Auto-discovery mechanism for ACME client configuration

Abstract

A significant impediment to the widespread adoption of the Automated Certificate Management Environment (ACME) [RFC8555] is that ACME clients need to be pre-configured with the URL of the ACME server to be used. This often leaves domain owners at the mercy of their hosting provider as to which Certification Authorities (CAs) can be used. This specification provides a mechanism to bootstrap ACME client configuration from a domain's DNS CAA Resource Record [RFC8659], thus giving control of which CA(s) to use back to the domain owner.

Specifically, this document specifies two new extensions to the DNS CAA Resource Record: the "discovery" and "priority" parameters. Additionally, it registers the URI "/.well-known/acme" at which all compliant ACME servers will host their ACME directory object. By retrieving instructions for the ACME client from the authorized CA(s), this mechanism allows for the domain owner to configure multiple CAs in either load-balanced or fallback prioritizations which improves user preferences and increases diversity in certificate issuers.

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://vanbroup.github.io/acme-auto-discovery/draft-vanbrouwershaven-acme-auto-discovery.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-vanbrouwershaven-acme-auto-discovery/.

Discussion of this document takes place on the ACME 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/vanbroup/acme-auto-discovery.

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 14 April 2024.

Copyright Notice

Copyright (c) 2023 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Conventions and Definitions
2. Introduction
3. Protocol Overview
4. CAA Record
   4.1. Extensions to the CAA Record
      4.1.1. The "discovery" Parameter
      4.1.2. The "priority" Parameter
   4.2. Examples
5. ACME Client Configuration
6. ACME Client Behavior
   6.1. Certificates with multiple domain names
       6.1.1. Selecting a CA through Compromise
7. Implementation Considerations

7.1. External Account Binding

7.1.1. Internal Account Binding using Domain Control Validation

7.1.2. Internal Account Binding using Email Address

7.2. Terms of Service and Acceptance

7.2.1. Implicit Acceptance of Terms of Service

7.2.2. Acceptance Through CAA Parameter

8. IANA Considerations

8.1. Well-Known URI for the ACME Directory

8.2. CAA Parameters

9. Security Considerations

9.1. Risks with Auto-Discovery of Authorized CAs

9.1.1. Unexpected Certificate Issuance

9.1.2. Issuance by the ‘wrong’ authorized CA

9.2. Malicious ACME Servers

9.3. ACME Keys

9.4. Use of DNS Security

10. References

10.1. Normative References

10.2. Informative References

Acknowledgments

Authors' Addresses

1. 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.

2. Introduction

The ACME protocol [RFC8555] offers a powerful framework for automating the issuance and validation of certificates, eliminating the need for user intervention. This capability has significantly streamlined the process of obtaining certificates for servers and infrastructure software. However, in shared environments, where multiple entities coexist, users often face limitations in modifying ACME client configurations. Consequently, they are forced to rely on manual certificate management systems or default Certificate Authorities (CAs) preconfigured for the shared environment.

This document introduces a mechanism to address the aforementioned challenge by enabling the automatic discovery of ACME client configurations for relevant domain names within a shared environment. The solution leverages the DNS Certification Authority Authorization (CAA) Resource Record [RFC8659] to identify the
authorized Certification Authorities capable of issuing certificates for a specific domain name or set of domain names.

By leveraging the power of CAA records, this mechanism empowers users with enhanced control and flexibility. Users can specify their preferences, choose from a broader range of certificate issuers, and even designate a backup Certification Authority.

This approach facilitates a more diverse and adaptable certificate management process within shared and managed environments. This document provides a detailed description of the proposed mechanism, along with its benefits and considerations.

Additionally, it outlines the security aspects associated with the use of CAA records for ACME client configuration discovery. Finally, this document presents IANA considerations and references relevant normative and informative documents.

There is previous work in this area in [I-D.tweedale-acme-discovery] which attempts to solve a similar ACME discovery problem for private networks and private ACME CAs.

**RFC Editor's Note:** Please remove this section prior to publication of a final version of this document.

The authors of the document considered both SRV and URI DNS resource record types as an alternative to the proposed Well-Known URI, see also: https://github.com/vanbroup/acme-auto-discovery/issues/15
1. The ACME client initiates a DNS lookup to retrieve the CAA record(s) according to [RFC8659].

