ACME Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2020

Y. Sheffer
Intuit
D. Lopez
O. Gonzalez de Dios
A. Pastor Perales
Telefonica I+D
T. Fossati
ARM

October 13, 2019

Support for Short-Term, Automatically-Renewed (STAR) Certificates in Automated Certificate Management Environment (ACME)
draft-ietf-acme-star-10

Abstract

Public-key certificates need to be revoked when they are compromised, that is, when the associated private key is exposed to an unauthorized entity. However the revocation process is often unreliable. An alternative to revocation is issuing a sequence of certificates, each with a short validity period, and terminating this sequence upon compromise. This memo proposes an ACME extension to enable the issuance of short-term and automatically renewed (STAR) X.509 certificates.

[RFC Editor: please remove before publication]

While the draft is being developed, the editor's version can be found at https://github.com/yaronf/I-D/tree/master/STAR.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The ACME protocol [RFC8555] automates the process of issuing a certificate to a named entity (an Identifier Owner or IdO). Typically, but not always, the identifier is a domain name.

If the IdO wishes to obtain a string of short-term certificates originating from the same private key (see [Topalovic] about why using short-lived certificates might be preferable to explicit revocation), she must go through the whole ACME protocol each time a new short-term certificate is needed - e.g., every 2-3 days. If done

this way, the process would involve frequent interactions between the registration function of the ACME Certification Authority (CA) and the identity provider infrastructure (e.g.: DNS, web servers), therefore making the issuance of short-term certificates exceedingly dependent on the reliability of both.

This document presents an extension of the ACME protocol that optimizes this process by making short-term certificates first class objects in the ACME ecosystem. Once the Order for a string of short-term certificates is accepted, the CA is responsible for publishing the next certificate at an agreed upon URL before the previous one expires. The IdO can terminate the automatic renewal before the negotiated deadline, if needed - e.g., on key compromise.

For a more generic treatment of STAR certificates, readers are referred to [I-D.nir-saag-star].

1.1. Name Delegation Use Case

The proposed mechanism can be used as a building block of an efficient name-delegation protocol, for example one that exists between a CDN or a cloud provider and its customers [I-D.ietf-acme-star-delegation]. At any time, the service customer (i.e., the IdO) can terminate the delegation by simply instructing the CA to stop the automatic renewal and letting the currently active certificate expire shortly thereafter.

Note that in the name delegation use case the delegated entity needs to access the auto-renewed certificate without being in possession of the ACME account key that was used for initiating the STAR issuance. This leads to the optional use of unauthenticated GET in this protocol (Section 3.4).

1.2. Terminology

IdO Identifier Owner, the owner of an identifier, e.g.: a domain
name, a telephone number.
STAR Short-Term and Automatically Renewed X.509 certificates.

1.3. Conventions used in this document

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.

2. Protocol Flow

The following subsections describe the three main phases of the protocol:

- Bootstrap: the IdO asks an ACME CA to create a short-term and automatically-renewed (STAR) certificate (Section 2.1);
- Auto-renewal: the ACME CA periodically re-issues the short-term certificate and posts it to the star-certificate URL (Section 2.2);
- Termination: the IdO requests the ACME CA to discontinue the automatic renewal of the certificate (Section 2.3).

2.1. Bootstrap

The IdO, in its role as an ACME client, requests the CA to issue a STAR certificate, i.e., one that:

- Has a short validity, e.g., 24 to 72 hours. Note that the exact definition of "short" depends on the use case;
- Is automatically renewed by the CA for a certain period of time;
- Is downloadable from a (highly available) location.

Other than that, the ACME protocol flows as usual between IdO and CA. In particular, IdO is responsible for satisfying the requested ACME challenges until the CA is willing to issue the requested certificate. Per normal ACME processing, the IdO is given back an Order resource associated with the STAR certificate to be used in subsequent interaction with the CA (e.g., if the certificate needs to be terminated.)

The bootstrap phase ends when the ACME CA updates the Order resource to include the URL for the issued STAR certificate.

2.2. Refresh

The CA issues the initial certificate after the authorization completes successfully. It then automatically re-issues the certificate using the same CSR (and therefore the same identifier and public key) before the previous one expires, and publishes it to the URL that was returned to the IdO at the end of the bootstrap phase. The certificate user, which could be either the IdO itself or a delegated third party, as described in [I-D.ietf-acme-star-delegation], obtains the certificate (Section 3.3) and uses it.

