Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 20, 2022

O. Friel
R. Barnes
Cisco
R. Shekh-Yusef
Auth0
M. Richardson
Sandelman Software Works

December 17, 2021

ACME Integrations
draft-ietf-acme-integrations-06

Abstract

This document outlines multiple advanced use cases and integrations that ACME facilitates without any modifications or enhancements required to the base ACME specification. The use cases include ACME integration with EST, BRSKI and TEAP.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

ACME [RFC8555] defines a protocol that a certification authority (CA) and an applicant can use to automate the process of domain name ownership validation and X.509 (PKIX) certificate issuance. The protocol is rich and flexible and enables multiple use cases that are not immediately obvious from reading the specification. This document explicitly outlines multiple advanced ACME use cases including:

- ACME integration with EST [RFC7030]

- ACME integration with BRSKI [RFC8995]

- ACME integration with BRSKI Default Cloud Registrar [I-D.ietf-anima-brski-cloud]

- ACME integration with TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

The integrations with EST, BRSKI (which is based upon EST), and TEAP enable automated certificate enrollment for devices.

ACME for subdomains [I-D.ietf-acme-subdomains] outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from an ACME server where the client has fulfilled a challenge against a parent domain, but does not need to fulfil a challenge against the explicit subdomain. This is a useful optimization when ACME is used to issue certificates for large numbers of devices as it reduces the domain ownership proof traffic (DNS or HTTP) and ACME traffic overhead, but is not a necessary requirement.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in DNS Terminology [RFC8499] and are reproduced here:

o Label: An ordered list of zero or more octets that makes up a portion of a domain name. Using graph theory, a label identifies one node in a portion of the graph of all possible domain names.

o Domain Name: An ordered list of one or more labels.

o Subdomain: "A domain is a subdomain of another domain if it is contained within that domain. This relationship can be tested by seeing if the subdomain's name ends with the containing domain's name." (Quoted from [RFC1034], Section 3.1) For example, in the host name "nnn.mmm.example.com", both "mmm.example.com" and "nnn.mmm.example.com" are subdomains of "example.com". Note that the comparisons here are done on whole labels; that is, "ooo.example.com" is not a subdomain of "oo.example.com".

o Fully-Qualified Domain Name (FQDN): This is often just a clear way of saying the same thing as "domain name of a node", as outlined above. However, the term is ambiguous. Strictly speaking, a fully-qualified domain name would include every label, including the zero-length label of the root: such a name would be written "www.example.net." (note the terminating dot). But, because every name eventually shares the common root, names are often written relative to the root (such as "www.example.net") and are still called "fully qualified". This term first appeared in [RFC0819]. In this document, names are often written relative to the root.

The following terms are used in this document:

- o BRSKI: Bootstrapping Remote Secure Key Infrastructures [RFC8995]

- o Certification Authority (CA): An organization that is responsible for the creation, issuance, revocation, and management of Certificates. The term applies equally to both Roots CAs and Subordinate CAs

- o CMS: Cryptographic Message Syntax [RFC5652]

- o CMC: Certificate Management over CMS [RFC5272]

- o CSR: Certificate Signing Request

- o EST: Enrollment over Secure Transport [RFC7030]

- o MASA: Manufacturer Authorized Signing Authority as defined in [RFC8995]

- o RA: PKI Registration Authority

- o TEAP: Tunneled Extensible Authentication Protocol [RFC7170]

- o TLV: Type-Length-Value format defined in TEAP

3. ACME Integration with EST

EST [RFC7030] defines a mechanism for clients to enroll with a PKI Registration Authority by sending CMC messages over HTTP. EST section 1 states:

"Architecturally, the EST service is located between a Certification Authority (CA) and a client. It performs several functions traditionally allocated to the Registration Authority (RA) role in a PKI."

EST section 1.1 states that:

"For certificate issuing services, the EST CA is reached through the EST server; the CA could be logically "behind" the EST server or embedded within it."