   1. The DNS resolver responds with the CAA record for each domain, specifying the authorized CAs capable of issuing certificates, along with their priorities and other optional parameters.

   2. The ACME client analyzes the valid CAA records for the domain, ignoring any it cannot process, and selects the CA with the highest priority.

   3. The ACME client will download the ACME directory from the well-known location of the issuer-domain-name of the selected CA (https://[issuer-domain-name]/.well-known/acme)

   4. If the directory object indicates that an External Account Binding is required, but this is not configured on the ACME
client, the client will try to determine an alternative common CA in step 2.

1. If no alternative CA can be found, the process with end with a failure and the user SHOULD be notified.

5. The ACME client continues normal operation according to \[RFC8555\].

4. CAA Record

4.1. Extensions to the CAA Record

This document defines the "discovery" and "priority" CAA parameters in the context of ACME auto discovery.

4.1.1. The "discovery" Parameter

The "discovery" parameter is used to control the auto-discovery functionality of the record in the context of this document.

The value of this parameter, if specified, MUST be a lower-case Boolean, where "true" indicates that this record can be used for auto discovery, and "false" indicates that this record should not be used for auto discovery.

When this parameter is not specified the client MUST assume that discovery is enabled.

4.1.2. The "priority" Parameter

The value of this parameter, if specified, MUST contain an integer greater than zero, where the value "1" represents the highest priority, and subsequent values like "2", "3", and so on, indicate progressively lower priorities. Where records specify an equal priority, their usage SHOULD be randomized.

In the case that this parameter is not specified, the entry will be considered to have a lower priority than all entries which specify any priority.

4.2. Examples

This section shows some examples of how CAA records can be configured in the context of ACME auto discovery. CAA records are used to authorize Certification Authorities (CAs) to issue certificates for a specific domain name.

A simple CAA record allows the issuance of certificates from a single designated CA. In the following example, the CAA record for
the domain "example.com" authorizes the CA "ca.example" to issue certificates. However, it does not specify any backup CA. Consequently, if the authorized CA is unable to issue the requested certificate, the certificate issuance will fail.

example.com CAA 0 issue "ca.example"

By default, when multiple CAA records are present, the CAs are randomized to distribute the load. However, some users may have preferences regarding the order in which CAs are attempted for certificate issuance. To explicitly specify the order, the "priority" parameter can be used.

In the next example, the domain "example.com" has two CAA records. The CAA record with "ca2.example" has a higher priority value of 1, indicating it should be attempted first. The CAA record with "ca1.example" has a lower priority value of 2, indicating it should be attempted second.

example.com CAA 0 issue "ca1.example; priority=2"
example.com CAA 0 issue "ca2.example; priority=1"

CAA records that do not explicitly specify a priority are automatically assigned the lowest priority. In cases where multiple CAA records have the same priority, the usage will be randomized.

Consider the following example, where the domain "example.com" has three CAA records. The CAA record with "ca1.example" has no specified priority, and thus it is assigned the lowest priority. The CAA records with "ca2.example" and "ca3.example" both have a priority of 1. In this scenario, the ACME client will first attempt to obtain its configuration from either "ca2.example" or "ca3.example", selected at random. If both of those fail, it will fall back to "ca1.example". If all attempts fail, the certificate issuance will ultimately fail.

example.com CAA 0 issue "ca1.example"
example.com CAA 0 issue "ca2.example; priority=1"
example.com CAA 0 issue "ca3.example; priority=1"

Furthermore, it is possible to configure CAA records to indicate a preference for specific types of certificates. In the following example, the domain "example.com" prefers Extended Validation (EV) certificates issued by "ca1.example". If the issuance of an EV certificate fails, the ACME client will attempt to obtain any type of certificate from "ca1.example". If that also fails, it will then try to obtain any type of certificate from "ca2.example".
When an ACME client requests the issuance of a wildcard certificate, the issuewild CAA property takes precedence over each issue property when specified, see also section 4.3 of RFC8659. The following example specifies that only ca3.example can issue certificates for "example.com" or "sub.example.com". However ca3.example is not permitted to issue for "example.com" or "sub.example.com". In the case the issuewild property was not specified all listed CAs would be authorized to issue wildcards for this domain.

```
example.com CAA 0 issue "ca1.example; priority=1 validationmethods=ca-ev
example.com CAA 0 issue "ca1.example; priority=2"
example.com CAA 0 issue "ca2.example; priority=3"
```

