draft-ietf-acme-acme

IETF 95

Closed since last IETF meeting

We did some stuff!

52 pull requests merged since IETF 94, with **14** different authors!

Highlights...

We deleted some stuff!

#108 Remove account recovery
#106 Revert #102
#102 Replace in-band account recovery with `meta`

> No replacement defined for now (despite “meta” proposal)
> Might come back (see below)

#101 Remove proof-of-possession challenge

> Probably only makes sense to come back in an extension

We improved account management!

#39 Add roll-over for account keys

Both keys indicate agreement with the transaction
Roll-over request signed by new key, then old key
Potential revision (see below)

#98 Add deletion of accounts and authz

```
POST /reg/asdf HTTP/1.1
{ “resource”: “reg”, “delete”: true }
/* signed as JWS */
```

We fixed a security issue!

#71 Change TLS-SNI to require hostname not specified by validator

Observation by Jehiah Czebotar that the structure of tls-sni-01 allows a careless server operator to create an insecure configuration

- Validation response can be created based on validation request
- Validation server could just respond to any request

Created tls-sni-02 that follows the same pattern as http-01 and dns-01 challenges

- Server needs to be explicitly configured for each challenge

We added more metadata!

#72 Add metadata to directory

```
GET /acme/directory HTTP/1.1
200 OK
{"meta": {"terms": "https://example.com/acme/terms", …}, …}
```

#37 Add an SCT link relation

For better support of Certificate Transparency

```
Link: <https://example.com/acme/sct/asdf>;rel="ct-sct"
```

Open issues

Miscellaneous

#77 Add conditional authz creation/search [PR] **merge**

#104 Add section about rate limits [PR] **merge**

#47 Clarify how to handle bad new-certificate requests **implement?**

- When should CA error vs. modify request?
- Proposal: MUST error if it cannot deliver as specified

Challenges

#111 Out-of-band challenge	**implement**
#89 dns-01 walk-up	**defer**
#88 http-01 and dns-01 challenges: just use account key	**defer**
#4 Define a new challenge type that runs on a dedicated port	**defer**

Karthik's Analysis

Karthik Bhargavan posted some analysis to the list, from which I made a few issues:

#112 Replace "resource" field with "url" **implement**

```
OLD: { "resource": "new-authz", ... }
NEW: { "url": "https://example.com/acme/new-authz", ... }
```

#113 Reverse the order of signatures in roll-over **implement**

```
OLD: Sign(K_old_priv, Sign(K_new_priv, Fingerprint(K_old_pub)))
NEW: Sign(K_new_priv, Sign(K_old_priv, Fingerprint(K_new_pub)))
```

#114 Specify MAC-based recovery (assuming MAC key) **implement?**

```
NEW: Sign(K_new_priv, MAC(K_recovery_priv, Fingerprint(K_new_pub)))
```

Wildcards

#73 Support authorizations for wildcard names

#74 Wildcard support [PR]

> Current text forbids the use of wildcard identifiers in new-authz requests
>
> How can we enable wildcard issuance?

Preconditions?

Current ACME is proactive: Get authz before you ask for a cert

We could also have a way for the CA to say “That’s not quite right, please do X”

1. ~~Client sends new-authz request~~
2. Client sends new-cert request
3. Server responds with a list of things the client needs to do (e.g., authz)

ACME for pre-ACME CAs

To what degree do we want to support legacy CAs in this iteration of ACME?

Rough proposal posted to the mailing list in December

- Pro: Could allow some current CAs to move to ACME
- Con: Significantly more complex

Solve now or punt?

The Original Proposal

```
Registration
 |
 +--* Authorization
 |
 +--* Certificate
```

```
Registration
 |
 +--* Authorization
 |
 +--* Order
       |
       +--* Authorization
       |
       +--* CertificateRequest
             |
             +--* CertificateInstance
```

Deaggregating

“I think there is a high risk of putting too much policy into the protocol” — pzb

Maybe we can address this with a few smaller issues:

- Preconditions
- Scoping for authorizations
- Grouping of new-cert requests / multiple CSRs

Is this something we want to tackle now, or defer to a possible v2?

Where do we go from here?

Rough Proposal

Clear out issue list by ~May / early June

Get at least two clients and two servers interoperating (ideally with a high degree of test coverage for the spec)

Last Call before IETF 96

At IETF 96, discuss LC comments, any extensions