When the CA is logically "behind" the EST RA, EST does not specify how the RA communicates with the CA. EST section 1 states:

"The nature of communication between an EST server and a CA is not described in this document."

This section outlines how ACME could be used for communication between the EST RA and the CA. The example call flow leverages [I-D.ietf-acme-subdomains] and shows the RA proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain. This is an optimization that reduces DNS and ACME traffic overhead. The RA could of course prove ownership of every explicit client certificate identifier. The example also illustrates using the ACME DNS challenge type, but this integration is not limited to DNS challenges.

The call flow illustrates the client calling the EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the client should include in the CSR that the client sends in the /simpleenroll API. CSR Attributes handling are discussed in Section 7.2.

The call flow illustrates the EST RA returning a 202 Retry-After response to the client's simpleenroll request. This is an optional step and may be necessary if the interactions between the RA and the ACME server take some time to complete. The exact details of when the RA returns a 202 Retry-After are implementation specific.

```
+--------+             +--------+          +--------+    +-----+
| Client |             | EST RA |          |  ACME  |    | DNS |
+--------+             +--------+          | Server |    +-----+
    |                       |              +--------+       |
    |                       |                   |           |
               STEP 1: Pre-Authorization of parent domain
    |                       |                   |           |
    |                       | POST /newAuthz    |           |
    |                       |  "example.com"    |           |
    |                       |------------------>|           |
    |                       |                   |           |
    |                       | 201 authorizations|           |
    |                       |<------------------|           |
    |                       |                   |           |
    |                       | Publish DNS TXT   |           |
    |                       | "example.com"     |           |
    |                       |------------------------------>|
    |                       |                   |           |
    |                       | POST /challenge   |           |
    |                       |------------------>|           |
    |                       |                   | Verify    |
    |                       |                   |---------->|
    |                       | 200 status=valid  |           |
    |                       |<------------------|           |
    |                       |                   |           |
    |                       | Delete DNS TXT    |           |
```

```
       |                      | "example.com"        |            |
       |                      |---------------------------------->|
       |                      |                      |            |
                 STEP 2: Client enrolls against RA
       |                      |                      |            |
       | GET /csrattrs        |                      |            |
       |--------------------->|                      |            |
       |                      |                      |            |
       | 200 OK               |                      |            |
       | SEQUENCE {AttrOrOID} |                      |            |
       | SAN OID:             |                      |            |
       | "client.example.com" |                      |            |
       |<---------------------|                      |            |
       |                      |                      |            |
       | POST /simpleenroll   |                      |            |
       | PCSK#10 CSR          |                      |            |
       | "client.example.com" |                      |            |
       |--------------------->|                      |            |
       |                      |                      |            |
       | 202 Retry-After      |                      |            |
       |<---------------------|                      |            |
       |                      |                      |            |
                 STEP 3: RA places ACME order
       |                      |                      |            |
       |                      | POST /newOrder       |            |
       |                      | "client.example.com" |            |
       |                      |--------------------->|            |
       |                      |                      |            |
       |                      | 201 status=ready     |            |
       |                      |<---------------------|            |
       |                      |                      |            |
       |                      | POST /finalize       |            |
       |                      | PKCS#10 CSR          |            |
       |                      | "client.example.com" |            |
       |                      |--------------------->|            |
       |                      |                      |            |
       |                      | 200 OK status=valid  |            |
       |                      |<---------------------|            |
       |                      |                      |            |
       |                      | POST /certificate    |            |
       |                      |--------------------->|            |
       |                      |                      |            |
       |                      | 200 OK               |            |
       |                      | PKCS#7               |            |
       |                      | "client.example.com" |            |
       |                      |<---------------------|            |
       |                      |                      |            |
                 STEP 4: Client retries enroll
```

```
       |                      |                      |           |
       | POST /simpleenroll   |                      |           |
       | PCSK#10 CSR          |                      |           |
       | "client.example.com" |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 200 OK               |                      |           |
       | PKCS#7               |                      |           |
       | "client.example.com" |                      |           |
       |<---------------------|                      |           |
```

4. ACME Integration with BRSKI

BRSKI [RFC8995] is based upon EST [RFC7030] and defines how to autonomically bootstrap PKI trust anchors into devices via means of signed vouchers. EST certificate enrollment may then optionally take place after trust has been established. BRKSI voucher exchange and trust establishment are based on EST extensions and the certificate enrollment part of BRSKI is fully based on EST. Similar to EST, BRSKI does not define how the EST RA communicates with the CA. Therefore, the mechanisms outlined in the previous section for using ACME as the communications protocol between the EST RA and the CA are equally applicable to BRSKI.