The refresh process (Figure 1) goes on until either:

- o IdO explicitly terminates the automatic renewal (Section 2.3); or
- o Automatic renewal expires.

```
      Certificate             ACME/STAR
      User                    Server
      |     Retrieve cert     |                     [...]
      |---------------------->|                      |
      |                       +------.              /
      |                       |      |             /
      |                       | Automatic renewal :
      |                       |      |             \
      |                       |<-----'              \
      |     Retrieve cert     |                      |
      |---------------------->|            short validity period
      |                       |                      |
      |                       +------.              /
      |                       |      |             /
      |                       | Automatic renewal :
      |                       |      |             \
      |                       |<-----'              \
      |     Retrieve cert     |                      |
      |---------------------->|            short validity period
      |                       |                      |
      |                       +------.              /
      |                       |      |             /
      |                       | Automatic renewal :
      |                       |      |             \
      |                       |<-----'              \
      |                       |                      |
      |         [...]         |                    [...]
```

Figure 1: Auto renewal

2.3. Termination

The IdO may request early termination of the STAR certificate by sending a cancellation request to the Order resource, as described in Section 3.1.2. After the CA receives and verifies the request, it shall:

- o Cancel the automatic renewal process for the STAR certificate;
- o Change the certificate publication resource to return an error indicating the termination of the issuance;
- o Change the status of the Order to "canceled".

Note that it is not necessary to explicitly revoke the short-term certificate.

```
     Certificate                                     ACME/STAR
     User                   IdO                      Server
     |                       |                       |
     |                       |      Cancel Order     |
     |                       +---------------------->|
     |                       |                       +-------.
     |                       |                       |       |
     |                       |                       |  End auto renewal
     |                       |                       |  Remove cert link
     |                       |                       |  etc.
     |                       |                       |       |
     |                       |         Done          |<------'
     |                       |<----------------------+
     |                       |                       |
     |                                               |
     |              Retrieve cert                    |
     +---------------------------------------------->|
     |              Error: autoRenewalCanceled       |
     |<----------------------------------------------+
     |                                               |
```

Figure 2: Termination

3. Protocol Details

This section describes the protocol details, namely the extensions to the ACME protocol required to issue STAR certificates.

3.1. ACME Extensions

This protocol extends the ACME protocol, to allow for automatically renewed Orders.

3.1.1. Extending the Order Resource

The Order resource is extended with a new "auto-renewal" object that MUST be present for STAR certificates. The "auto-renewal" object has the following structure:

- start-date (optional, string): the earliest date of validity of the first certificate issued, in [RFC3339] format. When omitted, the start date is as soon as authorization is complete.
- end-date (required, string): the latest date of validity of the last certificate issued, in [RFC3339] format.
- lifetime (required, integer): the maximum validity period of each STAR certificate, an integer that denotes a number of seconds. This is a nominal value which does not include any extra validity time due to server or client adjustment (see below).

- lifetime-adjust (optional, integer): amount of "left pad" added to each STAR certificate, an integer that denotes a number of seconds. The default is 0. If present, the value of the notBefore field that would otherwise appear in the STAR certificates is pre-dated by the specified number of seconds. See also Section 4.1 for why a client might want to use this control and Section 3.5 for how the effective certificate lifetime is computed. The value reflected by the server, together with the value of the lifetime attribute, can be used by the client as a hint to configure its polling timer.
- allow-certificate-get (optional, boolean): see Section 3.4.

These attributes are included in a POST message when creating the Order, as part of the "payload" encoded object. They are returned when the Order has been created, and the ACME server MAY adjust them at will, according to its local policy (see also Section 3.2).

The optional notBefore and notAfter fields defined in Section 7.1.3 of [RFC8555] MUST NOT be present in a STAR Order. If they are included, the server MUST return an error with status code 400 "Bad Request" and type "malformedRequest".

Section 7.1.6 of [RFC8555] defines the following values for the Order resource's status: "pending", "ready", "processing", "valid", and "invalid". In the case of auto-renewal Orders, the status MUST be "valid" as long as STAR certificates are being issued. We add a new status value: "canceled", see Section 3.1.2.

A STAR certificate is by definition a dynamic resource, i.e., it refers to an entity that varies over time. Instead of overloading the semantics of the "certificate" attribute, this document defines a new attribute "star-certificate" to be used instead of "certificate".

- star-certificate (optional, string): A URL for the (rolling) STAR certificate that has been issued in response to this Order.

3.1.2. Canceling an Auto-renewal Order

An important property of the auto-renewal Order is that it can be canceled by the IdO, with no need for certificate revocation. To cancel the Order, the ACME client sends a POST to the Order URL as shown in Figure 3.

```
POST /acme/order/ogfr8EcolOT HTTP/1.1
Host: example.org
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/gw06UNhKfOve",
    "nonce": "Alc00Ap6Rt7GMkEl3L1JX5",
    "url": "https://example.com/acme/order/ogfr8EcolOT"
  }),
  "payload": base64url({
    "status": "canceled"
  }),
  "signature": "g454e3hdBlkT4AEw...nKePnUyZTjGtXZ6H"
}
```

Figure 3: Canceling an Auto-renewal Order

After a successful cancellation, the server MUST NOT issue any additional certificates for this Order.

