Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2020

O. Friel
R. Barnes
Cisco
T. Hollebeek
DigiCert

October 23, 2019

ACME for Subdomains
draft-friel-acme-subdomains-00

Abstract

This document outlines how ACME can be used by a client to obtain a certificate for a subdomain identifier from a certificate authority. The client has fulfilled a challenge against a parent domain but does not need to fulfil a challenge against the explicit subdomain as certificate authority policy allows issuance of the subdomain certificate without explicit subdomain ownership proof.

Status of This Memo

This Internet-Draft will expire on April 25, 2020.

Copyright Notice

Table of Contents

1. Introduction

ACME [RFC8555] defines a protocol that a certificate 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 how ACME can be used to issue subdomain certificates, without requiring the ACME client to explicitly fulfil an ownership challenge against the subdomain identifiers - the ACME client need only fulfil an ownership challenge against a parent domain identifier.

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 used in this document:

- CA: Certificate Authority

- CSR: Certificate Signing Request

- FQDN: Fully Qualified Domain Name

3. ACME Workflow and Identifier Requirements

A typical ACME workflow for issuance of certificates is as follows:

1. client POSTs a newOrder request that contains a set of "identifiers"

2. server replies with a set of "authorizations" and a "finalize" URI

3. client sends POST-as-GET requests to retrieve the "authorizations", with the downloaded "authorization" object(s) containing the "identifier" that the client must prove control of

4. client proves control over the "identifier" in the "authorization" object by completing the specified challenge, for example, by publishing a DNS TXT record

5. client POSTs a CSR to the "finalize" API

6. server replies with an updated order object that includes a "certificate" URI

7. client sends POST-as-GET request to the "certificate" URI to download the certificate

ACME places the following restrictions on "identifiers":

- section 7.1.4: the only type of "identifier" defined by the ACME specification is a fully qualified domain name: "The only type of identifier defined by this specification is a fully qualified domain name (type: "dns"). The domain name MUST be encoded in the form in which it would appear in a certificate."

- Section 7.4: the "identifier" in the CSR request must match the "identifier" in the newOrder request: "The CSR MUST indicate the exact same set of requested identifiers as the initial newOrder request."

- Sections 8.3: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via HTTP: "Construct a URL by populating the URL template ... where the domain field is set to the domain name being verified"

- Section 8.4: the "identifier", or FQDN, in the "authorization" object must be used when fulfilling challenges via DNS: "The client constructs the validation domain name by prepending the label "_acme-challenge" to the domain name being validated."

ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects.

4. ACME Issuance of Subdomain Certificates

As noted in the previous section, ACME does not mandate that the "identifier" in a newOrder request matches the "identifier" in "authorization" objects. This means that the ACME specification does not preclude an ACME server processing newOrder requests and issuing certificates for a subdomain without requiring a challenge to be fulfilled against that explicit subdomain. ACME server policy could allow issuance of certificates for a subdomain to a client where the client only has to fulfil an authorization challenge for the parent domain. The relevant sections from current CA/Browser baseline requirements are given in section Appendix A.

This allows a flow where a client proves ownership of, for example, "example.com" and then successfully obtains a certificate for "sub.example.com". The ACME pre-authorization flow makes most sense for this use case, and that is what is illustrated in the following call flow.

The client could pre-authorize for the parent domain once, and then issue multiple newOrder requests for certificates for multiple subdomains. This call flow illustrates the client only placing one newOrder request.

The call flow illustrates the DNS-based proof of ownership mechanism, but the subdomain workflow is equally valid for HTTP based proof of ownership.

```
   +--------+             +------+     +-----+
   | Client |             | ACME |     | DNS |
   +--------+             +------+     +-----+
       |                      |            |
    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: Place order for subdomain
       |                      |           |
       | POST /newOrder       |           |
       | "sub.example.com"    |           |
       |--------------------->|           |
       |                      |           |
       | 201 status=ready     |           |
       |<---------------------|           |
       |                      |           |
       | POST /finalize       |           |
       | CSR "sub.example.com"|           |
       |--------------------->|           |
       |                      |           |
       | 200 OK status=valid  |           |
       |<---------------------|           |
       |                      |           |
       | POST /certificate    |           |
       |--------------------->|           |
       |                      |           |
       | 200 OK               |           |
       | PKI "sub.example.com"|           |
       |<---------------------|           |
```

4.1. newOrder and newAuthz Handling

Servers may consider validation of a parent domain sufficient authorization for a subdomain. If a server has such a policy and a client is already authorized for the parent domain then:

- If the client submits a newAuthz request for a subdomain: The server MUST return status 200 (OK) response. The response body is the existing authorization object for the parent domain with status set to "valid".