The following call flow shows how ACME may be integrated into a full BRSKI voucher plus EST enrollment workflow. For brevity, it assumes that the EST RA has previously proven ownership of a parent domain and that pledge certificate identifiers are a subdomain of that parent domain. The domain ownership exchanges between the RA, ACME and DNS are not shown. Similarly, not all BRSKI interactions are shown and only the key protocol flows involving voucher exchange and EST enrollment are shown.

Similar to the EST section above, the client calls EST /csrattrs API before calling the EST /simpleenroll API. This enables the server to indicate what fields the pledge should include in the CSR that the client sends in the /simpleenroll API. Refer to section {csr-attributes} for more details.

The call flow illustrates the RA returning a 202 Retry-After response to the initial EST /simpleenroll API. This may be appropriate if processing of the /simpleenroll request and ACME interactions takes some time to complete.

```
   +--------+             +--------+             +--------+     +------+
   | Pledge |             | EST RA |             |  ACME  |     | MASA |
   +--------+             +--------+             | Server |     +------+
       |                      |                  +--------+        |
```

```
       |                      |                      |           |
                  NOTE: Pre-Authorization of "example.com" is complete
       |                      |                      |           |
                  STEP 1: Pledge requests Voucher
       |                      |                      |           |
       | POST /requestvoucher |                      |           |
       |--------------------->|                      |           |
       |                      | POST /requestvoucher |           |
       |                      |--------------------------------->|
       |                      |                      |           |
       |                      | 200 OK Voucher       |           |
       |                      |<---------------------------------|
       | 200 OK Voucher       |                      |           |
       |<---------------------|                      |           |
       |                      |                      |           |
                  STEP 2: Pledge enrolls against RA
       |                      |                      |           |
       | GET /csrattrs        |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 200 OK               |                      |           |
       | SAN:                 |                      |           |
       | "pledge.example.com" |                      |           |
       |<---------------------|                      |           |
       |                      |                      |           |
       | POST /simpleenroll   |                      |           |
       | PCSK#10 CSR          |                      |           |
       | "pledge.example.com" |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 202 Retry-After      |                      |           |
       |<---------------------|                      |           |
       |                      |                      |           |
                  STEP 3: RA places ACME order
       |                      |                      |           |
       |                      | POST /newOrder       |           |
       |                      | "pledge.example.com" |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 201 status=ready     |           |
       |                      |<---------------------|           |
       |                      |                      |           |
       |                      | POST /finalize       |           |
       |                      | PKCS#10 CSR          |           |
       |                      | "pledge.example.com" |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 200 OK status=valid  |           |
```

```
       |                      |<---------------------|           |
       |                      |                      |           |
       |                      | POST /certificate    |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 200 OK               |           |
       |                      | PKCS#7               |           |
       |                      | "pledge.example.com" |           |
       |                      |<---------------------|           |
       |                      |                      |           |
                STEP 4: Pledge retries enroll
       |                      |                      |           |
       | POST /simpleenroll   |                      |           |
       | PCSK#10 CSR          |                      |           |
       | "pledge.example.com" |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 200 OK               |                      |           |
       | PKCS#7               |                      |           |
       | "pledge.example.com" |                      |           |
       |<---------------------|                      |           |
```

5. ACME Integration with BRSKI Default Cloud Registrar

BRSKI Cloud Registrar [I-D.ietf-anima-brski-cloud] specifies the behaviour of a BRSKI Cloud Registrar, and how a pledge can interact with a BRSKI Cloud Registrar when bootstrapping. Similar to the local domain registrar BRSKI flow, ACME can be easily integrated with a cloud registrar bootstrap flow.