When the Order is canceled, the server:

- MUST update the status of the Order resource to "canceled" and MUST set an appropriate "expires" date;
- MUST respond with 403 (Forbidden) to any requests to the star-certificate endpoint. The response SHOULD provide additional information using a problem document [RFC7807] with type "urn:ietf:params:acme:error:autoRenewalCanceled".

Issuing a cancellation for an Order that is not in "valid" state is not allowed. A client MUST NOT send such a request, and a server MUST return an error response with status code 400 (Bad Request) and type "urn:ietf:params:acme:error:autoRenewalCancellationInvalid".

The state machine described in Section 7.1.6 of [RFC8555] is extended as illustrated in Figure 4 (State Transitions for Order Objects).

```
       pending --------------+
          |                  |
          | All authz        |
          | "valid"          |
          V                  |
        ready ---------------+
          |                  |
          | Receive          |
          | finalize         |
          | request          |
          V                  |
      processing ------------+
          |                  |
          | First            |
          | certificate      | Error or
          | issued           | Authorization failure
          V                  V
        valid             invalid
          |
          | STAR
          | Certificate
          | canceled
          V
       canceled
```

Figure 4

Explicit certificate revocation using the revokeCert interface (Section 7.6 of [RFC8555]) is not supported for STAR certificates. A server receiving a revocation request for a STAR certificate MUST return an error response with status code 403 (Forbidden) and type "urn:ietf:params:acme:error:autoRenewalRevocationNotSupported".

3.2. Capability Discovery

In order to support the discovery of STAR capabilities, the "meta" field inside the directory object defined in Section 9.7.6 of [RFC8555] is extended with a new "auto-renewal" object. The "auto-renewal" object MUST be present if the server supports STAR. Its structure is as follows:

- o min-lifetime (required, integer): minimum acceptable value for auto-renewal lifetime, in seconds.
- o max-duration (required, integer): maximum delta between the auto-renewal end-date and start-date, in seconds.
- o allow-certificate-get (optional, boolean): see Section 3.4.

An example directory object advertising STAR support with one day min-lifetime and one year max-duration, and supporting certificate fetching with an HTTP GET is shown in Figure 5.

```
{
   "new-nonce": "https://example.com/acme/new-nonce",
   "new-account": "https://example.com/acme/new-account",
   "new-order": "https://example.com/acme/new-order",
   "new-authz": "https://example.com/acme/new-authz",
   "revoke-cert": "https://example.com/acme/revoke-cert",
   "key-change": "https://example.com/acme/key-change",
   "meta": {
     "terms-of-service": "https://example.com/acme/terms/2017-5-30",
     "website": "https://www.example.com/",
     "caa-identities": ["example.com"],
     "auto-renewal": {
       "min-lifetime": 86400,
       "max-duration":  31536000,
       "allow-certificate-get": true
     }
   }
}
```

Figure 5: Directory object with STAR support

3.3. Fetching the Certificates

The certificate is fetched from the star-certificate endpoint with POST-as-GET as per [RFC8555] Section 7.4.2, unless client and server have successfully negotiated the "unauthenticated GET" option described in Section 3.4. In such case, the client can simply issue a GET to the star-certificate resource without authenticating itself to the server as illustrated in Figure 6.

```
     GET /acme/cert/mAt3xBGaobw HTTP/1.1
     Host: example.org
     Accept: application/pem-certificate-chain

     HTTP/1.1 200 OK
     Content-Type: application/pem-certificate-chain
     Link: <https://example.com/acme/some-directory>;rel="index"
     Cert-Not-Before: Thu, 3 Oct 2019 00:00:00 GMT
     Cert-Not-After: Thu, 10 Oct 2019 00:00:00 GMT

     -----BEGIN CERTIFICATE-----
     [End-entity certificate contents]
     -----END CERTIFICATE-----
     -----BEGIN CERTIFICATE-----
     [Issuer certificate contents]
     -----END CERTIFICATE-----
     -----BEGIN CERTIFICATE-----
     [Other certificate contents]
     -----END CERTIFICATE-----
```

Figure 6: Fetching a STAR certificate with unauthenticated GET

The Server SHOULD include the "Cert-Not-Before" and "Cert-Not-After" HTTP header fields in the response. When they exist, they MUST be equal to the respective fields inside the end-entity certificate. Their format is "HTTP-date" as defined in Section 7.1.1.2 of [RFC7231]. Their purpose is to enable client implementations that do not parse the certificate.

Following are further clarifications regarding usage of these header fields, as per [RFC7231] Sec. 8.3.1. All apply to both headers.

