ForceHTTPS

Scribe: Chenggang Wu

October 28, 2015

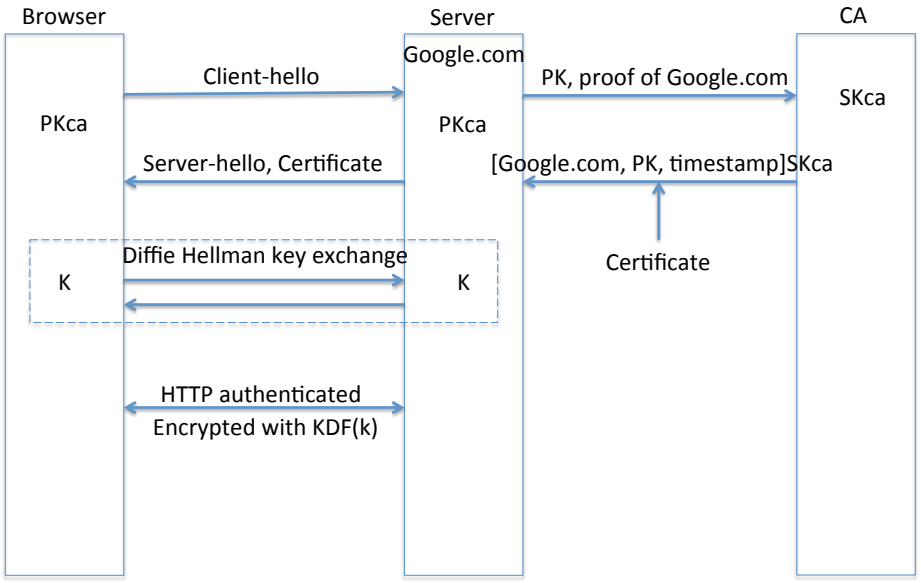
HTTPS

- Threat model:
  - Routers, DNS servers could be compromised
  - Passive Attacker: Reads traffic between browser and server
  - Active Attacker: Modifies/injects data in traffic
- Goals: Protect
  - Data center over network
  - Code/ data in browser
  - UI seen by user
- We can use encryption & authentication mechanisms to protect data center over the network, but how can we be certain about the identity of the server we are talking to?
- Solution: Use certificate
  - Trusted CA's are hardcoded in the browser
  - PK of certificate authority is hardcoded in browser
  - Can protect against compromised DNS server if the attacker don't have certificate to IPattacker
- Figure 1 shows how secure communication between the client and the server is established

ForceHTTPS

- Force usage of HTTPS for a site that uses ForceHTTPS
- Web server admin sets cookie with Forcehttps flag
- Client browser has ForceHTTPS extension
- Client browser:
  - Redirect non-HTTPS connections to HTTPS connections
  - Avoid loading resources with HTTP
  - All TLS warnings or errors terminate connection

Figure 1: TLS



Problems with HTTPS

- Weak cryptography
  - ForceHTTPS does not help because all it does is to ensure using HTTPS
  - Solution: Blacklist weak crypto such as MD5 and SHA-1
- Authenticating the server
  - Security depend on the least secure CA
  - Blacklist CAs works but it has delays (updating and downloading the list take time)
  - Online certificate status protocol (OCSP) is used for obtaining the revocation status of an X.509 digital certificate
  - Users ignore certificate mismatch errors (ForceHTTPS helps but not very functional)
- Mixing HTTP & HTTPS content
  - Proposal: Include hash of resource in HTTPS and check hash when retrieving content through HTTP
  - ForceHTTPS either doesn't allow or rewrite URL to HTTPS

- Protecting cookies: Secure cookies
  - ForceHTTPS converts HTTP access to HTTPS to avoid leaking the cookie
  - Problem: First visit to site is via HTTP so we have to trust on first use
  - Solution:
    - HTTP Strict-Transport-Security:
      - Web sites can include a special header in an HTTP(S) response that enforces browser to talk over HTTPS with server
      - Disadvantage: Actual first request can still be over HTTP, which is insecure
    - HTTPS everywhere:
      - Contains a list of sites known to have HTTPS, so the initial request can use HTTPS to the listed sites as well
      - Disadvantage: The list cannot scale to cover the entire Web
- Users don't check where they enter credentials
  - ForceHTTPS helps with the lock icon
  - ForceHTTPS does not help with the domain name