Workgroup: Network Working Group
Internet-Draft:
draft-tweedale-acme-discovery-01
Published: 16 November 2020
Intended Status: Standards Track
Expires: 20 May 2021
Authors: F. Tweedale
Red Hat

Automated Certificate Management Environment (ACME) Service Discovery

Abstract

This document specifies a DNS-based Service Discovery (DNS-SD) profile that enables Automated Certificate Management Environment (ACME) clients to locate ACME servers in their network environment.

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 20 May 2021.

Table of Contents

1. Introduction

Automatic Certificate Management Environment [ACME] specifies a protocol by which a client may, in an automatable way, prove control of identifiers and obtain a certificate from an Certificate Authority (the ACME server). However, it did not specify a mechanism by which a client can locate a suitable ACME server. It is assumed that a client will be configured to use a particular ACME server, or else default to some well known, publicly accessible ACME service.

In some environments, such as corporate networks, it may be impossible for ACME clients to obtain certificates from a publicly accessible ACME servers, or an organisation may prefer clients to use a particular server. Explicitly configuring ACME clients to use a particular ACME server presents an administrative burden. Furthermore, a service discovery mechanism could allow newly

connected systems to opportunistically locate an ACME server and acquire certificates, without operator (human or otherwise) intervention.

This document specifies a mechanism by which ACME clients can locate an ACME server using DNS-Based Service Discovery [DNS-SD]. Network administrators can advertise one or more ACME servers and express their endorsed capabilities (identifier types and validation methods) and priorities. Capable clients can discover the advertised services and use the most preferred service that satisfies its requirements and is reachable.

2. Terminology

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-SD Profile

3.1. Service Instance Name

A DNS-SD Service Instance Name has the form:

<Instance> . <Service> . <Domain>

The <Service> portion of the Service Instance Name SHALL be "_acme-server._tcp".

The <Instance> portion of the Service Instance Name MAY be arbitrary [Net-Unicode] text. That is, this specification does not further constrain what is allowed by [DNS-SD].

3.2. PTR Records (Service Instance Enumeration)

ACME clients discover ACME service instances by querying DNS PTR [RFC1035] records with the name "_acme-server._tcp.<Domain>", where <Domain> is a "parent domain" [RFC8552] known to or derived by the client. Network administrators enable ACME Service Discovery by creating such PTR records.

The target of each PTR record MUST be an ACME Service Instance Name, which MUST have the same <Service> portion. The Service Instance Name SHOULD have the same <Domain> portion as the PTR owner name. Administrators should delegate Service Instance Resolution to other domains with caution; doing so may remove control of service priorities and capability endorsement to a third party. Clients MUST ignore a Service Instance Name if its <Domain> portion differs from

the <Domain> portion owner of the PTR record, unless explicitly configured otherwise.

3.3. SRV Records

Each ACME service, identified by its Service Instance Name, MUST have an SRV [DNS-SRV] record giving the domain name and TCP port where an ACME server may be found. Each service instance SHOULD have exactly one SRV record.

This specification alters the semantics of the SRV priority field from that given by [DNS-SRV] and [DNS-SD]. For ACME Service Discovery, the scope of the SRV priority field is the set of all SRV records for all Service Instance Names enumerated for the parent domain. This allows network administrators to establish an order of preference among multiple distinct ACME service instance.

Because of the altered semantics of the SRV priority field, implementers SHALL ignore the recommendation of [DNS-SD] that where a single service instance is described by exactly one SRV record, the priority and weight fields of the SRV record should be set to zero.

3.4. TXT Records

Each ACME service, identified by its Service Instance Name, MUST have a TXT [RFC1035] record giving additional data about the service. Each service instance SHOULD have exactly one TXT record.

The TXT record MUST be structured according to [DNS-SD] Section 6. Attributes and their interpretations are set out in the following subsections. The order of the attributes in the TXT record is insignificant.

