Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

T. Fossati
Alcatel-Lucent
H. Tschofenig
ARM Ltd.

March 9, 2015

Resource Directory Names for Certificate Mode DTLS
draft-fossati-core-certmode-rd-names-00

Abstract

This memo describes the use of Resource Directory names in CoAP Certificate Mode DTLS for the purpose of verifying the identity of a server by a client endpoint.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Table of Contents

1. Introduction

Today, many Internet of Things (IoT) deployments consist of an IoT device that interacts with a cloud service infrastructure. (This deployment model is described in Section 2.2 of [I-D.iab-smart-object-architecture].)

If TLS/DTLS is used to mutually authenticate the device and the cloud server, then the guidance in [I-D.ietf-dice-profile] - which, in turn, takes [RFC7252] recommendations into account - should be followed.

In particular, according to Section 9.1.3.3 of [RFC7252], a client that receives a certificate from the server must check that the authority of the requested URI matches "at least one of the authorities of any CoAP URI found in a field of URI type in the SubjectAltName (SAN) set. If there is no SubjectAltName in the certificate, then the authority of the request URI must match the

Common Name (CN) found in the certificate [...]."

According to Section 4.2.1.6 of [RFC5280] an URI that includes an authority - such as a 'coaps' URI - needs to include a fully qualified domain name (FQDN), or an IP literal as its host part.

(So, an IoT device that wants to talk to a CoAP server at coaps://example.com will expect to receive a certificate with a matching URI in either the content of the SAN extension or the CN.)

The combination of the two requirements above, together with text in Section 3 of [RFC6066] which only allows FQDN hostname of the server in the ServerName field, basically binds Certificate Mode DTLS to either DNS, or static host tables containing FQDN's mappings, or some other system for lookup of registered names which is able to fully mimic the DNS naming scheme.

While DNS can be taken for granted in the Web, CoAP networks do not mandate its presence. In fact, there are IoT deployments where the server infrastructure is located in a home or residential environment in which IoT devices interact with the server solely in the local network (see also Section 2.1 of [I-D.iab-smart-object-architecture]).

Since static configuration is not generally a viable option, in order to cope with scenarios like the one described above there is a need to define some kind of stable, non-DNS, identifier that can be used for 'coaps' URIs in Certificate Mode DTLS as a fall-back in case DNS is not deployed, or not understood by CoAP endpoints.

1.1. Challenges

There seem to be at least four challenges that need to be solved to make sure that the IoT device is indeed talking to a server whose X.509 certificate identity can be compared with the requested CoAP URI:

1. what identifiers should be used in the certificate?

2. What identifier should be contained in the hostname part of the endpoint URI?

3. What identifier should be communicated in the SNI during the TLS/DTLS exchange?

4. How can the identifier in the CoAP URI be mapped to an IP address?

The way the Web solves these problems is by assuming that the name of an application service is based on a DNS domain name, as stated in [RFC6125]. The identifiers used in the certificate and in the SNI are then FQDN's.

In order to offer a solution for the CoAP space this document suggests the use of Resource Directory endpoint names (and domains) as an alternative to DNS names.

2. Terminology and Requirements Language

This specification requires the reader to be familiar with the terminology used in documents produced by the CoRE, TLS, and PKIX working groups.

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 [RFC2119].

3. Resource Directory Names and Domains

In CoAP networks, a Resource Directory (RD) [I-D.ietf-core-resource-directory] is an entity that acts as a centralized store where protocol endpoints can register and lookup links to resources that are made available in the network. The RD defines the concept of an "endpoint name" which identifies a given Endpoint (i.e. web server) within a given "domain". Under the assumption of its uniqueness, an endpoint name/domain can be used as a stable host component for CoAP authorities.

3.1. Uniqueness Guarantee

An endpoint name is guaranteed to be unique within the associated domain. If the domain is elided during registration, the RD should

assure its uniqueness within an implicit default domain.

3.2. Authority Format

3.2.1. Requirements

The syntax for RD name authorities has been designed to satisfy the following requirements:

REQ#1: full compatibility with URI reg-name syntax;

REQ#2: support identifiers from different and independently administered sources (e.g. those defined in OMA spec, EUI-64 [EUI-64], etc.);

REQ#3: allow for an optional "domain" under which a given name exists (for compatibility with current RD spec).

3.2.2. Syntax

The following ABNF reuses 'port' from [RFC3986]; ALPHA and DIGIT from [RFC5234].

```
RD-char = ALPHA / DIGIT / "-" / "_" / "~" / "!" /
          "$" / "&" / "'" / "(" / ")" / "*" /
          "," / ";" / "="
RD-ns = ALPHA *(ALPHA / DIGIT / "-")  ; the name-space
RD-name = 1*RD-char
RD-domain = 1*63RD-char
RD-authority = [ RD-ns "+" ] RD-name [ "." RD-domain ] [ ":" port ]
```

Note that RD-char is the set of chars allowed in reg-name (REQ#1) from which the two following characters have been removed:

- the dot ("."), which is used to introduce the domain component (REQ#3);
- the plus ("+"), which is used to encode namespace information along with the name in an unambiguous way (REQ#2).

If RD-ns is present, then the length of RD-ns and RD-name MUST be less then 63 chars.

Percent encoding MUST NOT be used if not needed, i.e. it can be used only to encode non otherwise allowed chars.

3.2.3. Examples

- eui-64+01-23-45-67-89-ab-cd-ef
- imei+123456789012345
- imei+123456789012345:9876
- uuid+64d5ecfa-addc-4695-ac6e-36e8b18de4b9
- eui-64+01-23-45-67-89-ab-cd-ef.local:1234
- name.domain:1234

3.2.4. Uri-Host and Uri-Port Considerations

When RD-authority is used in a 'coaps' URI, its value is the same as the ServerName.name included (and successfully validated) by the client in the associated DTLS handshake (see Section 3.3).

