CAPS: Smoothly Transitioning to a More Resilient Web PKI

Stephanos Matsumoto\(^\ast\), Jay Bosamiya\(^\wedge\), Yucheng Dai\(^\wedge\), Paul van Oorschot\(^\S\), Bryan Parno\(^\wedge\)

\(^\ast\)Olin College of Engineering, \(^\wedge\)Carnegie Mellon University, \(^\S\)Carleton University
Introduction: MITM Attacks in the Web PKI

Alice

Eve

https://bob.com

ServerCertificate (Bob's CA)

ServerCertificate' (Eve's CA)

(user/pass)

(sensitive info')

(encrypted communication)

(encrypted communication)

(user/pass)

(sensitive info)
Introduction: Deployability vs. MITM Prevention

- Certificate Transparency (RFC 6962)
  - Easy to detect misissued certificates
  - Widely deployed in today's Web
  - PKI remains largely the same as today
  - Does not prevent MITM attacks
  - No way to communicate "true" keys

- ARPKI (Basin et al., CCS 2014)
  - Prevents MITM despite $n$ malicious CAs
  - Substantially changes certificate issuance
  - Formally proven security guarantees
  - Misconfiguration leads to inaccessibility
  - Flag day to switch to new authentication

How can we smoothly deploy improvements to the Web PKI to prevent MITM attacks?

- Prevent MITM attacks by establishing "authoritative keys"
- Keep domains accessible even if misconfigured/attacked
- Resist downgrade attacks that force use of the current PKI
Certificates with Automated Policies and Signaling

- Use global, public logging as a communication channel for domains and CAs
- Signal deployment of both HTTPS and CAPS using a global view of the Web PKI
- Leverage the existing PKI to establish authoritative public keys for each domain
- Bootstrap deployment of Web PKI improvements using authoritative public keys
CAPS: Architecture and Overview

Client

TLS Handshake

Domain

Certificate

CA

Logs

Certificate Database

Collect scans

Log Aggregators

Collect certificates

Signaling Set
Data structure to indicate which domains are accessible over HTTPS and over CAPS

Policy Proof
Signed message indicating the max number of CAs that certify a public key for a domain

Transmit (and updates)

Fetch policy proofs

Log certificate
CAPS: Constructing the Signaling Set

Log Aggregators

Set
ca.foo
com.bar
com.baz
com.foo

Generate standard DAFSA
Compact long edges
An authoritative public key is any public key backed by a maximal number of independent certificate chains.

At least 2 CA private keys need to be compromised to forge this cert.

Certificate is valid for 2744 days.

Domains can tune their policy "strength" to their desired level of security against MITM attacks.

If misconfiguration or attack makes a domain inaccessible, it simply obtains more certs.

Current certificate issuance procedures stay as is – logs and scans publicize information.

Public key fingerprint: 6ac3c336...

Domain fingerprint: a83dd0cc...
**CAPS: Connection Establishment**

Alice

Check signaling set for com.bob
- If no HTTPS, use HTTP
- If HTTPS but not multicert, use regular TLS
- Otherwise, use CAPS extension

Hello (with CAPS extension: $k$)

ServerCertificate
  $k$ policy proofs
  $c - 1$ additional certificate chains

https://bob.com
### Evaluation: Signaling Set Construction

64,050,329 DNS names as of July 2018

<table>
<thead>
<tr>
<th>Representation</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext</td>
<td>1,509</td>
</tr>
<tr>
<td>zpaq (best-case compression)</td>
<td>104</td>
</tr>
<tr>
<td>DAFSA</td>
<td>190</td>
</tr>
<tr>
<td>DAFSA + zpaq</td>
<td>148</td>
</tr>
</tbody>
</table>

zpaq has better on-disk size, but DAFSA can be used as-is in memory when querying.
Evaluation: Connection Establishment Overhead

- **nginx/curl**: isolate effects of CAPS signaling set/extension
- **localhost**: loopback interface (lower bound on added latency)
- **WAN**: single client and VPS (realistic estimate of added latency)
- ~5% increase in latency on average
  - **localhost**: ~1.2ms
  - **WAN**: ~11ms
Conclusions

CAPS uses global view of Web PKI to **signal deployment and prevent downgrade attacks**

Flexible CAPS policies prevent MITM attacks **while avoiding pitfalls of misconfiguration**

CAPS imposes only **modest** overhead on storage, memory, and handshake latency

CAPS provides for a **secure, smooth transition to a more resilient Web PKI**

Thank you!

smatsumoto@olin.edu

https://github.com/syclops/caps