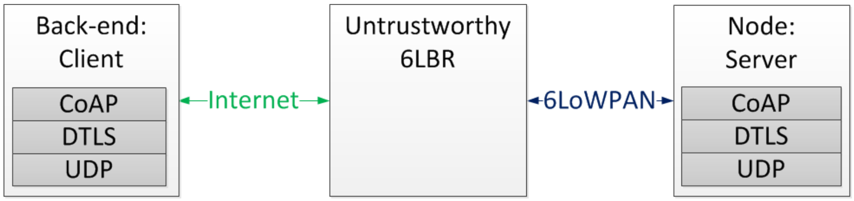

Figure 1: Back-end is using CoAP to communicate with node

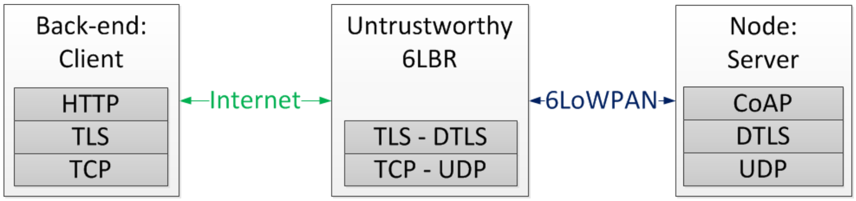

Figure 2: Back-end is using HTTP to communicate with the node

3. Security Considerations and Threat Model

Based on the architectures as described in Section 2, we identify two major security goals:

1. Ensure E2E security between the end-hosts.
2. Protect all the hosts in the LLN during an E2E security connection. This includes the protection of the LLN host and the LLN itself from (flooding, replay, amplification, etc) attacks, the protection of the host in the Internet, as well as the protection of the 6LBR.

When analyzing how to achieve these high-level security goals, one has to find answers to two main questions that define our threat model: (a) *How to perform an E2E handshake?* (b) *Who is the attacker and what attacks can be launched?*

3.1 Security Issues in End-to-End Handshake

In the case of a CoAP/CoAP use case as shown in Figure 1, DTLS can be applied E2E. For the HTTP/CoAP use case as illustrated in Figure 2, performing an E2E handshake remains a challenge; in particular, it is not clear whether a (partial) mapping between TLS and DTLS can be performed. This is even more complex because the CoAP devices in the LLN do not know whether the key establishment requests originate either from a HTTP or a CoAP device. In fact, the handshake phase is the most vulnerable period to attacks since the end-hosts have not yet authenticated to each other, and the communication is un-encrypted.

3.2 Attackers and Attack Models

The LLN and the LLN host are prone to flooding and resource exhaustion attacks because it consists of devices that are resource-constrained. We consider the attacker to be an Internet host, attempting to start multiple E2E handshakes with the LLN host. The DTLS handshake as it is, would allow the attacker to exhaust the resources of battery-powered devices in the LLN because a cookie request would be sent by the LLN host for each request; additionally malicious traffic would flood the network. This is just one of the many attack scenarios. Other examples, not considered further in this paper, could include attacks in which the attacker is in the LLN network, fragmentation issues of the security messages that make the verification of individual fragments impossible, etc.

In general, such scenarios lead to a key observation for the threat model: to ensure both the security goals above, it is just not enough to have a handshake between the end-hosts, but additional measures are needed in the 6LBR to protect the LLN hosts and the LLN itself. For instance, the 6LBR should be able to verify (i) whether the exchanged requests between two devices are located outside and inside the 6LoWPAN/CoAP network; or (ii) that requests coming from a client are valid in order to prevent (or limit the effect of), e.g., an energy exhausting attack, and still, the system needs to be designed such that it works even if the 6LBR cannot be completely trusted with the secret keys of the LLN nodes in its network.

4. Approaches to E2E security

In order to protect the LLN, attempts to flood the LLN from the Internet must be stopped at the 6LBR. This would require additional functionalities in the 6LBR to filter messages, thus allowing only authenticated and authorized messages into the LLN. Requiring the two end hosts and the 6LBR to mutually authenticate each other can partially mitigate the attacks. A well-known method is tunneling [2]. In the CoAP/CoAP use case, a DTLS-DTLS tunnel can be established. First the back-end creates a secure channel using DTLS to the 6LBR. When successful, the back-end establishes a second DTLS connection through the first connection to the CoAP device. The DTLS-DTLS tunnel requires changes in the network stack of the back-end. Therefore it is not suitable when the back-end is a legacy system, i.e., the HTTP/CoAP use case. Even though a TLS-DTLS tunnel can be created in the HTTP/CoAP use case, this would also require changes in the network stack of the back-end. Additionally, the CoAP device in the LLN would require prior knowledge of the back-end services that it is interacting with (whether it's HTTP or CoAP), and the 6LBR is required to perform the mapping of TLS to DTLS during the (D)TLS handshake.

Assuming that the two end hosts share a common secret-key established in the handshake, message verification and replay detection cannot be done in the 6LBR due to the fact that it does not have access to the secret key. Therefore, this must be devolved to the CoAP device. For the same reason, it is apparent in the HTTP/CoAP use case that the HTTP-CoAP mapping cannot be done by the 6LBR, and a dedicated HTTP-CoAP proxy in the LLN needs to be deployed.

5. Conclusions and future work

Through our observation and analysis, providing E2E security in IoT is not so trivial, mainly due to many possible usage scenarios, i.e, CoAP/CoAP and HTTP/CoAP mediated by a 6LBR, that have different constraints and requirements. Our analysis also reveals that having a secure E2E connection between two end hosts only provides a secure communication channel; the LLN can still be vulnerable to resource exhaustion, flooding, replay and amplification attacks, since the 6LBR typically does not perform any authentication. We have shown that tunneling can be useful to partially mitigate the resource exhaustion problem, where the 6LBR acts as a shield for the LLN by authenticating the source of the message and only forwards the relevant messages into the LLN. However, having two independent tunnels would incur a lot of communication overhead, we hope to find alternative solutions or methods that can optimize tunneling in the future.

The focus of this paper has been mainly on the CoAP/CoAP use case, and the HTTP/CoAP use case poses a greater challenge as it also involves secure protocol translation, which is not an easy task when the payload is encrypted. It remains a challenge to perform HTTP-CoAP mapping (either at the 6LBR or a dedicated proxy in the LLN) in a secure manner if true end-to-end security is required.

References

[1] E. Rescorla, N. Modadugu. Datagram Transport Layer Security. RFC4347, April 2006.

[2] Strategies to Secure End-to-End Communication - And Their Application to SCTP-Based Communication**,** R. Seggelmann, M. Tüxen, E.P. Rathgeb, *PIK - Praxis der Informationsverarbeitung und Kommunikation*, ISSN: 1865-8342, Vol. 34, No. 4, December 2012 (To Appear)