



Move your website to **HTTPS**

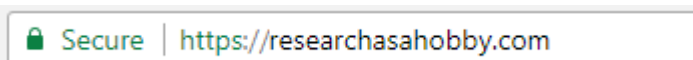
Tutorial from Michael Bely and ResearchAsAHobby.com

You are a website owner and it's likely that you did not care much about getting SSL and moving your website to HTTPS. But starting from October 2017 Google Chrome indicates non-HTTPS web pages as not secure the following way:



My tutorial (including video) will help you to get rid of this warning by moving your WordPress site to HTTPS easily and avoiding pitfalls. With this article even a person with little technical knowledge can switch to HTTPS correctly, easily and in a safe way.

And this is how your website URL may look like after migration to HTTPS:



This tutorial is based on my practical experience.

And this is quite a long article because I included **practical tips and tricks** which will let you move your website to HTTPS **on your own and for free**. I also mention paid options which can make the process even easier.

I've written this tutorial first of all for shared hosting users with WordPress websites. However, other users may find this article useful too as it contains universal practical advice.

In this tutorial I focus on the **most technical part** of moving your site to HTTPS:

- getting HTTPS in from of your website URL;

- fixing HTTPS issues on your website to get rid of the ‘non-secure’ warnings (so-called ‘mixed content’ issues).

Also I cover **other business concerns and options** that you need to be aware of when migrating your site to HTTPS:

- choosing right SSL certificate;
- SEO concerns like duplicate content, your site positions in Google search, existing backlinks to the HTTP version of your site;
- losing your site social shares and likes;
- affiliate tracking issues (if your affiliate target URL is HTTP);
- your site speed concerns when using HTTPS;
- enhanced website security when using HTTPS.

I'll show you that you can move your site to HTTPS the safe and right way easily even if you are far from being a programmer 😊

Contents

- Tutorial intro
- 1. Choose the SSL certificate that will fit you the best
- 2. Make a backup of your site or/and test on a clone site
- 3. Decide where from you will take the SSL certificate and install it
 - If you want a free SSL certificate
 - If you want a paid SSL certificate
 - Check your SSL certificate
- 4. Update URLs (change HTTP to HTTPS)
 - 4.1. General information
 - 4.2. Options you have
 - 4.3. Set HTTPS in the WordPress settings (Site URL and Home URL)
 - 4.4. Updating inner links to HTTPS in your website content
 - 4.5. Make sure your website pages do not have ‘mixed content’ issue
 - 4.6. Fixing ‘mixed content’ issues on your HTTPS pages
 - 4.6.1. General overview of the process and video walk-through
 - 4.6.2. Using Google Developer Tools to find mixed content
 - 4.6.3. Replacing insecure links in the files
 - 4.6.4. Viewing page source code to locate mixed content
 - 4.6.5. Replacing insecure links in the content (advanced)
 - 4.6.6. Using Search & Replace plugin
- 5. Managing redirects from HTTP to HTTPS version of your website
- 6. Check out robots.txt – whether HTTPS pages are blocked
- 7. Make sure your site map now contains HTTPS pages
- 8. Let Webmaster Tools and Google Analytics know about your migration to HTTPS
- 9. Setting up your CDN and/or cloud firewall for HTTPS
- 10. SEO concerns connected with migrating to HTTPS
- 11. Loosing social media shares after migrating to HTTPS
- 12. Speed concerns after migrating to HTTPS
- 13. Additional steps and other information
- Conclusion

By the way, here's a disclosure: There are some affiliate links on this page for the products which I mention: [Really Simple SSL pro](#) plugin that simplifies your site migration to HTTPS. Also I feature some of the [the hosting companies](#) which I recommend as well as a couple of other products. In other words, I get paid if you click on the links and make a purchase. All such links open in new window/tab; no software/program will be installed to your computer. (This is a standard notice required by affiliate programs terms.) Please note that I mention the products not as an advertisement, but as my recommendation.

Tutorial intro

The non-secure warning I've shown above appears in your visitor's browser *if your HTTP web page has an entry field* (like a contact form, comments section, password field etc).

This warning sucks as it may negatively impact your website visits and conversions. Non-technical visitors may simply be frightened off and leave your website as soon as they see it.

So, it makes a good sense to move your site to HTTPS, even if earlier it had been not that necessary.

I've written this tutorial to help you solve this issue and let your website look legitimate and safe in your visitors' eyes again. I will show you how to move your site to HTTPS easily and correctly.

When moving your site to HTTPS, there's a risk of making your site slower or lose SEO value. As well as **there are some other pitfalls** I mentioned [above](#). But doing everything correctly will let you avoid it all.

When done right, HTTPS even can be beneficial for your website SEO. At least Google [mentioned](#) it so. I would not count on it much though, but anyway HTTPS is one of the positive ranking signals. More about HTTPS and SEO you can read [below](#).

The process of moving your website to HTTPS the easiest and the cheapest (free) way consists of the following aspects:

1. You need to use a hosting which supports a free SSL (more on it in the [next chapter](#)).
2. You make some simple adjustments in your WordPress dashboard or/and in some service files in your WordPress installation.
3. You make sure that you use elements on your website in a safe way.
4. You adjust settings in your Google Webmaster console and Google Analytics.

I'll show you below how to do it all for free.

By the way, free SSL has some restrictions (e.g. multiple domains, subdomains, not all hosts support free SSL). In some cases you will need to use a paid SSL certificate. More on it below in the next section.

By the way, **if you prefer using a paid SSL certificate**, the procedure of moving your website to HTTPS will look similar:

1. You buy a SSL certificate from your hosting or the 3d party (more on it in [this chapter](#)).
2. You install the SSL certificate in your hosting account (a non-technical can do it; see more details in [this chapter](#)).
And then go the same steps as in the case with a free SSL certificate:
3. You make some simple adjustments in your WordPress dashboard or/and in some service files in your WordPress installation.
4. You make sure that you use elements on your website in a safe way.
5. You adjust settings in your Google Webmaster console and Google Analytics.

So, let's get down to the step-by-step tutorial now.

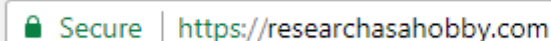
1. Choose the SSL certificate that will fit you the best

To put it super simply, [SSL](#) certificate is a security technology that let your site use HTTPS. Sometimes in the Internet you may notice the term "[TLS](#)". It's basically a synonym to "SSL" as far as it concerns moving your site to HTTPS. So just don't be scared off if you see SSL/TLS instead of just SSL.

