draft-fieau-https-delivery-delegation

Frédéric Fieau - Orange
Iuniana Oprescu - Orange

IETF 93 Prague 2015

Rationale

- Delegation of content delivery occurs both in the clear HTTP and encrypted HTTPS in interconnection
- Several methods for redirection exist:
  - HTTP 3xx redirections
  - URL rewriting
  - API mode (includes scripts with AJAX requests & JSONP)
  - DNS-layer redirection
- In HTTPS, the redirection should be unnoticeable to the end-user, without triggering security warnings in browsers

What works

- HTTP redirection between uCDN A and dCDN B:
  - HTTP → HTTPS: This is an upgrade and should be accepted by the browser if the certificates are valid and trusted (e.g., not self-signed).
  - HTTPS → HTTPS: In this case, the uCDN domain A redirects the browsers' request to dCDN domain B. The browser forms an initial TLS connection to domain A, receives a secure delegation, and then forms a new security association with domain B. The browser implementation determines how transparent the delegation may be to an end user. However, mainstream browser implementations support seamless secure redirection via HTTP 3xx responses.

What works also

- URL rewriting:
  - In some cases, when a web page is rendered on the browser side, embedded URLs in the page are modified in order to point towards new web locations. This modification is typically caused by a script embedded in the page. Alternatively, a server-side script of some kind could modify embedded URLs before the page is retrieved by a browser.

- Manifest rewriting for HTTP Adaptive Streaming:
  - The uCDN domain A replies to the end-user with a manifest file that redirects the browsers' request to dCDN domain B where the various content chunks are hosted. A more complex scenario is when both uCDN A and dCDN B serve different chunks for the same content requested by the user.

In both scenarios, a security warning might be issued by the browser to inform the user that the content is being served from a different domain.

What works again

- API mode:
  - In this scenario, the initial web page could be located on domain A, whereas the contents requested by the script are hosted on a secondary domain B.
  - The cross-domain (CORS) issues can be fixed with the header “Access-Control-Allow- Origin.”
  - Same security considerations are applicable as in the HTTP redirection case, where the certificates need to be valid and trusted to avoid warnings.

Issues arise

- DNS redirection
  - The DNS resolver, when it queries for the hostname associated with the uCDN URL, will be served a DNS response (such as CNAME) that will direct the client to the dCDN. However, in an HTTPS environment, this will result in the browser receiving a domain other than the one originally specified by the URL inputted by the end user.

    Consequently, this discrepancy between the requested domain and the provided certificates will almost certainly result in a security failure when the browser attempts to negotiate TLS with the web server it contacts. This output is due to the switch in the domain name (from uCDN A to dCDN B) that is indistinguishable from a potential malicious attacker.

Topology hiding

- potential leaking of information about the structure and IP addresses of the dCDN actually delivering the content (see “probes” in draft-ietf-cdni-redirection)

Some solutions

- HARD problem as described by R. Barnes

- DNSSEC → secure DNS redirections
- Attribute certificates → authorization from uCDN to delegate traffic delivery to the dCDN
- give the private keys to the dCDN

- Key server and forwarding of the session key?

Next steps

- There seems to be interest in the topic
- Add some clarification text on the solutions
- Merge with draft-slovetskiy-cdni-https-delegation-approaches ?