BRSKI cloud registrar is flexible and allows for multiple different local domain discovery and redirect scenarios. In the example illustrated here, the extension to [RFC8366] Vouchers which is defined in [I-D.ietf-anima-brski-cloud], and allows the specification of a bootstrap EST domain, is leveraged. This extension allows the cloud registrar to specify the local domain RA that the pledge should connect to for the purposes of EST enrollment.

Similar to the sections above, the client calls EST /csrattrs API before calling the EST /simpleenroll API.

```
+--------+             +--------+          +--------+    +----------+
| Pledge |             | EST RA |          |  ACME  |    | Cloud RA |
+--------+             +--------+          | Server |    |  / MASA  |
    |                                      +--------+    +----------+
    |                                                          |
         NOTE: Pre-Authorization of "example.com" is complete
    |                                                          |
```

```
            STEP 1: Pledge requests Voucher from Cloud Registrar
       |                                                         |
       | POST /requestvoucher                                    |
       |-------------------------------------------------------->|
       |                                                         |
       | 200 OK Voucher (includes 'est-domain')                  |
       |<--------------------------------------------------------|
       |                      |                      |           |
            STEP 2: Pledge enrolls against local domain RA
       |                      |                      |           |
       | GET /csrattrs        |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 200 OK               |                      |           |
       | SAN:                 |                      |           |
       | "pledge.example.com" |                      |           |
       |<---------------------|                      |           |
       |                      |                      |           |
       | POST /simpleenroll   |                      |           |
       | PCSK#10 CSR          |                      |           |
       | "pledge.example.com" |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 202 Retry-After      |                      |           |
       |<---------------------|                      |           |
       |                      |                      |           |
            STEP 3: RA places ACME order
       |                      |                      |           |
       |                      | POST /newOrder       |           |
       |                      | "pledge.example.com" |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 201 status=ready     |           |
       |                      |<---------------------|           |
       |                      |                      |           |
       |                      | POST /finalize       |           |
       |                      | PKCS#10 CSR          |           |
       |                      | "pledge.example.com" |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 200 OK status=valid  |           |
       |                      |<---------------------|           |
       |                      |                      |           |
       |                      | POST /certificate    |           |
       |                      |--------------------->|           |
       |                      |                      |           |
       |                      | 200 OK               |           |
       |                      | PKCS#7               |           |
```

```
       |                      | "pledge.example.com" |           |
       |                      |<---------------------|           |
       |                      |                      |           |
            STEP 4: Pledge retries enroll
       |                      |                      |           |
       | POST /simpleenroll   |                      |           |
       | PCSK#10 CSR          |                      |           |
       | "pledge.example.com" |                      |           |
       |--------------------->|                      |           |
       |                      |                      |           |
       | 200 OK               |                      |           |
       | PKCS#7               |                      |           |
       | "pledge.example.com" |                      |           |
       |<---------------------|                      |           |
```

6. ACME Integration with TEAP

TEAP [RFC7170] defines a tunnel-based EAP method that enables secure communication between a peer and a server by using TLS to establish a mutually authenticated tunnel. TEAP enables certificate provisioning within the tunnel. TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] defines extensions to TEAP that includes additional Type-Length-Value (TLV) elements for certificate enrollment and BRSKI handling inside the TEAP tunnel. Neither TEAP [RFC7170] or TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski] define how the TEAP server communicates with the CA.

This section outlines how ACME could be used for communication between the TEAP server and the CA. The example call flow leverages [I-D.ietf-acme-subdomains] and shows the TEAP server proving ownership of a parent domain, with individual client certificates being subdomains under that parent domain.

The example illustrates the TEAP server sending a Request-Action TLV including a CSR-Attributes TLV instructing the peer to send a CSR-Attributes TLV to the server. This enables the server to indicate what fields the peer should include in the CSR that the peer sends in the PKCS#10 TLV.