Special entities called Certificate Authorities issue SSL certificate for your website and your hosting installs it on your account. Some hosts make the SSL issuing and installation hassle-free (or even automatic and bu default) for their clients.

There are **different options and limitations that you can get with different types of SSL certificates**. To put it simply, you can get:

1. The simplest SSL certificate that allows your domain name have HTTPS and a padlock icon in its URL.
 - If you don't want anything special (like the options below), and just HTTPS is enough for you, then a free certificate (e.g. Let's Encrypt) will do fine for you. This type of SSL is called **Domain Validation**.



This is how a domain with Domain Validation SSL certificate looks like in Google Chrome

- Note: a free SSL from a well-established Certificate Authority does not normally support subdomains (at least until [2018](#)).
- Multiple domains are also not supported by free SSLs at the moment on shared hosting. You need either a VPS or a reseller plan or a paid SSL that supports multiple domains.

SSL that provides HTTPS not just for your single domain, but also for your subdomains or/and multiple domains. This type of SSL is called **Wildcard Certificate**. For your website visitors it looks like the Domain Validation certificate displayed above.

SSL that is issued by a recognized authority which verifies your organization. Your authenticated organization details will be available in your visitor's browser's security settings. This type of SSL is called **Organization Validation**.

- In the browser URL your visitors will see the same as in case with Domain Validation SSL. But in the certificate information (available through browser settings) it will be noted that this is organization validation, not just domain validation.

SSL that displays your business name beside your website URL in a green bar or font (referred to as "secure green browser bar" sometimes). This type of SSL is called Extended Validation.



A2 Hosting, Inc [US] | https://www.a2hosting.com

This is how a domain with Extended Validation SSL certificate looks like in Google Chrome

To get a feeling of prices and to compare different types of SSL certificates, have a look at [this Namecheap's page](#). Namecheap is a reputable domain registrar which offers a really nice wide range of SSL certificates for very affordable prices. There you can add to comparison multiple SSL certificates to see stronger sides of each option.

By the way, it's a good idea to use a free SSL certificate installed by your hosting. And if you need a paid SSL certificate, you can buy it from your hosting or by one of the SSL resellers. Namecheap is a good and affordable place to make purchase.

After all, if you don't want to pay for a SSL certificate, you don't have much choice. You then just need your hosting to support any free SSL certificate. Most likely it will be Let's Encrypt certificate (probably managed with AutoSSL tool provided by cPanel).

Have a look at a short overview of [Let's Encrypt limitations](#). Still, Let's Encrypt SSL certificate is often a good choice for simple blogs and non-commercial websites.

However, some hosts do not want to support any free SSL certificates, but offer only paid SSL certificates instead. So, if you use such shared hosting, the only way for you to move your website to HTTPS is either to pay for the cheapest SSL provided by your hosting or switch to the host which supports a free SSL. By the way, you can find [my recommended hosts here](#).

2. Make a backup of your site or/and test on a clone site

The process of moving your site to HTTPS includes modification of your website installation. And it's very likely you will need to edit internal links in your website. This means you will modify your database or/and website files during the process.

So, if you have not made a **fresh backup of both your database and files**, it's time to do it.

By the way, I have an article devoted to [WordPress backup solutions here](#).

It's also a good idea to make a [full backup](#) of your site (including all your hosting settings). A full backup contains not only your website data, but also other your hosting account data including SSL settings.

Besides, **if you don't want to risk and would like to test how you can migrate your site to HTTPS** before making any changes on your actual website, I can suggest making a clone of your site and test switching to HTTPS on it first. To do that, you need to do the following:

1. Sign up with some affordable hosting which supports free SSL (e.g. Let's Encrypt). Paying for just one month is enough. Note that you can't have two domains with Let's Encrypt SSL certificates under one hosting account. That's why you need another hosting account even if your current hosting plan supports add-on domains or subdomains.
2. Register a new domain (the domain for your tests on the new hosting). By the way, my favorite domain registrars including cheap ones are [here](#).
3. Clone your website to a new domain ([here's my tutorial](#) how to do it).
4. And test your migration to HTTPS on your clone site.

Alright, finished with the preparation. Let's get to the tutorial now!

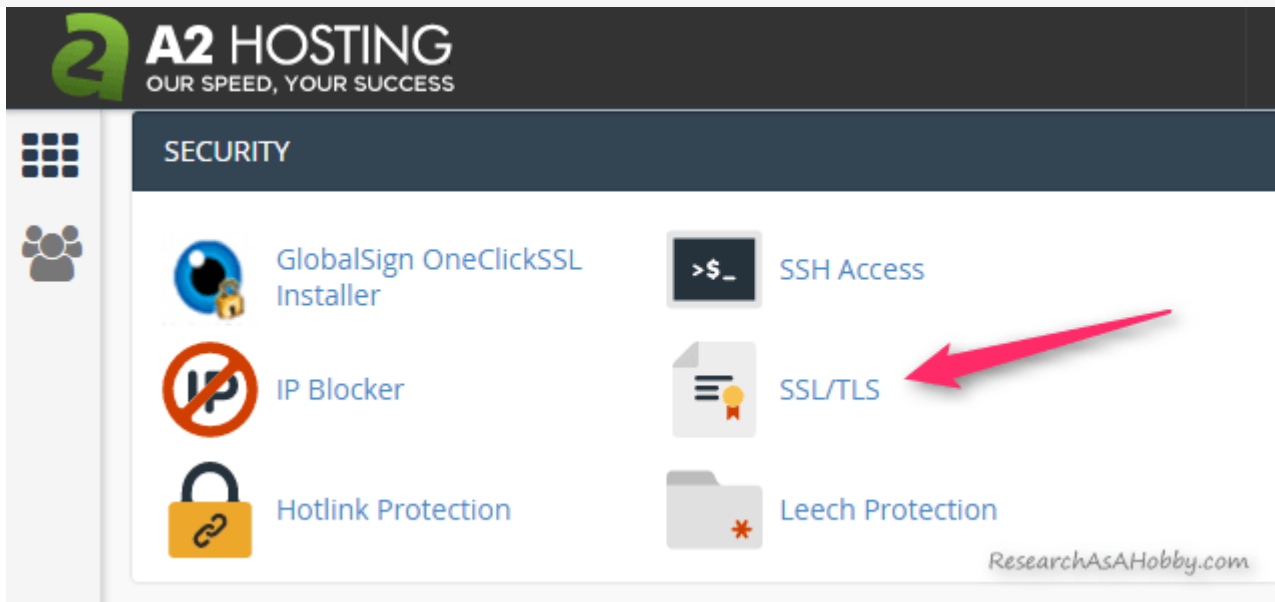
3. Decide where from you will take the SSL certificate and install it

