Demo: Enabling En-Route Filtering for End-to-End Encrypted CoAP Messages

Klara Seitz
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
Klara.Seitz@student.hpi.de

Sebastian Serth
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
Sebastian.Serth@student.hpi.de

Konrad-Felix Krentz
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
Konrad-Felix.Krentz@hpi.de

Christoph Meinel
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
Christoph.Meinel@hpi.de

ABSTRACT

IoT devices usually are battery-powered and directly connected to the Internet. This makes them vulnerable to so-called path-based denial-of-service (PDoS) attacks. For example, in a PDoS attack an adversary sends multiple Constrained Application Protocol (CoAP) messages towards an IoT device, thereby causing each IoT device along the path to expend energy for forwarding this message. Current end-to-end security solutions, such as DTLS or IPsec, fail to prevent such attacks since they only filter out inauthentic CoAP messages at their destination. This demonstration shows an approach to allow en-route filtering where a trusted gateway has all necessary information to check the integrity, decrypt and, if necessary, drop a message before forwarding it to the constrained mote. Our approach preserves precious resources of IoT devices in the face of path-based denial-of-service attacks by remote attackers.

CCS CONCEPTS

• **Security and privacy** → **Network security**; *Block and stream ciphers*; *Hash functions and message authentication codes*; • **Networks** → **Application layer protocols**; • **Hardware** → Power and energy;

KEYWORDS

Internet of Things, Path-Based Denial-of-Service Attacks (PDoS), Constrained Application Protocol (CoAP)

ACM Reference Format:
Klara Seitz, Sebastian Serth, Konrad-Felix Krentz, and Christoph Meinel. 2017. Demo: Enabling En-Route Filtering for End-to-End Encrypted CoAP Messages. In *Proceedings of SenSys '17, Delft, Netherlands, November 6–8, 2017,* 2 pages.
https://doi.org/10.1145/3131672.3136960

SenSys '17, November 6–8, 2017, Delft, Netherlands

ACM ISBN 978-1-4503-5459-2/17/11…$15.00
https://doi.org/10.1145/3131672.3136960

1 INTRODUCTION

The Internet of Things (IoT) comprises constrained devices, called IoT devices or motes. Motes typically offer few resources and are battery-powered. A challenge for data exchange is hence minimizing the energy consumption, which may be achieved via special protocols, such as the *Constrained Application Protocol (CoAP)* [7] and 6LoWPAN. The latter compresses IPv6 headers and handles packet fragmentation. CoAP, on the other hand, is a RESTful web transfer protocol with a small message overhead but lacks support for encryption. Therefore, it has to be combined with DTLS or IPsec. However, while both, DTLS and IPsec protect sensitive data neither of them protects against *path-based denial-of-service (PDoS)* attacks. For example, attackers can replay packets from outside the IoT network and only the destination mote can decrypt and then do integrity checks. Therefore, intermediate nodes have to forward these packets, forcing them to expend their precious energy.

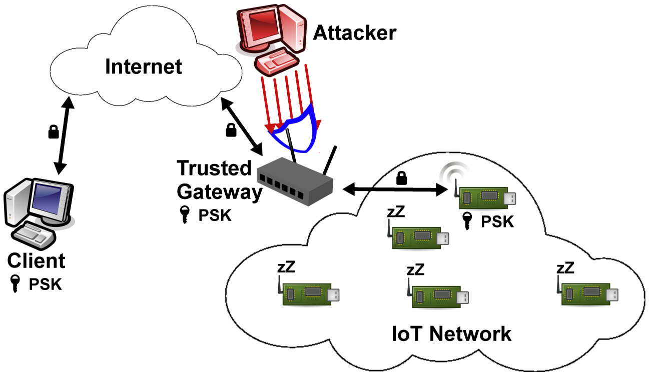

Figure 1: We enable a trusted gateway to inspect encrypted messages and drop malicious packets. This enhances the PDoS protection for battery-powered IoT devices.

We introduce a trusted gateway (see Figure 1) which has access to the pre-shared key (PSK) used for encryption. The gateway checks the integrity of the message, decrypts the payload and, based on an inspection, drops or forwards the unaltered message. Our approach preserves the end-to-end security while allowing to filter en-route before a packet reaches its destination. This method saves battery of IoT devices in the face of PDoS attacks by remote attackers.

2 RELATED WORK

End-to-End Security. End-to-end security can be achieved with different protocols. One of them is the protocol suite IPsec [2] on the Internet layer, which can be combined with 6LoWPAN [5]. Alternatively, the CoAP specification recommends the use of *Datagram Transport Layer Security (DTLS)* [6]. This allows encryption of the complete application layer and an integrity check of the message. Even though there exist lightweight implementations for both, IPsec and DTLS, they still add a fairly large communication overhead. Our approach provides similar security mechanisms within CoAP at the application layer. Since we only encrypt the payload, other security mechanisms (such as SMACK [1]) can still be applied.

