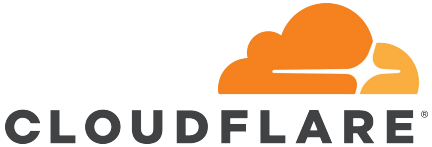


A view from the edge

CA/B Forum F2F Meeting — October 19, 2016

Patrick Donahue (@prdonahue), Security Engineering Team

Overview

1. Introduction
2. Data collection methodology and period
3. Observations from sampling of 100mm+ ClientHellos
4. Observations from 5mm+ origin servers
5. Quick updates: TLS 1.3 and Automatic HTTPS Rewrites
6. Cloudflare and CA/B: some requests for 2017
7. Cloudflare and CA/B: some plans for 2017
8. Questions/requests for data

Quick Introduction and Goals

1. General CA/B observations
2. Security Engineering/HTTPS at Cloudflare
3. Who are we/how do we have this data? i.e., where do we “sit” on the internet and how do we interact with this ecosystem?
4. Scale/geo distribution of request processing: PoPs, ratio of HTTP(S)
5. Our goals with respect to web PKI and this ecosystem
6. Actionable data
7. Where we’re headed (as it relates to CAs and browsers)

Edge Data Collection: Methodology & Periods

Collection:

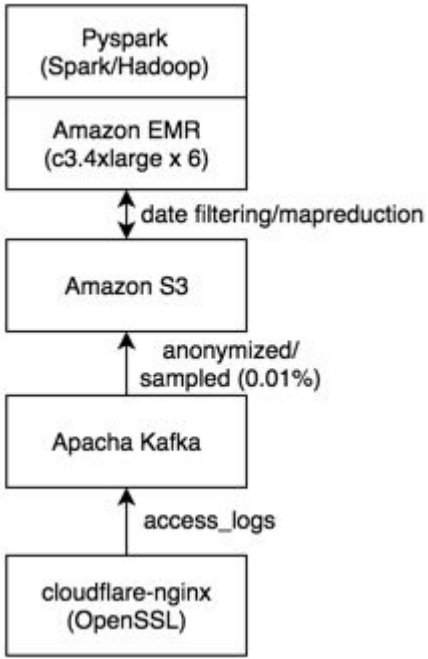
1. Access log extended with ClientHello data
2. Sampling of 0.01% inbound requests
   (~4+mm req/sec to sample CHs from)

Period:

1. March 14, 2016 vs. October 13, 2016 (7 mos. later)
2. Caveat: our traffic patterns != yours

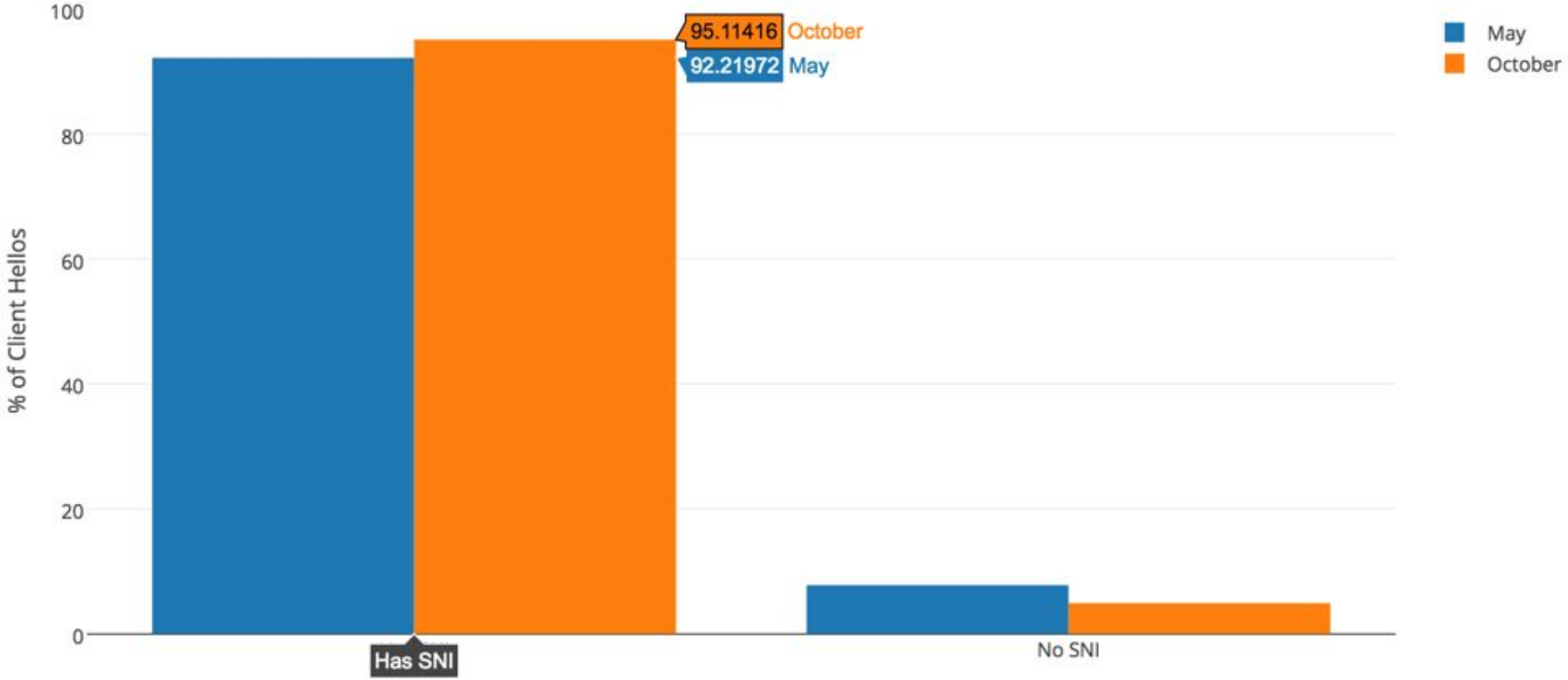
Analysis:

1. Apache Spark cluster (Amazon EMR) w/Pyspark



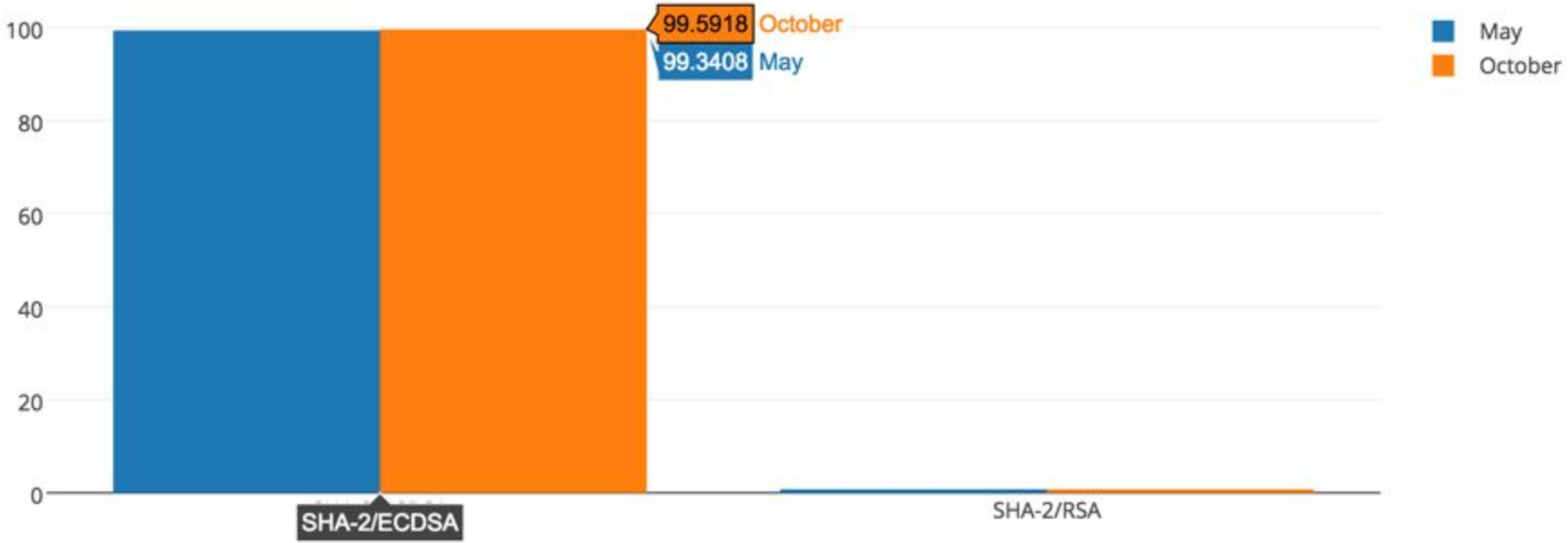
CLOUDFLARE

Edge: Server Name Indication



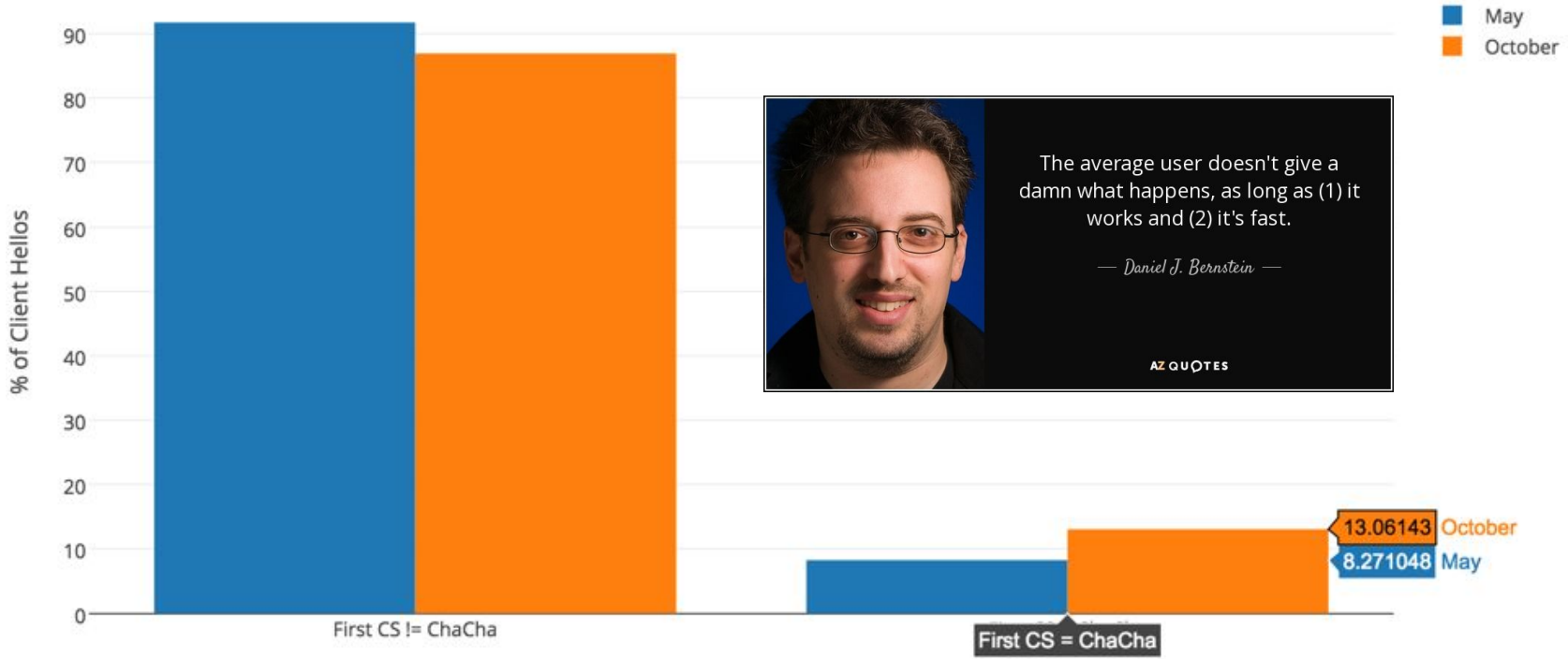
CLOUDFLARE

Edge: ECDSA Signature Algo. Support (1.2 only)