If you want a free SSL certificate

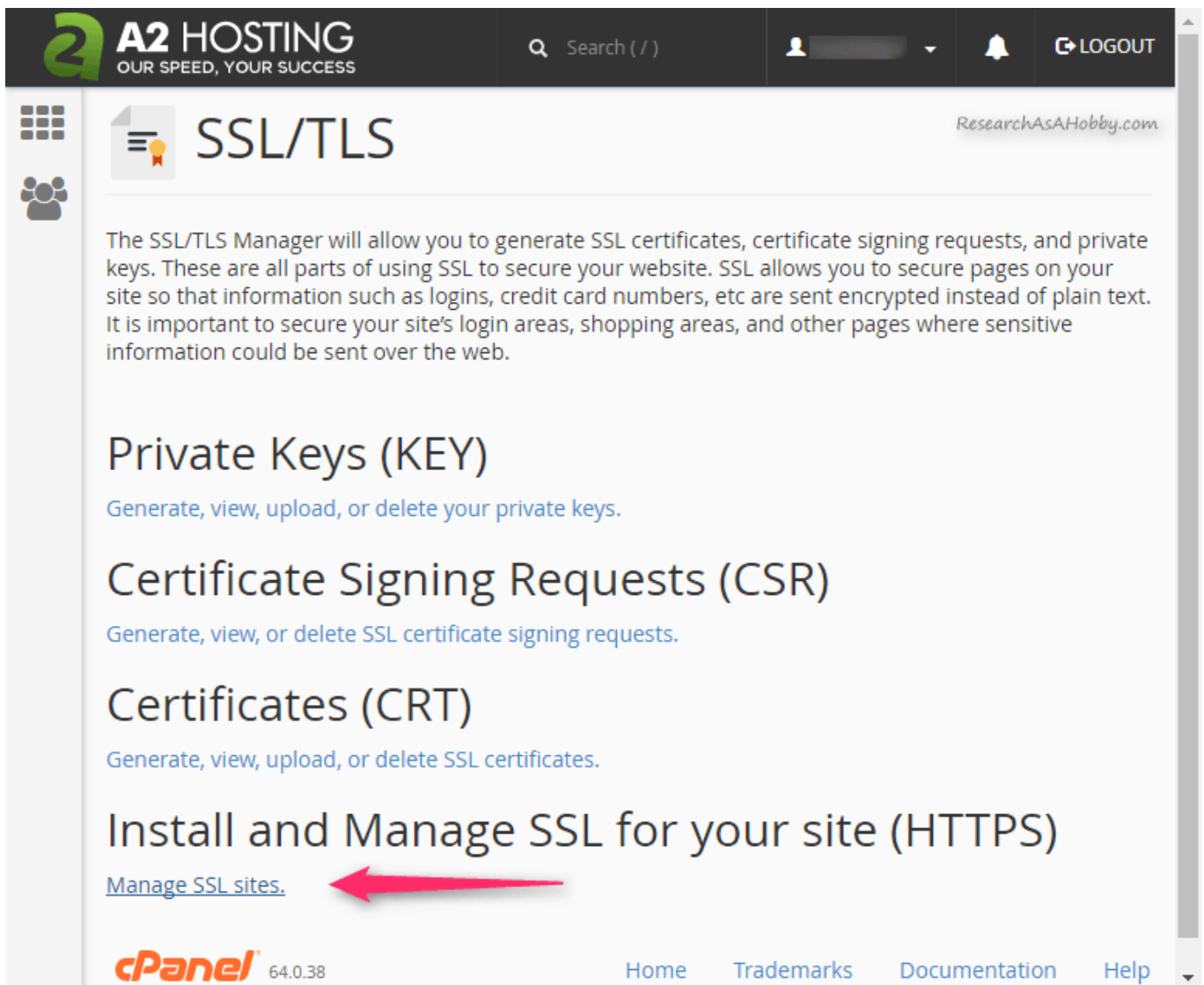
As I've mentioned above, the easiest and often the free way is to get SSL certificate provided by your hosting. In this case it's very likely that you don't need to do anything since SSL certificate is already installed on your hosting account. And you are ready to move you site to HTTPS.

Here are the screenshots below on which you can see **how to make sure a SSL certificate is installed on your hosting account**.

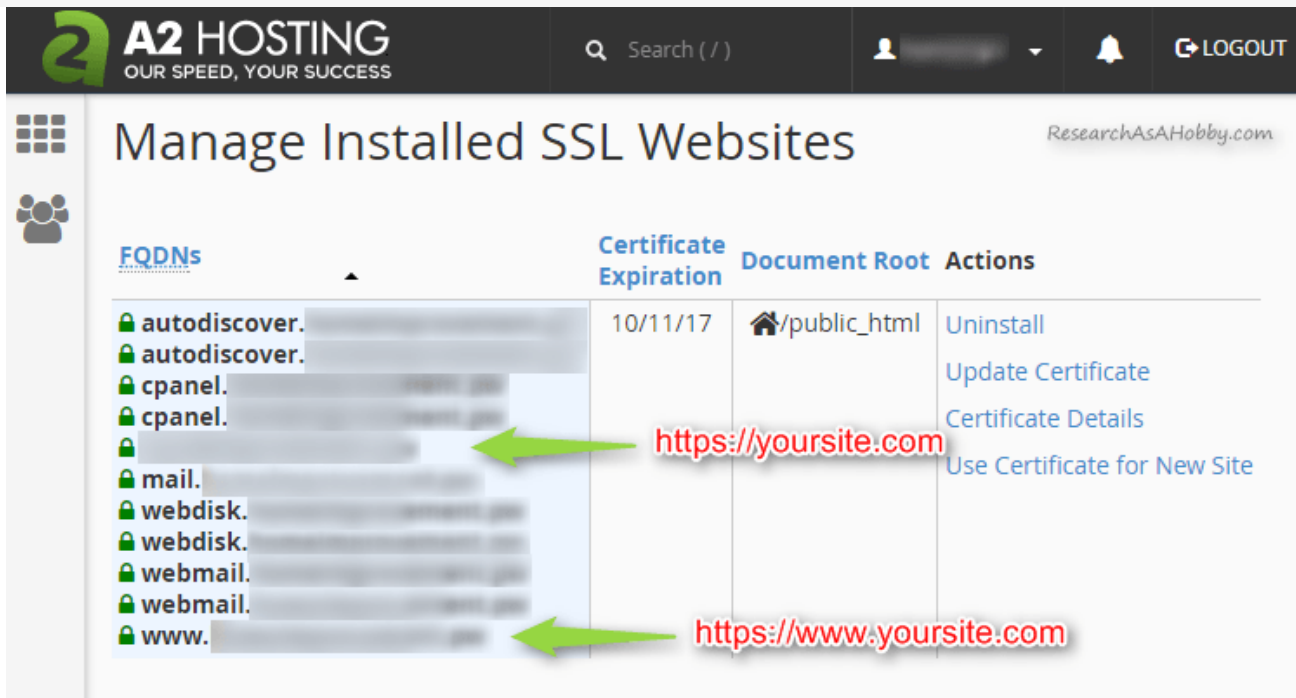
In you cPanel, select *SSL/TLS*:



Then click “Manage SSL sites” under “Install and Manage SSL for your site (HTTPS)”:



And then you will see your domains and subdomains on which SSL is installed. Make sure your domain names (e.g. *yoursite.com* and *www.yoursite.com*) are in the list:



That's it for this step! You have just made sure that SSL is installed for your website. So, hosting has already done its part of the work. As you see, you don't need to do anything on server side if you want to use a free SSL certificate that is provided by your hosting.

Also, you may see [this small article](#) and [this short comparison table](#) in which a free Let's Encrypt certificate is compared with paid SSL certificate types which are often used by website owners. The table shows briefly the most important limitations of a free SSL certificate.

If you want a paid SSL certificate

You may want to use a paid SSL certificate on your website for one or more reasons I've mentioned [above](#).

Where to buy SSLs and why?

Before all, you need to know that SSL certificates are issued by [Certificate Authorities](#). But the usual way you get a SSL certificate is buying it from a reseller. The reseller can be your hosting, domain registrar or some other 3d party.

The advantages of paid SSL certificates are the following:

- More functionality is available for some SSL types (works for multiple domains/subdomains).
- May have warranty (insurance for your visitor against loss of money when making a purchase on your site; read a short note [here](#)).
- Giving your site more trust in the eyes of your visitors (I've illustrated it in this section [above](#)).

Buying a SSL certificate from your hosting is a good idea as your hosting technical support will be more happy to help you in case you have questions.

If you are using [A2Hosting](#) (many of my readers use these large and well-established hosts), just go to the corresponding page and choose the SSL certificate you need:

- [SSL by A2Hosting](#).

You can also contact the hosting support if you have any questions.

Buying a SSL certificate from a third party sounds reasonable if your host does not offer paid SSLs. Also, you may be looking for a particular kind of SSL certificate or just looking for a more affordable one than what your hosting can offer.

For example, I use *Namecheap* domain registrar which has a wide variety of affordable SSL certificates. You can check out the offers [here](#). By the way, you don't need to have your domain registered with a 3d party SSL provider to buy a SSL certificate from it.

By the way, before buying SSL certificate from a 3d party make sure that your hosting allows it. Some hosts allow only SSL certificates if you buy them from those hosts only. Also get to know how you can install a third party SSL on your hosting. Contact your host if you hesitate.

By the way, **if you use a CDN/Firewal/Proxy**, then you may want to contact the provider and make sure you don't need to upgrade. The matter is that it's quite common that the cheapest CDN/Firewall/Proxy plans include only free SSL certificate. And if you want to take advantage of your paid SSL certificate, you may need a more advanced CDN/Firewall/proxy plan.

Installing SSL certificate which you bought from the 3d party is not difficult. Here are the instructions:

- [from A2Hosting](#),
- and [this one from Namecheap](#) if you use some other host with SSL/TLS Manager (also, SSL/TLS Wizard) available in your cPanel.

In case you don't have an option to setup SSL in your shared hosting account, contact your host's technical support.

Check your SSL certificate

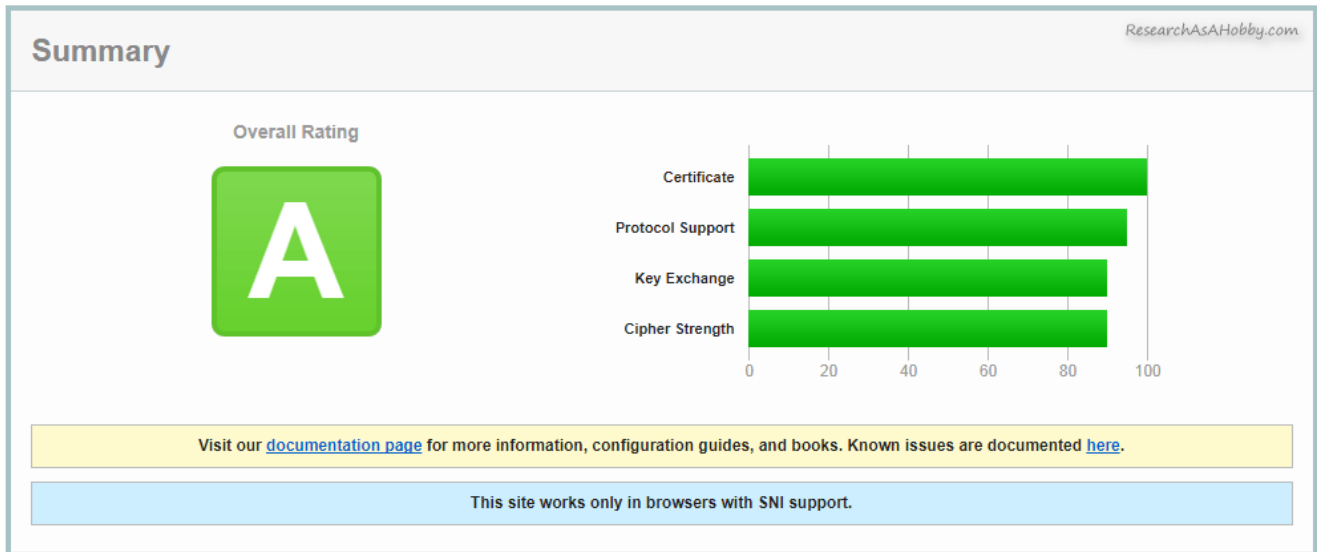
Once a SSL certificate is installed on your hosting account, you can check it.