En-Route Filtering. Current en-route filtering methods identify and drop packets sent from a compromised IoT device towards the Internet. These approaches attach a signature to the messages, which is checked by forwarding motes and may result in dropping the packet or isolating the compromised mote [4]. Our approach implements en-route filtering in a reverse way where faulty messages from an attacker outside the IoT network are prevented from reaching the motes. This prevents path-based denial-of-service attacks by remote attackers while preserving the end-to-end security.

3 OUR APPROACH: A TRUSTED GATEWAY

Basically, our approach works as follows. A legitimate client encrypts the payload of a CoAP message and adds special options to the CoAP message (cf. Figure 2) including an HMAC. Before entering the IoT network, the message is received by the gateway. It re-calculates the HMAC and compares the result with the received one. Only if both match, the integrity of the message has been proven and the payload may be decrypted for deeper inspection. Additionally, the gateway ensures that the CoAP message was not replayed. If the CoAP message is valid, it is forwarded to the desired IoT device or dropped otherwise. The IoT device then decrypts the payload and processes the CoAP message as usual. Both, the trusted gateway and the IoT device, require access to the PSK that is used for decrypting the payload and for generating the HMAC.

Accordingly, our approach involves adding custom options to the CoAP header, which we explain in the following. Currently, replayed messages cannot be distinguished from valid retransmitted messages since the CoAP header only increases the message ID with every new CoAP message. Therefore, we add a retransmission counter, which is incremented with every retransmission. In addition, we insert a boot counter, which represents the number of boot cycles of the client. Together, the message ID, the retransmission counter and the boot counter enable our gateway to filter out replayed CoAP messages en-route by comparing them with the ones of the last forwarded CoAP message from the client.

Also, as CoAP does not define any native encryption support, the attacker has full access to the content of a recorded packet. Instead of encrypting the whole application layer like DTLS and IPsec, we propose to encrypt the CoAP payload only, which reduces the processing overhead as a side effect. Our implementation uses an AES-128-based encryption with hardware acceleration. To allow encryption and to ensure integrity, we propose to set up a pre-shared key (PSK) between each pair of client and mote prior to the communication. During initial setup, the PSKs are enumerated and senders identify themselves by adding their so-called Client ID to CoAP messages, as shown in Figure 2. Lastly, clients add a *hash-based message authentication code (HMAC)* to CoAP messages. This HMAC is calculated using a key derived from the PSK and the complete CoAP message – except for the HMAC field – as input. As depicted in Figure 2, our implementation uses five custom CoAP options to support the decryption and prevent PDoS, as well as replay attacks by remote attackers. Using custom CoAP options is fully compliant with the CoAP specification. Beyond that, our approach avoids session key establishment, which saves communication overhead.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Client ID   |          Boot Counter         |Retrans. Count.|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   HMAC (8 bytes)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Encr. Algorithm| ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Encrypted Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: A simplified representation of a CoAP packet including our custom options and an encrypted payload.

4 DEMONSTRATION

We demonstrate the feasibility of our approach in a video available at https://youtu.be/gP3CBrqLItY. Our setup contains two *Open-Motes CC2538 Rev. A1* with a custom version of the Contiki OS[1] representing a gateway and a CoAP server. In addition, we adapted the Firefox plugin *Copper* [3] to support our proposals[2].

This work enables a trusted gateway to filter end-to-end-encrypted CoAP traffic based on a full packet inspection. Our future work will concentrate on identifying malicious packets on the gateway and evaluating different encryption algorithms.

REFERENCES

[1] C. Gehrmann, M. Tiloca, and R. Höglund. 2015. SMACK: Short message authentication check against battery exhaustion in the Internet of Things. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on.* IEEE, 274–282.

[2] S. Kent and K. Seo. 2005. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard). (Dec. 2005), 101 pages. https://doi.org/10.17487/RFC4301 Updated by RFCs 6040, 7619.

[3] M. Kovatsch. 2011. Demo abstract: Human-coap interaction with copper. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on.* IEEE, 1–2.

[4] A. Kumar and A. R. Pais. [n. d.]. En-Route Filtering Techniques in Wireless Sensor Networks: A Survey. *Wireless Personal Communications* ([n. d.]), 1–43.

[5] Shahid Raza, Simon Duquennoy, Tony Chung, Dogan Yazar, Thiemo Voigt, and Utz Roedig. 2011. Securing communication in 6LoWPAN with compressed IPsec. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on.* IEEE, 1–8.

[6] E. Rescorla and N. Modadugu. 2012. Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard). (Jan. 2012), 32 pages. https://doi.org/10.17487/RFC6347 Updated by RFCs 7507, 7905.

[7] Z. Shelby, K. Hartke, and C. Bormann. 2014. The Constrained Application Protocol (CoAP). RFC 7252 (Proposed Standard). (June 2014), 112 pages. https://doi.org/10.17487/RFC7252 Updated by RFC 7959.

[1] https://github.com/MrSerth/contiki

[2] https://github.com/MrSerth/Copper