To disable auto discovery for a particular record users can set the discovery parameter to false, in the example below this will ensure that the ACME client will only try to obtain a certificate from ca1.example and ignore ca2.example.

```
example.com CAA 0 issue "ca1.example"
example.com CAA 0 issue "ca2.example; discovery=false"
```

Implementers and operators should carefully configure CAA records according to their specific requirements and considerations.

5. ACME Client Configuration

To enable the ACME client to obtain the necessary configuration information for interacting with the authorized Certification Authority (CA)'s ACME server, a mechanism leveraging the well-known directory is proposed.

The well-known directory is a standardized location within the domain's web server where clients can discover specific resources or configurations. In the context of ACME client configuration retrieval, a copy of the ACME directory object or a redirect to it is placed in the well-known directory of the CA's domain, which is specified as a constraint in the CAA record. This allows the ACME client to conveniently retrieve the required configuration.

For instance, when the CAA record restricts certificate issuance to the CA "ca.example" for the domain "example.com", the ACME client retrieves the ACME directory object as specified in Section 7.1.1 of ACME RFC8555 from the URL "https://ca.example/.well-known/acme".

While an alternative consideration was to include the ACME server address directly as a parameter in the CAA record, it was determined...
that this approach could introduce clutter and significantly increase the size of the record. Additionally, a rigid binding between the CAA record and the ACME server address may present challenges if the CA needs to change its server address in the future.

Thus, the approach outlined in this document, utilizing the well-known directory for ACME client configuration retrieval, offers flexibility for CAs to manage and update their ACME server addresses while maintaining a concise and focused CAA record.

It is important for implementers and operators to ensure the availability and accessibility of the ACME directory object within the well-known directory to facilitate successful ACME client configuration retrieval.

6. ACME Client Behavior

Prior to establishing a connection with the default ACME server or a pool of ACME servers, the ACME client verifies the presence of any configured CA Authorization records (CAA) as defined in [RFC8659]. If a CAA record is found, the ACME client will attempt to obtain a certificate from the CA with the highest priority. If the certificate issuance attempt fails, the client will proceed to lower-priority CAs in an attempt to obtain the certificate.

In the event of a failed attempt to obtain a certificate from a particular CA, the ACME client employs a retry mechanism to ensure successful certificate acquisition. However, in cases where certain CAs are known to be temporarily unavailable, the ACME client MAY choose to ignore those CAs for a limited period of time. By temporarily excluding unresponsive CAs from the issuance process, the client can optimize its certificate acquisition strategy and enhance overall efficiency. This approach helps mitigate potential delays caused by unresponsive CAs and allows the client to focus on viable options for obtaining the required certificate.

ACME clients SHOULD notify the user if the enrollment of a certificate from a specific CA fails multiple times, even if the client successfully obtains a certificate from an alternative CA. This notification is essential to ensure that users are promptly informed about recurring enrollment failures, allowing them to take appropriate measures. By providing such notifications, clients enable users to assess and address any underlying issues, seek alternative solutions, or make informed decisions regarding their certificate management processes.

In order to promote the adoption of ACME in Enterprise environments, it is crucial for implementers and operators to recognize the
significance of External Account Bindings. The section dedicated to External Account Bindings provides valuable information and guidelines for effectively incorporating this feature.

6.1. Certificates with multiple domain names

When the ACME client initiates a certificate request for multiple domain names, it is required to check the CAA records for each domain. The purpose of this check is to identify a Certification Authority (CA) that is authorized by all the domain names intended to be included in the certificate. If, during the evaluation of CAA records, no common CA can be identified that satisfies the authorization requirements of all the domain names, the certificate issuance process will fail.

