M. Richardson
Sandelman Software Works
P. van der Stok
vanderstok consultancy
P. Kampanakis
Cisco Systems

October 18, 2018

Constrained Join Proxy for Bootstrapping Protocols
draft-vanderstok-anima-constrained-join-proxy-00

Abstract

This document defines a protocol to securely assign a pledge to an owner, using an intermediary node between pledge and owner. This intermediary node is known as a "constrained-join-proxy".

This document extends the work of [ietf-anima-bootstrapping-keyinfra] by replacing the Circuit-proxy by a stateless constrained join-proxy, that transports routing information.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Enrolment of new nodes into constrained networks with constrained nodes present is described in [I-D.ietf-anima-bootstrapping-keyinfra] and makes use of Enrolment over Secure Transport (EST) [RFC7030]. The specified solutions use https and may be too large in terms of code space or bandwidth required. Constrained devices in constrained networks [RFC7228] typically implement the IPv6 over Low-Power Wireless personal Area Networks (6LoWPAN) [RFC4944] and Constrained Application Protocol (CoAP) [RFC7252].

CoAP has chosen Datagram Transport Layer Security (DTLS) [RFC6347] as the preferred security protocol for authenticity and confidentiality of the messages. A constrained version of EST, using Coap and DTLS, is described in [I-D.ietf-ace-coap-est].

DTLS is a client-server protocol relying on the underlying IP layer to perform the routing between the DTLS Client and the DTLS Server. However, the new "joining" device will not be IP routable until it is authenticated to the network. A new "joining" device can only initially use a link-local IPv6 address to communicate with a neighbour node using neighbour discovery [RFC6775] until it receives the necessary network configuration parameters. However, before the device can receive these configuration parameters, it needs to authenticate itself to the network to which it connects. In [I-D.ietf-anima-bootstrapping-keyinfra] Enrolment over Secure Transport (EST) [RFC7030] is used to authenticate the joining device. However, IPv6 routing is necessary to establish a connection between joining device and the EST server.

This document specifies a Join-proxy and protocol to act as intermediary between joining device and EST server to establish a connection between joining device and EST server.

This document is very much inspired by text published earlier in [I-D.kumar-dice-dtls-relay].

2. Terminology

The following terms are defined in [RFC8366], and are used identically as in that document: artifact, imprint, domain, Join Registrar/Coordinator (JRC), Manufacturer Authorized Signing Authority (MASA), pledge, Trust of First Use (TOFU), and Voucher.

3. Requirements Language

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant STuPiD implementations.

4. Join Proxy functionality

As depicted in the Figure 1, the joining Device, or pledge (P), is more than one hop away from the EST server (E) and not yet authenticated into the network. At this stage, it can only communicate one-hop to its nearest neighbour, the Join proxy (J) using their link-local IPv6 addresses. However, the Pledge (P) needs to communicate with end-to-end security with a Registrar hosting the EST server (E) to authenticate and get the relevant system/network parameters. If the Pledge (P) initiates a DTLS connection to the EST server whose IP address has been pre-configured, then the packets are dropped at the Join Proxy (J) since the Pledge (P) is not yet admitted to the network or there is no IP routability to Pledge (P) for any returned messages.

```
                 ++++
                 |E |----       +--+        +--+
                 |  |    \      |J |........|P |
                 ++++     \-----|  |        |  |
               EST server       +--+        +--+
               REgistrar      Join Proxy   PLedge
                                           "Joining" Device
```

Figure 1: multi-hop enrolment.

Furthermore, the Pledge (P) may wish to establish a secure connection to the EST server (E) in the network assuming appropriate credentials are exchanged out-of-band, e.g. a hash of the Pledge (P)'s raw public key could be provided to the EST server (E). However, the Pledge (P) is unaware of the IP address of the EST-server (E) to initiate a DTLS connection and perform authentication with.

A DTLS connection is required between Pledge and EST server. To overcome the problems with non-routability of DTLS packets and/ or discovery of the destination address of the EST Server to contact, the Join Proxy is introduced. This Join-Proxy functionality is configured into all authenticated devices in the network which may act as the Join Proxy for newly joining nodes. The Join Proxy allows for routing of the packets from the Pledge using IP routing to the intended EST Server.

5. Join Proxy specification

In this section, the constrained Join Proxy functionality is specified using DTLS and coaps. When a joining device as a client attempts a DTLS connection to the EST server, it uses its link-local IP address as its IP source address. This message is transmitted one-hop to a neighbour node. Under normal circumstances, this message would be dropped at the neighbour node since the joining device is not yet IP routable or it is not yet authenticated to send messages through the network. However, if the neighbour device has the Join Proxy functionality enabled, it routes the DTLS message to a specific EST Server. Additional security mechanisms need to exist to prevent this routing functionality being used by rogue nodes to bypass any network authentication procedures.

