IETF
Internet-Draft
Intended status: Standards Track
Expires: September 19, 2019

K. Moriarty
Dell EMC

March 29, 2019

ACME Client Extension
draft-moriarty-acme-client-00

Abstract

Automated Certificate Management Environment (ACME) core protocol addresses the use case of web server certificates for TLS. This document extends the ACME protocol to support end user client, device client, and code signing 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 September 19, 2019.

Table of Contents

1. Introduction

ACME [RFC8555] 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.

ACME was designed for web server certificates with the possibility to create extensions for other use cases and certificate types. End user and device certificates may also benefit from automated management to ease the deployment and maintenance of these certificates type, thus the definition of the extension for that purpose in this document.

2. Identity Proofing for Client Certificates

As with the TLS certificates defined in the core ACME document, identity proofing for ACME issued end user client, device client, and code signing certificates was not covered in RFC8555.

Identity proofing for these certificate types present some challenges for process automation. NIST SP 800-63 r3 [NIST800-63r3] serves as guidance for identity proofing further detailed in NIST SP 800-63A [NIST800-63A] that may occur prior to the ability to automate certificate management via ACME or may obviate the need for it weighing end user privacy as a higher concern and allowing for credential issuance to be decoupled from identity proofing (IAL1).

Using this guidance, a CA might select from the identity proofing levels to assert claims on the issued certificates as follows from NIST SP 800-63 r3 [NIST800-63r3]:

"IAL1: There is no requirement to link the applicant to a specific real-life identity. Any attributes provided in conjunction with the authentication process are self-asserted or should be treated as such (including attributes a Credential Service Provider, or CSP, asserts to an RP).

IAL2: Evidence supports the real-world existence of the claimed identity and verifies that the applicant is appropriately associated with this real-world identity. IAL2 introduces the need for either remote or physically-present identity proofing. Attributes can be asserted by CSPs to RPs in support of pseudonymous identity with verified attributes.

IAL3: Physical presence is required for identity proofing. Identifying attributes must be verified by an authorized and trained representative of the CSP. As with IAL2, attributes can be asserted by CSPs to RPs in support of pseudonymous identity with verified attributes."

The certificate issuing CA may make this choice by certificate type issued. Once identity proofing has been performed, in cases where this is part of the process, and certificates have been issued, NIST SP 800-63 r3 [NIST800-63r3] has the following recommendations for authentication or in the context of ACME, management of issuance for subsequent client, device, or code-signing certificates:

"For services in which return visits are applicable, a successful authentication provides reasonable risk-based assurances that the subscriber accessing the service today is the same as that which accessed the service previously. The robustness of this confidence is described by an AAL categorization. NIST SP 800-63 B [NIST800-63B] addresses how an individual can securely authenticate to a CSP to access a digital service or set of digital services. SP 800-63B contains both normative and informative material.

The three AALs define the subsets of options agencies can select based on their risk profile and the potential harm caused by an attacker taking control of an authenticator and accessing agencies? systems. The AALs are as follows:

AAL1: AAL1 provides some assurance that the claimant controls an authenticator bound to the subscriber?s account. AAL1 requires either single-factor or multi-factor authentication using a wide range of available authentication technologies. Successful

authentication requires that the claimant prove possession and control of the authenticator through a secure authentication protocol.

AAL2: AAL2 provides high confidence that the claimant controls authenticator(s) bound to the subscriber?s account. Proof of possession and control of two distinct authentication factors is required through secure authentication protocol(s). Approved cryptographic techniques are required at AAL2 and above.

AAL3: AAL3 provides very high confidence that the claimant controls authenticator(s) bound to the subscriber?s account. Authentication at AAL3 is based on proof of possession of a key through a cryptographic protocol. AAL3 authentication SHALL use a hardware-based authenticator and an authenticator that provides verifier impersonation resistance; the same device MAY fulfill both these requirements. In order to authenticate at AAL3, claimants SHALL prove possession and control of two distinct authentication factors through secure authentication protocol(s). Approved cryptographic techniques are required."

If federations and assertions are used for authorizing certificate issuance, NIST SP 800-63 C [NIST800-63C] may be referenced for guidance on levels of assurance.

Existing PKI certification authorities (CAs) tend to use a set of ad hoc protocols for certificate issuance and identity verification. For each certificate usage type, a basic process will be described to obtain an initial certificate and for the certificate renewal process. If higher assurance levels are desired, the guidance from NIST SP 800-63 r3 [NIST800-63r3] may be useful and out-of-band identity proofing options are possible options for pre-authorization challenges or notifications.

3. Key Storage

[The following text may be left out in the next revision as it is decoupled already: A design goal for the automated workflow for these certificate types via ACME is to allow for use of the Key Management Interoperability Protocol (KMIP) for key management and storage or PKCS-11 for key storage. In the case of KMIP, the KMIP enterprise key manager could use ACME to communicate with the CA server, leaving the device communications between devices and the KMIP server. However, the use of ACME can be standalone integrating with the available client key storage method (for example, PKCS-#11) provided for accessibility and to prevent cost barriers for automating key management for some implementations. The ACME client on the device or system storing the code signing certificate would authenticate to

the CA running an ACME server to obtain initial certificates or renew certificates. With the proliferation of open source implementations of ACME for TLS server certificates, this seems like a reasonable goal.]

