

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

C. Wendt
D. Hancock
Comcast
M. Barnes
iconectiv
J. Peterson
Neustar Inc.
October 22, 2018

**TNAuthList profile of ACME Authority Token
draft-ietf-acme-authority-token-tnauthlist-01**

Abstract

This document defines a profile of the Automated Certificate Management Environment (ACME) Authority Token for the automated and authorized creation of certificates for VoIP Telephone Providers to support Secure Telephony Identity (STI) using the TNAuthList defined by STI certificates.

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 April 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	ACME new-order identifiers for TNAuthList	3
4.	TNAuthList Identifier Authorization	5
5.	TNAuthList Authority Token	6
5.1.	"iss" claim	7
5.2.	"exp" claim	7
5.3.	"jti" claim	7
5.4.	"atc" claim	7
5.5.	Acquiring the token from the Token Authority	8
5.6.	Token Authority Responsibilities	8
6.	Validating the TNAuthList Authority Token	8
7.	Usage Considerations	9
7.1.	Large number of Non-contiguous TNAuthList values	9
8.	Security Considerations	9
9.	IANA Considerations	9
10.	Acknowledgements	10
11.	References	10
11.1.	Normative References	10
11.2.	Informative References	11
	Authors' Addresses	11

[1.](#) Introduction

[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.

[[I-D.ietf-acme-authority-token](#)] extends ACME to provide a general method of extending the authority and authorization of entities to control a resource via a third party Token Authority beyond the Certification Authority.

This document addresses the STIR problem statement [[RFC7340](#)] which identifies the need for Internet credentials that can attest authority for the originator of VoIP calls in order to detect impersonation, which is currently an enabler for common attacks associated with illegal robocalling, voicemail hacking, and swatting. These credentials are used to sign PASSporTs [[RFC8225](#)], which can be carried in using protocols such as SIP [[RFC8224](#)]. Currently, the

only defined credentials for this purpose are the certificates specified in [\[RFC8226\]](#).

[\[RFC8226\]](#) describes certificate extensions suitable for associating telephone numbers and service provider codes with certificates. Specifically, the TN Authorization List defined in [\[RFC8226\]](#) [Section 9](#), defines the ability to associate a STI certificate with a specific set of Service Provider Codes (SPCs), Telephone Numbers (TNs), or Telephone Number ranges (TN ranges). Typically, these identifiers have been assigned to a Communications Service Provider (CSP) that is authorized to use a set of telephone numbers or telephone number ranges in association with a Service Provider Code as defined in [\[RFC8226\]](#). The SPC is a unique code or string managed by a national regulatory body that has the authority over those code-to-CSP associations.

2. Terminology

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

3. ACME new-order identifiers for TNAuthList

In [\[I-D.ietf-acme-acme\]](#), Section 7.4 defines the procedure that an ACME client uses to order a new certificate from a Certification Authority. The new-order request contains an identifier field that specifies the identifier objects the order corresponds to. This draft defines a new type of identifier object called TNAuthList. A TNAuthList identifier contains the identity information to be populated in the TN Authorization List of the new certificate. For the TNAuthList identifier, the new-order request MUST include a type set to the string "TNAuthList". The value of the TNAuthList identifier MUST be set to the details of the TNAuthList requested.

The format of the string that represents the TNAuthList MUST be constructed as a base64 [\[RFC4648\]](#) encoding of the TN Authorization List certificate extension ASN.1 object. The TN Authorization List certificate extension ASN.1 syntax is defined in [\[RFC8226\]](#) [section 9](#).

An example of an ACME order object "identifiers" field containing a TNAuthList certificate would look as follows,

```
"identifiers": [{"type": "TNAuthList", "value": "F83n2a...avn27DN3=="}]
```

where the "value" object string represents the arbitrary length base64 encoded string.