CLOUDFLARE

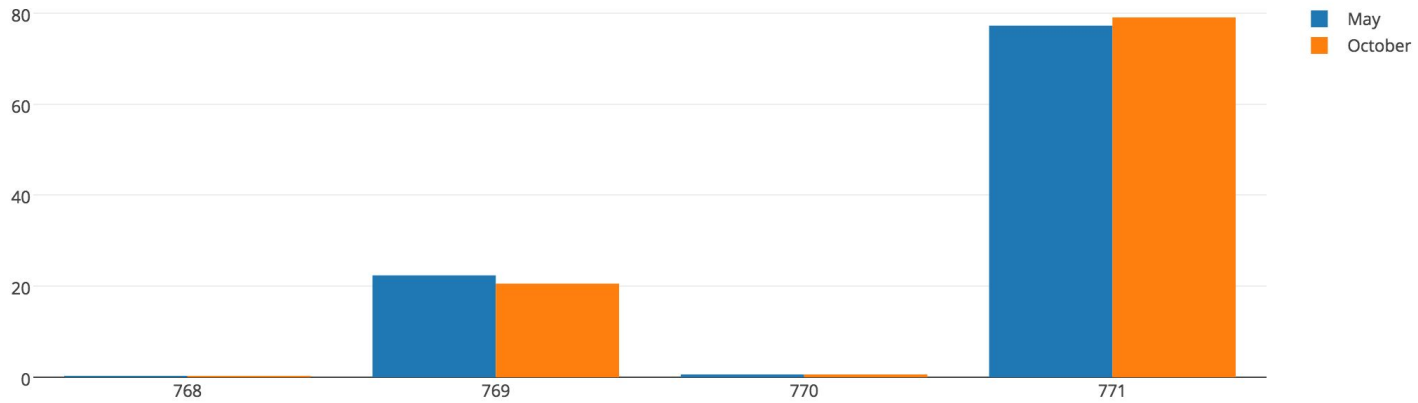
Edge: the rise of (prioritized) ChaCha



CLOUDFLARE

Edge: TLS version support

1. TLS 1.2 support (v=771) approaching 80%



CLOUDFLARE

Edge: SHA-1 Deprecation

1. Our logic: https://blog.cloudflare.com/tls-certificate-optimization-technical-details
2. Key metric: 1.8% of requests served with SHA-1/RSA signed certificates

Signature Algo.	Frequency (%)
SHA-2/RSA	61.4
SHA-2/ECDSA	36.8
SHA-1/RSA	1.8

CLOUDFLARE

Edge: Further analysis to be conducted

1. Bad/export cipher suites
2. Bad curves
3. % of requests routed through middleboxes/software (resulting in SHA-1)
4. What else would be helpful?

CLOUDFLARE

Origin Data Collection: Methodology & Periods

Collection:

1. Ran zgrab against 5mm+ origin zones. SNI is important here.
2. For simplicity, asked for “www” record. Resulting/analyzed set is unique certificates (after removing dupes, unresolvable/unreachable hostnames, etc.)

Period:

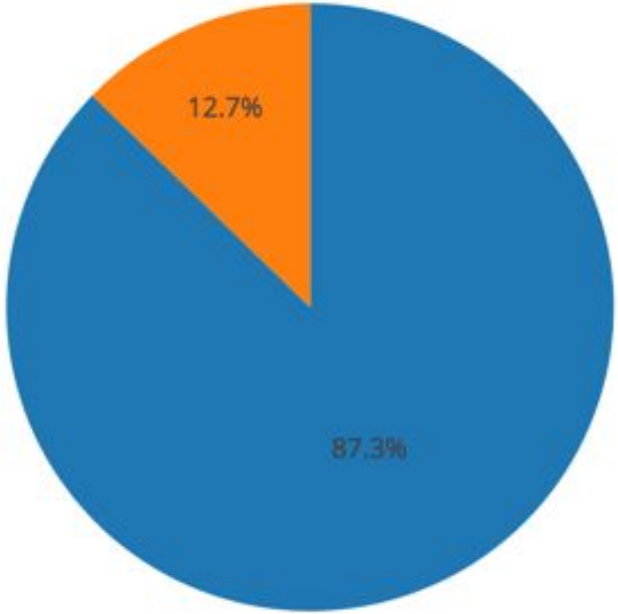
1. October 13, 2016

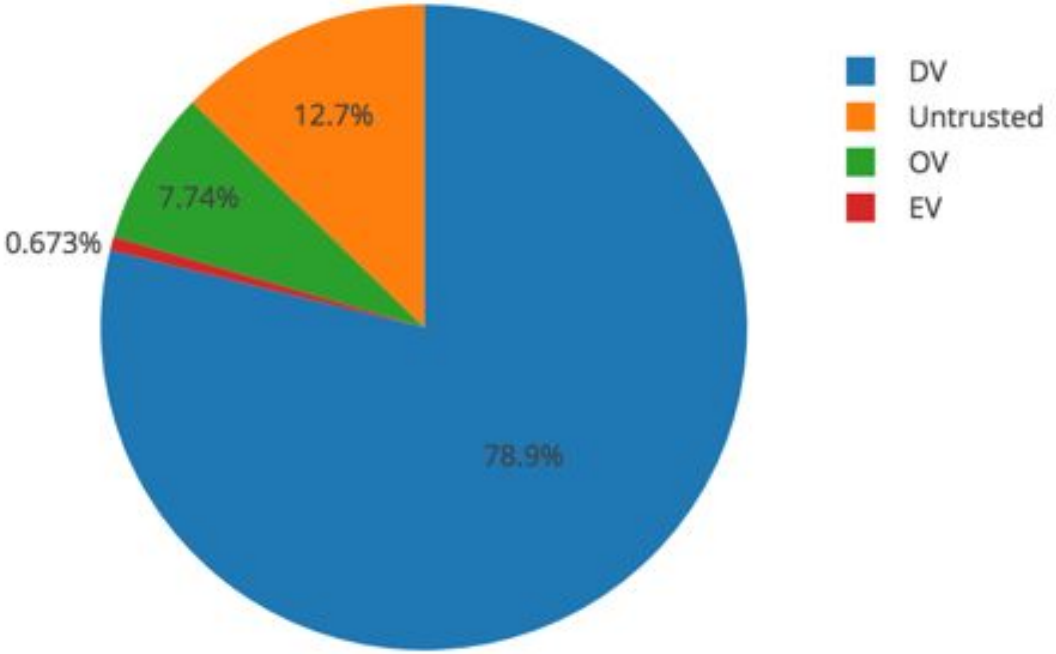
Analysis:

1. Apache Spark cluster (EMR) with TLS fingerprint engine