draft-ietf-websec-strict-transport-sec-01

Jeff “=JeffH” Hodges
IETF-80
Prague (Praha)

Status

- draft-ietf-websec-strict-transport-sec-01 submitted on 14-Mar-2011
- Addressed some known open issues

Normative Changes -00 → -01

- Changed "server" -> "host" where applicable, notably when discussing "HSTS Hosts". Left as "server" when discussing e.g. "http server"s.

- Changed the "URI Loading" section to be:

  "URI Loading and Port Mapping”

  - Explicitly specifies “port mapping”

Normative Changes -00 → -01 cont'd

- -00: 7.2. URI Loading
  Whenever the UA prepares to "load", also known as "dereference", any URI where the host production of the URI [RFC3986] matches that of a Known HSTS Server -- either as a congruent match or as a superdomain match where the superdomain Known HSTS Server has includeSubDomains asserted -- and the URI's scheme is "http", then the UA "MUST" replace the URI scheme with "https" before proceeding with the load.
  - http://example.org → https://example.org [ok]
    - implicit port 80 → implicit port 443
  - http://example.org:80 → https://example.org:80 [!ok]
    - explicit port 80 → explicit port 80
    - !ok because breaks standardized assigned HTTP ports

Normative Changes -00 → -01 cont'd

- -01: 7.2. URI Loading and Port Mapping
  Whenever the UA prepares to "load", also known as "dereference", any URI where the host component of the authority component of the URI [RFC3986] matches that of a Known HSTS Host -- either as a congruent match or as a superdomain match where the superdomain Known HSTS Host has includeSubDomains asserted -- and the URI's scheme is "http", then the UA MUST replace the URI scheme with "https" before proceeding with the load.

  Additionally, if the URI contains a port component [RFC3986] equal to "80", the UA MUST covert the port component to be "443". Otherwise, a present port component MUST be preserved.

  - http://example.org:80 → https://example.org:443 [ok]
    - explicit port 80 → explicit port 443
  - http://example.org:8080 → https://example.org: 8080 [ok]
    - explicit port 8080 → explicit port 8080

(still) Open Issues

- Julian notes that Effective Request URI is now manifested in HTTPbis (was leveraged from HSTS spec)
  - Should HSTS ref HTTPbis for this?
  - [I think yes (assuming they are on-schedule for finishing HTTPbis before Sol engulfs Gaia :)]
    - Update on the HTTPbis timeline?

(still) Open Issues cont'd

- Gerv suggested (a while back) a “LockCA” notion
  - i.e. cert and/or CA “pinning” (ie “LockCert”)
  - Several people have brought

LockCA

- Add directive to Strict-Transport-Security header field of “LockCA”
- Semantics are that UA remembers not only that site is secure-only, but also that its certs are issued by CA
    - From initial caching of HSTS info?
    - Supplied along with LockCA directive in header field?

LockCert

- Add directive to Strict-Transport-Security header field of “LockCert”
- Semantics are that UA remembers not only that site is secure-only, but also that this is its cert
  - Ie cache cert “fingerprint”
  - From initial caching of HSTS info?
  - Supplied along with LockCert directive in header field?

EVOnly

- Similar but different from LockCA
- There's operational issues with LockCA
    - Eg what if site wishes to change their CA?
- With EVOnly, UA notes that site's cert MUST be an EV cert.
    - Leverages EV infrastructure (CA/Browser Forum)
    - Site can change CA
- Issues
    - some IETF folks don't recognize CABF Guidelines as referenceable spec
    - Need IANA registry for EV CPS OIDs ?

Newly Raised Issues

- Decouple these two HSTS policy obligations..
  - Establish only secure connections to the HSTS Host – regardless of whether insecure connections are requested/indicated
  - Terminate secure connection establishment upon *any* error/warning
- Declined because they are both inherent to *this* policy.
  - If finer-grained policies are desired, need to invent them

Newly Raised Issues cont'd

- Need to be more explicit/clear in regards to notion of “cert verification” and errors/warnings thereof
  - i.e., HSTS does not prescribe any particular secure channel mechanism, nor certificate types, nor verification processes.
  - It simply states that if there's *any* issues with secure channel establishment, then hard fail.
- Nominally accepted, will endeavor to clarify spec appropriately

ToDo

- Put issues in the Tracker
- Ref HTTPbis for Effective Request URI ?
- Hash out issues on list and update spec appropriately