A Qualitative Comparison of SSL Validation Alternatives

OWASP AppSec Research 2013

August 22nd, 2013

Henning Perl, Sascha Fahl, Michael Brenner, and Matthew Smith
Leibniz Universität Hannover
Outline Of This Talk

- What's SSL again?
- Things broken in SSL
- So many solutions!
- The best solution (or why there isn't any yet)
- Our evaluation system
How SSL works

Client — Secure connection — Server

trusts

authenticates to

CAs

signs cert
How SSL works ...and breaks
SSL CA incidents

- In 2010, VeriSign was compromised, allowing the attackers to issue arbitrary certificates.
- In March 2011, an attacker from Iran was able to compromise the Comodo CA and get certificates for www.google.com, login.yahoo.com, login.skype.com, addons.mozilla.org, and login.live.com. A MITMA attack with at least one these certificate was observed.
- In August 2011, attackers used the DigiNotar CA to issue at least 200 fraudulent certificates and used them to impersonate web servers. The breach eventually lead to the exclusion of the CA from most browsers and operating systems.

⇒ weakest link security
Things broken in SSL

For sake of completeness

- **Users ignore warnings**
  (c.f. Sunshine et al., "Crying Wolf: An Empirical Study of SSL Warning Effectiveness")

- **Attacks against the cryptosystem**
  - Padding oracle attack ("Lucky Thirteen", S&P 2013)
  - Attacks against RC4 (Usenix 2013)

- **SSL stripping** (Marlinspike, Black Hat 2009)

- **SSL validation** / Weakest link CA security
Things broken in SSL

For sake of completeness

- **Users ignore warnings**
  (c.f. Sunshine et al., “Crying Wolf: An Empirical Study of SSL Warning Effectiveness”)

- **Attacks against the cryptosystem**
  - Padding oracle attack (“Lucky Thirteen”, S&P 2013)
  - Attacks against RC4 (Usenix 2013)

- **SSL stripping** (Marlinspike, Black Hat 2009)

- **SSL validation** / Weakest link CA security
Types of solutions:

- **Use of network perspective**
  Perspectives, Convergence

- **Keep a log of certificates**
  Sovereign Keys (SK), Certificate Transparency (CT), Accountable Key Infrastructure (AKI)

- **Serve certificates over DNS**
  DANE

- **Trust on first use**
  TACK
Network Perspective (Perspectives, Convergence)

globally distributed notaries

$N_1, N_2, N_3, N_4, N_5, \ldots, N_k$

Client

Server

✓ No extra software on server

✗ Network delay  ✗ Privacy

Perl, Fahl, Brenner, Smith | A Qualitative Comparison of SSL Validation Alternatives
Keep A Log Of Certificates  SK, CT, AKI

Client

Server

Secure connection

trusts

authenticates to

signs cert

CAs

Perl, Fahl, Brenner, Smith  A Qualitative Comparison of SSL Validation Alternatives
Keep A Log Of Certificates  SK, CT, AKI

Certificate Log

Certificate Log submits certificate

Proof of inclusion

trusts

authenticates to

Secure connection

Client

Server

CAs

signs cert

Perl, Fahl, Brenner, Smith | A Qualitative Comparison of SSL Validation Alternatives
**Keep A Log Of Certificates**  SK, CT, AKI

Certificate Log

Proof of inclusion

Client

Server

- **Certificate Log**
- **submits certificate**
- **trusts**
- **authenticates to**
- **secure connection**

- ✔ **No extra software on server**
- ✔ **No extra network delay**
- ✗ **Needs new infrastructure**
Serve Certificates Over DNS  

DANE

- CAs trusts signs cert
- Client authenticates to Secure connection
- Server

- No extra software on server
- Reuses infrastructure
- DNSSEC

Perl, Fahl, Brenner, Smith | A Qualitative Comparison of SSL Validation Alternatives
Serve Certificates Over DNS  DANE

Domain admin submits certificate

signs cert

certificate and DNS response

Domain admin

Client

Server

Secure connection

Perl, Fahl, Brenner, Smith | A Qualitative Comparison of SSL Validation Alternatives
Serve Certificates Over DNS

DNS Server

submits certificate

certificate and DNS response

Domain admin

signs cert

Client

Secure connection

Server

☑️ No extra software on server
☑️ reuses infrastructure

✗ DNSSEC

Perl, Fahl, Brenner, Smith | A Qualitative Comparison of SSL Validation Alternatives
Pinning  TACK

Pinning on TACK public key; TACK secret key signs actual cert.

First connection

Client -------- TACK public key -------- Server

Secure connection?

Subsequent connections

Client -------- TACK public key -------- Server

Secure connection
Pinning **TACK**

Pinning on **TACK public key**; TACK secret key signs actual cert.

---

First connection

- **Client**
- **TACK public key**
- **Server**
- Secure connection?

Subsequent connections

- **Client**
- **TACK public key**
- **Server**
- Secure connection

✓ No extra software on server ✓ no CAs (just selfsign)

✗ No protection on first visit
What do we draw from this?
Our Evaluation Scheme

Goals:

- Tool to compare solution
- Discussion about which properties are important
- Organize, formalize the debate
Our Evaluation Scheme

Goals:
- Tool to compare solution
- Discussion about which properties are important
- Organize, formalize the debate

Structure:
- One large table
- 12 Deployability Benefits
- 9 Security and Privacy Benefits
- Adversary Capabilities
  - Active MITMA required
  - Trusted CA certificate required
  - Compromising user chosen third parties required
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Ref.</th>
<th>Deployability</th>
<th>Security benefits</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL with CA-PKI (90's)</td>
<td>—</td>
<td>⬤ ● ● ● ● ● ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>Perspectives (2008)</td>
<td>[13]</td>
<td>● ● ● ○ ● ○ ○</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>DANE (2010)</td>
<td>[5]</td>
<td>● ○ ○ ● ○ ● ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>Convergence (2011)</td>
<td>[9]</td>
<td>● ● ● ● ○ ● ○</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>Sovereign Keys (2011)</td>
<td>[3]</td>
<td>● ○ ○ ○ ○ ○ ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>TACK (2012)</td>
<td>[8]</td>
<td>● ● ○ ○ ● ● ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
<tr>
<td>AKI (2013)</td>
<td>[6]</td>
<td>● ○ ○ ○ ● ● ●</td>
<td>○ ○ ○ ○ ○ ○ ○ ○</td>
<td>○ ○ ○ ○ ○ ○</td>
</tr>
</tbody>
</table>
Conclusion

- All proposals solve weakest link problem
- ...but in very different ways
- No clear winner
- Do we want/need/have to have CAs?
- Deployment is challenging
- Question: *When to fail hard?*