IETF
Internet-Draft
Intended status: Standards Track
Expires: November 14, 2019

K. Moriarty
DellEMC
M. Richardson
Sandelman

May 13, 2019

ACME Overview
draft-moriarty-acme-overview-00

Abstract

Automated Certificate Management Environment (ACME) core protocol addresses the use case of web server certificates for TLS and defines authentication challenge types to automate certificate issuance. This document describes the orthoganal nature of certificate types to challenge types for a better understanding of the applicability of challenge types to various certificate types.

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 November 14, 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. Although it was not explicitly stated, the challenge types defined in RFC8555 and any other ACME extensions may be used with any certificate type as deemed appropriate by the Certificate Authority management. The defined challenge types are orthogonal in nature to the certificate type that may be specified in the protocol document definition. For instance, it is possible to use a a pre-defined authentictaion challenge, such as a DNS resource record RFC8555 [RFC8555] originally specificed for web server certificates, to authenticate a device to be issued a client certificate. In this scenario, the only change necessary would be in the CSR. There is nothing in the ACME protocol challenge/response mechanism that specifies the certificate type. When designing extensions or revewing options to implement ACME, understanding that the certificate type is decoupled from the challenge type may be important. When designing for specific certificate types, it is logical to design for practical, reasonable challenges specific to

the certificate type to enable the secure automation of certificate management.

This document will cover some early design decisions, and authentication considerations such as identity proofing providing a summary for those interested in the history and decision points.

2. Key Storage

A design goal for the automated workflow for these certificate types via ACME is to allow for flexibility in the selection of key storage containers, for instance one may select the Key Management Interoperability Protocol (KMIP) [KMIP] for key management and storage, PKCS-11 for key storage, or a proprietary key store. This is particularly important and may be asked in regard to client certificates as flexibility to store certificates on a device or external to a device may be a concern and the flexibility to store keys using a chosen protocol may be important. 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. Several options for integrating ACME with other protocols will be presented in this draft, this section is specific to key storage and thus covers the open standards that enable key storage.

3. Why Not EST

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, for instance.

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.

4. Why EST, Bootstrapping with BRSKI

Bootstrapping Remote Secure Key Infrastructures (BRSKI) [I-D.ietf-anima-bootstrapping-keyinfra] is an extension to EST that has been deployed, but is still in the standardization process. Client device certificates typically require administrator interaction to request, retrieve, and install the initial certificate. The process being manual allows for control and understanding of the CA issuing certificates, the device, and verifying the security parameters including authenticating each party. While there are security advantages to the manual process, automation is necessary not only to scale deployment, but in some cases to make it possible. BRSKI defines a protocol to bootstrap the initial enrollment using device properties. This bootstrapping capability is critical when deploying devices at scale and enabling encryption via certificates. BRSKI allows devices to find their owner without being online first, and without preconfiguration. BRSKI does not change how EST authenticates the device; it just reduces it as described below.

BRSKI uses device information to "authenticate" for bootstrapping, namely a serial number entered into the subject field's DN and the subject-alt field may include the Trusted Platform Module (TPM) identifier, IDevID defined in Section 7.2.9 of RFC4108 [RFC4108]. If the subject field is present, it takes precedence over the subject-alt field. These identifiers are included in a voucher or voucher-requests (cryptographically protected artifact using a digital signature) to uniquely identify the device, or in BRSKI terms, the 'pledge'. It should be noted that to date, the identifier in the TPM has not been used in practice as the TPM identifier has not been

present in systems where EST with the BRSKI extension have been used. A registrar may be used with a local policy and it is also possible to use a Manufacturer Authorized Signing Authority (MASA) service for the authentication process. MASA has been the default (See section 5 of BRSKI [I-D.ietf-anima-bootstrapping-keyinfra]. The pledge only imprints when the voucher can be validated.