Hence, there is no need to include explicit Uri-Host and Uri-Port Options in requests associated to the same security context [[CREF1: This updates Sections 6.4 and 6.5 of [RFC7252]]].

If any of Uri-Host or Uri-Port is included in the request, then its value MUST match the corresponding value set in the established security context.

3.3. SNI Name Type and Server Name Syntax

In order to encode RD authorities in a ServerNameList, the extension_data field of the server_name extension is expanded to allow a RDAuthority in a ServerName:

```
struct {
```

```
        NameType name_type;
        select (name_type) {
            case host_name: HostName;
            case rd_authority: RDAuthority;
        } name;
    } ServerName;

    enum {
        host_name(0),
        rd_authority(1),
        (255)
    } NameType;

    opaque RDAuthority<1..2^16-1>;
```

RDAuthority, the data structure associated with the rd_authority NameType, is a variable-length vector that begins with a 16-bit length field indicating the length of the following RD authority. The RD authority is represented as a byte string using ASCII encoding. It MUST NOT contain any percent-encoded character other than for those characters not explicitly allowed by the grammar in Section 3.2.

3.4. New OID arc for CoAP

This OID designates the OID arc for CoAP-related OIDs assigned by future IETF action, including those introduced by the present document:

```
id-coap OBJECT IDENTIFIER ::= { id-pkix coap(TODO) }
```


3.5. OtherName type-id and value Syntax

A X.509 Server Certificate intended to be used for resources served by a RD authority MUST contain an otherName SAN identified using a type-id of 'id-rdauthority-san':

```
id-rdauthority-san OBJECT IDENTIFIER ::= { id-coap 2 }
```

The value field of the otherName MUST contain an RD authority (Section 3.2), encoded as a IA5String.

4. Client Behaviour

1) Send extended ClientHello containing:

   a) server_name extension with one (and one only) ServerName, case-insensitive matching the authority of the URI to be requested;

   b) Any other potentially useful extension, e.g. client_certificate_url;

2) Verify that the intended server name is indeed one of the identities bound to the presented certificate, by checking that the name in the SAN otherName of type id-rdauthority-san case-insensitive matches the authority requested via server_name;

3) Upon receiving the CertificateRequest message, send the certificate via a Certificate message - or CertificateURL message, if the client_certificate_url extension has been successfully negotiated during the "hello" phase;

4) Send ClientKeyExchange and then CertificateVerify to complete the mutual authentication process.

5. Server Behaviour

1) Server receives extended ClientHello carrying a server_name extension, and uses the given server_name (with a rd_authority NameType) to select the appropriate certificate. The selected certificate MUST include a SAN otherName with an id-rdauthority-san type-id and value, which MUST case-insensitive match the requested ServerName;

   a) If no certificate can be selected, the server MUST terminate the handshake by sending a fatal-level unrecognized_name(112) alert. [[CREF2: Prefer a single, hard failure, path over soft failure, or worse: ignoring the error altogether.

Rationale: do not waste time/energy; provide clear and prompt

diagnostic to the peer. It doesn't look like the condition that could be exploited by a timing attack.]]

b) If a matching certificate exist, the server SHALL include an extension of type "server_name" in the (extended) ServerHello message with an empty value.

2) The server MUST send the selected certificate back to the client in the Certificate message.

3) Server MUST then request the client certificate via a CertificateRequest message and conclude its negotiation with a ServerHelloDone message.

4) When server receives the Certificate message from the client then, depending on the specific application security policy, it MAY want to match one of the identities of the client against a configured ACL, and decide whether to continue or to tear down the session [[CREF3: TODO Which alert code to use if ACL check fails?]].

5) The server application running on top of DTLS MUST check the requested URI authority case-insensitive matches the requested server_name.

6. IANA Considerations

[[CREF4: Need to register a few new IDs, not sure where (IANA, PKIX registry, TLS registry)?]]

- id-coap
- OtherName.type-id::id-rdauthority-san
- NameType::rd_authority
- ServerName.name::RDAuthority

7. Security Considerations

It's the responsibility of the CA, by means of its Registration Authority component, to verify the identity of the requester before issuing a new certificate. In particular, the CA MUST ensure that no more than one certificate per SAN is valid at any given time. This should exclude the threat of a (possibly rogue) node to successfully impersonate another node's identity.

Security considerations from Section 11.1 of [RFC6066] fully apply.

8. Acknowledgements

TODO

9. References

9.1. Normative References

[EUI-64] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64)", November 2012.

[I-D.ietf-core-resource-directory] Shelby, Z. and C. Bormann, "CoRE Resource Directory", draft-ietf-core-resource-directory-02 (work in progress), November 2014.

[I-D.ietf-dice-profile] Tschofenig, H., "A Datagram Transport Layer Security (DTLS) 1.2 Profile for the Internet of Things", draft-ietf-dice-profile-04 (work in progress), August 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.

9.2. Informative References

[I-D.iab-smart-object-architecture]
Tschofenig, H., Arkko, J., Thaler, D., and D. McPherson, "Architectural Considerations in Smart Object Networking", draft-iab-smart-object-architecture-06 (work in progress), October 2014.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

Authors' Addresses

Thomas Fossati
Alcatel-Lucent
3 Ely Road
Milton, Cambridge CB24 6DD
Great Britain

Email: thomas.fossati@alcatel-lucent.com

Hannes Tschofenig
ARM Ltd.
110 Fulbourn Rd
Cambridge CB1 9NJ
Great Britain

Email: Hannes.tschofenig@gmx.net
URI: http://www.tschofenig.priv.at