A full new-order request would look as follows,

```
POST /acme/new-order HTTP/1.1
```

```
Host: example.com
```

```
Content-Type: application/jose+json
```

```
{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/1",
    "nonce": "5XJ1L3lEkMG7tR6pA00clA",
    "url": "https://example.com/acme/new-order"
  }),
  "payload": base64url({
    "identifiers": [{"type": "TNAuthList", "value": "F83n2a...avn27DN3==" }],
    "notBefore": "2016-01-01T00:00:00Z",
    "notAfter": "2016-01-08T00:00:00Z"
  }),
  "signature": "H6ZXtGjTZyUnPeKn...wEA4Tk1Bdh3e454g"
}
```

On receiving a valid new-order request, the CA creates an authorization object containing the challenge that the ACME client must satisfy to demonstrate authority for the identifiers specified by the new order (in this case, the TNAuthList identifier). The CA adds the authorization object URL to the "authorizations" field of the order object, and returns the order object to the ACME client in the body of a 201 (Created) response.

```
HTTP/1.1 201 Created
```

```
Replay-Nonce: MYAuvOpaoIiywTezizk5vw
```

```
Location: https://example.com/acme/order/1234
```

```
{
  "status": "pending",
  "expires": "2015-03-01T14:09:00Z",

  "notBefore": "2016-01-01T00:00:00Z",
  "notAfter": "2016-01-08T00:00:00Z",
  "identifiers": [{"type": "TNAuthList",
    "value": "F83n2a...avn27DN3==" }],

  "authorizations": [
    "https://example.com/acme/authz/1234"
  ],
  "finalize": "https://example.com/acme/order/1234/finalize"
}
```


4. TNAuthList Identifier Authorization

On receiving the new-order response, the ACME client queries the referenced authorization object to obtain the challenges for the identifier contained in the new-order request as shown in the following example request and response.

```
POST /acme/authz/1234 HTTP/1.1
Host: example.com
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": " https://example.com/acme/acct/1",
    "nonce": "uQpSjlRb4vQVCjVYAyyUWg",
    "url": "https://example.com/acme/authz/1234",
  }),
  "payload": "",
  "signature": "nuSDISbWG8mMgE7H...QyVUL68yzf3Zawps"
}

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

{
  "status": "pending",
  "expires": "2018-03-03T14:09:00Z",

  "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": "I1irfxKKXAsHtmzK29Pj8A"
    }
  ]
}
```

When processing a certificate order containing an identifier of type "TNAuthList", a CA MUST use the Authority Token challenge mechanism

defined in [[I-D.ietf-acme-authority-token](#)] to verify that the requesting ACME client has authenticated and authorized control over the requested resources represented by the "TNAuthList" value.

The challenge "token-authority" parameter is optional and only used in cases where the VoIP telephone network requires the CA to identify the Token Authority. This is currently not the case for the SHAKEN [[ATIS-1000080](#)] certificate framework governance, but may be used by other frameworks. If a "token-authority" parameter is present, then the ACME client MAY use the "token-authority" value to identify the URL representing the Token Authority that will provide the TNAuthList Authority Token response to the challenge. If the "token-authority" parameter is not present, then the ACME client MUST identify the Token Authority based on locally configured information or local policies.

The ACME client MUST respond to the challenge by posting the TNAuthList Authority Token to the challenge URL identified in the returned ACME authorization object, an example of which follows.

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

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/1",
    "nonce": "Q_s3MwoqT05TrdkM2MTDcw",
    "url": "https://boulder.example.com/acme/authz/asdf/0"
  }),
  "payload": base64url({
    "ATC": "DGyRejmCefe7v4N...vb29HhjjLPSgawiE"
  }),
  "signature": "9cbg5J01Gf5YLjjz...SpkUfcdPai9uVYYQ"
}
```

The specifics of the construction of the TNAuthList specific "ATC" token is defined in the next section.

5. TNAuthList Authority Token

The Telephone Number Authority List Authority Token (TNAuthList Authority Token) is an extension of the ACME Authority Token defined in [[I-D.ietf-acme-authority-token](#)].

The TNAuthList Authority Token Protected header MUST comply with the Authority Token Protected header as defined in [\[I-D.ietf-acme-authority-token\]](#).

The TNAuthList Authority Token Payload MUST include the mandatory claims and MAY include the optional claims defined for the Authority Token detailed in the next subsections.

[5.1.](#) "iss" claim

The "iss" claim is an optional claim. It can be used as a URL identifying the Token Authority that issued the TNAuthList Authority Token beyond the "x5u" Header claim that identifies the location of the certificate of the Token Authority used to validate the TNAuthList Authority Token.

[5.2.](#) "exp" claim

The "exp" claim contains the DateTime value of the ending date and time that the TNAuthList Authority Token expires.

[5.3.](#) "jti" claim

The "jti" claim contains a unique identifier for this TNAuthList Authority Token transaction.

[5.4.](#) "atc" claim

The "atc" claim is the only claim specifically defined in this document. It contains an array of three elements; a string set to "TNAuthList", the TNAuthList identifier "value" string, and a fingerprint.

The "fingerprint" value is a fingerprint, as defined in [\[RFC4949\]](#) of the ACME account credentials. Specifically, the fingerprint value is a secure one-way hash of the Distinguished Encoding Rules (DER) form of the public key corresponding to the key pair the SP used to create the account with the ACME server. The fingerprint value consists of the name of the hash function, which shall be 'SHA256' for this specification, followed by the hash value itself. The hash value is represented as a sequence of uppercase hexadecimal bytes, separated by colons. The number of bytes is defined by the hash function.

An example of the TNAuthList Authority Token is as follows,