A quick check is simply trying to open your website with HTTPS. Make sure your site loads. Perhaps, the layout may be broken, but anyway, in your browser you should see HTTPS (e.g. <https://yourwebsite.com>).

If your website redirects from HTTPS to HTTP, then you may want to temporarily disable this redirection (if you know how to do it) and test it again. It's likely that you will need to edit your `.htaccess` file. Contact your hosting support in case of doubts.

A deep SSL checking using [SSL testing tool from SSL Labs](#) let you see the technical details of your SSL certificate. Just go to that tool and enter your domain name. You will see a summary and

also a lot of technical details. What you want is simply seeing *A* or *A+* result in the summary. It's enough to say that your SSL is fine and you are ready to move on.



SSL test results summary (Let's Encrypt was tested)

However, if the test fails and you see an error or the summary shows something different from *A* or *A+*, then something is not good. You may want to contact your hosting for support in this case.

By the way, **if you use a CDN, a cloud firewall or a proxy**, see [below](#).

I hope your SSL certificate test has been successful. Now let's move on!

4. Update URLs (change HTTP to HTTPS)

4.1. General information

This is the technically most difficult part. However, in many cases it can be done in a few clicks.

Updating URLs from HTTP to HTTPS in a WordPress website basically consists of the following parts:

1. Updating WordPress settings.
2. Updating inner links in user content (i.e. links in your posts, pages to other posts, pages, inserted images, embedded third party content). Technically, these links are stored in the WP database.
3. Updating links to your website resources (such as CSS and Javascript files, image files etc used in your theme and plugins). Technically, these links are stored in your website's files on your hosting.

You don't need to update external links to make them HTTPS that go to other websites.

By the way, updating links from HTTP to HTTPS to the resources (images, files etc) on your HTTPS pages is called fixing '**mixed content**' issues.

For example, if you have a HTTPS page and inserted image which is linked as HTTP, this will be determined by Google as a mixed content. And you can see it the way I explain [below](#). This needs to be fixed. The embedded content also should be linked as HTTPS.

Otherwise, the page is not considered safe. It means no green padlock in the URL bar of your visitor's browser, and non-secure note there instead.

The whole process of updating links from HTTP to HTTPS in both your content and website system resources requires several steps. And in some cases it may seem a bit technical.

However, there are **simple ways** which can work in many cases.

- If you think you can't handle the technical part of moving your site to HTTPS, there's another way that may help you. There's [Really Simple SSL plugin](#). It's a free plugin which can make the work done. And its paid version is even better. I will explain the difference between free and paid version below.

Note that this plugins can work in many ways, but not always. In some cases you will need to fix mixed content manually ([see below how](#)) which have not been fixed automatically by the plugin.

4.2. Options you have

Here are the options that I suggest you can do to update your inner URLs to HTTPS. You need to choose one:

If you want to try avoiding technical things as much as possible:

- **A.** Try a plugin offered by your host to simplify the process (if your hosts offers it). Your hosting can assist you to some extent. In some cases you may need to fix 'mixed content' issues yourself (see [this section](#)).
- **B.** Use [Really Simple SSL pro](#) plugin, paid version. The paid version gives you a human technical support and the tool that simplifies finding mixed content issues (see more on this plugin feature [here](#)). In some cases you may need to fix 'mixed content' issues yourself (see [this section](#)). *Really Simple SSL pro* plugin just makes it easier by automatically crawling your site and finding the pages with mixed content which have not been fixed automatically. Also, it has some other nice features.
- **C.** Try using [Really Simple SSL](#) plugin, free version. This may work well on websites with HTTPS-ready WordPress themes, plugins and content. If the plugins does not work out of the box for all mixed content issues, you will need to either use option *B*, *D* or *E*.

If you want to have everything under control and do manually the job that the plugins in the above options do, then consider option *D*:

- **D.** Do all the work migrating your site to HTTPS manually without using any plugin. This approach of moving your site to HTTPS is the most professional. And although it's the most technically complicated compared to other options, it's a 'cleaner' way to go. And it's free. And if you are not afraid of doing some technical stuff such as running SQL update queries in your MySQL database on your WordPress website. And you may still be required to fix [mixed content issues](#). I'd recommend going this way.

If the above options do not make you happy or do not work out for you, then you can hire a technical person who can do the work for you:

- **E.** Find a freelancer who will move your website to HTTPS. However, don't forget, that you also need to make changes in Google Webmaster console and Google Analytics if you use it (read more [below](#)).

Besides, there's another option actually left:

- **F.** Create your website from scratch installing it on HTTPS. I include this option just in case so that you know about it. I guess this option makes sense if only you were going to re-do your website anyway. Obviously, in general it's much easier to migrate your website to HTTPS rather than re-do a website. Just be aware that if you re-do your site from scratch you still need to use properly coded theme and plugins which will not cause ['mixed content' issues](#).

After all, the whole matter of moving your website to HTTPS may sound to you quite technically complicated. But in fact from a practical point of view, it's not that difficult. You just need to understand what you are doing and in most cases it's pretty simple.

In fact, easy option *A*, *B* or *C* work in many cases perfectly. And you need just a couple of minutes to make it work. These are the most favorite options for most non-technical users.