- This header field is a single value, not a list.
- The header field is used only in responses to GET, HEAD and POST-as-GET requests, and only for MIME types that denote public key certificates.
- Header field semantics are independent of context.
- The header field is not hop-by-hop.
- Intermediaries MAY insert or delete the value;
- If an intermediary inserts the value, it MUST ensure that the newly added value matches the corresponding value in the certificate.
- The header field is not appropriate for a Vary field.
- The header field is allowed within message trailers.
- The header field is not appropriate within redirects.
- The header field does not introduce additional security considerations. It discloses in a simpler form information that is already available inside the certificate.

To improve robustness, the next certificate MUST be made available by the ACME CA at the URL pointed by "star-certificate" at the latest halfway through the lifetime of the currently active certificate. It is worth noting that this has an implication in case of cancellation: in fact, from the time the next certificate is made available, the cancellation is not completely effective until the "next" certificate also expires. To avoid the client accidentally entering a broken state, the notBefore of the "next" certificate MUST be set so that the certificate is already valid when it is published at the "star-certificate" URL. Note that the server might need to increase the auto-renewal lifetime-adjust value to satisfy the latter requirement. For a detailed description of the renewal scheduling logic, see Section 3.5. For further rationale on the need for adjusting the certificate validity, see Section 4.1.

The server MUST NOT issue any certificates for this Order with notAfter after the auto-renewal end-date.

For expired Orders, the server MUST respond with 403 (Forbidden) to any requests to the star-certificate endpoint. The response SHOULD provide additional information using a problem document [RFC7807] with type "urn:ietf:params:acme:error:autoRenewalExpired". Note that the Order resource's state remains "valid", as per the base protocol.

3.4. Negotiating an unauthenticated GET

In order to enable the name delegation workflow defined in [I-D.ietf-acme-star-delegation] as well as to increase the reliability of the STAR ecosystem (see Section 4.3 for details), this document defines a mechanism that allows a server to advertise support for accessing star-certificate resources via unauthenticated GET (in addition to POST-as-GET), and a client to enable this service with per-Order granularity.

Specifically, a server states its availability to grant unauthenticated access to a client's Order star-certificate by setting the allow-certificate-get attribute to true in the auto-renewal object of the meta field inside the Directory object:

o allow-certificate-get (optional, boolean): If this field is present and set to true, the server allows GET (and HEAD) requests to star-certificate URLs.

A client states its desire to access the issued star-certificate via unauthenticated GET by adding an allow-certificate-get attribute to the auto-renewal object of the payload of its newOrder request and setting it to true.

- allow-certificate-get (optional, boolean): If this field is present and set to true, the client requests the server to allow unauthenticated GET (and HEAD) to the star-certificate associated with this Order.

If the server accepts the request, it MUST reflect the attribute setting in the resulting Order object.

Note that even when the use of unauthenticated GET has been agreed, the server MUST also allow POST-as-GET requests to the star-certificate resource.

3.5. Computing notBefore and notAfter of STAR Certificates

We define "nominal renewal date" as the point in time when a new short-term certificate for a given STAR Order is due. Its cadence is a multiple of the Order's auto-renewal lifetime that starts with the issuance of the first short-term certificate and is upper-bounded by the Order's auto-renewal end-date (Figure 7).

```
    T      - STAR Order's auto-renewal lifetime
    end    - STAR Order's auto-renewal end-date
    nrd[i] - nominal renewal date of the i-th STAR certificate


                 .- T -.   .- T -.   .- T -.   .__.
                /       \ /       \ /       \ /  end
    -----------o---------o---------o---------o----X-------> t
              nrd[0]    nrd[1]    nrd[2]    nrd[3]
```

Figure 7: Nominal Renewal Date

The rules to determine the notBefore and notAfter values of the i-th STAR certificate are as follows:

```
    notAfter  = min(nrd[i] + T, end)
    notBefore = nrd[i] - max(adjust_client, adjust_server)
```

Where "adjust_client" is the min between the auto-renewal lifetime-adjust value ("la"), optionally supplied by the client, and the auto-renewal lifetime of each short-term certificate ("T"); "adjust_server" is the amount of padding added by the ACME server to make sure that all certificates being published are valid at the time of publication. The server padding is a fraction f of T (i.e., f * T with .5 <= f < 1, see Section 3.3):

```
    adjust_client = min(T, la)
    adjust_server = f * T
```

Note that the ACME server MUST NOT set the notBefore of the first STAR certificate to a date prior to the auto-renewal start-date.

3.5.1. Example

Given a server that intends to publish the next STAR certificate halfway through the lifetime of the previous one, and a STAR Order with the following attributes:

```
     "auto-renewal": {
       "start-date": "2019-01-10T00:00:00Z",
       "end-date": "2019-01-20T00:00:00Z",
       "lifetime": 345600,          // 4 days
       "lifetime-adjust": 259200    // 3 days
     }
```

The amount of time that needs to be subtracted from each nominal renewal date is 3 days - i.e., max(min(345600, 259200), 345600 * .5).

