Overview
Automated Certificate Management (ACME) Protocol

IP-NNI Task Force
Mary Barnes - iconectiv

ACME Overview

- ACME is a protocol being developed in IETF for Automated Certificate Management.

- ACME defines an extensible framework for automating the issuance and validation procedures for certificates:

  - Allows servers to obtain certificates without manual user interaction

- ACME protocol specifications:

  - Core protocol: draft-ietf-acme-acme

  - CAA extensions for more granular CA-specific policies: draft-ietf-acme-caa

  - Identifiers and Challenges for Telephone numbers: draft-peterson-acme-telephone

ACME Protocol model

- ACME uses HTTPS as a transport for Javascript Object Notation (JSON) Web Signature (JWS) objects (effectively a RESTful API):
  - ACME server runs at a Certification Authority (CA) and responds to client's actions if the client is authorized.
- ACME client uses the protocol to request certificate management actions.
- ACME client is represented by an "account key pair".
  - ACME client uses the private key to sign all messages to the server.
  - ACME server uses public to verify the authenticity and integrity of messages from the client.

ACME Protocol Resources

- ACME defines the following resource objects for representing information:
  - Directory object: contains URIs for each ACME operation
  - Registration object: metadata associated with account key pair
  - Application object: represents a client's request for a certificate – contains information about the requested certificate, the server's requirements and any (URL for) certificates (certificate resource) that have been issued.
  - Authorization object: contains the "challenges" (challenge resource) for identifier validation
  - Challenge resource: represents the challenge to prove control of the identifier
  - Certificate resource: represents the issued certificates

ACME Protocol Functions

- ACME uses different URLs (resources) for different management functions:
  - New nonce
  - New Registration
  - New Application
  - New Authorization
  - Revoke Certificate
  - Key change
- A single Directory URL is configured in client in order to get the Directory object containing the above URLs.

ACME Protocol Resource States

- Each resource object has a status field that reflects the state of the object and is used by the client and server to effect changes such as:
  - ACME server sets the status to “valid” in the Authorization object to indicate that the requestor of the certificate has been validated.
    - In the case of challenge/response, ACME client periodically GETs the Authorization object to determine if status is “valid”
  - ACME client sets the status to “deactivated” in the Registration object to deactivate an account

ACME Protocol - Status

- ACME protocol documents anticipated to be completed in IETF WG early 2017. RFCs published mid-2017.

- Proposal to re-charter to consider use of ACME for TNs and SPIDs in early 2017.

- Protocol implementation is well underway:
  - 46 ACME Client implementations with 14 different libraries available
  - Entrust has released a Beta version of an ACME server.
  - 12 ACME projects integrated with Let's Encrypt

Applying ACME to SHAKEN

- SHAKEN usage of ACME defines a new mechanism for the identifier validation challenge.

- SHAKEN service provider validation is based on a token mechanism.

- The protocol used to distribute the token is OUATH (another IETF developed protocol).

  - Uses a JWT (note this is different than the JWT included in the PASSporT).

Backup

ACME vs CMP

1) Encoding:

- CMP is based on ASN.1.
- ACME uses JSON so it's faster and easier to code and debug.

2) Deployment:

- CMP has been deployed primarily in an enterprise environment.
- ACME is intended for the open Internet (while still appropriate for a service provider network).

3) ACME includes provisions for verifying that an applicant for a certificate legitimately holds the identifiers they want to appear in the certificate.