Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

J. Peterson
Neustar
M. Barnes
iconectiv
D. Hancock
C. Wendt
Comcast

March 11, 2019

ACME Challenges Using an Authority Token
draft-ietf-acme-authority-token-02.txt

Abstract

Some proposed extensions to the Automated Certificate Management Environment (ACME) rely on proving eligibility for certificates through consulting an external authority that issues a token according to a particular policy. This document specifies a generic Authority Token challenge for ACME which supports subtype claims for different identifiers or namespaces that can be defined separately for specific applications.

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 September 12, 2019.

Table of Contents

1. Introduction

ACME [I-D.ietf-acme-acme] is a mechanism for automating certificate management on the Internet. It enables administrative entities to prove effective control over resources like domain names, and automates the process of generating and issuing certificates.

In some cases, proving effective control over an identifier requires an attestation from a third party who has authority over the resource, for example, an external policy administrator for a namespace other than the DNS application ACME was originally designed to support. In order to automate the process of issuing certificates for those resources, this specification defines a generic Authority Token challenge that ACME servers can issue in order to require clients to return such a token. The challenge contains a type indication that tells the client what sort of token it needs to acquire. It is expected that the Authority Token challenge will be usable for a variety of identifier types.

For example, the system of [I-D.ietf-acme-authority-token-tnauthlist] provides a mechanism that allows service providers to acquire certificates corresponding to a Service Provider Code (SPC) as defined in [RFC8226] by consulting an external authority responsible

for those codes. Furthermore, Communications Service Providers (CSPs) can delegate authority over numbers to their customers, and those CSPs who support ACME can then help customers to acquire certificates for those numbering resources with ACME. This can permit number acquisition flows compatible with those shown in [RFC8396]. Another, similar example would a mechanism that permits CSPs to delegate authority for particular telephone numbers to customers, as described in [I-D.ietf-acme-telephone].

2. Terminology

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

3. Challenges for an Authority Token

Proving that a device on the Internet has effective control over a non-Internet resource is not as straightforward as proving control over an Internet resources like a DNS zone or a web page. Provided that the issuer of identifiers in a namespace, or someone acting on the issuer's behalf, can implement a service that grants Authority Tokens to the people to whom it has issued identifiers, a generic token could be used as a response to an ACME challenge. This specification, therefore, defines an Authority Token issued by an authority over a namespace to an ACME client for delivery to a CA in response to a challenge. Authority over a hierarchical namespace can also be delegated, so that delegates of a root authority can themselves act as Token Authorities for certain types of names.

This architecture assumes a trust relationship between CAs and Token Authorities: that CAs are willing to accept the attestation of Token Authorities for particular types of identifiers as sufficient proof to issue a credential. It furthermore assumes that ACME clients have a relationship with Token Authorities which permits them to authenticate and authorize the issuance of Authority Tokens to the proper entities. This ACME challenge has no applicability to identifiers or authorities where those pre-associations cannot be assumed.

ACME challenges that support Authority Tokens therefore need to specify the type of token they require; CAs can even provide a hint in their challenges to ACME clients that tells them how to find a Token Authority who can issue tokens for a given namespace. This challenge type thus requires a new "tkauth-type" element, and may optionally supply a "token-authority" designating a location where tokens can be acquired. The set of "tkauth-type" values and the

semantic requirements for those tokens are tracked by an IANA registry.

3.1. Token Type Requirements

The IANA will maintain a registry of tkauth-types under a policy of Specification Required. In order to register a new tkauth-type, specifications must address the following requirements.

While Authority Token types do not need to be specific to a namespace, every token must carry enough information for a CA to determine the name that it will issue a certificate for. Some types of Authority Token types might be reusable for a number of different namespaces; other might be specific to a particular type of name. Therefore, in defining tkauth-types, future specifications must indicate how a token conveys to the CA the name(s) that the Token Authority is attesting that the ACME client controls.

