Automated Certificate Management Environment
Internet-Draft
Intended status: Standards Track
Expires: 16 November 2025

A. A. Chariton
A. A. Omidi
Independent Contributor
J. Kasten
Snowflake
F. Loukos
S. A. Janikowski
Google

15 May 2025

Automated Certificate Management Environment (ACME) DNS Labeled With ACME Account ID Challenge
draft-ietf-acme-dns-account-label-01

Abstract

This document outlines a new DNS-based challenge type for the ACME protocol that enables multiple independent systems to authorize a single domain name concurrently. By adding a unique label to the DNS validation record name, the dns-account-01 challenge avoids CNAME delegation conflicts inherent to the dns-01 challenge type. This is particularly valuable for multi-region or multi-cloud deployments that wish to rely upon DNS-based domain control validation and need to independently obtain certificates for the same domain.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://github.com/aaomidi/draft-ietf-acme-dns-account-challenge. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-acme-dns-account-label/.

Discussion of this document takes place on the WG Working Group mailing list (mailto:acme@ietf.org), which is archived at https://datatracker.ietf.org/wg/acme/about/. Subscribe at https://www.ietf.org/mailman/listinfo/acme/.

Source for this draft and an issue tracker can be found at https://github.com/aaomidi/draft-ietf-acme-dns-account-challenge.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 16 November 2025.

Copyright Notice

Table of Contents

1. Introduction

The dns-01 challenge specified in section 8.4 of [RFC8555] uses a single DNS authorization label (_acme-challenge) for domain validation. This single-label approach creates a limitation in domain validation: each domain can only delegate its validation to one ACME client at a time. Since delegation requires the use of CNAME records, of which only one can exist per DNS name, operators are forced to choose a single ACME challenge solver for their domain name.

This limitation becomes particularly problematic in modern deployment architectures. In multi-region deployments, separate availability zones serve the same content while avoiding cross-zone dependencies. These zones need to independently obtain and manage certificates for the same domain name. Similarly, during zero-downtime migrations, two different infrastructure setups may coexist for extended periods, with both requiring access to valid certificates. Other use cases include multi-CDN deployments and the provision of backup certificates for use when an active certificate must be quickly revoked.

This document specifies a new challenge type: dns-account-01, which addresses these operational needs. The dns-account-01 challenge incorporates the ACME account URL into the DNS validation record name, allowing multiple independent ACME clients to perform domain validation concurrently. Since these authorization labels depend on the ACME account KID ([RFC8555], Section 7.3), operators can generate and configure the necessary DNS records in advance.

This RFC does not deprecate the dns-01 challenge specified in [RFC8555]. The ability to complete the dns-account-01 challenge requires ACME server operators to deploy new code, making adoption of this challenge an opt-in process.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. DNS-ACCOUNT-01 Challenge

The dns-account-01 challenge allows a client to prove control of a domain name by provisioning a TXT resource record containing a designated value for a specific validation domain name. It leverages the ACME account URL to construct a unique but stable validation domain name. The ACME server validates control of the domain name by performing one or more DNS queries to this validation domain name, following CNAME records, to arrive at one or more TXT resource record. The ACME server verifies that the contents of one or more of these TXT record(s) match the digest value of the key authorization that is constructed from the token value provided in the challenge.

3.1. Challenge Definition

The challenge object contains the following fields:

* type (required, string): The string "dns-account-01".

* token (required, string): A random value that uniquely identifies the challenge. This value MUST have at least 128 bits of entropy. It MUST NOT contain any characters outside the base64url alphabet, including padding characters ("="). See [RFC4086] for additional information on additional requirements for secure randomness.

Example challenge object:

```
{
    "type": "dns-account-01",
    "url": "https://example.com/acme/chall/i00MGYwLWIx",
    "status": "pending",
    "token": "ODE4OWY4NTktYjhmYS00YmY1LTk5MDgtZTFjYTZmNjZlYTUx"
}
```

3.2. Challenge Fulfillment

To fulfill this challenge, a client performs the following steps:

1. Construct Key Authorization - Construct a key authorization [RFC8555], Section 8.1 from the token value provided in the challenge and the client’s account key - Compute the SHA-256 digest [FIPS180-4] of the key authorization

2. DNS Record Creation - Construct the validation domain name by prepending the following two labels to the domain name being validated:

    "_" || base32(SHA-256(<ACCOUNT_URL>)[0:10]) || "._acme-challenge"

- SHA-256 is the SHA hashing operation defined in [RFC6234]
- [0:10] is the operation that selects the first ten bytes (bytes 0 through 9 inclusive) from the previous SHA-256 operation
- base32 is the operation defined in [RFC4648]
- ACCOUNT_URL is defined in [RFC8555], Section 7.3 as the value in the Location header field
- The || operator indicates concatenation of strings