To mitigate the risk of encountering failures in the certificate issuance process due to incompatible CAA records, it is crucial to ensure that certificates only include domain names that are under the control of the same entity. By maintaining a consistent ownership and control of the domain names included in a certificate, the likelihood of encountering authorization conflicts among CAA records is minimized. This practice promotes a more streamlined and reliable certificate issuance process, reducing the potential for errors and ensuring that the certificate accurately represents the domains controlled by a single entity.

The process with multiple domain names looks as follows:

1. The ACME client identifies the list of domain names for which a certificate is requested.

2. For each domain in the list, the ACME client initiates a DNS lookup to retrieve the CAA record(s) according to [RFC8659].

   1. The DNS resolver responds with the CAA record for each domain, specifying the authorized CAs capable of issuing certificates, along with their priorities and other optional parameters.

3. The ACME client analyzes the valid CAA records for each domain to identify a CA that is authorized by all included domains and which has the highest priority while it ignores any CAA records it cannot process.

   1. If all domains prioritize the same CA, the ACME client proceeds with step 4.

   2. If not all domains prioritize the same CA, the ACME client tries to find a compromise based on the highest overall preference.
3. If no compromise can be found, the process will end with a failure and the user **SHOULD** be notified.

4. The ACME client will download the ACME directory from the well-known location of the issuer-domain-name of the selected CA (https://[issuer-domain-name]/.well-known/acme)

5. If an External Account Binding is required but not configured the ACME client will try to determine an alternative CA in step 3.

   1. If no alternative CA can be found, the process with end with a failure and the user **SHOULD** be notified.

6. The ACME clients continues normal operation according to [RFC8555].

6.1.1. Selecting a CA through Compromise

In the example below, we have three domains: "one.example", "two.example", and "three.example". Among these domains, "one.example" and "three.example" prioritize "ca1.example", while "two.example" prioritizes "ca2.example" over "ca1.example". To select a Certification Authority (CA), a compromise on the priority needs to be established. Based on the priorities specified, "ca1.example" is preferred by two out of the three domains. Since "ca1.example" is authorized by all domains and has the highest overall preference, it is selected as the CA to continue the process.

```
one.example CAA 0 issue "ca1.example; priority=1"
one.example CAA 0 issue "ca2.example; priority=2"

two.example CAA 0 issue "ca1.example; priority=2"
two.example CAA 0 issue "ca2.example; priority=1"

three.example CAA 0 issue "ca1.example; priority=1"
three.example CAA 0 issue "ca2.example; priority=2"
```

7. Implementation Considerations

7.1. External Account Binding

Clients **SHOULD** provide users with the ability to configure and utilize external account bindings per CA or ACME server, as it offers enhanced security and flexibility in managing the certificate provisioning process.
External account bindings are not only crucial for users seeking to provision certificates with stronger authentication requirements such as Organization Validated (OV), Extended Validation (EV), and Qualified Website Authentication Certificates (QWAC), but they can also be applicable to Domain Validation (DV) certificates covered by a commercial agreement. By offering the configuration option for external account bindings, clients enable users to establish a secure association between their ACME accounts and external accounts, facilitating streamlined and authenticated certificate issuance processes. This flexibility accommodates a wide range of certificate types and use cases, ensuring that users can provision certificates with the appropriate level of authentication based on their specific requirements, whether they be DV certificates covered by a commercial agreement or certificates with higher levels of validation. Therefore, it is essential for clients to implement the external account binding configuration option to support the diverse needs of users in obtaining certificates with varying authentication levels.

It is crucial for Certification Authorities (CAs) to carefully consider the internal account binding mechanisms described in this document. This is especially critical given the current lack of widespread support for external account bindings in user interfaces, with only a few command line utilities offering such functionality. By recognizing and implementing the internal account binding approach, CAs can provide a viable alternative for users who may not have access to or be familiar with external account binding options. This will help ensure a seamless and secure account linkage process, even in situations where the availability of external account binding configurations is limited.

7.1.1. Internal Account Binding using Domain Control Validation

In addition to the external account binding mechanism, an alternative approach can be implemented by the CAs that offers distinct advantages, particularly in cases where service providers may not expose the account binding configuration options to their users. This alternative method leverages domain control validation as the initial step in the process and subsequently pauses to await confirmation from the account holder regarding the account binding. The email address associated with the ACME account can be utilized for differentiating between multiple Certification Authority (CA) accounts holding the same domain name.
The process to establish an internal account binding would be as follows:

1. The user adds the domain name to their Certification Authority (CA) account and completes the domain control verification process within this account.