The Join-proxy is stateless to minimize the requirements on the constrained Join-proxy device.

If an untrusted DTLS Client that can only use link-local addressing wants to contact a trusted end-point EST Server, it sends the DTLS message to the Join Proxy. The Join Proxy extends this message into a new type of message called Join ProxY (JPY) message and sends it on to the EST server. The JPY message payload consists of two parts:

- Header (H) field: consisting of the source link-local address and port of the Pledge (P), and

- Contents (C) field: containing the original DTLS message.

On receiving the JPY message, the EST Server retrieves the two parts. The EST Server transiently stores the Header field information. The EST server uses the Contents field to execute the EST server functionality. However, when the EST Server replies, it also extends its DTLS message with the header field in a JPY message and sends it back to the Join Proxy. The Header contains the original source link-local address and port of the DTLS Client from the transient state stored earlier (which can now be discarded) and the Contents field contains the DTLS message.

On receiving the JPY message, the Join Proxy retrieves the two parts. It uses the Header field to route the DTLS message retrieved from the Contents field to the Pledge.

The Figure 2 depicts the message flow diagram when the EST Server end-point address is known only to the Join Proxy:

```
+--------------+-----------+--------------+-----------------------+
| EST  Client  | Join Proxy |    EST server |        Message        |
|     (P)      |     (J)    |      (E)      |Src_IP:port|Dst_IP:port|
+--------------+-----------+--------------+-----------+-----------+
|      --ClientHello-->                    | IP_C:p_C  |IP_Ra:5684 |
|                    --JPY[H(IP_C:p_C),--> | IP_Rb:p_Rb|IP_S:5684  |
|                          C(ClientHello)] |           |           |
|                    <--JPY[H(IP_C:p_C),-- | IP_S:5684 |IP_Rb:p_Rb |
|                          C(ServerHello)] |           |           |
|      <--ServerHello--                    | IP_Ra:5684|IP_C:p_C   |
|              :                           |           |           |
|              :                           |     :     |    :      |
|                                          |     :     |    :      |
|      --Finished-->                       | IP_C:p_C  |IP_Ra:5684 |
|                    --JPY[H(IP_C:p_C),--> | IP_Rb:p_Rb|IP_S:5684  |
|                          C(Finished)]    |           |           |
|                    <--JPY[H(IP_C:p_C),-- | IP_S:5684 |IP_Rb:p_Rb |
|                          C(Finished)]    |           |           |
|      <--Finished--                       | IP_Ra:5684|IP_C:p_C   |
|              :                           |     :     |    :      |
+------------------------------------------+-----------+-----------+
IP_C:p_C = Link-local IP address and port of the Pledge
IP_S:5684 = IP address and coaps port of EST Server
IP_Ra:5684 = Link-local IP address and coaps port of Join Proxy
IP_Rb:p_Rb = IP address(can be same as IP_Ra) and port of Join Proxy

JPY[H(),C()] = Join ProxY message with header H and content C
```

Figure 2: constrained joining message flow.

6. Protocol

The JPY message is constructed as a payload with media-type application/multipart-core specified in [I-D.ietf-core-multipart-ct]. Header and Contents fields use different media formats:

1. header field: application/CBOR containing a CBOR array [RFC7049] with the pledge IPv6 Link Local address as a 16-byte binary value, the pledge’s UDP port number, if different from 5684, as a CBOR integer, and the proxy’s ifindex or other identifier for the physical port on which the pledge is connected.

2. Content field: Any of the media types specified in [I-D.ietf-ace-coap-est] dependent on the EST function that is requested:

* application/pkcs7-mime; smime-type=server-generated-key
* application/pkcs7-mime; smime-type=certs-only
* application/pkcs7-mime; smime-type=CMC-request
* application/pkcs7-mime; smime-type=CMC-response
* application/pkcs8
* application/csrattrs
* application/pkcs10

Examples are shown in Appendix A.

7. Discovery

It is assumed that Join-Proxy seamlessly provides a coaps connection between Pledge and coaps EST-server. An additional Registrar is needed to connect the Pledge to an http EST server, see section 8 of [I-D.ietf-ace-coap-est].

The Discovery of the coaps EST server by the Join Proxy follows section 6 of [I-D.ietf-ace-coap-est]. The discovery of the Join-Proxy by the Pledge is an extension to the discovery described in section 4 of [I-D.ietf-anima-bootstrapping-keyinfra]. In particular this section replaces section 4.2 of [I-D.ietf-anima-bootstrapping-keyinfra]. Three discovery cases are discussed: coap discovery, 6tisch discovery and GRASP discovery.

7.1. GRASP discovery

In the context of autonomous networks, discovery takes place via the GRASP protocol as described in [I-D.ietf-anima-bootstrapping-keyinfra]. The port number is.

