Lessons from ACME

How to win friends and influence people
make an specification that people will actually use

Richard Barnes + Aaron Gable, SAAG @ IETF 117
“In recent years, the SEC ADs have tried to showcase successful adoption of security-related technologies developed by the IETF during the face-to-face SAAG meeting.”
— Roman Danyliw
Has ACME been successful?
Percentage of Web Pages Loaded by Firefox Using HTTPS

(14-day moving average, source: Firefox Telemetry)

- Oct. 2015: First certificate issued with ACME draft-00
- Mar. 2019: RFC 8555
ACME Drives the Web PKI

CAs that support ACME have >95% market share

<table>
<thead>
<tr>
<th>Certificate Authority</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let’s Encrypt</td>
<td>62.7%</td>
</tr>
<tr>
<td>Sectego</td>
<td>12.9%</td>
</tr>
<tr>
<td>DigiCert Group</td>
<td>9.8%</td>
</tr>
<tr>
<td>GlobalSign</td>
<td>9.1%</td>
</tr>
<tr>
<td>GoDaddy Group</td>
<td>0.2%</td>
</tr>
<tr>
<td>Certum</td>
<td>0.7%</td>
</tr>
<tr>
<td>Actalis</td>
<td>0.4%</td>
</tr>
<tr>
<td>Secom Trust</td>
<td>0.4%</td>
</tr>
<tr>
<td>Entrust</td>
<td>0.1%</td>
</tr>
<tr>
<td>Others</td>
<td>3.7%</td>
</tr>
</tbody>
</table>

Vibrant Software Ecosystem

Server side:
- Boulder (Let’s Encrypt CA)
- Smallstep
- EJBCA

Client side:
- Integrations with web servers: Apache, Nginx, Caddy, Kubernetes
- Stand-alone: LEGO, acme.sh
Other Applications Have Emerged

Enable ACME with PKI secrets engine

Private PKI

Mobile Device Management

Email

Telephony

RFC 8823
Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates

Abstract
This document specifies identifiers and challenges required to enable the Automated Certificate Management Environment (ACME) to issue certificates for use by email users that want to use S/MIME.

TNAuthList profile of ACME Authority Token

Abstract
This document defines a profile of the Automated Certificate Management Environment (ACME) Authority Token for the automated and authorized creation of certificates for VoIP Telephone Providers to support Secure Telephony Identity (STI) using the TNAuthList defined by STI certificates.

HTTPS with TLS is the defacto standard for all web traffic today and production use cases require this level of security at a minimum.

You can configure the ACME Certificate payload to obtain certificates from a certificate authority (CA) for Apple devices enrolled in a mobile device management (MDM) solution. ACME is modern alternative to SCEP. It is a protocol for requesting and installing certificates. Use of ACME is required when using Managed Device Attestation.
Why has ACME been successful?
1. Ship it
2. Embrace and extend
3. Just enough extensibility
4. Formal validation
5. Just GREASE it
Ship It

What:

- Deploy early drafts to production
- Upgrade when the RFC gets done

In ACME: Let’s Encrypt shipped draft-00

Why:

- Build demand – show that it’s useful
- Kick-start the software ecosystem
- Learn things to feed into the standards process
- Don’t be afraid of the upgrade, it’s not that hard
  - … especially if you plan for it from the start!
Embrace and Extend

What:

● Take seriously the way things are done today
● Give folks a clear upgrade path to your protocol

In ACME: External account binding, order flow

Why:

● Incorporate the best of the prior art
● Make clear that people are welcome
● Reduce the barriers to using the standard
Just Enough Extensibility

What:

- Have extension points where they are natural
- Follow the shape of the use case
- YAGNI still applies

In ACME: Validation methods, identifier types

Why:

- Future-proofing
- Applicability to new use cases
- ... without adding confusion
Formal Validation

What:

- Get formal models to verify the security of the protocol

In ACME: Found bugs in of EAB, key roll-over

Why:

- Gets smart academic folks involved in the WG
- Often finds interesting bugs
- Builds trust in the standard
Just GREASE It

What:

- Add random values to exercise points where implementations are supposed to be flexible [RFC8701]

In ACME: Didn’t do this, wish we had

Why:

- If your protocol succeeds, there will be a whole ecosystem of clients / servers
- Need to have consistent expectations
- Developers respond to pain
Summary
1. Ship it
2. Embrace and extend
3. Just enough extensibility
4. Formal validation
5. Just GREASE it