

Workgroup: ACME Working Group
Internet-Draft: draft-aaron-acme-ari-03
Published: 26 July 2022
Intended Status: Standards Track
Expires: 27 January 2023
Authors: A. Gable
Internet Security Research Group

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

Abstract

This document specifies how an ACME server may provide suggestions 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 27 January 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Conventions and Definitions](#)
- [3. Extensions to the ACME Protocol: The "directory" Resource](#)
- [4. Extensions to the ACME Protocol: The "renewalInfo" Resource](#)
 - [4.1. Getting Renewal Information](#)
 - [4.2. Updating Renewal Information](#)
- [5. Security Considerations](#)
- [6. IANA Considerations](#)
 - [6.1. New Registries](#)
 - [6.2. ACME Resource Type](#)
 - [6.3. ACME Renewal Info Object Fields](#)
- [7. Normative References](#)
- [8. Informative References](#)
- [Appendix A. Example Certificates](#)
 - [A.1. Example End-Entity Certificate](#)
 - [Example CA Certificate](#)
- [Acknowledgments](#)
- [Author's Address](#)

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 for dynamic smearing of load, enabling a CA to respond to exceptional circumstances. For example, a CA could suggest that clients renew prior to a mass-revocation event to mitigate the impact of the revocation, or a CA could

suggest that clients renew earlier than they normally would to reduce the size of an upcoming mass-renewal spike.

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
<code>renewalInfo</code>	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

The "renewalInfo" resource is a new resource type introduced to ACME protocol. This new resource both allows clients to query the server for suggestions on when they should renew certificates, and allows

clients to inform the server when they have completed renewal (or otherwise replaced the certificate to their satisfaction).

4.1. Getting Renewal Information

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 a forward slash and the base64url-encoded [RFC4648] bytes of a DER-encoded CertID ASN.1 sequence [RFC6960]. Trailing '=' characters MUST be stripped.

For example, to request renewal information for the end-entity certificate given in Appendix A.1, issued by the CA certificate given in Appendix A.2, using SHA256, the client would make the following request (the path has been split onto multiple lines for readability):

```
GET https://example.com/acme/renewal-info/  
    MFswCwYJYIZIAWUDBAIBCCeWLRusNLb--vmW0kxm34qDjTMWkc  
    3utIhOMoMwKdqbgQg2iikWysZrD-6c88HMZ6vhIHZPamChLlzGH  
    eZ7pTS8jYCCD6jRWhlRB8c
```

The ACME Server **MAY** restrict the hash algorithms which it accepts (for example, only allowing SHA256 to limit the number of potential cache keys); if it receives a request whose embedded `hashAlgorithm` field contains an unacceptable OID, it **SHOULD** respond with HTTP status code 400 (Bad Request).

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.

`explanationURL` (string, optional): A URL pointing to a page which may explain why the suggested renewal window is what it is. For example, it may be a page explaining the CA's dynamic load-balancing strategy, or a page documenting which certificates are affected by a mass revocation event. Conforming clients **SHOULD** provide this URL to their operator, if present.

```
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"
  },
  "explanationURL": "https://example.com/docs/example-mass-reissuance-event"
}
```

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** attempt renewal at a time of their choosing based on the suggested renewal window. The following algorithm is **RECOMMENDED** for choosing a renewal time:

1. Select a uniform random time within the suggested window.
2. If the selected time is in the past, attempt renewal immediately.
3. Otherwise, if the client can schedule itself to attempt renewal at exactly the selected time, do so.
4. Otherwise, if the selected time is before the next time that the client would wake up normally, attempt renewal immediately.
5. Otherwise, sleep until the next normal wake time, re-check ARI, and return to Step 1.

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.

4.2. Updating Renewal Information

To update the renewal status of a certificate, the client sends a POST request to the server's renewalInfo URL.

The body of the POST is a JWS object which is authenticated to an account as defined in [RFC8555], Section 6.2, and whose JSON payload has the following structure:

certID (required, string): The CertID of the certificate whose renewal information should be updated, in the base64url-encoded version of the DER format with trailing "=" stripped. Note: this is identical to the final path component constructed for GET requests above.

replaced (required, boolean): Whether or not the client considers the certificate to have been replaced. A certificate is considered replaced when its revocation would not disrupt any ongoing services, for instance because it has been renewed and the new certificate is in use, or because it is no longer in use. Clients SHOULD NOT send a request where this value is false.

```
POST /acme/renewal-info HTTP/1.1
Host: example.com
Content-Type: application/jose+json
```

```
{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/evOfKhNU60wg",
    "nonce": "JHb54aT_KTXBWQ0zGYkt9A",
    "url": "https://example.com/acme/renewal-info"
  }),
  "payload": base64url({
    "certID": "MFswCwYJ...RWhlRB8c",
    "replaced": true
  }),
  "signature": "Q1bURgJoEslbD1c5...3pYdSMLio57mQNN4"
}
```

The server **MUST** verify that the request is signed by the account key of the Subscriber to which the certificate was originally issued. If the server accepts the request and the update succeeds, it responds with HTTP status code 200 (OK). If the update is rejected or fails,

for example because the certificate has already been marked as replaced, the server returns an error.

The server might use this renewal update to inform a number of processes, such as: not sending renewal reminder notifications for certificates that have been marked as replaced; sending empty or error responses to subsequent requests for the certificate's renewal information; or confidently revoking certificates subject to a mass revocation without fear of disrupting the Subscriber's operations.

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 Section 10.1.

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 document

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 document

Table 3

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", DOI 10.17487/RFC2119, BCP 14, RFC 2119, 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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", DOI 10.17487/RFC8174, RFC 8174, BCP 14, 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)", DOI 10.17487/RFC8555, RFC 8555, March 2019, <<https://www.rfc-editor.org/info/rfc8555>>.