EDNote: to be specified

7.2. 6tisch discovery

The discovery of EST server by the pledge uses the enhanced beacons as discussed in [I-D.ietf-6tisch-enrollment-enhanced-beacon].

7.3. Coaps discovery

In the context of a coap network without Autonomous Network support, discovery follows the standard coap policy. The Pledge can discover a Join-Proxy by sending a link-local multicast message to ALL CoAP Nodes with address FF02::FD. Multiple or no nodes may respond. The handling of multiple responses and the absence of responses follow section 4 of [I-D.ietf-anima-bootstrapping-keyinfra].

The presence and location of (path to) the join-proxy resource are discovered by sending a GET request to "/.well-known/core" including a resource type (rt) parameter with the value "brski-proxy" [RFC6690]. Upon success, the return payload will contain the root resource of the Join-Proxy resources. It is up to the implementation to choose its root resource; throughout this document the example root resource /est is used. The example below shows the discovery of the presence and location of voucher resources.

```
REQ: GET coap://[FF02::FD]/.well-known/core?rt=brski-proxy

RES: 2.05 Content
</est>; rt="brski-proxy"
```

Port numbers, not returned in the example, are assumed to be the default numbers 5683 and 5684 for coap and coaps respectively (sections 12.6 and 12.7 of [RFC7252]. Discoverable port numbers MAY be returned in the <href> of the payload.

8. Security Considerations

It should be noted here that the contents of the CBOR map are not protected, but that the communication is between the Proxy and a known registrar (a connected UDP socket), and that messages from other origins are ignored.

9. IANA Considerations

This document needs to create a registry for key indices in the CBOR map. It should be given a name, and the amending formula should be IETF Specification.

9.1. Resource Type registry

This specification registers a new Resource Type (rt=) Link Target Attributes in the "Resource Type (rt=) Link Target Attribute Values" subregistry under the "Constrained RESTful Environments (CoRE) Parameters" registry.

rt="brski-proxy". This EST resource is used to query and return the supported EST resource of a join-proxy placed between Pledge and EST server.

10. Acknowledgements

Much of this text is inspired by [I-D.kumar-dice-dtls-relay]. Many thanks for the comments by Brian Carpenter.

11. Changelog

11.1. 00 to 00

- o added payload examples in appendix
- o discovery for three cases: AN, 6tisch and coaps

12. References

12.1. Normative References

[I-D.ietf-6tisch-enrollment-enhanced-beacon]
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join and Enrollment Information", draft-ietf-6tisch-enrollment-enhanced-beacon-00 (work in progress), July 2018.

[I-D.ietf-ace-coap-est]
Stok, P., Kampanakis, P., Kumar, S., Richardson, M., Furuhed, M., and S. Raza, "EST over secure CoAP (EST-coaps)", draft-ietf-ace-coap-est-06 (work in progress), October 2018.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-16 (work in progress), June 2018.

[I-D.ietf-core-multipart-ct]
Fossati, T., Hartke, K., and C. Bormann, "Multipart Content-Format for CoAP", draft-ietf-core-multipart-ct-02 (work in progress), August 2018.

[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>.

[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>.

[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>.

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/info/rfc8366>.

12.2. Informative References

[duckling] Stajano, F. and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks", 1999, <https://www.cl.cam.ac.uk/~fms27/papers/1999-StajanoAnd-duckling.pdf>.

[I-D.kumar-dice-dtls-relay] Kumar, S., Keoh, S., and O. Garcia-Morchon, "DTLS Relay for Constrained Environments", draft-kumar-dice-dtls-relay-02 (work in progress), October 2014.

[pledge] Dictionary.com, ., "Dictionary.com Unabridged", 2015, <http://dictionary.reference.com/browse/pledge>.

[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>.

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

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[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>.

Appendix A. Example payloads

Examples are extensions of two examples shown in [I-D.ietf-ace-coap-est].

EDNote:
provisional stake holder examples to be improved and corrected.

A.1. cacerts

The request from Join-Proxy to EST-server looks like:

```
Get coaps://192.0.2.1/est/crts
content format = 287
payload =
[60,[L-L IPv6, port, ident]]
```

The response will then be

```
 2.05 Content
 content format = 287
 payload =
 [60, [L-L IPv6, port, ident], 281, <DTLS encoded certificate>]
```

A.2. serverkeygen

The request from Join-Proxy to EST-server looks like:

```
Get coaps://192.0.2.1/est/skg
content format = 287
payload =
[60,[L-L IPv6, port, ident], 286, <DTLS encoded request>]
```

The response will then be

```
 2.05 Content
 content format = 287
 payload =
 [60, [L-L IPv6, port, ident],
  284, <DTLS encoded preamble>,
  281, <DTLS encoded key>]
```

Authors' Addresses

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Peter van der Stok
vanderstok consultancy

Email: consultancy@vanderstok.org

Panos Kampanakis
Cisco Systems

Email: pkampana@cisco.com