2. The user initiates the ACME process for certificate issuance or a new authorization.

3. The ACME client validates domain control for the requested domain.

4. Upon successful domain control validation the CA does not mark the authorizations as completed but awaits the completion of the account binding.

5. The CA, having confirmed domain control, uses the provided email address associated with the ACME account to distinguish between different CA accounts that have confirmed control over the same domain name.

6. The account holder receives a notification or prompt and confirms the account binding within the CA's system.

7. Once the account binding is confirmed, the CA marks the ACME authorization as completed and the ACME client proceeds with the remaining steps of the ACME process, such as finalizing the certificate issuance.

It is important to note that before the ACME process can start, the domain name must be added to the CA account and the domain control validation process must be successfully completed. This ensures that the domain ownership is verified for both accounts before proceeding with the account binding.

### 7.1.2. Internal Account Binding using Email Address

When ACME clients provide the email address associated with the user's account during the creation of a new account, Certification Authorities (CAs) can utilize this email address to establish an internal account binding between the ACME account and the corresponding customer account within their internal systems. This approach offers an alternative method for establishing the account linkage, particularly in cases where the ACME integration's user interface does not provide explicit external account binding configuration options.

However, it is crucial to acknowledge that this internal account binding mechanism introduces potential vulnerabilities, particularly
in relation to phishing attacks. It is imperative to exercise caution when utilizing this mechanism since the email address associated with the ACME account is not verified, and the account binding request can be initiated by any party. Careful consideration should be given to the security implications of relying solely on the email address for establishing the account linkage.

7.2. Terms of Service and Acceptance

The terms of service associated with different CAs can vary, and it is important to consider how these terms are handled within the context of auto-discovery.

7.2.1. Implicit Acceptance of Terms of Service

As the ACME client is not explicitly controlled by the user in a shared environment, the user's explicit approval of the terms of service presented by the CA becomes challenging. In the absence of a direct user interaction with the ACME client, it is assumed that the user accepts the terms of service by explicitly configuring the CAA record to authorize the CA.

CAs will typically provide documentation indicating how to configure a domain's CAA record for ACME auto-discovery and are encouraged to note in alongside those instructions that doing so will be taken as implicit agreement to the Terms of Service, and also to include a direct link to those Terms of Service.

ACME clients are strongly encouraged to display the relevant terms of service for the obtained certificates to ensure users have visibility into the associated obligations and restrictions. This helps users make informed decisions about their certificate management and ensures compliance with the terms of service set by the authorized CA.

7.2.2. Acceptance Through CAA Parameter

One potential enhancement to address the explicit acceptance of terms of service is the inclusion of a CAA parameter called "termsOfServiceAgreed". This parameter would provide a direct mechanism for users to indicate their agreement to the terms of service.

However, it is important to consider the trade-offs associated with adding this type of data to the CAA record. The inclusion of additional attributes can be perceived as clutter and may increase the complexity of configuring the CAA record. Therefore, the authors of this document recommend relying on the implicit acceptance of the terms of service.
By configuring the CAA record to authorize a specific CA, users implicitly indicate their acceptance of the associated terms of service. This approach strikes a balance between simplicity and compliance with the CA's requirements. It is crucial for ACME clients to display the relevant terms of service for the obtained certificates, ensuring that users have visibility and can make informed decisions regarding their certificate management.

8. IANA Considerations

8.1. Well-Known URI for the ACME Directory

The following value has been registered in the "Well-Known URIs" registry (using the template from [RFC5785]):

URI suffix: acme
Change controller: IETF
Specification document(s): RFC XXXX, Section Y.Z
Related information: N/A

RFC Editor's Note: Please replace XXXX above with the RFC number assigned to this document

8.2. CAA Parameters

As per [RFC8659], the parameter namespace for the CAA "issue" and "issuewild" Properties has CA-defined semantics, and the identifiers within that namespace may be freely and arbitrarily assigned by a CA. This document merely specifies recommended semantics for parameters of the names "discovery" and "priority", which CAs may choose to adopt.