4. Why Not EST

[These discussions have happened already for the core ACME protocol, expect this to be removed for the next version:

Enrollment over Secure Transport (EST) [RFC7030] and OpenStack's Keystone are options for automating client certificates. [OpenSSL can be combined with libest to automate the management of client certificates.]

The authentication options used in EST to obtain a client certificate are described in [RFC7030] Section 2.2 and are stated as follows:

> TLS with a previously issued client certificate (e.g., an existing certificate issued by the EST CA);
>
> TLS with a previously installed certificate (e.g., manufacturer-installed certificate or a certificate issued by some other party);
>
> Certificate-less TLS (e.g., with a shared credential distributed out-of-band);
>
> HTTP-based with a username/password distributed out-of-band.

Although a fine a protocol, none of these options enable the protocol to establish authentication of the entity (device, user, owner of code signing certificate) without a pre-established and external process to the protocol. In some cases, higher levels of assertion are necessary and EST may be more suited for those purposes or additional out-of-band processing could be used in conjunction with ACME if adopted widely for the automation of client certificate management.]

5. Device Certificates

A device certificate is a client certificate issued to a device identified through device credentials such as an IP address, hostname, or MAC address. This process is separate from an end user client certificate that may be stored on a device, but identifies a person using the device described in the next subsection. While there are automated processes in place today for device certificate renewal, most are specific to the CA and not open standards. The

general workflow is similar to that described in RFC8555 with the differences being in the CSR, requesting a client certificate. [IP addresses may be necessary for some devices and it may be best to extend [I-D.ietf-acme-ip] to cover varying CSR types that include client certificates for devices explicitly.]

A typical process to obtain a device certificate may be similar to the following workflow described in the introduction of RFC8555 with the exception of certificate type and usage.

[Is an additional type definition helpful to distinguish that this is for a client certificate?]

6. End USer Client Certificates

[Should this be done in ACME? I'm leaning towards no.]

A client certificate used to authenticate an end user may be used for mutual authentication in TLS or another example would be with EAP-TLS. The client certificate in this case may be stored in a browser, PKCS-#11 container, KMIP, or another key container. To obtain an end user client certificate, there are several possibilities to automate authentication of an identity credential presumably tied to an end user.

[Several authentication options are intentionally provided for review and discussion by the ACME working group.]

A trusted federated service that ties the user to an email address with a reputation of the user attached to the email may be possible. One such example might be the use of a JWT signed OAuth token.

Risk based authentication used for identity proofing with red herring questions is a third option that could utilize public information on individuals to authenticate.

Just use FIDO and don't create anything new. FIDO provides a mechanism to have unique certificate based access for client authentication to web sites and they are working on non-web. Identity proofing is intentionally decoupled from authentication in this model as that is in line with NIST 800-63r3 recommendations for privacy protections of the user. The credential in this case is authenticated and would be consistent for it's use, but the identity proofing for that credential is not performed. Obviously, identity proofing is more important for some services, like financial applications where tying the user to the identity for access to financial information is important. However, is automated identity proofing important for any user certificate or should it remain

decoupled where it could be automated by a service offering or is there a need for a standardized mechanism to support it for user certificates?

7. CodeSigning Certificates

The process to retrieve a code signing certificate is similar to that of a web server certificate, with differences primarily in the CSR request and the resulting certificate properties. [The storage and access of a code signing certificate must be protected and is typically done through hardware, a hardware security module (HSM) which likely has a PKCS#11 interface. A code signing certificate may either be a standard one or an extended validation (EV) certificate.]

[For automation purposes, the process described in this document will follow the standard process and any out-of-band preprocessing can increase the level of the issued certificate if the CA offers such options and has additional identity proofing mechanisms (in band or out-of-band).]

Strict vetting processes are necessary for many code signing certificates to provide a high assurance on the signer. In some cases, issuance of a standard CodeSigning certificate will be appropriate and no additional "challenges" [RFC8555 Section 8] will be necessary. In this case, the standard option could be automated very similar to Web server certificates with the only changes being in the CSR properties. However, this may not apply to all scenarios, such as those requiring EV certificates with the possibility for required out-of-band initial authentication.

Organization validation is required for standard code signing certificates from most issuers. The CSR is used to identify the organization from the included domain name in the request. The resulting certificate, however, instead contains the organization's name and for EV certificates, other identifying information for the organization. For EV certificates, this typically requires that the domain is registered with the Certificate Authority provider, listed in CAA [RFC6844], and administrators for the account are named with provided portal access for certificate issuance and management options.

