Public Key Pinning in TLS

Gabor Toth, Tjebbe Vlieg

February 6, 2013

Problems with X.509 PKI

- Security breaches certificate authorities *(e.g. COMODO, DigiNotar)*
- Issuance of intermediate CA certificates to wrong entities *(e.g. TÜRKTRUST)*
- Government controlled CAs could issue certificates for use in MitM attacks

Trust-on-first-Use (TOFU)

- First encountered public key or certificate is trusted
- Warning if public key changed

Examples:

- OpenSSH
- Certificate Patrol

Notary Services

- Notary services probe certificates of hosts from different network locations
- Client verifies public key or certificate using notary servers

Examples:

- Perspectives
- Convergence
- Crossbear

Pinning Protocols

- A protocol is used by the server to publish a pinned public key or certificate
- This key must be used in subsequent sessions

Examples:

- DNS-Based Authentication of Named Entities (DANE)
- Trust Assertions for Certificate Keys (TACK)
- Public Key Pinning Extension for HTTP (websec-key-pinning)

Research Question

- How can we provide additional TLS certificate verification methods for applications, to bridge the gap until a proper pinning protocol becomes widespread?

Public key pinning with TOFU

- Long-term solution is the use of pinning protocols
- Interim solution is pinning with a TOFU scheme
- Pinning functionality should be available to all applications
- Implemented as a library instead of a browser add-on

TLS libraries on Linux

- The most popular ones are OpenSSL, GnuTLS, and NSS
- They provide certificate chain verification functionality
- Different implementations using different trust stores
- Sharing trust policy is desired

Steps of verifying certificates

- first verify certificate chain using a pinning protocol
- if not available
  - verify chain against local trust store
    - revocation lists
    - trusted CAs
    - manually trusted or blacklisted certificates
  - if successfully verified, check local pinning database

Storage model

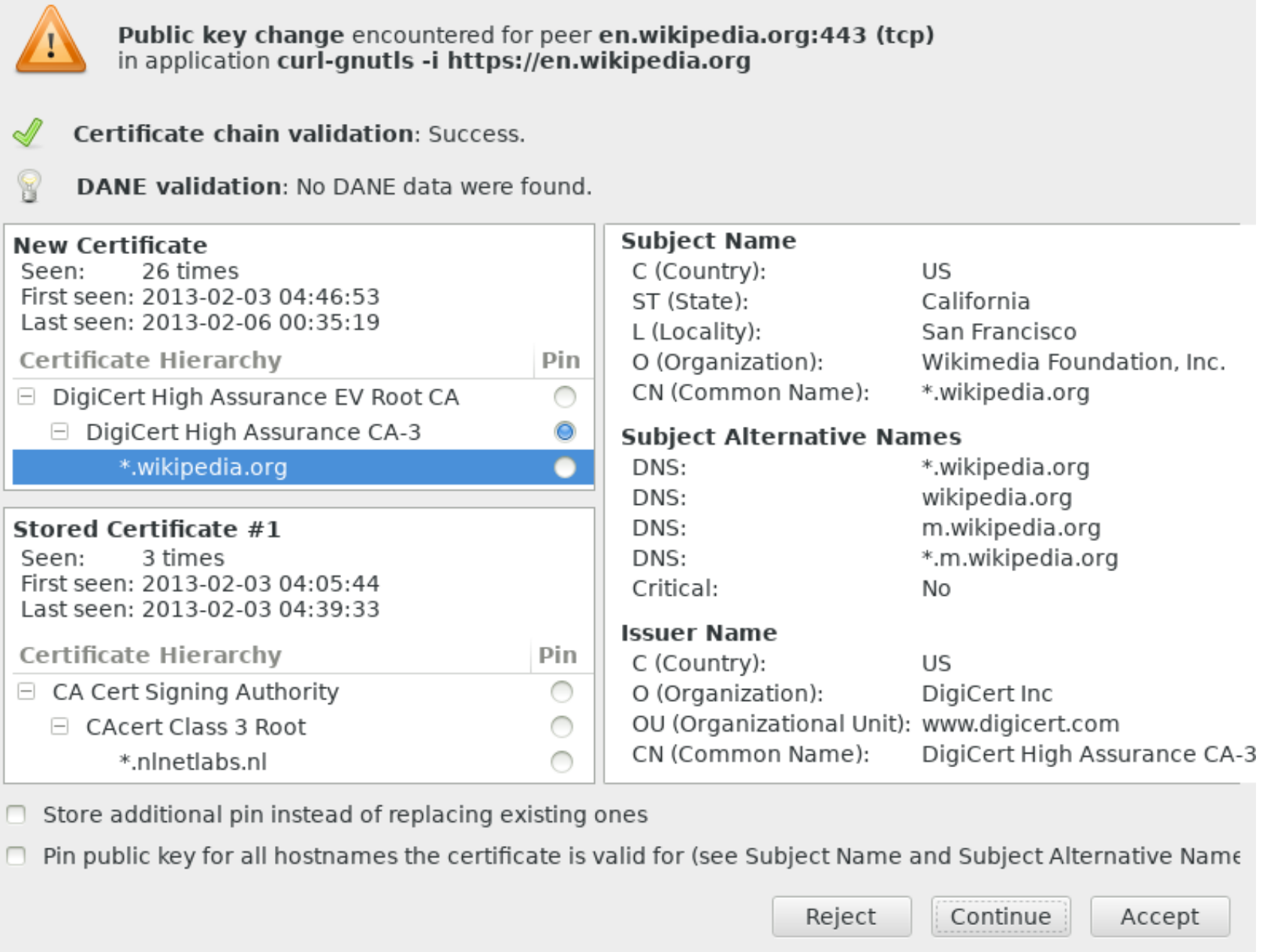
- Local database with pinning information
- Peers associated with one or more pinned public keys
- Some large sites use multiple active certificates for the same host
- Store each certificate encountered for a peer

