When HTTPS Meets CDN

A Case of Authentication in Delegated Service

Jinjin Liang¹, Jian Jiang¹, Haixin Duan¹, Kang Li², Tao Wan³, Jianping Wu¹

¹ Tsinghua University  ² University of Georgia  ³ Huawei Canada
• This work is about **HTTPS, CDN** and how they fail to work together.

• Our contributions
  – Identified the gap between HTTPS and CDN, and exposed a number of security issues in current practice through extensive measurements
  – Proposed a lightweight and flexible solution to fill the gap.
Our Sensitive Information Is Transmitted Over the Web
The Website Is Using HTTPS

- HTTP over SSL/TLS
  - Authentication, Encryption
- Server Certificate serves as website identity
  - Domain Validation (DV)
  - Organization Validation (OV)
- Extended Validation (EV)
The Website Is Using CDN

• Caching servers around the world
  – performance
  – security

• Request Routing
  – URL Rewriting
  – DNS Based
    • CNAME
    • Domain Hosting
The Website Is Using CDN

• Caching servers around the world
  – performance
  – security

• Request Routing

CNAME  online.citibank.com.edgekey.net.
IN CNAME  e5035.b.akamaiedge.net.
A  184.28.156.106

• Domain Hosting
When HTTPS Meets CDN

- From 2 parties to 3 parties
- Break into Frontend and Backend
Backend Communication

User → Frontend → CDN → Backend → Website
The Current Practice in Backend

- Experiment on 5 CDNs in Nov. 2013

<table>
<thead>
<tr>
<th>CDN Provider</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDN77</td>
<td>HTTP</td>
</tr>
<tr>
<td>CDN.NET</td>
<td>HTTP</td>
</tr>
<tr>
<td>CloudFlare</td>
<td>HTTPS, not validate certificate</td>
</tr>
<tr>
<td>Incapsula</td>
<td>HTTPS, not validate certificate</td>
</tr>
<tr>
<td>CloudFront</td>
<td>HTTPS, not validate common name</td>
</tr>
</tbody>
</table>
A Security Incident We (CCERT) handled

- CERNET; April 15, 2014
- Victims: a large CDN and a famous website
- A MITM attack injected a fake JS file into CDN caching server
Backend Should Use HTTPS and Validate Certificate

• Response from industries
  – CloudFlare (Fixed)
  – CloudFront (Fixed)
  – Incapsula (Fixing)

• Customers want to use self-signed certificates?
  – Still could be validated
    • Ask customers to upload their certificates explicitly
    • Or use Trust on First Use (like SSH)
Frontend Communication

Backend is secure with standard HTTPS
Broken HTTPS Authentication in DNS Based Request Routing

website.com != cdn.com

Root Cause: Browser does not know the delegation from website to CDN!
Survey on CDNs and Websites

• 20 popular CDN providers

<table>
<thead>
<tr>
<th>Support DNS Routing</th>
<th>Support HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>19</td>
</tr>
</tbody>
</table>

• Alexa Top 1M websites
  – 10,721 use CDN and HTTPS

<table>
<thead>
<tr>
<th>Invalid Certificate</th>
<th>Valid Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status 200</td>
<td>Other</td>
</tr>
<tr>
<td>15%</td>
<td>54%</td>
</tr>
<tr>
<td>69%</td>
<td></td>
</tr>
</tbody>
</table>
Custom Certificate (Type I)

- Website’s CA
- Website’s Cert
  - CN: website.com
- Website
  - Upload Certificate And Private Key
- CDN
- HTTPS
- User Browser

- Have to share private key
- Heavy key management overhead
- Inefficient revocation
Custom Certificate (Type II)

Not covered in the paper

- Heavy key management overhead
- Inefficient issuance and revocation
Custom Certificate (Type II)