While nothing precludes use cases where an ACME client is itself a Token Authority, an ACME client will typically need a protocol to request and retrieve an Authority Token. The Token Authority will require certain information from an ACME client in order to ascertain that it is the right entity to request a certificate for a particular name. The protocols used to request an Authority Token MUST convey to the Token Authority the identifier type and value from the ACME challenge, as well as the binding (see Section 3.3), and those MUST be reflected in the Authority Token. A baseline mechanism for how the Token Authority authenticates and authorizes ACME clients to receive Authority Tokens is given in Section 5.

Because the assignment of resources can change over time, demonstrations of authority must be regularly refreshed. Definitions of a tkauth-type MUST specify how they manage the freshness of authority assignments. Typically, a CA will expect a regular refreshing of the token.

3.2. Authority Token Scope

An Authority Token is used to answer a challenge from an ACME server, upon a request for the issuance of a certificate. It could be that the AT is requested from the Token Authority after a challenge has been received, or it could be that the AT was acquired prior to the initial ACME client request. A Token Authority could grant to a client a Token that has the exact same scope as the requested certificate; alternatively, an Authority Token could attest to all of the resources that the client is eligible to receive certificates for, which could be a superset of the scope of the requested certificate.

For example, imagine a case where an Authority for DNS names knows that a client is eligible to receive certificates for "example.com" and "example.net". The client asks an ACME server for a certificate for "example.com", the server directs the client to acquire an Authority Token from the Authority. When the client sends an acquisition request (see Section 5) to the Authority, the Authority could issue a token scoped just to "example.com", or a token that attests the client is eligible to receive certificates for both "example.com" or "example.net". The advantage of the latter is that if, at a later time (but one within the expiry of the JWT), the client wanted to acquire a certificate for "example.net", it would not have to return to the Authority, as the Token effectively pre-authorized the issuance of that certificate.

Applications of the Authority Token to different identifier types might require different scopes, so registrations of tkauth-types should be clear if and how a scope greater than that of the requested certificate would be conveyed in a token.

3.3. Binding Challenges

Applications that use the Authority Token need a way to correlate tokens issued by an Authority with the proper ACME client, to prevent replay or cut-and-paste attacks using a token issued for a different purpose. To mitigate this, Authority Tokens contain a binding signed by an Authority; an ACME server can use the binding to determine that a Token presented by a client was in fact granted by the Authority based on a request from the client, and not from some other entity.

Binding an Authority Token to a particular ACME account entails that the Token could be reused up until its expiry for multiple challenges issued by an ACME server. This might be a desirable property when using short-lived certificates, for example, or in any cases where the ACME server issues challenges more frequently that an Authority Token can or should issue tokens, or in cases where the Authority Token scope (see Section 3.2) is broad, so certificates with a more narrow scope may periodically be issued.

For some identifier types, it may be more appropriate to bind the Authority Token to a nonce specific to the challenge rather than to an ACME account fingerprint. Any specification of the use of the nonce for this purpose is left to the identifier type profile for the Authority Token.

4. ATC tkauth-type Registration

This draft registers a tkauth-type of "ATC", for the Authority Token Challenge. Here the "ATC" tkauth-type signifies a standard JWT token [RFC7519] using a JWS-defined signature string [RFC7515]. This may be used for any number of different identifier types given in ACME challenges. The "atc" element (defined below) lists the identifier type used by tokens based on ATC. The use of "ATC" is restricted to JWTs, if non-JWT tokens were desired for ACME challenges, a different tkauth-type should be defined for them.

For this ACME Authority Token usage of JWT, the payload of the JWT OPTIONALLY contain an "iss" indicating the Token Authority that generated the token, if the "x5u" element in the header does not already convey that information; typically, this will be the same location that appeared in the "token-authority" field of the ACME challenge. In order to satisfy the requirement for replay prevention the JWT MUST contain a "jti" element, and an "exp" claim. In addition to helping to manage replay, the "jti" provides a convenient way to reliably track with the same "ATC" Authority Token is being used for multiple challenges over time within its set expiry.

The JWT payload must also contain a new JWT claim, "atc", for Authority Token Challenge, which contains three mandatory elements in an array: the identifier type, the identifier value, and the binding. The identifier type and value are those given in the ACME challenge and conveyed to the Token Authority by the ACME client. Following the example of [I-D.ietf-acme-authority-token-tnauthlist], this could be the TNAuthList, as defined in [RFC8226], that the Token Authority is attesting. Practically speaking, that scope may comprise a list of Service Provider Code elements, telephone number range elements, and/or individual telephone numbers. For the purposes of the "ATC" tkauth-type, the binding is assumed to be a fingerprint of the ACME credential for the account used to request the certificate, but the specification of how the binding is generated is left to the identifier type profile for the Authority Token.