Verification process

- Go through entries stored for a peer
- Check pinned public keys against certificate chain to be verified

Notifications

- Show a dialog when a certificate change occurs
- Accept: pin public key at the chosen level
- Reject: mark public key as rejected, causes validation failure
- Continue: accept just once, do not pin it


Public key change encountered for peer en.wikipedia.org:443 (tcp)
in application curl-gnutls -i https://en.wikipedia.org
Certificate chain validation: Success.
DANE validation: No DANE data were found.
New Certificate
Seen: 26 times
First seen: 2013-02-03 04:46:53
Last seen: 2013-02-06 00:35:19
Certificate Hierarchy
Pin
DigiCert High Assurance EV Root CA
DigiCert High Assurance CA-3
*.wikipedia.org
Stored Certificate #1
Seen: 3 times
First seen: 2013-02-03 04:05:44
Last seen: 2013-02-03 04:39:33
Certificate Hierarchy
Pin
CA Cert Signing Authority
CAcert Class 3 Root
*.nlnetlabs.nl
Subject Name
C (Country): US
ST (State): California
L (Locality): San Francisco
O (Organization): Wikimedia Foundation, Inc.
CN (Common Name): *.wikipedia.org
Subject Alternative Names
DNS: *.wikipedia.org
DNS: wikipedia.org
DNS: m.wikipedia.org
DNS: *.m.wikipedia.org
Critical: No
Issuer Name
C (Country): US
O (Organization): DigiCert Inc
OU (Organizational Unit): www.digicert.com
CN (Common Name): DigiCert High Assurance CA-3
Store additional pin instead of replacing existing ones
Pin public key for all hostnames the certificate is valid for (see Subject Name and Subject Alternative Name
Reject
Continue
Accept

Usability

- Default pin level can be set: end entity, issuer CA, root CA
- Increasing pin level reduces the amount of notifications

Questions?