* Provision a DNS TXT record with the base64url digest value under the constructed domain validation name

1. Challenge Response - Respond to the ACME server with an empty object ({}) to acknowledge that the challenge can be validated by the server

Example DNS record for domain example.org with account URL https://example.com/acme/acct/ExampleAccount:

```
_ujmmovf2vn55tgye._acme-challenge.example.org 300 IN TXT "LoqXcYV8...jxAjEuX0.9jg46WB3...
fm21mqTI"
```

Example response to server:

```
POST /acme/chall/Rg5dV14Gh1Q
Host: example.com
Content-Type: application/jose+json

{
  "protected": base64url({
    "alg": "ES256",
    "kid": "https://example.com/acme/acct/ExampleAccount",
    "nonce": "SS2sSl1PtspvFZ08kNtzKd",
    "url": "https://example.com/acme/chall/Rg5dV14Gh1Q"
  }),
  "payload": base64url({}),
  "signature": "Q1bURgJoEslbD1c5...3pYdSMLio57mQNN4"
}
```

3.3. Server Validation

Upon receiving the challenge response, the server:

1. Performs the typical JWS validation.

2. Constructs and stores the key authorization

3. Computes the SHA-256 digest [FIPS180-4] of the stored key authorization

4. Computes the validation domain name using the KID value from the JWS message

5. Queries for TXT records at the validation domain name

6. Verifies that one TXT record matches the computed digest value

The validation succeeds only if all verifications pass. The server MUST mark the challenge as invalid if any verification fails.

The client SHOULD de-provision the resource record(s) provisioned for this challenge once the challenge is complete, i.e., once the "status" field of the challenge has the value "valid" or "invalid".

3.4. Errors

The server SHOULD follow the guidelines set in [RFC8555], Section 6.7 for error conditions that occur during challenge validation.

If the server is unable to find a TXT record for the validation domain name, it SHOULD include the account URL it used to construct the validation domain name in the problem document. Clients MUST NOT use or rely on the presence of this field to construct the validation domain name.

3.5. Implementation Considerations

As this challenge creates strong dependency on the kid account identifier, the server SHOULD ensure that the account identifier is not changed during the lifetime of the account. This contains the entire URI, including the ACME endpoint domain name, port, and full HTTP path.

4. Security Considerations

The same security considerations apply for the integrity of authorizations ([RFC8555], Section 10.2) and DNS security ([RFC8555], Section 11.2) as in the original specification for dns-01.

To allow for seamless account key rollover without the label changing, the dynamic part of the label depends on the ACME account and not the account key. This allows for long-lived labels, without the security considerations of keeping the account key static.

In terms of the construction of the account label prepended to the domain name, there is no need for a cryptographic hash. The goal is to simply create a long-lived and statistically distinct label of minimal size. SHA-256 was chosen due to its existing use in the dns-01 challenge ([RFC8555], Section 8.1).

The first 10 bytes were picked as a tradeoff: the value needs to be short enough to stay lower than the size limits for DNS ([RFC1035], Section 2.3.4), long enough to provide sufficient probability of collision avoidance across ACME accounts, and just the right size to have Base32 require no padding. As the algorithm is used for a uniform distribution of inputs, and not for integrity, we do not consider the trimming a security issue.

5. IANA Considerations

5.1. ACME Validation Method

The "ACME Validation Methods" registry is to be updated to include the following entries:

label: dns-account-01
identifier-type: dns
ACME: Y
Reference: This document

6. References

6.1. Normative References

[FIPS180-4] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", August 2015, <https://csrc.nist.gov/publications/detail/fips/180/4/final>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[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/rfc/rfc2119>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/rfc/rfc4086>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/rfc/rfc6234>.

[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/rfc/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/rfc/rfc8555>.

6.2. Informative References

[I-D.draft-ietf-dnsop-domain-verification-techniques] Sahib, S. K., Huque, S., Wouters, P., and E. Nygren, "Domain Control Validation using DNS", Work in Progress, Internet-Draft, draft-ietf-dnsop-domain-verification-techniques-06, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-domain-verification-techniques-06>.

Acknowledgments

Authors' Addresses

Antonios A. Chariton
Independent Contributor
Email: daknob@daknob.net

Amir A. Omidi
Independent Contributor
Email: amir@aaomidi.com

James Kasten
Snowflake
Email: james.kasten@snowflake.com

Fotis Loukos
Google
Email: fotisl@google.com

Stanislaw A. Janikowski
Google
Email: stanwise@google.com