The notBefore and notAfter of each short-term certificate are:

notBefore	notAfter
2019-01-10T00:00:00Z	2019-01-14T00:00:00Z
2019-01-11T00:00:00Z	2019-01-18T00:00:00Z
2019-01-15T00:00:00Z	2019-01-20T00:00:00Z

The value of the notBefore is also the time at which the client should expect the new certificate to be available from the star-certificate endpoint.

4. Operational Considerations

4.1. The Meaning of "Short Term" and the Impact of Skewed Clocks

"Short Term" is a relative concept, therefore trying to define a cut-off point that works in all cases would be a useless exercise. In practice, the expected lifetime of a STAR certificate will be counted in minutes, hours or days, depending on different factors: the underlying requirements for revocation, how much clock synchronization is expected among relying parties and the issuing CA, etc.

Nevertheless, this section attempts to provide reasonable suggestions for the Web use case, informed by current operational and research experience.

Acer et al. [Acer] find that one of the main causes of "HTTPS error" warnings in browsers is misconfigured client clocks. In particular, they observe that roughly 95% of the "severe" clock skews - the 6.7% of clock-related breakage reports which account for clients that are more than 24 hours behind - happen to be within 6-7 days.

In order to avoid these spurious warnings about a not (yet) valid server certificate, site owners could use the auto-renewal lifetime-adjust attribute to control the effective lifetime of their Web facing certificates. The exact number depends on the percentage of the "clock-skewed" population that the site owner expects to protect - 5 days cover 97.3%, 7 days cover 99.6% - as well as the nominal auto-renewal lifetime of the STAR Order. Note that exact choice is also likely to depend on the kinds of client that is prevalent for a given site or app - for example, Android and Mac OS clients are known to behave better than Windows clients. These considerations are clearly out of scope of the present document.

In terms of security, STAR certificates and certificates with OCSP must-staple [RFC7633] can be considered roughly equivalent if the STAR certificate's and the OCSP response's lifetimes are the same. Given OCSP responses can be cached on average for 4 days [Stark], it is RECOMMENDED that a STAR certificate that is used on the Web has an "effective" lifetime (excluding any adjustment to account for clock skews) no longer than 4 days.

4.2. Impact on Certificate Transparency (CT) Logs

Even in the highly unlikely case STAR becomes the only certificate issuance model, discussion with the IETF TRANS Working Group and Certificate Transparency (CT) logs implementers suggests that existing CT Log Server implementations are capable of sustaining the resulting 100-fold increase in ingestion rate. Additionally, such a future, higher load could be managed with a variety of techniques (e.g., sharding by modulo of certificate hash, using "smart" load-balancing CT proxies, etc.). With regards to the increase in the log size, current CT log growth is already being managed with schemes like Chrome's Log Policy [OBrien] which allow Operators to define their log life-cycle; and allowing the CAs, User Agents, Monitors, and any other interested entities to build-in support for that life-cycle ahead of time.

4.3. HTTP Caching and Dependability

When using authenticated POST-as-GET, the HTTPS endpoint from where the STAR certificate is fetched can't be easily replicated by an on-path HTTP cache. Reducing the caching properties of the protocol makes STAR clients increasingly dependent on the ACME server

availability. This might be problematic given the relatively high rate of client-server interactions in a STAR ecosystem and especially when multiple endpoints (e.g., a high number of CDN edge nodes) end up requesting the same certificate. Clients and servers should consider using the mechanism described in Section 3.4 to mitigate the risk.

When using unauthenticated GET to fetch the STAR certificate, the server SHALL use the appropriate cache directives to set the freshness lifetime of the response (Section 5.2 of [RFC7234]) such that on-path caches will consider it stale before or at the time its effective lifetime is due to expire.

5. Implementation Status

Note to RFC Editor: please remove this section before publication, including the reference to [RFC7942] and [I-D.sheffer-acme-star-request].

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

5.1. Overview

The implementation is constructed around 3 elements: STAR Client for the Name Delegation Client (NDC), STAR Proxy for IdO and ACME Server for CA. The communication between them is over an IP network and the HTTPS protocol.

The software of the implementation is available at: https://github.com/mami-project/lurk

The following subsections offer a basic description, detailed information is available in https://github.com/mami-project/lurk/blob/master/proxySTAR_v2/README.md

5.1.1. ACME Server with STAR extension

This is a fork of the Let's Encrypt Boulder project that implements an ACME compliant CA. It includes modifications to extend the ACME protocol as it is specified in this draft, to support recurrent Orders and cancelling Orders.

