Secondary Certificates

IETF 102 – Montréal
Since London....
Draft 00

C

Request (HEADERS...)

CERTIFICATE_REQUEST

CERTIFICATE

USE_CERTIFICATE

Response (HEADERS...)

S

Draft 01

C

Request (HEADERS...)

CERTIFICATE_REQUEST

CERTIFICATE_REQUIRED[N]

CERTIFICATE

USE_CERTIFICATE[N]

Response (HEADERS...)

S
More explicit

Draft 00
• CERTIFICATE contained flag for AUTOMATIC_USE
  • Servers MAY consider this certificate for any request the client makes
  • If challenged, client doesn’t know whether server already tried that certificate
• Servers MUST set AUTOMATIC_USE on all certificates

Draft 01
• USE_CERTIFICATE contains flag for UNSOLICITED
  • Indicates that server hasn’t asked for a certificate, but client is offering one just in case
  • If server challenges for certificate, client knows the proffered one didn’t work
• If you want certificate used on every request, send USE_CERTIFICATE with every request
More explicit

**Draft 01**
Clients probe for certificates by sending CERTIFICATE_NEEDED for an idle stream
  • ...because the stream will be used for a request to that origin
  • ...and the request can’t proceed until the client sees the cert

**Draft 02**
Clients probe for certificates by sending CERTIFICATE_NEEDED for stream 0
  • ...because the server certificate is a property of the connection
More delegation to TLS

Draft 00

• CERTIFICATE_REQUEST carries OID filters to describe desired cert
• CERTIFICATE carries cert chain
• CERTIFICATE_PROOF carries signature proving possession of cert
• USE_CERTIFICATE without Cert-ID refuses request
More delegation to TLS

**Draft 00**
- CERTIFICATE_REQUEST carries OID filters to describe desired cert
- CERTIFICATE carries cert chain
- CERTIFICATE_PROOF carries signature proving possession of cert
- USE_CERTIFICATE without Cert-ID refuses request

**Draft 01**
- CERTIFICATE_REQUEST carries Exported Authenticator Request
- CERTIFICATE carries Exported Authenticator
  - Includes cert chain + proof
- CERTIFICATE_PROOF
- USE_CERTIFICATE without Cert-ID refuses request
More delegation to TLS

**Draft 00**
- CERTIFICATE_REQUEST carries OID filters to describe desired cert
- CERTIFICATE carries cert chain
- CERTIFICATE_PROOF carries signature proving possession of cert
- USE_CERTIFICATE without Cert-ID refuses request

**Draft 01**
- CERTIFICATE_REQUEST carries Exported Authenticator Request
- CERTIFICATE carries Exported Authenticator
  - Includes cert chain + proof
- CERTIFICATE_PROOF
- USE_CERTIFICATE without Cert-ID refuses request

**Draft 02**
- CERTIFICATE_REQUEST carries Exported Authenticator Request
- CERTIFICATE carries Exported Authenticator
- Exported Authenticator with empty cert chain refuses request
Mitigate MitM

Draft 01

SETTING_HTTP_CERT_AUTH: 1

SETTING_HTTP_CERT_AUTH: 1

CERTIFICATE:

(Exported Authenticator)

Draft 02

SETTING_HTTP_CERT_AUTH: (TLS Exporter)

Invalid!

Invalid!
Open Issues
Binding of Frame Types

CERTIFICATE_REQUEST
- Request-ID
- Exported Auth Request
  - request_context:
    - Request-ID
    - (entropy)
  - Certificate properties...

CERTIFICATE
- Cert-ID
- Exported Authenticator
  - request_context:
    - Request-ID
    - (entropy)
  - Certificate chain...

CERTIFICATE_NEEDED
- Stream ID
- Request-ID

USE_CERTIFICATE
- Stream ID
- Cert-ID
Binding of Frame Types

CERTIFICATE_REQUEST
- Request-ID

Exported Auth Request
- request_context:
  - Request-ID (entropy)

Certificate properties...

CERTIFICATE
- Cert-ID

Exported Authenticator
- request_context:
  - Request-ID (entropy)

Certificate chain...

CERTIFICATE_NEEDED
- Stream ID
- Request-ID

USE_CERTIFICATE
- Stream ID
- Cert-ID
Binding of Frame Types

Verify context [...] contains the Request-ID of a previously-sent CERTIFICATE_REQUEST frame.

Requests and authenticators need not be one-to-one.
Questions

• Should cross-responses be permitted in the first place?
  • “You asked for a certificate for ‘example.com’ and a certificate for ‘images.example.com’ – this certificate covers both.”

• Should the CERTIFICATE frame explicitly contain the Request-ID of the CERTIFICATE_REQUEST?
  • Can be retrieved from the Exported Authenticator
  • Parity with CERTIFICATE_REQUEST structure
The Fly in the Ointment
Misissued Certificates

Status Quo

With Secondary Certs

Revoke with extreme prejudice!

Who issued that cert?

Induced navigation

Induced navigation

Who issued that cert?
Key Compromise

**Status Quo**

- DNS or IP Routing Subversion
- Alice
- Mallory
- Bob
- Bob.com

**With Secondary Certs**

- Induced navigation
- Alice
- Mallory
- Bob.com
- Mallory.net
- Bob.com

C’mon, Angry Birds!
Path forward?

Permit Normal Certificates
• Misissued certificates are harder to trace
  • Attacker’s domain doesn’t need to be included
• Compromised certificates are easier to use
  • Attacker doesn’t need to hijack the TCP connection, just get you to browse his site

Require New Certificate Properties
• Existing certificates not useful with Secondary Certs
  • Slows deployment once feature supported
• If the property makes the certificate “less secure,” would anyone do it?
Proposal: Pin to Primary Domain(s)

- Define new certificate extension
- Required for server certificates to be used with Secondary Certificates
- Indicates which primary domains (from TLS handshake) the certificate can be used with
  - Could be wildcard
- Reject server certificates that don’t include the extension
  - ...or that are used under a different primary certificate
- Do we need something for client certs?