- If the client submits a newOrder request for a subdomain: The server MUST return a 201 (Created) response. The response body is

an order object with status set to "ready" and links to the unexpired authorizations against the parent domain.

If a server has such a policy and a client is not authorized for the parent domain then:

- o If the client submits a newAuthz request for a subdomain: The server MUST return a status 201 (Created) response. The response body is a newly created authorization object for the parent domain with status set to "pending".

- o If the client submits a newOrder request for a subdomain: The server MUST return a status 201 (Created) response. The response body is an order object with status set to "pending" and links to newly created authorizations objects against the parent domain.

[[TODO: This section documents a change from RFC8555, which states that the identifier in the newAuthz request MUST match that in the authorization object.

Additionally, 200 response code is used here in one scenario instead of a 201 response. However, this is arguably an under-specification in RFC8555, and has been reported in https://www.rfc-editor.org/errata/eid5861.

These two items need a review.]]

4.2. Examples

In order to illustrate subdomain behaviour, let us assume that a client wishes to get certificates for subdomain identifiers "sub0.example.com", "sub1.example.com" and "sub2.example.com" under parent domain "example.com", and CA policy allows certificate issuance of these subdomain identifiers while only requiring the client to fulfil an ownership challenge for parent domain "example.com". Let us also assume that the client has not yet proven ownership of parent domain "example.com".

1. The client POSTs a newOrder request for identifier "sub0.example.com"

The server creates an authorization object for identifier "example.com". The server replies with a 201 (Created) response. The response body is an order object with status set to "pending" and a link to newly created authorization object against the parent domain "example.com". Therefore, the server is instructing the client to fulfil a challenge against domain identifier "example.com"

in order to obtain a certificate including identifier "sub0.example.com".

The client completes the challenge for "example.com", POSTs a CSR to the order finalize URI, and downloads the certificate.

1. The client POSTs a newOrder request for identifier "sub1.example.com"

The server replies with a 201 (Created) response. The response body is an order object with status set to "ready" and a link to the unexpired authorization against the parent domain "example.com".

The client POSTs a CSR to the order finalize URI, and downloads the certificate.

1. The client POSTs a newAuthz request for identifier "sub2.example.com"

The server replies with a 200 (OK) response. The response body is the previously created authorization object for "example.com" with status set to "valid".

5. Directory Object Metadata Fields Registry

[[TODO: is this required?]]

An ACME server can advertise support of issuance of subdomain certificates by including the boolean field "implicitSubdomainAuthorization" in its "ACME Directory Metadata Fields" registry. If not specified, then no default value is assumed. If an ACME server supports issuance of subdomain certificates, it can indicate this by including this field with a value of "true".

Field Name	Field Type	Reference
implicitSubdomainAuthorization	boolean	RFC XXXX

6. IANA Considerations

[[TODO: register implicitSubdomainAuthorization?]]

7. Security Considerations

[[TODO]]

8. Informative References

[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>.

[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>.

[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>.

Appendix A. CA Browser Forum Baseline Requirements

The CA/Browser Forum Baseline Requirements version 1.6.5 states:

- Section: "1.6.1 Definitions": Authorization Domain Name: The Domain Name used to obtain authorization for certificate issuance for a given FQDN. The CA may use the FQDN returned from a DNS CNAME lookup as the FQDN for the purposes of domain validation. If the FQDN contains a wildcard character, then the CA MUST remove all wildcard labels from the left most portion of requested FQDN. The CA may prune zero or more labels from left to right until encountering a Base Domain Name and may use any one of the intermediate values for the purpose of domain validation.

- Section: "3.2.2.4.7 DNS Change": Once the FQDN has been validated using this method, the CA MAY also issue Certificates for other FQDNs that end with all the labels of the validated FQDN. This method is suitable for validating Wildcard Domain Names.

Authors' Addresses

Owen Friel
Cisco

Email: ofriel@cisco.com

Richard Barnes
Cisco

Email: rlb@ipv.sx

Tim Hollebeek
DigiCert

Email: tim.hollebeek@digicert.com