The implementation understands the new "recurrent" attributes as part of the Certificate issuance in the POST request for a new resource. An additional process "renewalManager.go" has been included in parallel that reads the details of each recurrent request, automatically produces a "cron" Linux based task that issues the recurrent certificates, until the lifetime ends or the Order is canceled. This process is also in charge of maintaining a fixed URI to enable the NDC to download certificates, unlike Boulder's regular process of producing a unique URI per certificate.

5.1.2. STAR Proxy

The STAR Proxy has a double role as ACME client and STAR Server. The former is a fork of the EFF Certbot project that implements an ACME compliant client with the STAR extension. The latter is a basic HTTP REST API server.

The STAR Proxy understands the basic API request with a server. The current implementation of the API is defined in draft-ietf-acme-star-01. Registration or Order cancellation triggers the modified Certbot client that requests, or cancels, the recurrent generation of certificates using the STAR extension over ACME protocol. The URI with the location of the recurrent certificate is delivered to the STAR client as a response.

5.2. Level of Maturity

This is a prototype.

5.3. Coverage

A STAR Client is not included in this implementation, but done by direct HTTP request with any open HTTP REST API tool. This is expected to be covered as part of the [I-D.sheffer-acme-star-request] implementation.

This implementation completely covers STAR Proxy and ACME Server with STAR extension.

5.4. Version Compatibility

The implementation is compatible with version draft-ietf-acme-star-01. The implementation is based on the Boulder and Certbot code release from 7-Aug-2017.

5.5. Licensing

This implementation inherits the Boulder license (Mozilla Public License 2.0) and Certbot license (Apache License Version 2.0).

5.6. Implementation experience

To prove the concept all the implementation has been done with a self-signed CA, to avoid impact on real domains. To be able to do it we use the FAKE_DNS property of Boulder and static /etc/hosts entries with domains names. Nonetheless this implementation should run with real domains.

Most of the implementation has been made to avoid deep changes inside of Boulder or Certbot, for example, the recurrent certificates issuance by the CA is based on an external process that auto-configures the standard Linux "cron" daemon in the ACME CA server.

The reference setup recommended is one physical host with 3 virtual machines, one for each of the 3 components (client, proxy and server) and the connectivity based on host bridge.

Network security is not enabled (iptables default policies are "accept" and all rules removed) in this implementation to simplify and test the protocol.

5.7. Contact Information

See author details below.

6. IANA Considerations

[[RFC Editor: please replace XXXX below by the RFC number.]]

6.1. New Registries

This document requests that IANA create the following new registries:

- ACME Order Auto Renewal Fields (Section 6.4)

- ACME Directory Metadata Auto Renewal Fields (Section 6.6)

All of these registries are administered under a Specification Required policy [RFC8126].

6.2. New Error Types

This document adds the following entries to the ACME Error Type registry:

Type	Description	Reference
autoRenewalCanceled	The short-term certificate is no longer available because the auto-renewal Order has been explicitly canceled by the IdO	RFC XXXX
autoRenewalExpired	The short-term certificate is no longer available because the auto-renewal Order has expired	RFC XXXX
autoRenewalCancellationInvalid	A request to cancel a auto-renewal Order that is not in state "valid" has been received	RFC XXXX
autoRenewalRevocationNotSupported	A request to revoke a auto-renewal Order has been received	RFC XXXX

6.3. New fields in Order Objects

This document adds the following entries to the ACME Order Object Fields registry:

Field Name	Field Type	Configurable	Reference
auto-renewal	object	true	RFC XXXX
star-certificate	string	false	RFC XXXX

6.4. Fields in the "auto-renewal" Object within an Order Object

The "ACME Order Auto Renewal Fields" registry lists field names that are defined for use in the JSON object included in the "auto-renewal" field of an ACME order object.

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
- Configurable: Boolean indicating whether the server should accept values provided by the client
- Reference: Where this field is defined

Initial contents: The fields and descriptions defined in Section 3.1.1.

Field Name	Field Type	Configurable	Reference
start-date	string	true	RFC XXXX
end-date	string	true	RFC XXXX
lifetime	integer	true	RFC XXXX
lifetime-adjust	integer	true	RFC XXXX
allow-certificate-get	boolean	true	RFC XXXX

6.5. New fields in the "meta" Object within a Directory Object

This document adds the following entry to the ACME Directory Metadata Fields:

Field Name	Field Type	Reference
auto-renewal	object	RFC XXXX

6.6. Fields in the "auto-renewal" Object within a Directory Metadata Object

The "ACME Directory Metadata Auto Renewal Fields" registry lists field names that are defined for use in the JSON object included in the "auto-renewal" field of an ACME directory "meta" object.

Template:

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

Initial contents: The fields and descriptions defined in Section 3.2.

