ACME Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2018

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

March 03, 2018

Support for Short-Term, Automatically-Renewed (STAR) Certificates in Automated Certificate Management Environment (ACME)
draft-ietf-acme-star-03

Abstract

Public-key certificates need to be revoked when they are compromised, that is, when the associated private key is exposed to an attacker. 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) 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

Table of Contents

1. Introduction 3
1.1. Name Delegation Use Case 4
1.2. Terminology 4
1.3. Conventions used in this document 4
2. Protocol Flow 4
2.1. Bootstrap 5
2.2. Refresh 5
2.3. Termination 6
3. Protocol Details 7
3.1. ACME Extensions 7
3.1.1. Extending the Order Resource 7
3.1.2. Canceling a Recurrent Order 8
3.2. Capability Discovery 9
3.3. Fetching the Certificates 10
4. Operational Considerations 11
4.1. Short Term and the Impact of Skewed Clocks 11
4.2. Impact on Certificate Transparency (CT) Logs 11
5. Implementation Status 12
5.1. Overview 12
5.1.1. ACME Server with STAR extension 12
5.1.2. STAR Proxy 13
5.2. Level of Maturity 13
5.3. Coverage 13
5.4. Version Compatibility 13
5.5. Licensing 14
5.6. Implementation experience 14
5.7. Contact Information 14
6. IANA Considerations 14
6.1. New ACME Error Types 14
6.2. New ACME Order Object Fields 15
6.3. Not-Before and Not-After HTTP Headers 15

7. Security Considerations 16
7.1. Denial of Service Considerations 16
7.2. Additional Considerations TBD 16
8. Acknowledgments 16
9. References 17
9.1. Normative References 17
9.2. Informative References 17
Appendix A. Document History 19
A.1. draft-ietf-acme-star-03 19
A.2. draft-ietf-acme-star-02 19
A.3. draft-ietf-acme-star-01 19
A.4. draft-ietf-acme-star-00 19
A.5. draft-sheffer-acme-star-02 19
A.6. draft-sheffer-acme-star-01 19
A.7. draft-sheffer-acme-star-00 20
A.8. draft-sheffer-acme-star-lurk-00 20
Authors' Addresses 20

1. Introduction

The ACME protocol [I-D.ietf-acme-acme] automates the process of issuing a certificate to a named entity (an Identity Owner or IdO). Typically, but not always, the identity is a domain name and we may refer to the entity as a Domain Name Owner (DNO).

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 natural 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.sheffer-acme-star-request]. 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.

1.2. Terminology

IdO Identifier Owner, the owner of an identifier, e.g.: a domain name, a telephone number.
DNO Domain Name Owner, a type of IdO whose identifier is a domain name.
STAR Short-Term, Automatically Renewed X.509 certificates.
NDC Name Delegation Client, an entity to which the identifier owned by the IdO is delegated for a limited time. Examples include a CDN edge cache, a cloud provider's load balancer or Web Application Firewall (WAF).

1.3. Conventions used in this document

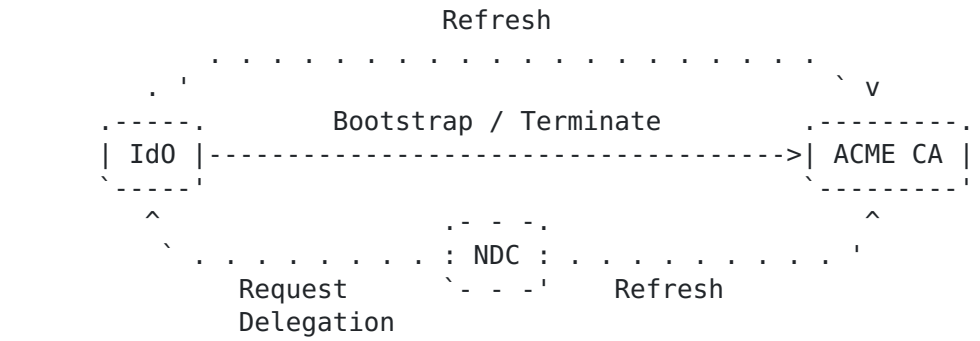
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 a public URL (Section 2.2);
- Termination: the IdO requests the ACME CA to discontinue the automatic renewal of the certificate (Section 2.3).