While ACME allows for the client to directly establish an account with a CA, an initial process for this step may assist with the additional requirements for EV certificates and assurance levels typically required for code signing certificates. For standard certificates, with a recommendation for additional vetting through extended challenge options to enable ACME to establish the account directly. In cases where code signing certificates are used heavily

for an organization, having the portal access accessible replaced with ACME authenticated client access with extra challenges for authentication may be an option to automate the functionality.

To improve the vetting process, ACME's optional use of CAA [RFC6844] with the Directory "meta" data "caaIdentities" ([RFC8555] Section 9.7.6) assists with the validation that a CA may have issue certificates for any particular domain and is RECOMMENDED for use with code signing certificates for this additional level of validation checking on issued certificates.

CAA helps as anyone verifying a certificate used for code signing can verify that the CA used has been authorized to issue certificates for that organization. CSR requests for code signing certificates typically contain a Common Name (CN) using a domain name that is replaced with the organization name to have the expected details displayed in the resulting certificate. Since this work flow already occurs, there is a path to automation and validation via an existing ACME type, "dns".

As noted in RFC8555, "the external account binding feature (see Section 7.3.4) can allow an ACME account to use authorizations that have been granted to an external, non-ACME account. This allows ACME to address issuance scenarios that cannot yet be fully automated, such as the issuance of "Extended Validation" certificates."

The ACME challenge object, [RFC8555] Section 7.1.5 is RECOMMENDED for use for Pre-authorization ([RFC8555] Section 7.4.1).

Questions for reviewers:

[Is there interest to set a specific challenge object for CodeSigning Certificates? Or should this be left to individual CAs to decide and differentiate? The current challenge types defined in RFC8555 include HTTPS (provisioning HTTP resources) and DNS (provisioning a TXT resource record). Use of DNS may be possible, but the HTTP resource doesn't necessarily make sense. Since the process to retrieve an EV CodeSigning certificate usually requires proof of the organization and validation from one of 2 named administrators, SMS or email may be needed as defined challenge types. AN organization may want to tie this contact to a role rather than a person and that consideration should be made in the design as well as implementation by organizations.]

ACME provides an option for notification of the operator via email or SMS upon issuance/renewal of a certificate after the domain has been validated as owned by the requestor. This option is RECOMMENDED due to the security considerations of code signing certificates as a way

to limit or reduce the possibility of a third party gaining access to a code signing certificate inappropriately. [Development of additional challenge types is likely to support this for pre-authorization, which would better match the security considerations for this certificate type.]

Since DNS is used to identify the organization in the request, the identifier "type" ([RFC8555]Section 7.4) is set to dns, not requiring any additions to the ACME protocol for this type of certificate. The distinction lies in the CSR, where the values are set to request a CodeSigning certificate for a client certificate. [Question: Is it helpful to define an identifier for the administrator or for the developer to distinguish the certificate type in ACME and not just the CSR?]

KeyUsage (DigitalSignature) and ExtendedKeyUsage (CodeSigning) in the CSR MUST be set to the correct values for the CA to see the request is for a Code Signing certificate. The Enhanced Key Usage SHOULD be set to show this is a client certificate., using OID "1.3.6.1.5.5.7.3.2". The CN MUST be set to the expected registered domain with the CA account.

An advantage of ACME is the ability to automate rollover to allow for easy management of short expiry times on certificates. The lifetime of CodeSigning certificates is typically a year or two, but automation could allow for shorter expiry times becoming feasible.

Automation of storage to an HSM, which typically requires authentication is intentionally left out-of-scope.

8. Pre-authorization

Additional challenge types are defined here for the verification of administrors at an organization requesting CodeSigning certificates. SMS and email are both defined and may be used singularly or in combination as the ACME protocol allows for multiple pre-authorization challenges to be issued.

TBD

9. Security Considerations

This will likely be full of considerations and is TBD for revision one.

10. IANA Considerations

This memo includes no request to IANA, yet.

11. Contributors

12. References

12.1. Normative References

[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>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/info/rfc7030>.

[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>.

[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/info/rfc8555>.

12.2. Informative References

[I-D.ietf-acme-ip] Shoemaker, R., "ACME IP Identifier Validation Extension", draft-ietf-acme-ip-05 (work in progress), February 2019.

12.3. URL References

[NIST800-63A] US National Institute of Standards and Technology, "https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63a.pdf".

[NIST800-63B] US National Institute of Standards and Technology, "https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf".

[NIST800-63C]
US National Institute of Standards and Technology,
"https://nvlpubs.nist.gov/nistpubs/SpecialPublications/
NIST.SP.800-63c.pdf".

[NIST800-63r3]
US National Institute of Standards and Technology,
"https://nvlpubs.nist.gov/nistpubs/SpecialPublications/
NIST.SP.800-63-3.pdf".

Appendix A. Change Log

Note to RFC Editor: if this document does not obsolete an existing RFC, please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Author's Address

Kathleen M. Moriarty
Dell EMC
176 South Street
Hopkinton
US

EMail: Kathleen.Moriarty@dell.com