Field Name	Field Type	Reference
min-lifetime	integer	RFC XXXX
max-duration	integer	RFC XXXX
allow-certificate-get	boolean	RFC XXXX

6.7. Cert-Not-Before and Cert-Not-After HTTP Headers

The "Message Headers" registry should be updated with the following additional values:

Header Field Name	Protocol	Status	Reference
Cert-Not-Before	http	standard	RFC XXXX, Section 3.3
Cert-Not-After	http	standard	RFC XXXX, Section 3.3

7. Security Considerations

7.1. No revocation

STAR certificates eliminate an important security feature of PKI which is the ability to revoke certificates. Revocation allows the administrator to limit the damage done by a rogue node or an adversary who has control of the private key. With STAR certificates, expiration replaces revocation so there is potential for lack of timeliness in the revocation taking effect. To that end, see also the discussion on clock skew in Section 4.1.

It should be noted that revocation also has timeliness issues, because both CRLs and OCSP responses have nextUpdate fields that tell relying parties (RPs) how long they should trust this revocation data. These fields are typically set to hours, days, or even weeks in the future. Any revocation that happens before the time in nextUpdate goes unnoticed by the RP.

One situation where the lack of explicit revocation could create a security risk to the IdO is when the Order is created with start-date some appreciable amount of time in the future. Recall that when authorizations have been fulfilled, the Order moves to the "valid" state and the star-certificate endpoint is populated with the first cert (Figure 4). So, if an attacker manages to get hold of the private key as well as of the first (post-dated) certificate, there is a time window in the future when they will be able to successfully impersonate the IdO. Note that cancellation is pointless in this case. In order to mitigate the described threat, it is RECOMMENDED that IdO place their Orders at a time that is close to the Order's start-date.

More discussion of the security of STAR certificates is available in [Topalovic].

7.2. Denial of Service Considerations

STAR adds a new attack vector that increases the threat of denial of service attacks, caused by the change to the CA's behavior. Each STAR request amplifies the resource demands upon the CA, where one Order produces not one, but potentially dozens or hundreds of certificates, depending on the auto-renewal "lifetime" parameter. An attacker can use this property to aggressively reduce the auto-renewal "lifetime" (e.g. 1 sec.) jointly with other ACME attack vectors identified in Sec. 10 of [RFC8555]. Other collateral impact is related to the certificate endpoint resource where the client can retrieve the certificates periodically. If this resource is external to the CA (e.g. a hosted web server), the previous attack will be reflected to that resource.

Mitigation recommendations from ACME still apply, but some of them need to be adjusted. For example, applying rate limiting to the initial request, by the nature of the auto-renewal behavior cannot solve the above problem. The CA server needs complementary mitigation and specifically, it SHOULD enforce a minimum value on auto-renewal "lifetime". Alternatively, the CA can set an internal certificate generation processes rate limit. Note that this limit has to take account of already-scheduled renewal issuances as well as new incoming requests.

7.3. Privacy Considerations

In order to avoid correlation of certificates by account, if unauthenticated GET is negotiated (Section 3.4) the recommendation in Section 10.5 of [RFC8555] regarding the choice of URL structure applies, i.e. servers SHOULD choose URLs of certificate resources in a non-guessable way, for example using capability URLs [W3C.WD-capability-urls-20140218].

8. Acknowledgments

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI). This support does not imply endorsement.

Thanks to Ben Kaduk, Richard Barnes, Roman Danyliw, Jon Peterson, Eric Rescorla, Ryan Sleevi, Sean Turner, Alexey Melnikov, Adam Roach, Martin Thomson and Mehmet Ersue for helpful comments and discussions that have shaped this document.

9. References

9.1. 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>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <https://www.rfc-editor.org/info/rfc7234>.

[RFC7807] Nottingham, M. and E. Wilde, "Problem Details for HTTP APIs", RFC 7807, DOI 10.17487/RFC7807, March 2016, <https://www.rfc-editor.org/info/rfc7807>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

9.2. Informative References

[Acer] Acer, M., Stark, E., Felt, A., Fahl, S., Bhargava, R., Dev, B., Braithwaite, M., Sleevi, R., and P. Tabriz, "Where the Wild Warnings Are: Root Causes of Chrome HTTPS Certificate Errors", DOI 10.1145/3133956.3134007, 2017, <https://acmccs.github.io/papers/p1407-acerA.pdf>.

[I-D.ietf-acme-star-delegation] Sheffer, Y., Lopez, D., Pastor, A., and T. Fossati, "An ACME Profile for Generating Delegated STAR Certificates", draft-ietf-acme-star-delegation-01 (work in progress), August 2019.

[I-D.nir-saag-star] Nir, Y., Fossati, T., Sheffer, Y., and T. Eckert, "Considerations For Using Short Term Certificates", draft-nir-saag-star-01 (work in progress), March 2018.

[I-D.sheffer-acme-star-request] Sheffer, Y., Lopez, D., Dios, O., Pastor, A., and T. Fossati, "Generating Certificate Requests for Short-Term, Automatically-Renewed (STAR) Certificates", draft-sheffer-acme-star-request-02 (work in progress), June 2018.

