Why Use HTTPS?

Google Chrome is marking non-HTTPS sites as "Not secure", this is just one of many good reasons to secure a website.

Why Use HTTPS? Learning Objectives

After reading this article you will be able to:

- Outline the changes to HTTPs traffic
- Explain The history of the steps google made to get here
- Explain the myths of HTTPs and the truth

Related Content

HTTPS

What is SSL?

What is an SSL Certificate?

SSL Handshake

Keyless SSL

What is the difference between HTTP and HTTPS?

HTTPS is HTTP with TLS encryption. HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses, making it safer and more secure. A website that uses HTTPS has https:// in the beginning of its URL instead of http://, like https://www.cloudflare.com.

So, why should websites use HTTPS?

Reason No. 1: Website using HTTPS are more trustworthy for users.

A website using HTTPS is like a restaurant displaying a "Pass" from the local food safety inspector: potential customers can trust that they can patronize the business without experiencing massively negative effects. And in this day and age, using HTTP is essentially like displaying a "Fail" food safety inspection sign: there's no guarantee that something terrible won't happen to a customer.

HTTPS uses the SSL/TLS protocol to encrypt communications so that attackers can't steal data. SSL/TLS also confirms that a website server is who it says it is, preventing impersonations. This stops multiple kinds of cyber attacks (just like food safety prevents illness).

Even though some users may be unaware of the benefits of SSL/TLS, modern browsers are making sure they're aware of the trustworthiness of a website no matter what.

Chrome and other browsers mark all HTTP websites as "not secure."

Google incrementally took steps to nudge websites towards incorporating HTTPS over a number of years. Google also uses HTTPS as a quality factor in how they return search results; the more secure the website, the less likely the visitor will be making a mistake by clicking on the link Google provided.

Starting in July 2018 with the release of Chrome 68, all unsecured HTTP traffic has been flagged in the URL bar as “not secure”. This notification appears for all websites without a valid SSL certificate. Other browsers have followed suit.

Reason No. 2: HTTPS is more secure, for both users and website owners.

With HTTPS, data is encrypted in transit in both directions: going to and coming from the origin server. The protocol keeps communications secure so that malicious parties can't observe what data is being sent. As a result usernames and passwords can't be stolen in transit when users enter them into a form. If websites or web applications have to send sensitive or personal data to users (for instance, bank account information), encryption protects that data as well.

Reason No. 3: HTTPS authenticates websites.

Users of rideshare apps such as Uber and Lyft don't have to get into an unfamiliar car on faith, just because the driver says they're there to pick them up. Instead the apps tell them information about the driver, like their name and appearance, what kind of car they drive, and the license plate number. User can check these things and be certain they are getting into the right car, even though every rideshare car is different and they've never seen the driver before.

Similarly, when a user navigates to a website, what they're actually doing is connecting to faraway computers that they don't know about, maintained by people they've never seen. An SSL certificate, which enables HTTPS, is like that driver information in the rideshare app. It represents external verification by a trustworthy third party that a web server is who it claims to be.

This prevents attacks in which an attacker impersonates or spoofs a website, making users think they're on the site they intended to reach when actually they're on a fake site. HTTPS authentication also does a lot to help a company website appear legitimate, and that influences user attitudes towards the company itself. (Users can check if a website is properly using HTTPS by testing it at the Cloudflare Diagnostic Center.)

HTTPS myth-conceptions

Many websites have been slow to adopt HTTPS. To explore why this is the case we have to look at the history.

When HTTPS initially began rolling out, proper implementation was hard, slow, and expensive; it was hard to implement properly, slowed down Internet requests, and increased costs by requiring expensive certificate services. None of these impediments remain true, but a lingering fear still exists for a lot of website owners, which has impeded some taking the leap into better security. Let’s explore some of the myths about HTTPS.