from Zakir Durumeric and off-the-shelf OS/browser fingerprinting

CLOUDFLARE

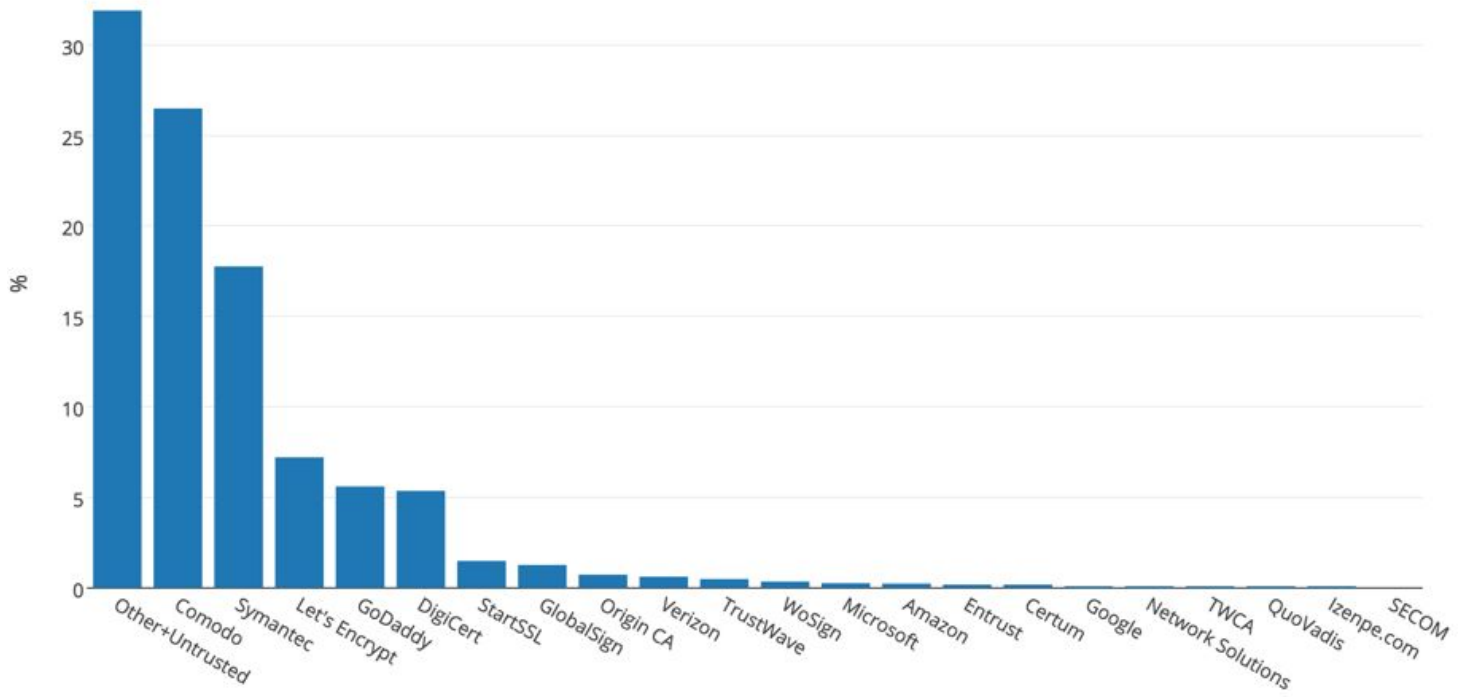
Origin Certs: Browser Trust & Validation Type