So for example:

```
{ "typ":"JWT",
  "alg":"ES256",
  "x5u":"https://authority.example.org/cert"}
  {
    "iss":"https://authority.example.org/authz",
    "exp":1300819380,
    "jti":"id6098364921",
    "atc":{"TnAuthList","F83n2a...avn27DN3==",
    "SHA256 56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3:BA:B9:19:81:F8:50:
     9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3"} }
```

Optionally, the "atc" element may contain a fourth element, "ca". If present, the "ca" element indicates that the Token Authority is granting permission to issue a certification authority certificate rather than an end-entity certificate for the names in question. This permits subordinate delegations from the issued certificate. The "atc" object in the example above would then look like:

```
 "atc":{"TnAuthList","ca","""F83n2a...avn27DN3==",
 "SHA256 56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3:BA:B9:19:81:F8:50:
  9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3"} }
```

5. Acquiring a Token

The acquisition of a Authority Token requires a network interface, apart from potential use cases where the entity that acts as an ACME client itself also acts as a Token Authority trusted by the ACME server. Implementations compliant with this specification MUST support an HTTPS REST interface for Authority Token acquisition as described below, though other interfaces MAY be supported as well.

5.1. Basic REST Interface

In order to request an Authority Token from a Token Authority, a client sends an HTTPS POST request. Different services may organize their web resources in domain-specific ways, but the resource locator should specify the account of the client, an identifier for the service provider, and finally a locator for the token.

```
POST /at/account/:id/token HTTP/1.1
Host: authority.example.com
Content-Type: application/json
```

The body of the POST request will minimally contain a JSON fingerprint object for the ACME client, for example:

```
{
  "fingerprint":"SHA256 56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3 \
   :BA:B9:19:81:F8:50:9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3"
}
```

It is understood if this minimal JSON object is provided that the client is requesting the Token Authority to issue a token that attests the entire scope of authority to which the client is entitled. The client may also request an AT with some subset of its own authority via an optional "scope" element in this JSON object. The way that "scope" is defined will necessarily be specific to the identifier type. For the TNAuthlist identifier type, for example, an object requesting an AT with authority for only a single telephone number might look like:

```
{
 "fingerprint":"SHA256 56:3E:CF:AE:83:CA:4D:15:B0:29:FF:1B:71:D3 \
   :BA:B9:19:81:F8:50:9B:DF:4A:D4:39:72:E2:B1:F0:B9:38:E3",
 "scope":"12125551000"
}
```

Finally, the JSON object may also contain a optional element "ca" which signifies that the client is requesting that the Token Authority issue an AT with the "ca" flag set, as described in Section 4.

After an HTTPS-level challenge to verify the identity of the client and subsequently making an authorization decision, in the success case the Token Authority returns a 200 OK with a body of type "application/json" containing the Authority Token.

6. Using an Authority Token in a Challenge

Taking the identifier example of TNAuthList from [I-D.ietf-acme-authority-token-tnauthlist], an ACME for this tkauth-type challenge might for example look as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json
Link: <https://example.com/acme/some-directory>;rel="directory"

{
 "status": "pending",

 "identifier": {
    "type": "TNAuthList",
    "value": "F83n2a...avn27DN3=="
  },
  "challenges": [
  {
    "type": "tkauth-01",
    "tkauth-type": "ATC",
    "token-authority": "https://authority.example.org/authz",
    "url": "https://boulder.example.com/authz/asdf/0"
    "token": "IlirfxKKXAsHtmzK29Pj8A" }
  ],
}
```

Entities receiving this challenge know that they can, as a proof, acquire an ATC token from the designated Token Authority (specified in the "token-authority" field), and that this authority can provide tokens corresponding to the identifier type of "TNAuthList".

Once the ATC has been acquired by the ACME Client, it can be posted back to the URL given by the ACME challenge.