This diagram presents the entities that are (or may be) involved in the protocol and their interactions during the different phases.



Note that there might be a distinct NDC entity (e.g., a CDN edge cache) that uses a separate channel to request the IdO to set up a name delegation. The protocol described in [I-D.sheffer-acme-star-request] may be used for this purpose.

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) public link without requiring any special authorization.

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 URL for the issued 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 IdO obtains a confirmation from the ACME CA that includes a certificate endpoint.

2.2. Refresh

The CA automatically re-issues the certificate using the same CSR (and therefore the same identifier and public key) before it 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.sheffer-acme-star-request], obtains the certificate and uses it.

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

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

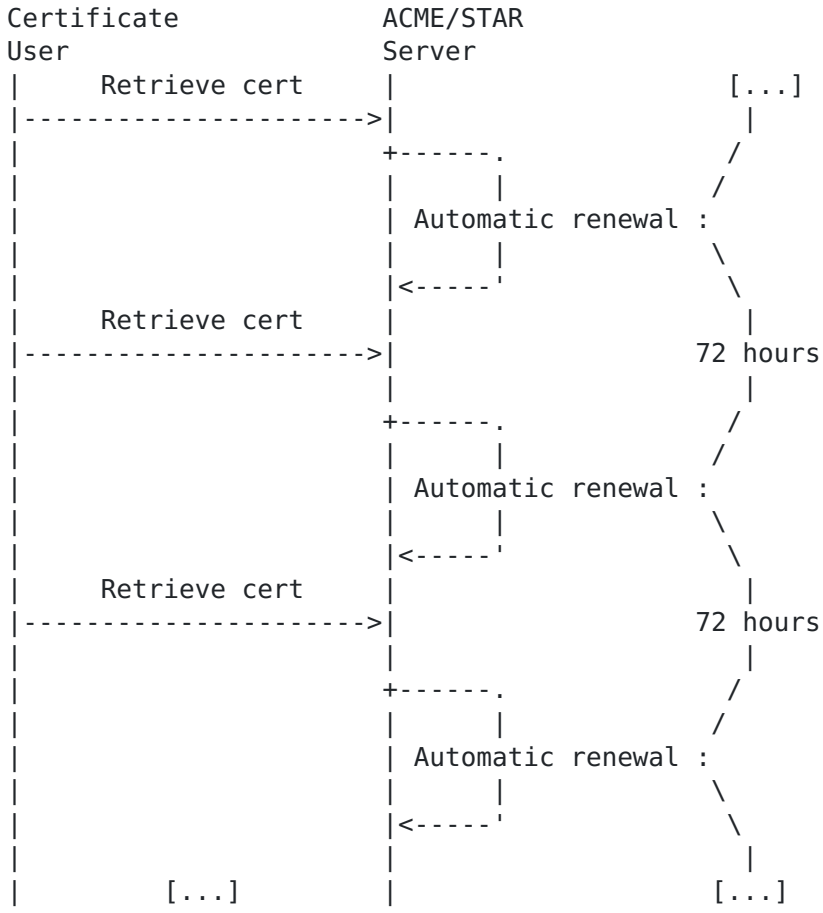


Figure 1: Auto renewal

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:

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

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

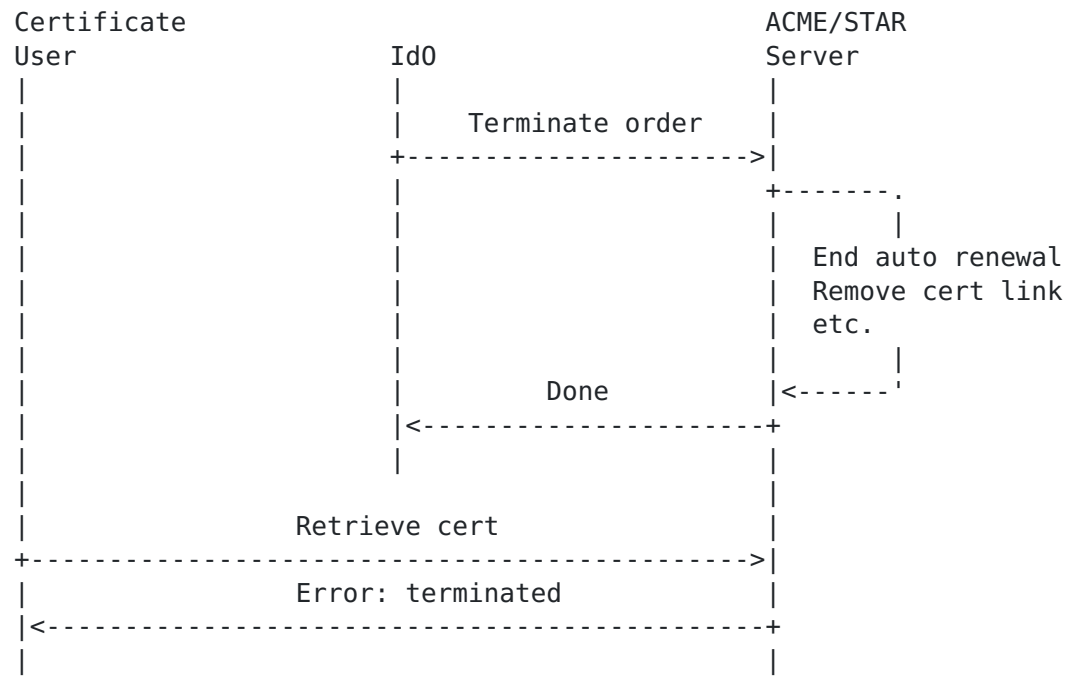


Figure 2: Termination

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 recurrent orders.

3.1.1. Extending the Order Resource

The order resource is extended with the following attributes:

```
{
  "recurrent": true,
  "recurrent-start-date": "2016-01-01T00:00:00Z",
  "recurrent-end-date": "2017-01-01T00:00:00Z",
  "recurrent-certificate-validity": 604800
}
```

- recurrent: MUST be true for STAR certificates.
- recurrent-start-date: the earliest date of validity of the first certificate issued, in [RFC3339] format. This attribute is

optional. When omitted, the start date is as soon as authorization is complete.

- recurrent-end-date: the latest date of validity of the last certificate issued, in [RFC3339] format.
- recurrent-certificate-validity: the maximum validity period of each STAR certificate, an integer that denotes a number of seconds.

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 MUST NOT be present in a STAR order.

ACME defines the following values for the order resource's status: "invalid", "pending", "processing", "valid". In the case of recurrent 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.

3.1.2. Canceling a Recurrent Order

An important property of the recurrent 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:

```
POST /acme/order/1 HTTP/1.1
Host: acme-server.example.org
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/1",
    "nonce": "5XJ1L3lEkMG7tR6pA00clA",
    "url": "https://example.com/acme/order/1"
  }),
  "payload": base64url({
    "status": "canceled"
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4TklBdh3e454g"
}
```

The server MUST NOT issue any additional certificates for this order, beyond the certificate that is available for collection at the time of deletion.

Immediately after 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 certificate endpoint. The response SHOULD provide additional information using a problem document [RFC7807] with type "urn:ietf:params:acme:error:recurrentOrderCanceled".

Issuing a cancellation for an order that is not in "valid" state has undefined semantics. 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:recurrentCancellationInvalid".

3.2. Capability Discovery

In order to support the discovery of STAR capabilities, The directory object of an ACME STAR server MUST contain the following attributes inside the "meta" field:

- star-enabled: boolean flag indicating STAR support. An ACME STAR server MUST include this key, and MUST set it to true if the feature is enabled.
- star-min-cert-validity: minimum acceptable value for recurrent-certificate-validity, in seconds.
- star-max-renewal: maximum delta between recurrent-end-date and recurrent-start-date, in seconds.