3.4.1. "path" attribute (ACME Directory Path)

The "path" attribute gives the path at which the ACME directory resource is located on the HTTP server identified by the service instance's SRV record. The attribute value MUST be a valid [URI]. This attribute is REQUIRED.

3.4.2. "i" attribute (ACME Identifier Types)

The "i" attribute gives a list of ACME identifier types supported by the service. Its value MUST be a comma-separated list of ACME identifier types, without whitespace. The list MAY be empty, and SHOULD only include values registered in the IANA ACME Identifier Type registry [IANA-ACME-ID].

The list of identifier types MAY be a subset of the identifier types actually supported by the ACME server. As such, this attribute constitutes the network administrators' endorsement to use the service instance for the listed identifier types only, but does not offer a means of enforcement. Clients MUST ignore services whose "i" attribute does not list the identifier type(s) they require.

The "i" attribute is REQUIRED. An empty list of identifier means that the network administrators acknowledge the presense of the ACME service, but do not endorse its use. Clients MUST ignore a service instance if its "i" attribute is not present, or present with no value, or present with an empty value.

3.4.3. "v" attribute (ACME Validation Methods)

The "v" attribute gives a list of ACME validation methods (also called "challenge types") supported by the service. Its value MUST be a comma-separated list of ACME validation methods, without whitespace. The list MAY be empty, and SHOULD only include values registered in the IANA ACME Validation Methods registry [IANA-ACME-VAL].

The list of validation methods MAY be a subset of the validation methods actually supported by the ACME server. As such, this attribute constitutes the network administrators' endorsement to use only the listed validation methods with this service, but does not offer a means of enforcement.

The "v" attribute is OPTIONAL. If the "v" attribute is present with a value (including an empty value), and that value does not include a validation method the client is capable and willing to use, the client MUST ignore the service instance. If the "v" attribute is present with no value, the client MUST regard it as having an empty value. If the "v" value is not present, the service is implicitly endorsed for all validation methods; the client SHALL assume that the server will support a validation method that the client is capable and willing to use.

3.5. Examples

An organisation operates a corporate ACME server "https://ca.corp.example/acme" for issuing both TLS server certificates (identifier type "dns") and user S/MIME certificates (identifier type "email").

In case their own ACME service cannot be reached, the administrators will advise clients to fall back to the public "Certs 4 All" service at "https://certs4all.example/acme/v2". This service only supports "dns" identifiers.

The following DNS configuration achieves these goals:

```
$ORIGIN corp.example.

_acme-server._tcp PTR CorpCA._acme-server._tcp
_acme-server._tcp PTR C4A._acme-server._tcp

CorpCA._acme-server._tcp SRV 10 0 443 ca.corp.example.
CorpCA._acme-server._tcp TXT "path=/acme" "i=email,dns"

C4A._acme-server._tcp    SRV 20 0 443 certs4all.example.
C4A._acme-server._tcp    TXT "path=/acme/v2" "i=dns"
```

Note that the "CorpCA" SRV priority of 10 ensures that "dns" clients will first attempt to use the "CorpCA" service. If "CorpCA" is unavailable they will try "C4A", which has an SRV priority of 20.

4. Client Behaviour

4.1. When to Perform Service Discovery

If an ACME client provides for explicit configuration of an ACME server, and such configuration is provided, the client MUST use the configured ACME server and MUST NOT perform service discovery.

Otherwise, if an ACME client supports service discovery, in the absense of explicit configuration of an ACME server the client MAY attempt to locate an ACME server using the mechanisms specified in this document. A client MAY refuse to perform service discovery unless its configuration explicitly enables it.

4.2. Candidate Parent Domains

To perform service discovery, the ACME client needs a prioritised list of candidate parent domains. The client will perform DNS-Based Service Discovery in each parent domain until a suitable service is found, or the list is exhausted.

If an ACME client provides for explicit configuration of parent domains to use for service discovery, and such configuration is provided, the candidate parent domains SHALL be the configured values.