Although not explicitly illustrated in this call flow, the Peer and TEAP Server could exchange BRSKI TLVs, and a BRSKI integration and voucher exchange with a MASA server could take place over TEAP. Whether a BRSKI TLV exchange takes place or not does not impact the ACME specific message exchanges.

```
+------+                +-------------+            +--------+   +-----+
| Peer |                | TEAP-Server |            |  ACME  |   | DNS |
```

```
   +------+                +-------------+          | Server |   +-----+
       |                          |                 +--------|       |
       |                          |                       |           |
                 STEP 1: Pre-Authorization of parent domain
       |                          |                       |           |
       |                          | POST /newAuthz        |           |
       |                          |  "example.com"        |           |
       |                          |---------------------->|           |
       |                          |                       |           |
       |                          | 201 authorizations    |           |
       |                          |<----------------------|           |
       |                          |                       |           |
       |                          | Publish DNS TXT       |           |
       |                          | "example.com"         |           |
       |                          |---------------------------------->|
       |                          |                       |           |
       |                          | POST /challenge       |           |
       |                          |---------------------->|           |
       |                          |                       | Verify    |
       |                          |                       |---------->|
       |                          | 200 status=valid      |           |
       |                          |<----------------------|           |
       |                          |                       |           |
       |                          | Delete DNS TXT        |           |
       |                          | "example.com"         |           |
       |                          |---------------------------------->|
       |                          |                       |           |
       |                          |                       |           |
                 STEP 2: Establsh EAP Outer Tunnel
       |                          |                       |           |
       |  EAP-Request/            |                       |           |
       |   Type=Identity          |                       |           |
       |<-------------------------|                       |           |
       |                          |                       |           |
       |  EAP-Response/           |                       |           |
       |   Type=Identity          |                       |           |
       |------------------------->|                       |           |
       |                          |                       |           |
       |  EAP-Request/            |                       |           |
       |   Type=TEAP,             |                       |           |
       |   TEAP Start,            |                       |           |
       |   Authority-ID TLV       |                       |           |
       |<-------------------------|                       |           |
       |                          |                       |           |
       |  EAP-Response/           |                       |           |
       |   Type=TEAP,             |                       |           |
       |   TLS(ClientHello)       |                       |           |
       |------------------------->|                       |           |
```

```
       |                          |                       |            |
       |  EAP-Request/            |                       |            |
       |   Type=TEAP,             |                       |            |
       |   TLS(ServerHello,       |                       |            |
       |   Certificate,           |                       |            |
       |   ServerKeyExchange,     |                       |            |
       |   CertificateRequest,    |                       |            |
       |   ServerHelloDone)       |                       |            |
       |<-------------------------|                       |            |
       |                          |                       |            |
       |  EAP-Response/           |                       |            |
       |   Type=TEAP,             |                       |            |
       |   TLS(Certificate,       |                       |            |
       |   ClientKeyExchange,     |                       |            |
       |   CertificateVerify,     |                       |            |
       |   ChangeCipherSpec,      |                       |            |
       |   Finished)              |                       |            |
       |------------------------->|                       |            |
       |                          |                       |            |
       |  EAP-Request/            |                       |            |
       |   Type=TEAP,             |                       |            |
       |   TLS(ChangeCipherSpec,  |                       |            |
       |   Finished),             |                       |            |
       |   {Crypto-Binding TLV,   |                       |            |
       |   Result TLV=Success}    |                       |            |
       |<-------------------------|                       |            |
       |                          |                       |            |
       |  EAP-Response/           |                       |            |
       |   Type=TEAP,             |                       |            |
       |   {Crypto-Binding TLV,   |                       |            |
       |   Result TLV=Success}    |                       |            |
       |------------------------->|                       |            |
       |                          |                       |            |
       |  EAP-Request/            |                       |            |
       |   Type=TEAP,             |                       |            |
       |   {Request-Action TLV:   |                       |            |
       |     Status=Failure,      |                       |            |
       |     Action=Process-TLV,  |                       |            |
       |     TLV=CSR-Attributes,  |                       |            |
       |     TLV=PKCS#10}         |                       |            |
       |<-------------------------|                       |            |
       |                          |                       |            |
               STEP 3: Enroll for certificate
       |                          |                       |            |
       |  EAP-Response/           |                       |            |
       |   Type=TEAP,             |                       |            |
       |   {CSR-Attributes TLV}   |                       |            |
       |------------------------->|                       |            |
```

```
       |                         |                      |           |
       |  EAP-Request/           |                      |           |
       |   Type=TEAP,            |                      |           |
       |   {CSR-Attributes TLV}  |                      |           |
       |<------------------------|                      |           |
       |                         |                      |           |
       |  EAP-Response/          |                      |           |
       |   Type=TEAP,            |                      |           |
       |   {PKCS#10 TLV:         |                      |           |
       |   "client.example.com"} |                      |           |
       |------------------------>|                      |           |
       |                         | POST /newOrder       |           |
       |                         | "client.example.com" |           |
       |                         |--------------------->|           |
       |                         |                      |           |
       |                         | 201 status=ready     |           |
       |                         |<---------------------|           |
       |                         |                      |           |
       |                         | POST /finalize       |           |
       |                         | PKCS#10 CSR          |           |
       |                         | "client.example.com" |           |
       |                         |--------------------->|           |
       |                         |                      |           |
       |                         | 200 OK status=valid  |           |
       |                         |<---------------------|           |
       |                         |                      |           |
       |                         | POST /certificate    |           |
       |                         |--------------------->|           |
       |                         |                      |           |
       |                         | 200 OK               |           |
       |                         | PKCS#7               |           |
       |                         | "client.example.com" |           |
       |                         |<---------------------|           |
       |                         |                      |           |
       |  EAP-Request/           |                      |           |
       |   Type=TEAP,            |                      |           |
       |   {PKCS#7 TLV,          |                      |           |
       |    Result TLV=Success}  |                      |           |
       |<------------------------|                      |           |
       |                         |                      |           |
       |  EAP-Response/          |                      |           |
       |   Type=TEAP,            |                      |           |
       |   {Result TLV=Success}  |                      |           |
       |------------------------>|                      |           |
       |                         |                      |           |
       |  EAP-Success            |                      |           |
       |<------------------------|                      |           |
```