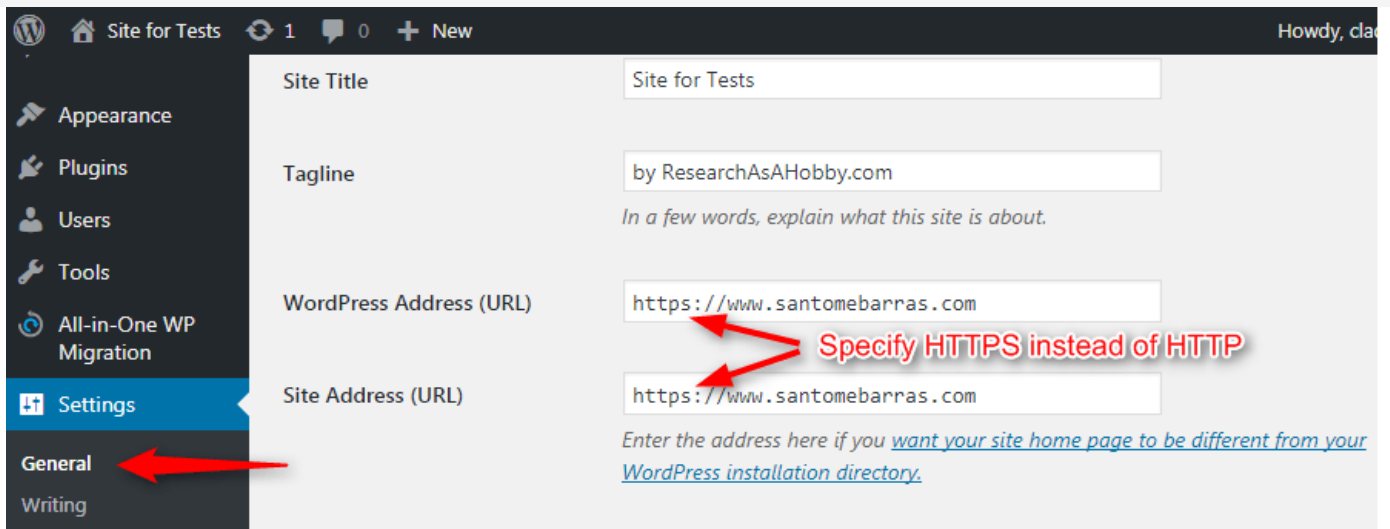
However, there are disadvantages of the three methods (A, B, C). They are the following:

1. These options do not solve 'mixed content' issues if your website or the plugins you use are not HTTPS-ready. In this case you need to do manual actions. It means that you need to change some insecure HTTP links to HTTPS manually in the code. I explain it in more details in option *D*. [Really Simple SSL pro](#) plugin simplifies identification of such issues.
2. These three options require installing and constant using a plugin which is a little but still a drawback. Although it does not make your site slower, it is additional piece of code. And your HTTPS website functioning depends on it.

So, if you are a perfectionist or technically pedantic, then you will want to avoid using one more plugin which you are able to do without. This will require more technical efforts of replacing HTTP with HTTPS inside your website. But it's not really difficult. See how to do it (the option *D*) in the next chapter.

4.3. Set HTTPS in the WordPress settings (Site URL and Home URL)

Go to your *WordPress dashboard / Settings / General*. And in in the fields "WordPress Address (URL)" and "Site Address (URL)" set HTTPS instead of HTTP for your website URL. See an example on the image below:



Set HTTPS instead of HTTP in your website URL

The same effect takes place if you change *siteurl* and *home* values in the *wp_options* table. (No need to do it, the image below is just for informational purposes):

option_id	option_name	option_value	autoload
1	siteurl	https://www.santomebarras.com	yes
2	home	https://www.santomebarras.com	yes
3	blogname	Site for Tests	yes
4	blogdescription	by ResearchAsAHobby.com	yes
5	users_can_register	0	yes

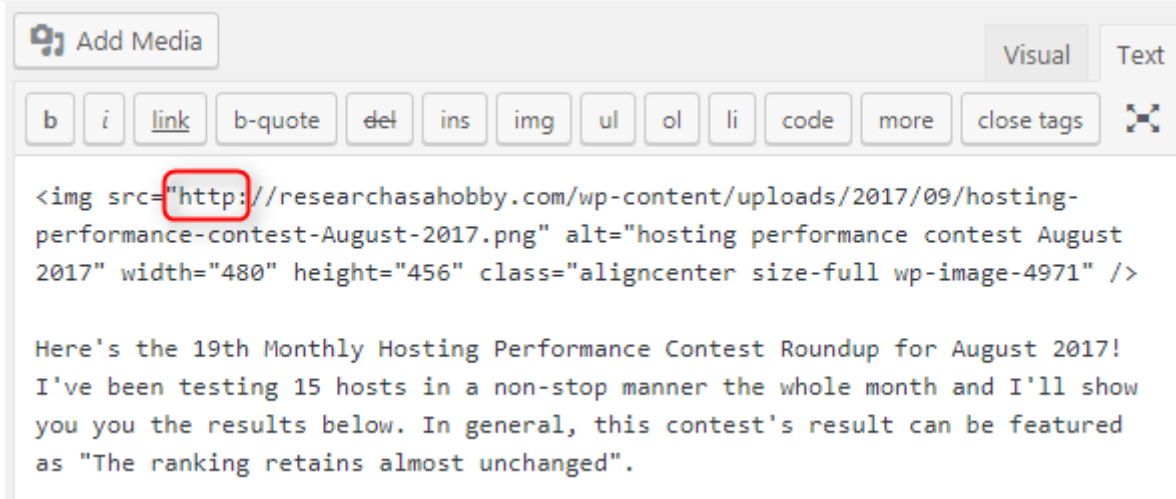
Website URL settings in the WordPress Options table

That's it with this step. Very easy!

4.4. Updating inner links to HTTPS in your website content

Here's what it basically means. There are links from your pages and posts to other posts and pages on your website. You inserted these links when you wrote your articles, pages etc on your website. And these links are HTTP.

And you need to turn these links into HTTPS. You can go to each page and edit the HTML text of your articles and other places manually to do that.



HTTP links in user content needs to be changed to HTTPS

Manually changing *http://yourwebsite.com* to *https://yourwebsite.com* can probably makes sense if you have just a small number of pages. But in case of tens and hundreds of pages or more you may want to have an automatic way to do it.