Otherwise, there are a variety of ways an ACME client could choose candidate parent domains, including:

- The host's fully-qualified domain name with one or more labels removed from the left.

- The "search" domains from the host's DNS configuration.

*The Kerberos [RFC4120] realm of the host.

*The result of a PTR lookup on one of the host's non-loopback IP addresses, with one or more labels removed from the left.

An ACME client MAY use any or all of these or other suitable methods for identifying candidate parent domains. If multiple candidate parent domains are identified the client MUST establish an order of preference among them. If any candidate parent domain A is a subdomain of another candidate parent domain B, the client MUST preference A higher than B.

4.3. DNS-SD Queries and Validation

Service discovery begins with the most preferred candidate parent domain. For each candidate parent domain, the client performs DNS-SD Service Instance Enumeration and Service Instance Resolution until a suitable server is found, or the candidate parent domains are exhausted.

4.3.1. Service Instance Enumeration

The ACME client SHALL query the DNS PTR records for "<Service>.<Domain>" where <Service> is "_acme-server._tcp" and <Domain> is the candidate parent domain name. For each record returned, the client SHALL verify that the target is an ACME Service Instance Name, i.e. that is has the form:

<Instance>.<Service>.<TargetDomain>

where instance is arbitrary Net-Unicode text, and SHALL ignore targets that are not valid ACME Service Instance Names.

If <TargetDomain> is different from <Domain>, the network administrator of <Domain> has delegated control of the location, priority and service attributes of the service instance to <TargetDomain>, which may be a third party. Clients MUST ignore a Service Instance Name if its <Domain> portion differs from the <Domain> portion owner of the PTR record, unless explicitly configured otherwise.

4.3.2. Service Instance Resolution

The ACME client now has a set of ACME Service Instance Names. For each ACME Service Instance Name, the client SHALL query the SRV and TXT records for that name, and collect the results as (SRV,TXT) pairs. The client could do this sequentially, or with some degree of concurrency. The client SHALL ignore any service instance that is missing either the SRV or TXT record (or both). Although each service instance SHOULD have exactly one SRV record and exactly TXT

record, if multiple SRV and/or multiple TXT records are returned, the client SHALL use the cartesian product of these.

The client MUST exclude any service instances whose TXT "path" attribute is missing or invalid, or whose "i" or "v" attributes do not contain acceptable values.

4.3.3. Verifying the Server

The client now has a list of suitable ACME service instances represented as (SRV,TXT) pairs. The client SHALL attempt to contact servers in an order determined by the SRV priority and weight fields, according to [DNS-SRV].

For each attempt, the client SHALL construct the URI:

```
https://<Target>:<Port><Path>
```

where <Target> is the SRV target, <Port> is the SRV port value and <Path> is the value of the TXT "path" attribute. If the SRV value is 443 the client MAY omit ":<Port>". The client SHALL perform an HTTPS [HTTP] GET request for this URI and SHALL attempt to parse the response body as an ACME directory object. If successful, service discovery has succeeded; the client SHALL use the constructed URI as the ACME server, and SHOULD NOT process the remaining service instances or candidate parent domains.

If none of the service instances yield a valid ACME directory object, service discovery for the current parent domain has failed. Failure modes include:

- *No PTR records at "_acme-server._tcp.<Domain>"
- *No eligible service instances, according to the TXT attributes
- *All HTTPS requests to eligible service instances either failed or did not response with a valid ACME directory object.

In this case, the client MAY retry service discovery with the next most preferred candidate parent domain. The client MAY continue retrying until no candidate parent domains remain, or MAY give up earlier (e.g. after a fixed number of attempts).

If service discovery does not succeed, an ACME client MAY fall back to a default ACME server (e.g. a publicly accessible ACME server).

4.4. ACME Operations

An ACME client MAY record (cache) the URI of the ACME server located via service discovery and MAY use the cached server for new account

and new order operations, without performing service discovery each time.