7. ACME Integration Considerations

7.1. Service Operators

The goal of these integrations is enabling issuance of certificates with identifiers in a given domain by an ACME server to a client. It is expected that the EST RA or TEAP servers that the client sends certificate enrollment requests to are operated by the organization that controls the domains. The ACME server is not necessarily operated by the organization that controls the domain.

7.2. CSR Attributes

In all integrations, the client MUST send a CSR Attributes request to the EST or TEAP server prior to sending a certificate enrollment request. This enables the server to indicate to the client what attributes, and what attribute values, it expects the client to include in the subsequent CSR request. For example, the server could instruct the peer what Subject Alternative Name entries to include in its CSR.

EST [RFC7030] is not clear on how the CSR Attributes response should be structured, and in particular is not clear on how a server can instruct a client to include specific attribute values in its CSR. [I-D.richardson-lamps-rfc7030-csrattrs] clarifies how a server can use CSR Attributes response to specify specific values for attributes that the client should include in its CSR.

Servers MUST use this mechanism to tell the client what identifiers to include in CSR request. ACME [RFC8555] allows the identifier to be included in either CSR Subject or Subject Alternative Name fields, however [I-D.ietf-uta-use-san] states that Subject Alternative Name field MUST be used. This document aligns with [I-D.ietf-uta-use-san] and Subject Alternate Name field MUST be used. The identifier must be a subdomain of a domain that the server has control over and can fulfill ACME challenges against. The leftmost part of the identifier MAY be a field that the client presented to the server in an IEEE 802.1AR [IDevID].

Servers MAY use this field to instruct the client to include other attributes such as specific policy OIDs. Refer to EST [RFC7030] section 2.6 for further details.