```
{ "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"]  
}
```

Similar to the definition for the TNAuthList identifier "value" string, the identifier value in the "atc" claim must contain the base64 encoding of the TN Authorization List certificate extension ASN.1 object.

5.5. Acquiring the token from the Token Authority

The specifics of how the token is acquired from the authority is out of the scope of this document

5.6. Token Authority Responsibilities

When the Token Authority creates the TnAuthList Authority Token, it is the responsibility of the Token Authority to validate that the information contained in the ASN.1 TNAuthList accurately represents the SPC or telephone number resources the ACME client is authorized to represent.

6. Validating the TNAuthList Authority Token

Upon receiving a response to the challenge, the ACME server MUST perform the following steps to determine the validity of the response.

- o Verify that the token contained in the Payload "ATC" field is an TNAuthList Authority Token.
- o Verify the TNAuthList Authority Token signature using the public key of the certificate referenced by the token's "x5u" parameter.
- o Verify that "atc" claim contains an identifier type of "TNAuthList",

- o Verify that the "atc" claim contains the equivalent base64 encoded TNAuthList certificate extension string value as the Identifier specified in the original challenge.
- o Verify that the remaining claims are valid (e.g., verify that token has not expired)
- o Verify that the "atc" claim "fingerprint" is valid

If all steps in the token validation process pass, then the CA MUST set the challenge object "status" to "valid". If any step of the validation process fails, the "status" in the challenge object MUST be set to "invalid".

7. Usage Considerations

7.1. Large number of Non-contiguous TNAuthList values

There are many scenarios and reasons to have various combinations of SPCs, TNS, and TN Ranges. [RFC8226] has provided a somewhat unbounded set of combinations. It's possible that a complex non-contiguous set of telephone numbers are being managed by a CSP. Best practice may be simply to split a set of non-contiguous numbers under management into multiple STI certificates to represent the various contiguous parts of the greater non-contiguous set of TNS, particularly if length of the set of values in identifier object grows to be too large.

8. Security Considerations

TBD

9. IANA Considerations

This document requests the addition of a new identifier object type that can be present in the identifier field of the ACME authorization object defined in [I-D.ietf-acme-acme].

+-----+-----+
Label Reference
+-----+-----+
TNAuthList RFCThis
+-----+-----+

10. Acknowledgements

We would like to thank Richard Barnes and Russ Housley for valuable contributions to this document.

11. References

11.1. 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-16](#) (work in progress), October 2018.
- [I-D.ietf-acme-authority-token]
Peterson, J., Barnes, M., Hancock, D., and C. Wendt, "ACME Challenges Using an Authority Token", [draft-ietf-acme-authority-token-00](#) (work in progress), July 2018.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 4648](#), DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, [RFC 4949](#), DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [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>>.
- [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>>.

11.2. Informative References

[ATIS-1000074]

ATIS/SIP Forum NNI Task Group, "Signature-based Handling of Asserted information using toKENS (SHAKEN) <https://access.atis.org/apps/group_public/download.php/32237/ATIS-1000074.pdf>", January 2017.

[ATIS-1000080]

ATIS/SIP Forum NNI Task Group, "Signature-based Handling of Asserted information using toKENS (SHAKEN) Governance Model and Certificate Management <https://access.atis.org/apps/group_public/download.php/32237/ATIS-1000080.pdf>", July 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>>.

Authors' Addresses

Chris Wendt
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: chris-ietf@chriswendt.net

David Hancock
Comcast

Email: davidhancock.ietf@gmail.com

Mary Barnes
iconectiv

Email: mary.ietf.barnes@gmail.com

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

Email: jon.peterson@neustar.biz