When storing data about accounts and orders, ACME clients SHOULD record the URI of the actual ACME server used. When retrieving or revoking certificates or performing account operations, the client SHOULD use the recorded URI to contact the ACME server and SHOULD NOT perform service discovery.

When renewing or replacing a certificate, if the recorded ACME server cannot be contacted or fails to issue a certificate, a client MAY perform service discovery to attempt to locate an alternative ACME server that may be able to issue the certificate.

5. IANA Considerations

5.1. "acme-server" Service Name Registration

Per [RFC6335], please add the following entry to the Service Name and Transport Protocol Port Number Registry [IANA-SN]:

Service Name	acme-server
Port Number	N/A
Transport Protocol(s)	tcp
Description	Automated Certificate Management Environment (ACME) server
Assignee	IESG <iesg@ietf.org>
Contact	IETF Chair <chair@ietf.org>
Reference	(this document)
Assignment Notes	Defined TXT keys: path, i, v

6. Security Considerations

6.1. TLS and Certificate Validation

Use of TLS is REQUIRED by [ACME]. [X.509] supports the uniformResourceIdentifier and [SRVName] name types in the Subject Alternative Name extension, and [RFC6125] describes the DNS-ID, URI-ID and SRV-ID identifier types and how to validate them against a server's X.509 certificates.

However, the uniformResourceIdentifier and SRVName name types are not in widespread use and not widely supported by TLS libraries or certificate authorities. [HTTP-TLS] does not describe the use of either of these name types for HTTP services. Therefore when an ACME server was located via service discovery its certificate MUST be validated according to both [X.509] and [RFC6125], using the target of the service's SRV record as the DNS-ID.

6.2. Parent Domain Selection

An attacker who is able to influence an ACME client's candidate parent domains can influence which ACME server the client uses, or cause service discovery to fail. The attacker could use this capability to perform a denial of service against the ACME client (i.e. the client cannot acquire or renew a certificate), or against parties that validate certificates issued to the client (because they do not trust the issuing CA or because the certificate is invalid in some way), or against a target ACME server (by directing many clients to it). ACME client implementers should carefully consider which methods of determining the parent domain(s) are appropriate for their use cases, and the security implications of their chosen methods.

An ACME client might derive candidate parent domains by removing one or more labels from the left side of some other DNS name (e.g. the host name of the client's machine). If too many labels are removed, the ACME client could perform DNS queries in zones outside the control of the organisation that operates the ACME client. As a result, the ACME client could locate and use an ACME server that the organisation does not intend.

To mitigate this risk, it is RECOMMENDED that clients limit the amount of label pruning that occurs. It is not possible to make a concrete recommendation that is suitable for all environments. Implementers must consider what is appropriate for their use cases and environments. The candidate parent domain ordering requirements also mitigate this risk.

6.3. DNS Security

Without ACME Service Discovery, an ACME client must be configured or hard-coded to use a particular ACME server, specified as the HTTPS URI of the server's directory resource. Typically the host will be a DNS name rather than an IP address, and one or more DNS queries are necessary to resolve the host's DNS name to an IP address.

When service discovery is used, the URI of the ACME server is obtained from a DNS URI record. If an attacker is able to spoof the _acme-server URI record for a candidate parent domain name, the attacker could cause service discovery to fail or could direct the client to an ACME server of the attacker's choosing. This could constitute a denial of service attack against the client, against parties that validate certificates issued to the client, or against the target server.

Therefore it is RECOMMENDED that URI records used for ACME Service Discovery be secured using DNSSEC. It is RECOMMENDED that ACME

clients make DNS URI queries via DNSSEC-validating stub or recursive resolvers.

Some methods of candidate parent domain selection may involve DNS queries. For example, a client could query PTR records to find a host name, from which it derives a candidate parent domain. Implementers must consider the security of DNS data used for parent domain selection.

6.4. Service Instance Delegation

As noted in [DNS-SD] Section 4.2, it is possible for a service enumeration in one domain to return the names of services in a different domain. It is necessary to consider the security implications for ACME Service Discovery in this scenario.