In the case where a private CA is used, it is possible to configure the environment to use EST with the BRSKI extension to fully automate certificate enrollment and management with bootstrapping. In some cases, it is possible to use EST and BRSKI alone when retreieving a certificate from a public CA. In this case, the Registration Authority (RA) or registrar, may be on the same system as the CA or the registrar has a protocol to communicate with the CA using a variation of EST such as fullcmc rather than the simpleenroll methods. If EST is not supported on the public CA, the registrar may need to communicate with the CA using another protocol that enables automated certificate management, namely ACME. This scenario may be true for other certificate and key management protocols, such as KMIP, where the device certificate management handled by the registrar is fully enabled to the devices or 'pledges', but communication to the CA uses ACME.

5. Why EST and BRSKI with ACME, or KMIP with ACME

Interoperability is the key factor that would drive a deployment mixing several protocols. Since there are a few choices and some are a better fit in certain enviornments, the method to fully automate with the avilable infrastructure may require a mix of protocols. Yes, this is a bit more complicated and a method like EST with BRSKI could solve this end-to-end, but support is needed at the public CA to make that possible. KMIP may also be used to manage certificates to devices and with a similar design, may utilize ACME for communications from the registrar to the CA. The use case is as follows:

- BRSKI provides the authentication of the new device, establishing trust for use with EST
- EST enables enrollment, where the device requests the certificate from the registrar
- The registrar invokes the ACME protocol to request a certificate from the CA.

ACME could possibly be used as a protocol in multiple use cases where the existing standard does not specify how the enrolment server communicates with the CA. In addition to the above described use

case for EST with BRSKI, EST and TEAP enrolment servers, with BRSKI and TEAP-BRSKI is a variations on extensions to those. Similarly, KMIP may be used for the first two steps, with the connection from the registrar to the CA using ACME.

6. Authentication Challenges and Certificate Types

A typical process to obtain a client certificate may be similar to the workflow described in the introduction of RFC8555 with the exception of certificate type and Enhanced Key Usage (EKU). Although RFC8555 was written with the web server use case in mind, the protocol was designed to allow for the defined authentication challenges to be orthogonal to the certificate type issued. The workflow may vary between certificate types as the issuance process may have specific requirements and as a result, may have unique authentication challenge options that alter the workflow. If the defined challenge types are appropriate for use to issue certificates of other types, this is possible without further specification. The implementation would use the appropriate certificate type, defined in the CSR. The ACME protocol does not define CSR contents, hence the certificate type and challenge, as well as pre-authorization challenges, are decoupled from the certificate type. The EKU are standard defined values and include:

- 1.3.6.1.5.5.7.3.1 Server Authentication
- 1.3.6.1.5.5.7.3.2 Client Authentication
- 1.3.6.1.5.5.7.3.3 Code Signing
- 1.3.6.1.5.5.7.3.4 Email Protection
- 1.3.6.1.5.5.7.3.5 IPSec End System
- 1.3.6.1.5.5.7.3.6 IPSec Tunnel
- 1.3.6.1.5.5.7.3.7 IPSec User
- 1.3.6.1.5.5.7.3.8 Time Stamping
- 1.3.6.1.5.5.7.3.9 OCSP Signing

7. Security Considerations

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

8. IANA Considerations

This memo includes no request to IANA, yet.

9. Contributors

Thank you to Owen Friel for your early comments and suggestions and to Richard Barnes for suggesting the need for such a document.

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/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>.

10.2. Informative References

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

[I-D.ietf-anima-bootstrapping-keyinfra] Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-19 (work in progress), March 2019.

[RFC4108] Housley, R., "Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages", RFC 4108, DOI 10.17487/RFC4108, August 2005, <https://www.rfc-editor.org/info/rfc4108>.

10.3. URL References

[KMIP] OASIS, "Key Management Interoperability Protocol Specification Version 1.4 http://docs.oasis-open.org/kmip/spec/v1.4/cos01/kmip-spec-v1.4-cos01.pdf".

[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.

Authors' Addresses

Kathleen M. Moriarty
DellEMC
176 South Street
Hopkinton
US

EMail: Kathleen.Moriarty@dell.com

Michael C. Richardson
Sandelman Software Works

EMail: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/