7.3. Certificate Chains and Trust Anchors

ACME [RFC8555] section 9.1 states that ACME servers may return a certificate chain to an ACME client where an end entity certificate is followed by certificates that certify it. The trust anchor

certificate MAY be omitted from the chain as it is assumed that the trust anchor is already known by the ACME client i.e. the EST or TEAP server.

7.3.1. EST /cacerts

EST [RFC7030] section 4.2.3 states that the /simpleenroll response contains "only the certificate that was issued". EST [RFC7030] section 4.1.3 states that the /cacerts response "MUST include any additional certificates the client would need to build a chain from an EST CA-issued certificate to the current EST CA TA".

Therefore, the EST server MUST return only the ACME end entity certificate in the /simpleenroll response. The EST server MUST return the remainder of the chain returned by the ACME server to the EST server in the /cacerts response to the client, appending the trust anchor root CA if necessary.

7.3.2. TEAP PKCS#7 TLV

TEAP [RFC7170] section 4.2.16 allows for download of a PKCS#7 certificate chain in response to a TEAP PKCS#10 TLV request. TEAP also allows for download of multiple PKCS#7 certificates in response to a TEAP Trusted-Server-Root TLV request.

The TEAP server MUST return the full ACME client certificate chain in the PKCS#7 response to the PKCS#10 TLV request. The TEAP server MUST return the ACME server trust anchor in a PKCS#7 response to a Trusted-Server-Root TLV request. As outlined in Section 7.4, the TEAP server SHOULD also return the trust anchor that was used for issuing its own identity certificate, if different from the ACME server trust anchor.

7.4. id-kp-cmcRA

BRSKI [RFC8995] mandates that the id-kp-cmcRA extended key usage bit is set in the Registrar (or EST RA) end entity certificate that the Registrar uses when signing voucher request messages sent to the MASA. Public ACME servers may not be willing to issue end entity certificates that have the id-kp-cmcRA extended key usage bit set. In these scenarios, the EST RA may be used by the pledge to get issued certificates by a public ACME server, but the EST RA itself will need an end entity certificate that has been issued by a different CA (e.g. an operator deployed private CA) and that has the id-kp-cmcRA bit set.

7.5. Error Handling

ACME [RFC8555] section 6.7 defines multiple errors that may be returned by an ACME server to an ACME client. TEAP [RFC7170] section 4.2.6 defines multiple errors that may be returned by a TEAP server to a client in an Error TLV. EST [RFC7030] section 4.2.3 defines how an EST server may return an error encoded in a CMC response, or may return a human readable error in the response body.

The following mapping from ACME errors to CMC [RFC5272] section 6.1.4 CMCFailInfo and TEAP [RFC7170] section 4.2.6 error codes is RECOMMENDED.

ACME	CMCFailInfo	TEAP Error Code
badCSR	badRequest	1025 Bad CSR
caa	badRequest	1025 Bad CSR
rejectedIdentifier	badIdentity	1024 Bad Identity In CSR
all other errors	internalCAError	1026 Internal CA Error

8. IANA Considerations

This document does not make any requests to IANA.

9. Security Considerations

This draft is informational and makes no changes to the referenced specifications. All security considerations from these referenced documents are applicable here:

- o EST [RFC7030]
- o BRSKI [RFC8995]
- o BRSKI Default Cloud Registrar [I-D.ietf-anima-brski-cloud]
- o TEAP [RFC7170] and TEAP Update and Extensions for Bootstrapping [I-D.lear-eap-teap-brski]

Additionally, all Security Considerations in ACME in the following areas are equally applicable to ACME Integrations.

It is expected that the integration mechanisms proposed here will primarily use the DNS-01 challenge documented in [RFC8555] section 8.4. The security considerations in RFC8555 says:

The DNS is a common point of vulnerability for all of these challenges. An entity that can provision false DNS records for a domain can attack the DNS challenge directly and can provision false A/AAAA records to direct the ACME server to send its HTTP validation query to a remote server of the attacker's choosing.