Consider an organisation that operates a corporate ACME server "https://ca.corp.example/acme" for issuing user "email" certificates, and intends to use the public ACME CA "https://certs4all.example/acme/v2" for "dns" certificates. If the public CA has DNS-SD service instance records in their own domain:

```
$ORIGIN certs4all.example.
C4A._acme-server._tcp SRV 10 0 443 certs4all.example.
C4A._acme-server._tcp TXT "path=/acme/v2" "i=dns"
```

then the network administrators could avoid maintaining variants of these records in their own domain, with a configuration such as:

```
$ORIGIN corp.example.

_acme-server._tcp PTR CorpCA._acme-server._tcp
_acme-server._tcp PTR C4A._acme-server._tcp.certs4all.example.

CorpCA._acme-server._tcp SRV 10 0 443 ca.corp.example.
CorpCA._acme-server._tcp TXT "path=/acme" "i=email"
```

This is a risky configuration because, for some of the service instances, a third party controls both SRV priority and weight, and the TXT attributes, which are used to select eligible service instances. In the configuration above, everything works as intended. ACME "email" clients go to "CorpCA" and "dns" clients go to "C4A". But if the administrators of certs4all.example change their service instance records to:

```
$ORIGIN certs4all.example.
C4A._acme-server._tcp SRV 5 0 443 certs4all.example.
C4A._acme-server._tcp TXT "path=/acme" "i=dns,email"
```

then the organisation's "email" clients will now prefer "C4A". This could lead to denial of service (C4A may not be trusted by mail agents and systems) or breaches of privacy (corporate email addresses will be exposed to the CA, and possibly to the world via Certificate Transparency [RFC6962])

For these reasons, delegation of service instance records to third parties is NOT RECOMMENDED. As stated elsewhere in this document, clients MUST ignore Service Instance Names whose <Domain> part differs from the parent domain that owns the PTR records, unless explicitly configured otherwise.

6.5. Multicast DNS

DNS-SD is compatible with Multicast DNS [RFC6762]. Devices on the local network can advertise their services by responding to mDNS Service Instance Enumeration (PTR) queries. For example, a client can search for printers by querying "_printer._tcp.local.", and printers respond with their Service Instance Names (and will also respond to requests for the associated SRV and TXT records).

There may be real use cases for ACME service discovery via DNS-SD/mDNS. But there are also risks. The same issues arise as for service instance delegation, but these are compounded because the parent domain is always "local." and service providers (devices) may be ephemeral. This increases the risk of denial of service for ACME clients and relying parties.

The author of this document does not wish to dissuade people from considering use cases and developing and analysing an ACME service discovery profile for DNS-SD/mDNS. It remains an open topic. This specification only requires that a client MUST NOT use DNS-SD/mDNS for ACME Service Discovery unless explicitly configured to do so.

7. Normative References

[ACME] 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/info/rfc8555>.

[DNS-SD] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[DNS-SRV] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[HTTP] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[Net-Unicode] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

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

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[X.509] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

8. Informative References

[HTTP-TLS] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[IANA-ACME-ID] IANA, "ACME Identifier Types", <https://www.iana.org/assignments/acme/acme.xhtml#acme-identifier-types>.

[IANA-ACME-VAL] IANA, "ACME Validation Methods", <https://www.iana.org/assignments/acme/acme.xhtml#acme-validation-methods>.

[IANA-SN] IANA, "Service Name and Transport Protocol Port Number Registry", <https://www.iana.org/assignments/service-names-port-numbers/>.

[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>.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, DOI 10.17487/RFC4120, July 2005, <https://www.rfc-editor.org/info/rfc4120>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[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>.

[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/info/rfc8174>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/info/rfc8552>.

[SRVName]
Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, DOI 10.17487/RFC4985, August 2007, <https://www.rfc-editor.org/info/rfc4985>.

Author's Address

Fraser Tweedale
Red Hat

Email: ftweedal@redhat.com