"I don't handle sensitive information on my website so I don't need HTTPS"

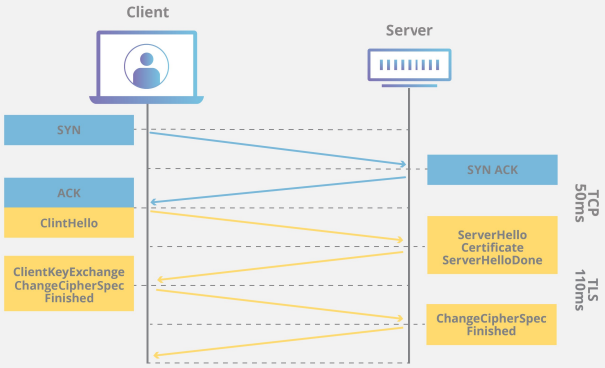
A common reason websites don't implement security is because they think it's overkill for their purposes. After all, if you're not dealing with sensitive data, who cares if someone is snooping? There are a few reasons that this is an overly simplistic view on web security. For example, some Internet service providers will actually inject advertising into HTTP-served websites. These ads may or may not be in line with the content of the website, and can potentially be offensive, aside from the fact that the website provider has no creative input or share of the revenue. These injected ads are no longer feasible once a site is secured.

Modern web browsers now limit functionality for sites that are not secure. Important features that improve the quality of the website now require HTTPS. Geolocation, push notifications and the service workers needed to run progressive web applications (PWAs) all require heightened security. This makes sense; data such as a user's location is sensitive and can be used for nefarious purposes.

"I don't want to damage my website's performance by increasing my page load times"

Performance is an important factor in both user experience and how Google returns results in search. Understandably, increasing latency is something to take seriously. Luckily, over time improvements have been made to HTTPS to reduce the performance overhead required to set up a encrypted connection.

When an HTTP connection occurs, there are a number of trips the connection needs to make between the client requesting the webpage and the server. Aside from the normal latency associated with a TCP handshake (shown in blue below), an additional TLS/SSL handshake (shown in yellow) must occur to use HTTPS.



Improvements can be implemented to reduce the total latency of creating a SSL connection, including TLS session resumption and TLS false start.

By using session resumption a server can keep a connection alive for longer by resuming the same session for additional requests. Keeping the connection alive saves time spent renegotiating the connection when the client requires an uncached origin fetch, reducing the total RTT by 50%.

Another improvement to the speed at which an encrypted channel can be created is to implement a process called TLS false start, which cuts down on the latency by sending the encrypted data before the client has finished authentication. For more information explore how TLS/SSL works on a CDN.

Last but not least, HTTPS unlocks performance enhancements using HTTP/2 that let you do cool things like server pushing and multiplexing which can greatly optimize performance for HTTP requests. In total there is a significant performance benefit for making the switch.

"It's too expensive for me to implement HTTPS"

At one point this may have been true, but now the cost is no longer a concern; Cloudflare offers websites the ability to encrypt transit free of charge. We were the first to provide SSL at no cost, and we continue to do so. By improving Internet security at large, we are able to help make the Internet safer and faster.

"I'm going to lose search ranking while migrating my site to HTTPS"

There are risks associated with website migration, and done improperly a negative SEO impact is possible. Potential pitfalls include website downtime, uncrawled webpages, and penalization for content duplication when two copies of the site exist at the same time. That said, websites can be migrated safely to HTTPS by following best practices.

Two of the most important migration practices are:

1) using 301 redirects and 2) the proper placement of canonical tags. By using server 301 redirects on the HTTP site to point to the HTTPS version, a website tells Google to move to the new location for all search and indexing purposes. By placing canonical tags on the HTTPS site only, crawlers such as Googlebot will know that the new secure content should be considered canonical going forward.

If you have a large number of pages and are concerned that the recrawl will take too long, reach out to Google and tell them how much traffic you're willing to put through your website. The network engineers will then crank up the crawl rate to help parse your site quickly and get it indexed.

+1 (888) 99 FLARE

About HTTPS

What is HTTPS?

Why Use HTTPS?

HTTP Security Gaps

Connection Not Private

About Encryption

What is Encryption?

Public Key Encryption

Asymmetric Encryption

Lava Lamp Encryption

Cryptographic Key

What is a Session Key?

SSL Glossary

What is Mixed Content?

SSL Handshake

What is an SSL Certificate?

SSL Certificate Types

Why Use TLS 1.3?

What is SNI?

What is Encrypted SNI?

What is Domain Spoofing?

Learning Center Navigation

Learning Center Home

DDoS Learning Center

CDN Learning Center

DNS Learning Center

Performance Learning Center

Security Learning Center

Serverless Learning Center

Bots Learning Center

Cloud Learning Center

Access Management Learning Center

Network Layer Learning Center

Privacy Learning Center

Video Streaming Learning Center