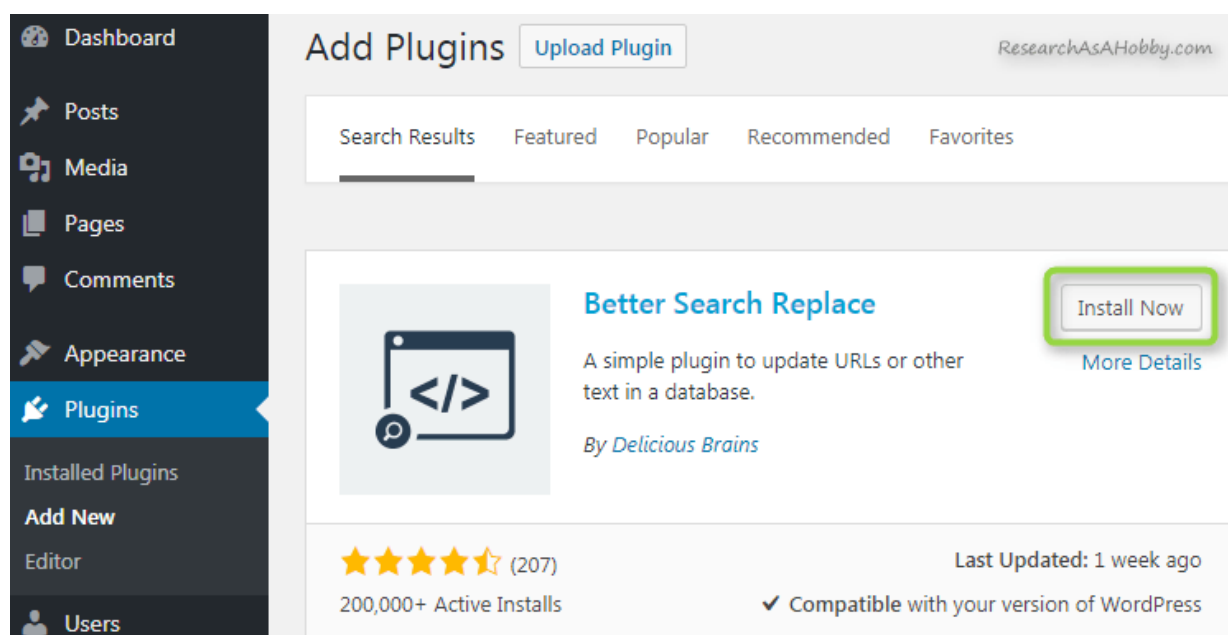
Moreover, if you use a page building plugin, forexample: (e.g. [ThriveArchitect](#), [Elementor](#), [BeaverBuidler](#) etc) manual editing of HTML code can be not very convenient. And automatic changing HTTP to HTTPS in links saves your time and efforts.

So, here comes a free [Better Search and Replace](#) plugin for WordPress. By the way, there are also other similar plugins. But I will use this one for my tutorial as it works fast and does what we need.

After you replace HTTP with HTTPS for your website's URLs which are mentioned in your website content's HTML code, you can remove the plugin.

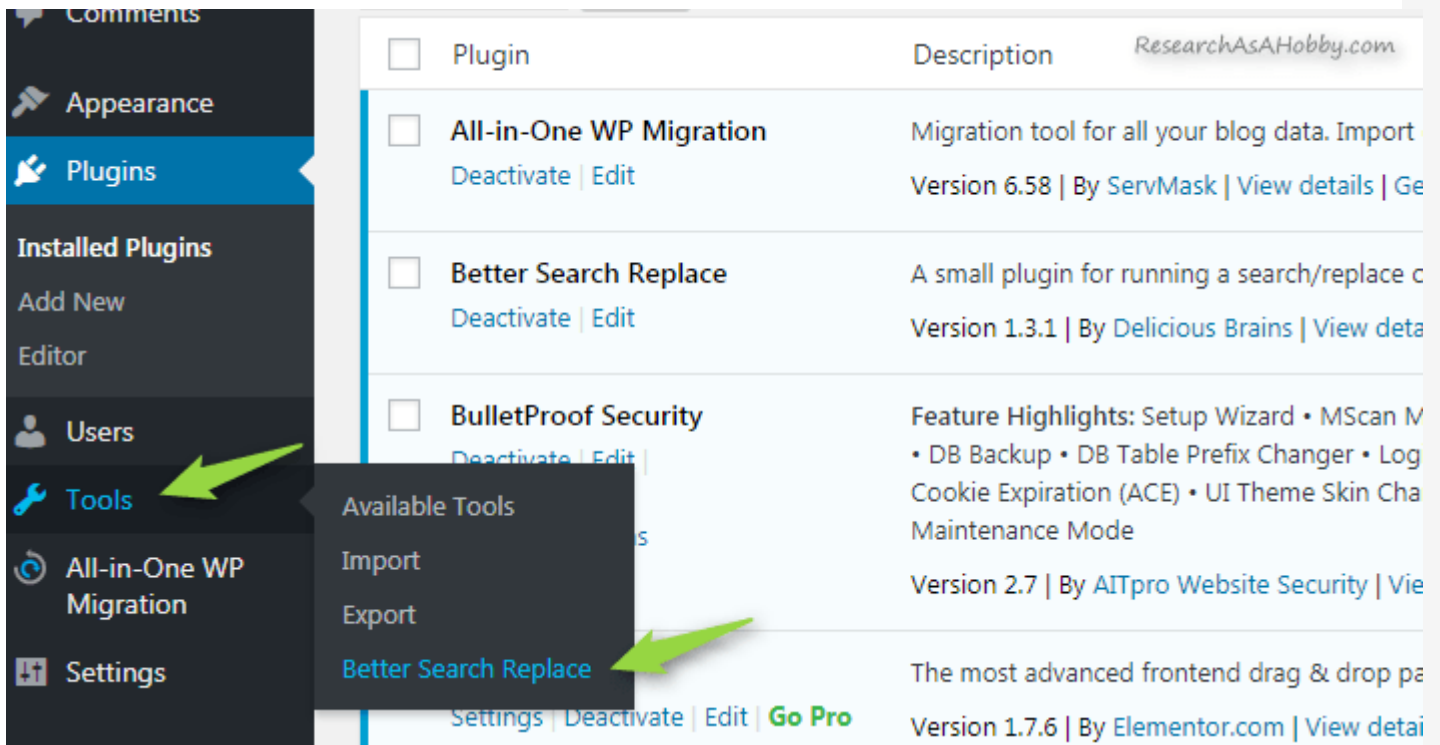
So, let's start.

Install and activate the plugin. Nothing special:



Installing the plugin to replace HTTP with HTTPS in your website content

Then go to the plugin settings. They are located at your *WordPress dashboard / Tools / Better Search Replace*:



Better Search Replace settings location

Now setup the plugin and run replacing the HTTP version of your website URLs with the HTTPS version as shown on the image below:

Better Search Replace

Search/Replace Settings Help

This tool allows you to search and replace text in your database (supports serialized arrays and objects).
To get started, use the form below to enter the text to be replaced and select the tables to update.
WARNING: Make sure you backup your database before using this plugin!