8. Informative References

[boulder] Internet Security Research Group, "Boulder", 2022, <<https://github.com/letsencrypt/boulder>>.

[lestaging] Internet Security Research Group, "Let's Encrypt Staging Environment", 2022, <<https://acme-staging-v02.api.letsencrypt.org/directory>>.

Appendix A. Example Certificates

A.1. Example End-Entity Certificate

```
-----BEGIN CERTIFICATE-----
MIIDMDCCAhigAwIBAgIIPqNFaGVEHxwwDQYJKoZIhvcNAQELBQAwIDeMBwGA1UE
AxMVbWluawNhhIHJvb3QyY2EgM2ExMzU2MB4XDTEyMDMxMjE0MDQxNjE3NTEw
VowFjEUMBIGA1UEAxMLZXhhbXBsZS5jb20wggeiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCgm9K/c+i12Pf0f8qhgx9n9SKqXq88c0m9ov9AVRbPA
OWAAewqX2yUAWI4LZBGEgzGzTATkiXfoJ3cN3k39cH6tBbb3iSPuEn70ZpIk9D+e
3Q9/hX+N/jlWkaTB/FNA+7aE5IVWhmdczYilXa10V9r+RcvACJt0gsipBZVJ4jfJ
HnWJJGRZzzxqG/xkQmpXxZ07n0PFc8SxYKWdfcgp+rjR2ogYhSz7BfKoVakGPbpX
vZ0uT9z4kkHra/Wjw1kQhtHoTXdAxH3qC2UjMz057Tx+otj0CxAv907CTJXISyWB
vEVcmTSZkHS3eZtvvIwPx7I30ITRkYk/tLl1MbyB3SiZAgMBAAGjeDB2MA4GA1Ud
DwEB/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIwDAYDVR0T
AQH/BAIwADAFBgNVHSMGDAWgBQ4zzDRUaXHVKq1STWkULGU4zGZpTAWBgNVHREE
DzANggtleGFtcGxlLmNvbTANBgkqhkiG9w0BAQsFAAOCAQEAx0aYvmCk7JYGNEXe
+hrOfKawkHYzwVA92cI/Oi6h+oSdHZ2UKzwFNf37cVKZ37FCrrv5pFP/xhhHvrNV
En0x4IaF70rnaTu5miZiUwuvRQP7ZGmGNFYbLTEF6/dj+WqyYdVaWzxRqHFu1ptC
TXysJCeyiGnR+K00j00Q9Zl05JUK30E4hagPLfaIpDDy6RXQt3ss0iNLuB1+I0tp
1URpvffLZQ8xPsEgOZyPW0cabTwJrtqBwily+lwPFn2mChUx846LwQfxtsXU/lJg
HX2RteNjx7YYNeX3Uf960mgo5an6vE8QNAsIoNHyrGyEmXDhTRe9mCHyiW2S7fZq
o9q12g==
-----END CERTIFICATE-----
```

Example CA Certificate

```
-----BEGIN CERTIFICATE-----
MIIDSzCCAjOgAwIBAgII0hNwtJ7Igr0wDQYJKoZIhvcNAQELBQAwIDEeMBwGA1UE
AxMVbWluaWNhIHJvb3QgY2EgM2ExMzU2MCAxDTIyMDMxNzE3NTEwOVoYDzIxMjIw
MzE3MTC1MTA5WjAgMR4wHAYDVQQDExVtaW5pY2Egcm9vdCBjYSAzYTEzNTYwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC3P6cxcCZ7FQ0QrYuigReSa8T
IOPNkmlmX90rTkPwjThiMNEETYK01ea99yXPK36LUHC60LmZ9jVQW2Ny1qwQC0y6
TrquhnwKgtkBMdAZBlySSEXydkL3r0jA4sf1W130/0LwhstU/yv0J8+pj7eSVOR3
zJBnYd1AqnXHRswQm299KXgqema7uwsa8cgjrXsBzAhrwrVY1VhpWFSv3lQRDFQg
c5Z/ZDV9i26qiaJsCCmdisJZWN7N2luUgxdRqzZ4Cr2Xoilg3T+hkb2y/d6ttsPA
kaSA+pq3q6Qa7/qfGdT5WuUkcHpvKNRwqnt9rCYlmG00r3hGgc42D/z1VvfAgMB
AAGjgYYwgYMwDgYDVR0PAQH/BAQDAgKEMBOGA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjASBgNVHRMBAf8ECDAGAQH/AgEAMBOGA1UdDgQWBBQ4zzDRUaXHVKq1
STWkULGU4zGZpTAFBgNVHSMEGDAWgBQ4zzDRUaXHVKq1STWkULGU4zGZpTANBgkq
hkiG9w0BAQsFAAOCAQEArbDhHjGedjb/YjU80aFTPWOMRjgyfQaPPgyxwX6Dsid
1i2H1x4ud4ntz3sTZZxdQIr0qt1IWTWVCjpStwGxaC+38SdreiTTwy/nikXGa/6W
ZyQRppR3agh/pl5LHV06GsJz3YHa7wQhEhj3xsRwa9VrRXgHbLGBPOFVRTHPjaPg
Gtsv2PN3f67DsPHF47ASqy0IRpLZPQmZIw6D3isJwfl+8CzvlB1ve00Q3uh08IJc
fspYQxvFBzYa64uKxNAJMi4Pby8cf4r36Wnb7cL4ho3f0HgAltxdw8jgibRzqZpQ
QKyxn2jX7kxeUDt0hFDJE8l0rhP73m66eBNzxe//FQ==
-----END CERTIFICATE-----
```

Acknowledgments

TODO acknowledge.

Author's Address

A. Gable
Internet Security Research Group

Email: aaron@letsencrypt.org