インポートファッション
アクメ貿易

ACME
-00 → -01

- Fixed Ayer's signature reuse vulnerability
- Fixed default vhost vulnerabilities
- Added versioning to challenge names
  - simpleHttp → http-01, etc.
- Forgot to remove the "DO NOT IMPLEMENT" caveat
SIGNATURE REUSE

- Issue: Reliance on non-standard properties of signature
- Solution: Remove the signature, just digest what you want
- Bonus: Consistency across validation mechanisms

`token.base64url(JWK_Thumbprint(accountKey))`

`DePg9...i1D_z.hG11p...NhkSE`
DEFAULT VIRTUAL HOST

- Issue: Some hosting platforms route TLS requests for an unknown server to a default virtual host
- Solution:
  - Remove `tls` option from HTTP validation
  - Add iterations to TLS SNI validation (revert?)
-01 → NOW

- Merged a couple of editorial PRs
- Remembered to remove the "DO NOT IMPLEMENT" caveat
MERGED!

- #18. Clarify encoding for certs in PoP challenge
- #24. Remove obsolete references to "Simple HTTP"
- #28. Update the caveat in the abstract
TODAY

- Issues
- Pull requests
Extra security is under enforcement
#23. ADD DOMAIN TO CHALLENGE1, CHALLENGE2, DOMAIN, ACME, INVALID

- Would provide a hint to TLS hosting layer as to where to send the request
- ... but no current stack would actually consume it
- ... and it risks running into the 255 byte limit
- **Proposal:** WONTFIX
#17. ADD RATELIMITED ERROR

- Errors are currently required to be in the urn:acme namespace
- Should we REQUIRE servers not to emit errors in this namespace that are not registered?
- If we make this requirement, what should servers do to extend the space
Currently, require text/plain or nothing
This raises the question of how to get the server to emit this content type
**Proposal:** Use a POST to registration URL
Currently, a registration has the same account key forever

Clients might want to periodically rotate

**Proposal:** Remove content type requirement
  - Have old key sign over new key
  - Have new key sign over original registration

#14. SUPPORT KEY ROLLOVER FOR ACCOUNT KEYS
POST /acme/reg/asdfasdf HTTP/1.1
Host: example.com

{    
    "newKey": {
        "resource": "new-reg",
        "registration": "/acme/reg/asdfasdf",
    }
    /* signed as JWK with new key */
}
/* signed as JWK with original key */
#25. ACME SHOULD EXPOSE AN ENDPOINT FOR CT SCT PROOFS

- SCT is provisional proof of inclusion in a CT log
- Send SCT in X.509, OCSP, or TLS extension
- TLS extension flavor requires explicit download
- **Proposal**: Add a Link header from the certificate resource
- Probably also note the other ways a CA can provide CT info

HTTP/1.1 200 OK
Content-Type: application/pkix-cert
Link: </acme/cert/c5111dc6>;rel="signed-certificate-timestamp"
#16. HTTP-01 PROTOCOL

- Actually three issues:
- "Base64" strings are actually "Base64url"
- Libraries often add a zero octet to big integers
- Complete example of key → key authorization
#16. HTTP-01 PROTOCOL

- **Proposed:**
- \( s/\text{Base64}/\text{Base64url}/g \)
- Clarify that the zero octet **MUST** be removed (cite JWK)
- Add a complete example (possibly in the context of a full protocol example appendix?)
#15. REQUEST CERTIFICATE LIFETIME

- Client should be able to request a certificate lifetime
- Design philosophy:
  ■ Use CSRs for:
    1. Things that the certified key pair needs to sign
    2. Things that can be expressed in a CSR
  ■ Use JSON in the new-certificate request for everything else
#15. REQUEST CERTIFICATE LIFETIME

Thus saith RFC 2986:

Note 4 - This document is not compatible with the certification request syntax for Privacy-Enhanced Mail, as described in RFC 1424 [5]. The syntax here differs in three respects: It allows a set of attributes; it does not include issuer name, serial number, or validity period; and it does not require an "innocuous" message to be signed. This document is designed to minimize request size, an important feature for certification authorities accepting requests on paper.
Proposed: Add some JSON to the new-certificate request to express either a life time (as a duration) or proposed notBefore / notAfter.

POST /acme/new-cert HTTP/1.1
Host: example.com
Accept: application/pkix-cert

{
    "resource": "new-cert",
    "csr": "5jNudRx6Ye4HzKEqT5...FS6aKdZeGsysoCo4H9P",
    "durationDays": "90",
    "notBefore": "2016-01-01T00:00:00",
    "notAfter": "2116-04-01T00:00:00"
}
In some hosting configs, TLS requests for an unknown server name go to a default host. If that default host can provision a cert that fulfils a TLS-SNI challenge, he can get a cert for any other host. Fix in -01 is to check a random set of hosts, assuming certs can't change fast enough. This is a lot of hassle, for marginal protection. **Proposal:** Remove default vhost protection from TLS-SNI.
#4. ALLOW PORTS OTHER THAN 443

- http-01 always connects on port 80
- tls-sni-01 always connects on port 443
- These can overlap with existing services
- Or an admin might not control them
#4. ALLOW PORTS OTHER THAN 443

No proposal, but some options

1. Do nothing. Continue to use 80/443
2. Define new port(s) just for ACME
3. Allow the server to specify acceptable ports, client picks
4. Define some list of acceptable ports
FIN