Example directory object advertising STAR support with one day star-min-cert-validity and one year star-max-renewal:

```
{
   "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"],
     "star-enabled": true,
     "star-min-cert-validity": 86400,
     "star-max-renewal":  31536000
   }
}
```

3.3. Fetching the Certificates

The certificate is fetched from the certificate endpoint, as per [I-D.ietf-acme-acme], Section 7.4.2.

```
GET /acme/cert/asdf HTTP/1.1
Host: acme-server.example.org
Accept: application/pkix-cert

HTTP/1.1 200 OK
Content-Type: application/pem-certificate-chain
Link: <https://example.com/acme/some-directory>;rel="index"
Not-Before: Mon, 1 Feb 2016 00:00:00 GMT
Not-After: Mon, 8 Feb 2016 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-----
```

The Server SHOULD include the "Not-Before" and "Not-After" HTTP headers 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.

To improve robustness, the next certificate MUST be made available by the ACME CA 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 latter also expires.

The server MUST NOT issue any additional certificates for this order beyond its recurrent-end-date.

Immediately after the order expires, the server MUST respond with 403 (Forbidden) to any requests to the certificate endpoint. The response SHOULD provide additional information using a problem document [RFC7807] with type "urn:ietf:params:acme:error:recurrentOrderExpired".

4. Operational Considerations

4.1. 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 browers 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, it is RECOMMENDED that site owners pre-date their Web facing certificates by 5 to 7 days. 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%. Note that exact choice is also likely to depend on the kind of clients 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 pre-dating to account for clock skews) no longer than 4 days.

4.2. Impact on Certificate Transparency (CT) Logs

Provided that the recommendations in Section 4.1 are followed, the increase in Certificate Transparency (CT) [RFC6962] log ingestion should be one order of magnitude in the worst case compared to the current state.

The input received from most members of the CT community when the issue was raised was that this should not represent a problem for the CT architecture.

5. Implementation Status

Note to RFC Editor: please remove this section before publication, including the reference to [RFC7942].

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 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 ACME Error Types

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

Type	Description	Reference
recurrentOrderCanceled	The short-term certificate is no longer available because the recurrent order has been explicitly canceled by the IdO	RFC XXXX
recurrentOrderExpired	The short-term certificate is no longer available because the recurrent order has expired	RFC XXXX
recurrentCancellationInvalid	A request to cancel a recurrent order that is not in state "valid" has been received	RFC XXXX

6.2. New ACME Order Object Fields

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

Field Name	Field Type	Configurable	Reference
recurrent	string	true	RFC XXXX
recurrent-start-date	string	true	RFC XXXX
recurrent-end-date	string	true	RFC XXXX
recurrent-certificate-validity	string	true	RFC XXXX

6.3. Not-Before and Not-After HTTP Headers

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

Header Field Name	Protocol	Status	Reference
Not-Before	http	standard	RFC XXXX
Not-After	http	standard	RFC XXXX

7. Security Considerations

7.1. 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 "recurrent-certificate-validity" parameter. An attacker can use this property to aggressively reduce the "recurrent-certificate-validity" (e.g. 1 sec.) jointly with other ACME attack vectors identified in Sec. 10 of [I-D.ietf-acme-acme]. 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 recurrent behavior cannot solve the above problem. The CA server needs complementary mitigation and specifically, it SHOULD enforce a minimum value on "recurrent-certificate-validity". Alternatively, the CA can set an internal certificate generation processes rate limit.

7.2. Additional Considerations TBD

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 Jon Peterson and Martin Thomson for helpful comments and discussions that have shaped this document.

9. References

9.1. Normative References

[I-D.ietf-acme-acme]
Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", draft-ietf-acme-acme-09 (work in progress), December 2017.

[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>.

[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>.

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.nir-saag-star]
Nir, Y., Fossati, T., and Y. Sheffer, "Considerations For Using Short Term Certificates", draft-nir-saag-star-00 (work in progress), October 2017.

[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-01 (work in progress), June 2017.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[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.w2spconf.com/2012/papers/w2sp12-final9.pdf>.

Appendix A. Document History

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

A.1. draft-ietf-acme-star-03

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

A.2. 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.3. 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.4. 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.5. 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.6. draft-sheffer-acme-star-01

- A terminology section.
- Some cleanup.

A.7. 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.8. 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
Nokia

EMail: thomas.fossati@nokia.com