[OBrien] O'Brien, D. and R. Sleevi, "Chromium Certificate Transparency Log Policy", 2017, <https://github.com/chromium/ct-policy>.

[RFC7633] Hallam-Baker, P., "X.509v3 Transport Layer Security (TLS) Feature Extension", RFC 7633, DOI 10.17487/RFC7633, October 2015, <https://www.rfc-editor.org/info/rfc7633>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[Stark] Stark, E., Huang, L., Israni, D., Jackson, C., and D. Boneh, "The case for prefetching and prevalidating TLS server certificates", 2012, <http://crypto.stanford.edu/~dabo/pubs/abstracts/ssl-prefetch.html>.

[Topalovic] Topalovic, E., Saeta, B., Huang, L., Jackson, C., and D. Boneh, "Towards Short-Lived Certificates", 2012, <http://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final9.pdf>.

[W3C.WD-capability-urls-20140218] Tennison, J., "Good Practices for Capability URLs", World Wide Web Consortium WD WD-capability-urls-20140218, February 2014, <http://www.w3.org/TR/2014/WD-capability-urls-20140218>.

Appendix A. Document History

[[Note to RFC Editor: please remove before publication.]]

A.1. draft-ietf-acme-star-10

IESG processing:

- More clarity on IANA registration (Alexey);
- HTTP header requirements adjustments (Adam);
- Misc editorial (Ben)

A.2. draft-ietf-acme-star-09

Richard and Ryan's review resulted in the following updates:

- STAR Order and Directory Meta attributes renamed slightly and grouped under two brand new "auto-renewal" objects;
- IANA registration updated accordingly (note that two new registries have been added as a consequence);
- Unbounded pre-dating of certificates removed so that STAR certs are never issued with their notBefore in the past;
- Changed "recurrent" to "autoRenewal" in error codes;
- Changed "recurrent" to "auto-renewal" in reference to Orders;
- Added operational considerations for HTTP caches.

A.3. draft-ietf-acme-star-08

- Improved text on interaction with CT Logs, responding to Mehmet Ersue's review.

A.4. draft-ietf-acme-star-07

- Changed the HTTP headers names and clarified the IANA registration, following feedback from the IANA expert reviewer

A.5. draft-ietf-acme-star-06

- Roman's AD review

A.6. draft-ietf-acme-star-05

- EKR's AD review
- A detailed example of the timing of certificate issuance and predating
- Added an explicit client-side parameter for predating
- Security considerations around unauthenticated GET

A.7. draft-ietf-acme-star-04

- WG last call comments by Sean Turner
- revokeCert interface handling
- Allow negotiating plain-GET for certs
- In STAR Orders, use star-certificate instead of certificate

A.8. draft-ietf-acme-star-03

- Clock skew considerations
- Recommendations for "short" in the Web use case
- CT log considerations

A.9. draft-ietf-acme-star-02

- Discovery of STAR capabilities via the directory object
- Use the more generic term Identifier Owner (IdO) instead of Domain Name Owner (DNO)
- More precision about what goes in the order
- Detail server side behavior on cancellation

A.10. draft-ietf-acme-star-01

- Generalized the introduction, separating out the specifics of CDNs.
- Clean out LURK-specific text.
- Using a POST to ensure cancellation is authenticated.
- First and last date of recurrent cert, as absolute dates. Validity of certs in seconds.
- Use RFC7807 "Problem Details" in error responses.
- Add IANA considerations.
- Changed the document's title.

A.11. draft-ietf-acme-star-00

- Initial working group version.
- Removed the STAR interface, the protocol between NDC and DNO. What remains is only the extended ACME protocol.

A.12. draft-sheffer-acme-star-02

- Using a more generic term for the delegation client, NDC.
- Added an additional use case: public cloud services.
- More detail on ACME authorization.

A.13. draft-sheffer-acme-star-01

- A terminology section.
- Some cleanup.

A.14. draft-sheffer-acme-star-00

- Renamed draft to prevent confusion with other work in this space.
- Added an initial STAR protocol: a REST API.
- Discussion of CDNI use cases.

A.15. draft-sheffer-acme-star-lurk-00

- Initial version.

Authors' Addresses

Yaron Sheffer
Intuit

EMail: yaronf.ietf@gmail.com

Diego Lopez
Telefonica I+D

EMail: diego.r.lopez@telefonica.com

Oscar Gonzalez de Dios
Telefonica I+D

EMail: oscar.gonzalezdedios@telefonica.com

Antonio Agustin Pastor Perales
Telefonica I+D

EMail: antonio.pastorperales@telefonica.com

Thomas Fossati
ARM

EMail: thomas.fossati@arm.com