Not covered in the paper

Website’s CA

<table>
<thead>
<tr>
<th>Issued To</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name (CN)</td>
<td><a href="http://www.apple.com">www.apple.com</a></td>
</tr>
<tr>
<td>Organization (O)</td>
<td>Apple Inc.</td>
</tr>
<tr>
<td>Organizational Unit (OU)</td>
<td>Internet Services for Akamai</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issued By</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name (CN)</td>
<td>Symantec Class 3 EV SSL CA - G3</td>
</tr>
<tr>
<td>Organization (O)</td>
<td>Symantec Corporation</td>
</tr>
<tr>
<td>Organizational Unit (OU)</td>
<td>Symantec Trust Network</td>
</tr>
</tbody>
</table>

- Inefficient issuance and revocation
Shared Certificate

CDN's CA

CDN's Cert

CN: cdn.com
SAN: website.com

Permit adding website.com into CDN's Certificate

CDN

HTTPS

Website

User Browser

- Improper security indicator (e.g. website has EV but CDN has DV/OV)
- Website cannot revoke the certificate
Permit adding website.com into CDN's Certificate

- Improper security indicator (e.g. website has EV but CDN has DV/OV)
- Website can not revoke the certificate
Case Study on Shared Certificate

- CDN: Incapsula (CA: GlobalSign)
  - Issuance: Email confirmation from CA
  - Revocation
    - Incapsula removed our website domain name in a new shared certificate
    - But our stale certificate was not revoked by CA
    - Contacted GlobalSign, but no response
- Incapsula said they would work on this problem with their CAs
Revocation Problem of Shared Certificate

- 1198 websites using shared certificate
- Certificate update, CRL and OCSP
- Last for 3 months
- 1865 certificate updates from 5 CDNs, but none was revoked
- Also discovered by Web PKI (NDSS 2014)
  - “this form of operation should be more strongly regulated”
Our Proposal for the Frontend
DANE with Delegation Semantics

How to improve authentication in the frontend
Background of DANE

- DNS-Based Authentication of Named Entities
- An alternative or complementary trust model
  - Problem of current CA model
  - Self-signed certificate
- Using DNS (TLSA) record to bind a certificate with a domain
  - Secured by DNSSEC
Idea: Put Delegation Token into DANE

Website’s Certificate
-----------------------
CN: website.com

Website

Binding Certificate

DANE
Domain: website.com
TLSA: website.com’s cert

DNSSEC
Idea: Put Delegation Token into DANE

Website’s Certificate
CN: website.com

Website

DNSSEC

DANE
Domain: website.com
TLSA: website.com’s cert

Binding Certificate

CDN

DNSSEC

DANE
Domain: website.com
TLSA: website.com’s cert

TLSA: cdn.com’s cert

Binding delegation

Binding Certificate
The Interaction of Authentication

Steps of standard web browsing:

1. User: https://website.com
2. Browser: website.com A?
3. CDN: CNAME website.cdn.com
4. CDN: TLS handshake starts
5. CDN: CDN’s Certificate
6. CDN: TLS handshake ends
7. CDN: HTTP GET
8. CDN: Content
9. Website: Content

---

User
Browser
CDN
Website
The Interaction of Authentication

User → Browser → CDN → Website

https://website.com

Steps of standard web browsing

Steps added/changed by DANE based solution
Proof of Concept

- Implemented as a Firefox Extension
Advantages and Limitation

• Lightweight and flexible
  – Only introduce one additional DNS request

• Delegation can be issued and revoked easily
  – Modifying TLSA record

• Deployability
  – Based on DANE, which is not deployed yet
  – But we believe DANE will be adopted in the near future
Conclusion

• Identified some problems of CDN’s current practice of HTTPS
  – Backend is vulnerable to MITM
  – Frontend has various issues, such as sharing private key, ignorance of certificate revocation, improper security indicator

• Proposed a new frontend solution based on DANE, which can avoid the existing issues

• It is not the end
  – CDN vendors should improve/standardize best practice
  – Raise further discussion in the community
Thank You!

Questions and Comments?