Please leave feedback using Google Docs' suggestion and comment features.

Delivering the news over HTTPS

SRCCON 2015 • June 25, 2015, 3PM CDT • Session Transcript • SRCCON Slides • BigWP Slides • WordCamp Slides

Who we are

Mike Tigas, *ProPublica* • @mtigas
Paul Schreiber, *FiveThirtyEight* • @paulschreiber

The New York Times News Challenge

If you run a news site, or any site at all, we'd like to issue a friendly challenge to you. Make a commitment to have your site fully on HTTPS by the end of 2015 and pledge your support with the hashtag #https2015.
—Eitan Konigsburg, Rajiv Pant and Elena Kvochko
"Embracing HTTPS," November 13, 2014

What does HTTPS do?

HTTPS (HyperText Transfer Protocol — Secure) encrypts the content of websites so that it can't be observed or altered in transit. It ensures:

- site identity (the server you're talking to is the correct server for that domain)
- page integrity (what the server sends is what you receive)
- privacy (eavesdroppers can't see what pages you are viewing, or their content)
- safety (assets you load — like JavaScripts — can't be hijacked and weaponized)

HTTP is deprecated

Mozilla (April 2015) and Chrome (December 2014) have both publicly committed to marking HTTP as non-secure. (See also "We're Deprecating HTTP And It's Going To Be Okay.")

Configuration

Buying Certificates

You need to purchase a certificate. There are dozens of vendors. You can pay a lot for a certificate, but you don't need to. Here are three good options:

- SSLs.com — sells certificates inexpensively. Starts at $5.
- SSLMate — a command-line tool that automatically purchases and installs certificates. Certs $15.
- Let's Encrypt — coming September 2015. Free encryption!

Amazon will soon start selling certificates. That looks interesting.

Renewal

You need to renew your certificate every year. If you don't, your readers will get scary browser warnings about expired certificates. Set yourself a calendar reminder a week or two before it expires.

SSLMate can automatically renew certificates for domains registered with Amazon Route 53, CloudFlare, DNSimple and DigitalOcean.

Certificate types

- Standard certificates support one domain name (i.e. srccon.org and www.srccon.org)
- Wildcard certificates support subdomains (i.e. *.srccon.org supports srccon.org, www.srccon.org, mail.srccon.org, shoeboxes.srccon.org). Note that won't work with second-level subdomains, such as extra.shoeboxes.srccon.org.
- Multi-domain certificates, aka Subject Alternative Name (SAN) certificates, support multiple unrelated names on the same certificate (i.e. www.srccon.org and www.ilikepuppies.com)

Extended Validation certificates require additional paperwork and get you a nice green bar in the browser with your company name. Standard certificates are *domain validation* certificates.

Server Name Indication (SNI)

If you're on a shared server and don't have your own IP address, the server uses SNI to allow this sort of sharing. Some really old browsers (like IE on Windows XP or Python 2.6) don't support SNI.

Enabling HTTPS and HSTS (HTTP Strict Transport Security)

Your site can support HTTPS in four ways, with increasing security:

- HTTPS is optionally available
- HTTPS is on by default
- HTTPS is on by default, and enforced with HSTS
- HTTPS is on by default, enforced with HSTS, and on the HSTS preload list

HSTS ensures that the browser performs *all* requests via HTTPS, preventing some requests from being hijacked.

A good way to transition from HTTPS by default to HSTS is to apply a very short `max-age` (minutes) and slowly adjust it higher (to a year) as you work out issues.

Cryptography

Hashes

Certificates are cryptographically signed with a hash.

Older certificates are signed with a SHA-1 hash. This is deprecated and will generate warnings and errors. Google, Mozilla and Microsoft have publicly documented their deprecation plans for SHA-1, and Qualys has a good summary. You must not use SHA-1 certificates after 2016.

Newer certificates must be signed with SHA-2 (aka SHA-256). Make sure your certificates are signed with SHA-2. If you can't switched to SHA-2 right away, ensure your SHA-1 certificates expire before 2016.

Server configuration

There's a saying that you shouldn't write your own crypto implementation. The same things goes for configuring your web server's HTTPS support. It's messy. There are a zillion options (forward secrecy, cipher suites, etc.). It's easy to get wrong.

Use a tool to generate your configuration:

- Mozilla SSL Configuration Generator
- SSLMate's `mkconfig` command

Compatibility

Some older operating systems or browsers don’t support SNI or SHA-2. Cloudflare and Globalsign have detailed articles explaining SHA-2 incompatibilities. The main culprits are Windows XP (SP 1 or 2), Android 2.x, and Python 2.7.8 and below. Python 2.7.9+ and Python 3.x are fine.

Content

Once your certificate and server are configured, you’ll need to make sure your site content works over HTTPS. If some of it doesn’t, you’ll get mixed-content warnings.

- New content you create from today on is relatively easy to serve over HTTPS
- Existing *static* content, such as images and videos, is often straightforward to move or redirect
- Existing article content can be tricky — there are often lots of ads and dependencies to track down and fix
- Make sure your APIs and RSS feeds are served over HTTPS and include HTTPS URLs
- Make sure your news apps, interactives and other embeddable content will work over HTTPS when others use them
- Ask external folks whose content you embed to support HTTPS. This is sometimes hard.