CLOUDFLARE

Origin Certs: Distribution by Cert. Authority (CA)



CLOUDFLARE

Origin Certs: Distribution by Cert. Authority (CA)

1. Neither Let's Encrypt (7.21%) nor our Origin CA (0.72%) have existed for very long, but now represent ~8% of certificates seen on origin.

2. StartSSL/WoSign represents ~2% of certs we see → customer communication

3. Mapping of intermediates to companies done with publicly available data set.

4. Other category includes the following untrusted CAs:
   a. cPanel, Inc. Certification Authority (5.5%)
   b. Parallels Panel (2.1%)
   c. Dreamhost (1.3%)
   d. Plesk (1.0%)

CLOUDFLARE

Origin Certs: Signature Algorithms

Signature Algo.	Frequency (%)
SHA-256/RSA	79.85
Untrusted	12.74
SHA-1/RSA	7.15
MD5/RSA	0.18
SHA-512/RSA	0.037
SHA-256/ECDSA	0.036

CLOUDFLARE

Origin: HTTP Daemons

1. Customers (still) need assistance installing certificates
2. Usual suspects on the list, but lack of standardization holding some back
3. Certbot? Other options?

Daemon	Frequency (%)
Apache	39.25
NGINX	34.38
IIS	5.64
LiteSpeed	4.62
Cowboy	1.81
Other	14.3

CLOUDFLARE

Update from @maxystrom: Automatic Rewrites

What we're doing:

1. Rewriting URLs in HTML documents (HTTP→HTTPS) when URL matches a rule in the EFF's HTTPS Everywhere ruleset or Google's HSTS preload list

What we're seeing:

1. 5% of zones have opted in so far
2. Rewriting 30 million URLs/second to HTTPS

What's next:

1. Enable for all free zones (default on coming soon); more intelligent detection of HTTPS capabilities for sites on Cloudflare
2. "Unsafe" rewrites for active mixed content (currently dogfooding)

CLOUDFLARE

Update from @grittygrease: TLS 1.3

What we did:

1. Implemented TLS 1.3 Draft 16. Currently using same certificates as TLS 1.2. RSA certificates use PSS signatures only (no RSA key exchange or PKCS#1 1.5)

What we're seeing:

1. Massive traffic! About 1 req per second.

What's next:

1. Beacon experiment (enabled.tls13.com and disabled.tls13.com) to be used by major browser to test performance.
2. Long term implementation (e.g., BoringSSL)

CA/B and Cloudflare: our wish list for 2017

What we'd like to see (and/or get clarification on):

1. Policy clarification
    a. Issuance risk
        i. Clarification on policy CAs must follow when evaluating whether to issue certificate, i.e., brand/phishing checks
    b. Uniformity of "prohibited" TLDs
2. ACME
    a. Standardized way to issue from all CAs (see also: Netflix Lemur)
3. EV
    a. Process for issuing EV certificate to individuals, not just companies.
4. Misc: adoption of Must-Staple; TLS 1.3 only certs w/critical extension?
5. Browsers
    a. UI indicator: (standardization) plans. Special designation for CT? Treatment of sub resources w/older TLS versions

CLOUDFLARE

CA/B and Cloudflare: our plans for 2017

Some of what we plan to do:

1. CAA
    a. Support these records in our dashboard; use as monitoring input
2. CT
    a. Ensure 100% of our existing certificates in logs; 100% issued with SCTs
    b. Monitor logs and report to our customers when certificates issued
3. OCSP
    a. 100% of responses with stapled OCSP response (cache); Expect-Staple
4. TLS 1.3
    a. Long term implementation, e.g., BoringSSL, within cloudflare-nginx
5. Other
    a. Improved SaaS SSL offerings, EV sale w/in dashboard, CSP headers

CLOUDFLARE

Questions? Requests for data?