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

Although there is no requirement for a RFC 8659-compliant CA to process parameters, having a list of parameters whose recommended semantics have been defined would likely be useful. Therefor the authors of this document have the intention to establishing a CAA parameter registry in another document.

See also: https://github.com/vanbroup/acme-auto-discovery/issues/14

9. Security Considerations

9.1. Risks with Auto-Discovery of Authorized CAs

The mechanism described in this document relies on the DNS Certification Authority Authorization (CAA) Resource Record
to determine the authorized Certification Authorities (CAs) capable of issuing certificates for a given domain name(s). However, there are potential risks associated with the automatic provisioning of certificates without an explicit indication from the user.

9.1.1. Unexpected Certificate Issuance

Where the issuance of certificates is currently restricted through CAA records and certificates are provisioned through alternative means (i.e., manual or via a proprietary API) certificates can unexpectedly be replaced with a similar certificate or a certificate of a different type (e.g., DV versus EV) if the ACME client supports this new mechanism.

It's recommended that users who wish to obtain certificates attesting to more than domain validation (DV) control, restrict the validation method using a CA specific “validationmethods” CAA parameter value (e.g., “ca-ov”, “ca-ev”, “ca-qwac”) as specified by RFC8657.

9.1.2. Issuance by the ‘wrong’ authorized CA

In scenarios where a domain name authorizes multiple CAs without specifying a weight or preference attribute, there is a risk that the ACME client may unexpectedly request a certificate from one of the authorized CAs that was only included as backup. To mitigate this risk, it is recommended that users who have multiple CAA records explicitly configure the CAA record to include a weight or preference attribute to indicate their desired CA for certificate issuance.

Additionally, ACME clients should provide clear visibility and feedback to users regarding the CA from which certificates will be obtained, ensuring that it aligns with their expectations.

9.2. Malicious ACME Servers

One potential security risk associated with the mechanism defined in this document is the possibility of domain owners placing links to malicious ACME servers into their DNS CAA Resource Records in order to attack the infrastructure of hosting providers. Malicious actors could exploit vulnerabilities in the ACME client implementation or inject malicious code, potentially leading to unauthorized access or remote code execution on the client's system.

To minimize this risk, ACME clients must be written with a cautious and security-conscious approach when interacting with ACME servers. It is crucial not to blindly trust servers to behave securely and in accordance with the ACME protocol.
9.3. ACME Keys

To ensure a secure account binding per customer, it is essential that each customer possesses their own unique ACME key. The utilization of individual ACME keys allows for a distinct association between the customer's account and the established account binding.

If an account binding were to be established based on a shared ACME key, it could potentially lead to unauthorized users obtaining certificates using the same Certificate Authority (CA) based on the established account binding. This scenario poses a significant security risk and could result in the compromise of sensitive information or unauthorized certificate issuance.

To mitigate this risk, it is crucial to enforce the use of individual ACME keys for each customer. This ensures that the account binding is securely linked to the respective customer's account, preventing unauthorized access or misuse by other users. By maintaining separate ACME keys per customer, the integrity and confidentiality of the account binding process are upheld, enhancing the overall security posture of the system.

9.4. Use of DNS Security

The use of DNSSEC to authenticate CAA RRs is strongly RECOMMENDED but not required. In scenarios where DNSSEC is not utilized, there is a potential risk wherein the ACME client may be compelled to request a certificate from an alternative ACME server, which could be malicious in nature.

In the context of the public PKI, a compliant CA associated with the ACME server will deny such unauthorized requests if it was not delegated the authority through a CAA record. However, for ACME servers that operate outside the scope of public trust and may have malicious intentions, the ACME client can validate certificates against its local root store, thus identifying and mitigating this potential attack.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/


10.2. Informative References

[I-D.tweedale-acme-discovery]


Acknowledgments

TODO acknowledge.

Authors' Addresses

Paul van Brouwershaven
Entrust Limited
2500 Solandt Road – Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: paul.vanbrouwershaven@entrust.com
Mike Ounsworth
Entrust Limited
2500 Solandt Road – Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: mike.ounsworth@entrust.com