It is expected that the TEAP-EAP server/EST Registrar will perform DNS dynamic updates to a DNS primary server using [RFC3007] Dynamic updates, secured with either SIG(0), or TSIG keys.

A major source of vulnerability is the disclosure of these DNS key records. An attacker that has access to them, can provision their own certificates into the the name space of the entity.

For many uses, this may allow the attacker to get access to some enterprise resource. When used to provision, for instance, a (SIP) phone system this would permit an attacker to impersonate a legitimate phone. Not only does this allow for redirection of phone calls, but possibly also toll fraud.

Operators should consider restricting the integration server such that it can only update the DNS records for a specific zone or zones where ACME is required for client certificate enrollment automation. For example, if all IoT devices in an organisation enroll using EST against an EST RA, and all IoT devices will be issued certificates in a subdomain under iot.example.com, then the integration server could be issued a credential that only allows updating of DNS records in a zone that includes domains in the iot.example.com namespace, but does not allow updating of DNS records under any other example.com DNS namespace.

When performing challenge fulfilment via writing files to HTTP webservers, write access should only be granted to a specific set of servers, and only to a specific set of directories for storage of challenge files.

9.1. Denial of Service against ACME infrastructure

The intermediate node (the TEAP-EAP server, or the EST Registrar) should cache the resulting certificates such that if the communication with the pledge is lost, subsequent attempts to enroll will result in the cache certificate being returned.

As many ACME servers have per-day, per-IP and per-subjectAltName limits, it is prudent not to request identical certificates too often. This could be due to operator or installer error, with multiple configuration resets occurring within a short period of time.

The cache should be indexed by the complete contents of the Certificate Signing Request, and should not persist beyond the notAfter date in the certificate.

This means that if the private/public keypair changes on the pledge, then a new certificate will be issued. If the requested SubjectAltName changes, then a new certificate will be requested.

In a case where a device is simply factory reset, and enrolls again, then the same certificate can be returned.

10. Informative References

[CAB] CA/Browser Forum, "Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates", n.d., <https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.7.1.pdf>.

[I-D.ietf-acme-subdomains]
Friel, O., Barnes, R., Hollebeek, T., and M. Richardson, "ACME for Subdomains", draft-ietf-acme-subdomains-00 (work in progress), October 2021.

[I-D.ietf-anima-brski-cloud]
Friel, O., Shekh-Yusef, R., and M. Richardson, "BRSKI Cloud Registrar", draft-ietf-anima-brski-cloud-02 (work in progress), October 2021.

[I-D.ietf-uta-use-san]
Salz, R., "Update to Verifying TLS Server Identities with X.509 Certificates", draft-ietf-uta-use-san-00 (work in progress), April 2021.

[I-D.lear-eap-teap-brski]
Lear, E., Friel, O., Cam-Winget, N., and D. Harkins, "TEAP Update and Extensions for Bootstrapping", draft-lear-eap-teap-brski-06 (work in progress), August 2021.

[I-D.richardson-lamps-rfc7030-csrattrs]
Richardson, M., Harkins, D., Oheimb, D. D. V., and O. Friel, "Clarification of RFC7030 CSR Attributes definition", draft-richardson-lamps-rfc7030-csrattrs-01 (work in progress), October 2021.

[IDevID] IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", n.d., <https://1.ieee802.org/security/802-1ar>.

[RFC0819] Su, Z. and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC 819, DOI 10.17487/RFC0819, August 1982, <https://www.rfc-editor.org/info/rfc819>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[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>.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <https://www.rfc-editor.org/info/rfc3007>.

[RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <https://www.rfc-editor.org/info/rfc5272>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[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>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/info/rfc7170>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/info/rfc8995>.

Authors' Addresses

Owen Friel
Cisco

Email: ofriel@cisco.com

Richard Barnes
Cisco

Email: rlb@ipv.sx

Rifaat Shekh-Yusef
Auth0

Email: rifaat.s.ietf@gmail.com

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca