DELEGATING TLS CERTIFICATES TO A CDN: *DRAFT-SHEFFER-LURK-CERT-DELEGATION-00*

YARON SHEFFER, JULY 17, 2016

IETF LURK BOF, IETF-96

MOTIVATION

- The CDN Edge Server should not need to access an external box on each TLS connection

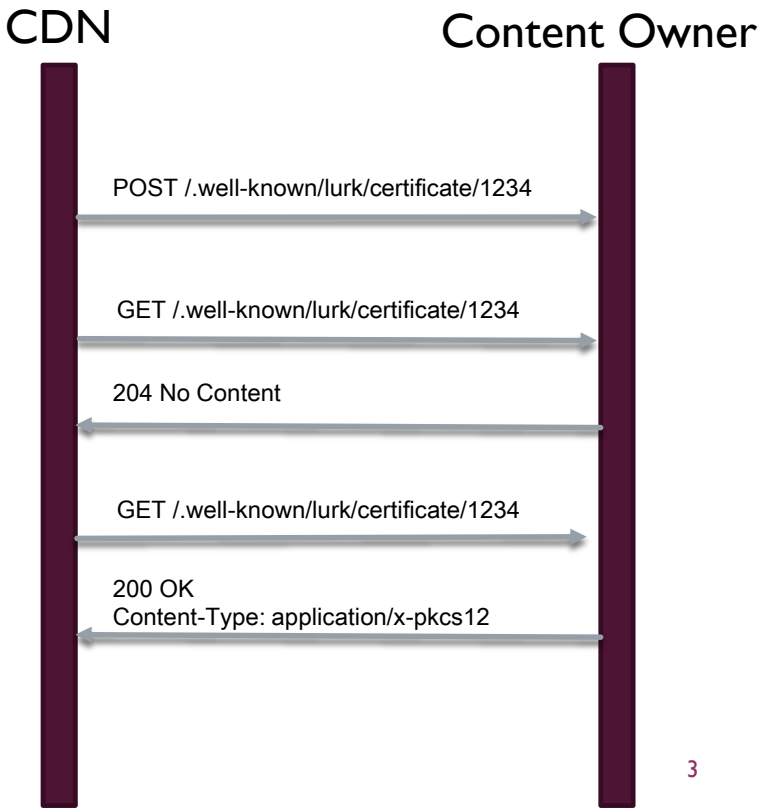
- Key Server on the content owner's network?
- The whole point of using a CDN is to avoid scalable infrastructure on the content owner's side

- Key server on the CDN infrastructure?
- We don't want the CDN to have the long-term private key, after all

- Still in keeping with LURK assumptions: no change to the client
  - Compared to subcerts proposal

SOLUTION OVERVIEW

- The CDN requests a new keypair/certificate from the content owner periodically, e.g. once a day
- The content owner should have a way to obtain a new cert every time
  - Will presumably use ACME
- Request/response with a simple REST API over HTTPS
  - Initial POST to create cert, then a periodic poll for results
  - Channel must be mutually authenticated
- Format: a password-protected PKCS#12 package, containing the (fresh) private key and signed certificate
  - Recommended certificate validity: 3 days



UPON COMPROMISE

- Revoke any extant short-term certificates
  - Could be eliminated if "revocation doesn't work"
- Stop issuing certificates to the CDN

SECURITY CONSIDERATIONS

- Discussion of how the content owner can prevent a rogue CDN from issuing its own certificates for the content owner's domain
  - In the presence of CAs that issue certs based on validating web server ownership
  - Such as ACME
- This is generally applicable, independent of the proposed use of short term certificates
- And relies on ACME mechanisms that are still undefined
  - Specifically, a way to restrict issuance of ACME certs to a specific authorization key
  - Plus wide-scale, reliable deployment of CAA

THANK YOU!