Workgroup: ACME Working Group
Internet-Draft: draft-aaron-acme-ari-01
Published: 8 November 2021
Intended Status: Standards Track
Expires: 12 May 2022
A A. Gable
u Internet Security Research Group
t
h
o
r
s
:

Automated Certificate Management Environment (ACME) Renewal Information (ARI) Extension

Abstract

This document specifies how an ACME server may provide hints to ACME clients as to when they should attempt to renew their certificates. This allows servers to mitigate load spikes, and ensures clients do not make false assumptions about appropriate certificate renewal periods.

Current Implementations

Draft note: this section will be removed by the editor before final publication.

Let's Encrypt's Staging environment (available at [lestaging], source at [boulder]) implements this draft specification.

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 https://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 12 May 2022.

Table of Contents

1. Introduction

Most ACME [RFC8555] clients today choose when to attempt to renew a certificate in one of three ways. They may be configured to renew at a specific interval (e.g. via cron); they may parse the issued certificate to determine its expiration date and renew a specific amount of time before then; or they may parse the issued certificate and renew when some percentage of its validity period has passed. The first two techniques create significant barriers against the issuing CA changing certificate lifetimes. All three techniques lead to load clustering for the issuing CA.

Being able to indicate to the client a period in which the issuing CA suggests renewal would allow both dynamic changes to the certificate validity period and proactive smearing of load. This document specifies a mechanism by which ACME servers may provide suggested renewal windows to ACME clients.

2. Conventions and Definitions

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.

3. Extensions to the ACME Protocol: The "directory" Resource

An ACME server which wishes to provide renewal information **MUST** include a new field, renewalInfo, in its directory object.

Field	URL in Value
renewalInfo	Renewal info

Table 1

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "newNonce": "https://example.com/acme/new-nonce",
  "newAccount": "https://example.com/acme/new-account",
  "newOrder": "https://example.com/acme/new-order",
  "newAuthz": "https://example.com/acme/new-authz",
  "revokeCert": "https://example.com/acme/revoke-cert",
  "keyChange": "https://example.com/acme/key-change",
  "renewalInfo": "https://example.com/acme/renewal-info",
  "meta": {
    "termsOfService": "https://example.com/acme/terms/2021-10-05",
    "website": "https://www.example.com/",
    "caaIdentities": ["example.com"],
    "externalAccountRequired": false
  }
}
```

4. Extensions to the ACME Protocol: The "renewalInfo" Resource

We define a new resource type, the "renewalInfo" resource, as part of the ACME protocol. To request the suggested renewal information for a certificate, the client sends a GET request to a path under the server's renewalInfo URL.

The full request URL is computed by concatenating the renewalInfo URL from the server's directory with the following case-insensitive hex-encoded (see [RFC4648], Section [RFC4648]) elements, separated by forward slashes:

- the SHA-1 hash of the issuer's public key (often included in the certificate as the Authority Key Identifier, see [RFC5280], Section [RFC5280]),

- the SHA-1 hash of the issuer's Distinguished Name, see [RFC5280], Section [RFC5280], and

- the certificate serial number.

These are the same components that make up the CertID sequence of an OCSPRequest [RFC6960], Section [RFC6960], with the caveat that the hash algorithm is restricted to SHA-1, in line with [RFC5019].

```
GET https://example.com/acme/renewal-info
        /254581685026383D3B2D2CBECD6AD9B63DB36663
        /06FE0BABD8E6746EFCC4730285F7A9487ED1344F
        /BCDF4596B6BDC523
```

The structure of an ACME renewalInfo resource is as follows:

suggestedWindow (object, required): A JSON object with two keys, "start" and "end", whose values are timestamps, encoded in the format specified in [RFC3339], which bound the window of time in which the CA recommends renewing the certificate.

```
HTTP/1.1 200 OK
Content-Type: application/json
Retry-After: "21600"

{
  "suggestedWindow": {
    "start": "2021-01-03T00:00:00Z",
    "end": "2021-01-07T00:00:00Z"
  }
}
```

The server **SHOULD** include a Retry-After header indicating the polling interval that the ACME server recommends. Conforming clients **SHOULD** query the renewalInfo URL again after the Retry-After period has passed, as the server may provide a different suggestedWindow.

Conforming clients **MUST** select a uniform random time within the suggested window to attempt to renew the certificate. If the selected time is in the past, the client **SHOULD** attempt renewal immediately. If the selected time is in the future, but before the next time that the client would wake up normally, the client **MAY** attempt renewal immediately. In all cases, renewal attempts are subject to the client's existing error backoff and retry intervals.

In particular, cron-based clients may find they need to increase their run frequency to check ARI more frequently. Those clients will need to store information about failures so that increasing their run frequency doesn't lead to retrying failures without proper backoff. Typical information stored should include: number of failures for a given order (defined by the set of names on the order), and time of the most recent failure.

If the client receives no response or a malformed response (e.g. an end timestamp which precedes the start timestamp), it **SHOULD** make its own determination of when to renew the certificate, and **MAY** retry the renewalInfo request with appropriate exponential backoff behavior.

5. Security Considerations

The extensions to the ACME protocol described in this document build upon the Security Considerations and threat model defined in [RFC8555], Section [RFC8555].

This document specifies that renewalInfo resources **MUST** be exposed and accessed via unauthenticated GET requests, a departure from RFC8555's requirement that clients must send POST-as-GET requests to fetch resources from the server. This is because the information contained in renewalInfo resources is not considered confidential,

and because allowing renewalInfo to be easily cached is advantageous to shed load from clients which do not respect the Retry-After header.

6. IANA Considerations

Draft note: The following changes to IANA registries have not yet been made.

6.1. New Registries

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, IANA has created the new "ACME Renewal Info Object Fields" registry (Section 6.4).

6.2. ACME Resource Type

Within the "Automated Certificate Management Environment (ACME) Protocol" registry, the following entry has been added to the "ACME Resource Types" registry.

Field Name	Resource Type	Reference
renewalInfo	Renewal Info object	This draft

Table 2

6.3. ACME Renewal Info Object Fields

The "ACME Renewal Info Object Fields" registry lists field names that are defined for use in ACME renewal info objects.

Template:

- Field name: The string to be used as a field name in the JSON object
- Field type: The type of value to be provided, e.g., string, boolean, array of string
- Reference: Where this field is defined

Initial contents:

Field Name	Field type	Reference
suggestedWindow	object	This draft

Table 3

7. Normative 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>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC5019] Deacon, A. and R. Hurst, "The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments", RFC 5019, DOI 10.17487/RFC5019, September 2007, <https://www.rfc-editor.org/info/rfc5019>.

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[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>.

8. Informative References

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[boulder] Internet Security Research Group, "Boulder", 2021, <https://github.com/letsencrypt/boulder>.

[lestaging] Internet Security Research Group, "Let's Encrypt Staging Environment", 2021, <https://acme-staging-v02.api.letsencrypt.org/directory>.

Acknowledgments

TODO acknowledge.

Author's Address

A. Gable
Internet Security Research Group

Email: aaron@letsencrypt.org