Embedding

Good news! Almost all of the content you can embed already supports HTTPS:

- Video: YouTube, Vimeo, Vine, Instagram
- Audio: SoundCloud, SoundCite
- Documents: DocumentCloud, Scribd,
- Comments: Facebook, Disqus
- Social Sharing: Facebook, Twitter, Google Plus
- Tweets
- Polls: PollDaddy

Sadly, some news organizations’ players don’t (yet!) support HTTPS embeds. We hope our friends at the *New York Times* and NPR make this available soon.

Ads

Major ad networks support HTTPS! In April, Google committed to HTTPS support for DoubleClick. The Internet Advertising Bureau (IAB) issued a call to action telling folks to get moving. Outbrain supports HTTPS.

Analytics

Analytics providers *definitely* support HTTPS. Google Analytics and Omniture (aka Adobe Marketing Cloud) work with HTTPS sites.

In Google Analytics, turn on the `forceSSL` flag and use an HTTPS (instead of a protocol-relative) URL for google-analytics.com (see example).

A/B testing

Optimizely and Adobe Target support HTTPS.

Facebook comments

Facebook’s commenting system considers `http://site/page/` and `https://site/page/` to be distinct pages. If you migrate to https, you will need to ensure your `data-href` attribute still uses http for these old stories — otherwise your comments will disappear. I consider this to be a bug in Facebook. Sadly, they disagree.

Facebook likes

Similarly, Facebook's like count considers the http and https version of your site to be distinct. Bug report.

Fonts

All the major web font providers support HTTPS, including Google, Adobe TypeKit, fonts.com, typography.com, MyFonts and FontDeck.

JavaScript libraries

The CDNs for major JavaScript libraries — such jQuery, D3 and Knight Lab — support HTTPS.

Protocol-relative URLs

If you have a site that works over both https and http, when including content, you can use:

- http URLs `<img src="http://mysite.com/images/foo.gif">`
- https URLs `<img src="https://mysite.com/images/foo.gif">`
- protocol-relative URLs `<img src="//mysite.com/images/foo.gif">`

Use https URLs whenever you can. Use protocol-relative URLs only for iframes (when both the site and iframe support both). Don't use http URLs.

Mixed-content warnings

Passive content (images, audio, video) will be let through with a warning. Active content (JavaScript, CSS, iframe, XHR, WebSockets) is blocked.

Proxies

If you need to include third-party images or other assets that aren't available over HTTPS, you can set up a proxy to handle this, like Google does with Gmail. GitHub created camo, a free proxy for providing images over HTTPS.

Canonical URLs

Be sure to identify the HTTPS version of the URL as canonical.

- set `<link rel="canonical" href="https://mysite.com/story-123">`
- add a separate entry to In Google WebMaster Central for the https version of your site
- set up 301 redirects for non-canonical URLs

Content Security Policy

You can set a Content Security Policy to either automatically upgrade requests to HTTPS or log insecure requests. Upgrading insecure requests is really handy when you have a lot of legacy content, though the 'upgrade-insecure-requests' feature is not yet widely supported in browsers.

SEO

Using HTTPS will improve your placement in Google search results.

Content Delivery Networks (CDNs)

CDNs — including CloudFront, CloudFlare, Fastly and Akamai — support https. Each provider has a different fee structure and process for managing certificates. WordPress.com will put your content on a CDN.

CloudFlare even offers free SSL.

Tools & Resources

Many good tools exist to help you test and implement HTTPS sites:

- SSL Labs will evaluate your web server's configuration
- shaaaaaaaaaaaaa.com checks if your certificates are using SHA-1 or SHA-2
- badssl.com provides samples of problematic secure sites (mixed content, expired certificates, etc.)
- The Mixed Content Scan script identifies site assets/resources which are still on HTTP
- The EFF's HTTPS Everywhere Atlas has a list of sites/tools and their HTTPS equivalents
- The HTTPS Everywhere Chrome extension will list requests that are still on HTTP
- The chrome://net-internals panel lets you inspect and tweak low-level network settings
- CIO.gov's The HTTPS-Only Standard, including the Guide to securing APIs
- Google's Use canonical URLs
- The OpenSSL Cookbook documents OpenSSL's most frequently used features and commands
- Ads: Google SSL Implementation Guide and Google SSL FAQ
- Load Impact's HTTP/2 speed test
- HTTP2 details
- 2015 was not the year of HTTPS for news organizations by Melody Kramer
- SSL/TLS certificate pricing chart (updated December 4, 2015)
- Server-Gated Cryptography (SGC), explained
- moarTLS, a Chrome extension that helps you discover non-secure links on your pages
- CFSSL lets you evaluate the TLS configuration of internal sites
- securityheaders.io evaluates the HTTP headers your server sends
- report-uri.io is a cloud service for handling Content-Security-Policy requests
- cspisawesome.com helps you generate a content security policy
- Bulletproof SSL & TLS, a book

Cost

You'll pay for the following:

- **Certificates:** $0-1400 / year (you could probably find more expensive options if you tried)
- **Hosting provider setup fee:** $25–500
- **CDN bandwidth fee:** CloudFront 33% more; Fastly and CloudFlare appear to have no surcharge; Akamai has a surcharge, but doesn't publicly post prices
- **CDN monthly fee:** Fastly charges $100–1500 per month

Performance

While googling for "https performance" may get you dire warnings … from 2008, it's no longer a real problem today. With smarter protocols (like SPDY, which requires HTTPS, and HTTP 2.0, which requires HTTPS in major browsers), you'll see *better* performance on your secure site.

Even Netflix found they could move to HTTPS (after rewriting parts of the FreeBSD kernel…you don't need to do that).