```
POST /acme/authz/asdf/0 HTTP/1.1
Host: boulder.example.com
Content-Type: application/jose+json

 {
  "protected": base64url({
  "alg": "ES256",
  "kid": "https://boulder.example.com/acme/reg/asdf",
  "nonce": "Q_s3MWoqT05TrdkM2MTDcw",
  "url": "https://boulder.example.com/acme/authz/asdf/0"
   }),
  "payload": base64url({
  "ATC": "evaGxfADs...62jcerQ"
     }),
  "signature": "5wUrDI3eAaV4wl2Rfj3aC0Pp--XB3t4YYuNgacv_D3U"
 }
```

The "ATC" field in this response contains the Authority Token.

7. Acknowledgements

We would like to thank you for your contributions to this problem statement and framework.

8. IANA Considerations

This document requests that the IANA registers a new ACME identifier type (per [I-D.ietf-acme-acme]) for the label "atc", for which the reference is [RFCThis].

This document further requests that the IANA create a registry for "token types" as used in these challenges, following the requirements in Section 3.1, pre-populated with the label of "ATC" per Section 4 with a value of [RFCThis].

9. Security Considerations

Per the guidance in [I-D.ietf-acme-acme], ACME transactions MUST use TLS, and similarly the HTTPS REST transactions used to request and acquire authority tokens MUST use TLS. These measures are intended to prevent the capture of Authority Tokens by eavesdroppers.

The capture of Authority Tokens by an adversary could enable an attacker to acquire a certificate from a CA. Therefore, all Authority Tokens MUST contain a field that identifies to the CA which ACME client requested the token from the authority; here that is the fingerprint specified in Section 4). All Authority Tokens must specify an expiry (of the token itself as proof for a CA, as opposed to the expiry of the name), and for some application, it may make sense of that expiry to be quite short. Any protocol used to retrieve Authority Tokens from an authority MUST use confidentiality to prevent eavesdroppers from acquiring an Authority Token.

10. 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-18 (work in progress), December 2018.

[I-D.ietf-acme-authority-token-tnauthlist]
Wendt, C., Hancock, D., Barnes, M., and J. Peterson, "TNAuthList profile of ACME Authority Token", draft-ietf-acme-authority-token-tnauthlist-01 (work in progress), October 2018.

[I-D.ietf-acme-service-provider]
Barnes, M. and C. Wendt, "ACME Identifiers and Challenges for VoIP Service Providers", draft-ietf-acme-service-provider-02 (work in progress), October 2017.

[I-D.ietf-acme-star]
Sheffer, Y., Lopez, D., Dios, O., Pastor, A., and T. Fossati, "Support for Short-Term, Automatically-Renewed (STAR) Certificates in Automated Certificate Management Environment (ACME)", draft-ietf-acme-star-05 (work in progress), March 2019.

[I-D.ietf-acme-telephone]
Peterson, J. and R. Barnes, "ACME Identifiers and Challenges for Telephone Numbers", draft-ietf-acme-telephone-01 (work in progress), October 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>.

[RFC7340] Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Telephone Identity Problem Statement and Requirements", RFC 7340, DOI 10.17487/RFC7340, September 2014, <https://www.rfc-editor.org/info/rfc7340>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC8224] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 8224, DOI 10.17487/RFC8224, February 2018, <https://www.rfc-editor.org/info/rfc8224>.

[RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <https://www.rfc-editor.org/info/rfc8225>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity Credentials: Certificates", RFC 8226, DOI 10.17487/RFC8226, February 2018, <https://www.rfc-editor.org/info/rfc8226>.

[RFC8396] Peterson, J. and T. McGarry, "Managing, Ordering, Distributing, Exposing, and Registering Telephone Numbers (MODERN): Problem Statement, Use Cases, and Framework", RFC 8396, DOI 10.17487/RFC8396, July 2018, <https://www.rfc-editor.org/info/rfc8396>.

Authors' Addresses

Jon Peterson
Neustar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@team.neustar

Mary Barnes
iconectiv

Email: mary.ietf.barnes@gmail.com

David Hancock
Comcast

Email: davidhancock.ietf@gmail.com

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net