Search for: **1 Specify your website URL (http version)**

Replace with: **2 Specify the desired (https) version of your website URL**

Select tables: wppx_bpspro_db_backup (0 MB) wppx_bpspro_login_security (0 MB) wppx_bpspro_mscan (0 MB) wppx_bpspro_seclog_ignore (0 MB) wppx_cjtoolbox_authors (0 MB) wppx_cjtoolbox_backups (0 MB) wppx_cjtoolbox_block_files (0 MB) wppx_cjtoolbox_block_pins (0 MB) wppx_cjtoolbox_block_templates (0 MB) **3 Select all tables**

Select multiple tables with Ctrl-Click for Windows or Cmd-Click for Mac.

Case-Insensitive? Searches are case-sensitive by default.

Replace GUIDs? If left unchecked, all database columns titled 'guid' will be skipped.

Run as dry run? **4 Uncheck it to let the changes take effect**
If checked, no changes will be made to the database, allowing you to check the results beforehand.

5 Click the button to replace

Better Search Replace settings to replace HTTP with HTTPS URLs in your website content

By the way, the settings on the “Settings” tab can be left by default.

Note: if you get a error during the execution, then it’s probably too much data to process. Try executing the update in several steps, e.g. process bigger tables separately or/and process tables in batches. Decreasing page size setting (on the “Settings” tab) also helps.

In many cases using this neat plugin is enough for replacing links in your website **database**. However this plugin (*Better Search Replace*) does not affect serialized data in the database. It means that some HTTP links still may stay unchanged. If this happens in your case, you will encounter it in the [further checking](#) and you can fix it as I show [below](#) using *Search&Replace* plugin.

That’s it for this step. You have replaced your HTTP version of your website URLs with the HTTPS version in your website content.

4.5. Make sure your website pages do not have ‘mixed content’ issue

On the previous step you have changed the URLs to HTTPS which were technically stored in the database. But some URLs of your website which are stored **in the files** may still be HTTP.

Also, there can be some other HTTP links which are stored in the serialized form (sort of compressed) in the database. You need to change these links to HTTPS too by editing your website content.

Insecure (HTTP) links to resources which are used on your site cause so-called **'mixed content'** issue. Mixed content issue takes place if a HTTPS (i.e. secure) page of your website contains HTTP (non-secure) elements. Such elements can be:

- Web fonts
- JavaScript includes
- CSS style sheets
- Image embeds
- Video embeds
- Audio embeds
- Iframe content

You need to find such places and fix them to make secure (i.e. using HTTPS links instead of HTTP links).

How you can fix mixed content?

Once you find the page with mixed content, you need to find out what file or link is causing the issue. And then you decide how to fix it:

Sometimes it's just enough to change the HTTP link to HTTPS (e.g. replace *http://expternalsite.com/some_file.css* with *https://expternalsite.com/some_file.css* if this file is available via HTTPS). Some of them you can fix manually by editing your content. Some of them you can fix using database editing plugins which search and replace text in the database. I explain how to do it [below](#).

- Sometimes you may realize that you just need to delete the outdated plugin that you don't really need. Moreover if the plugin contains potentially insecure content.
- In some cases it makes sense to contact a developer of that plugin and request the update.
- In some cases you may want to download an external resource (e.g. a file with fonts) which is available only via HTTP to your website. Thus you will be able to link to the resource locally using HTTPS.
- In some cases you can find an alternative to the insecure content (e.g. find a plugin with similar functionality but which does not cause mixed content issues).

So, whatever you decide to do, you need to replace the external links to insecure (HTTP) resources with secure (HTTPS) links.

How you can find the pages on your site with mixed content?

There's a number of ways how you can check that each and every page of your website is secure (i.e. with no insecure elements in it).

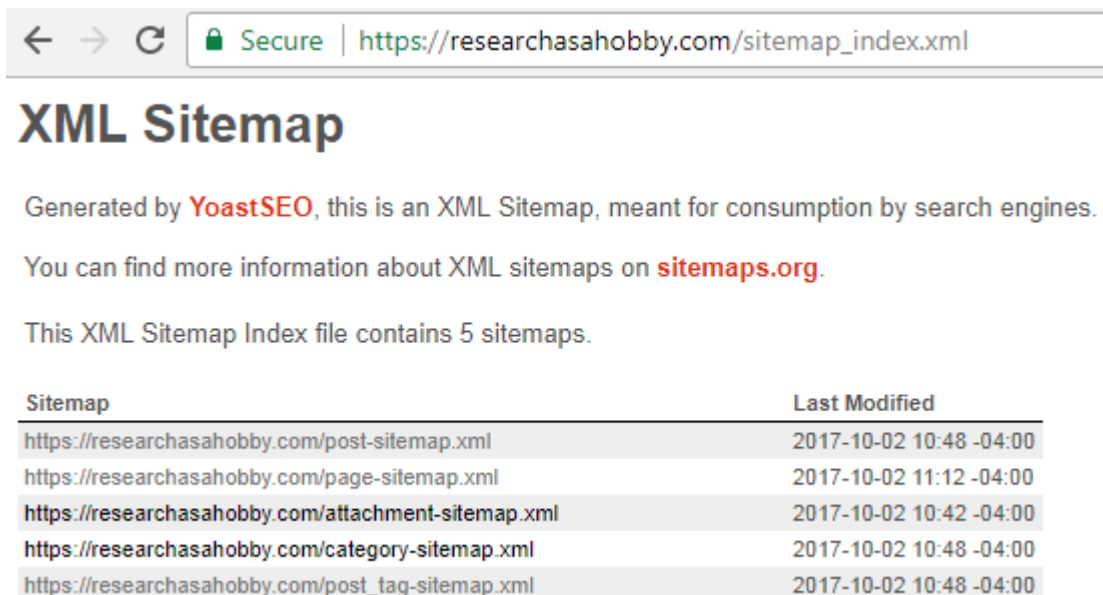
One of the most efficient ways is to use [Really Simple SSL pro](#) plugin functionality. [Here's the article](#) about it.

And as a totally free alternative, I'll show you right now **how to check each of your page manually**. All you need is just a Google Chrome browser.

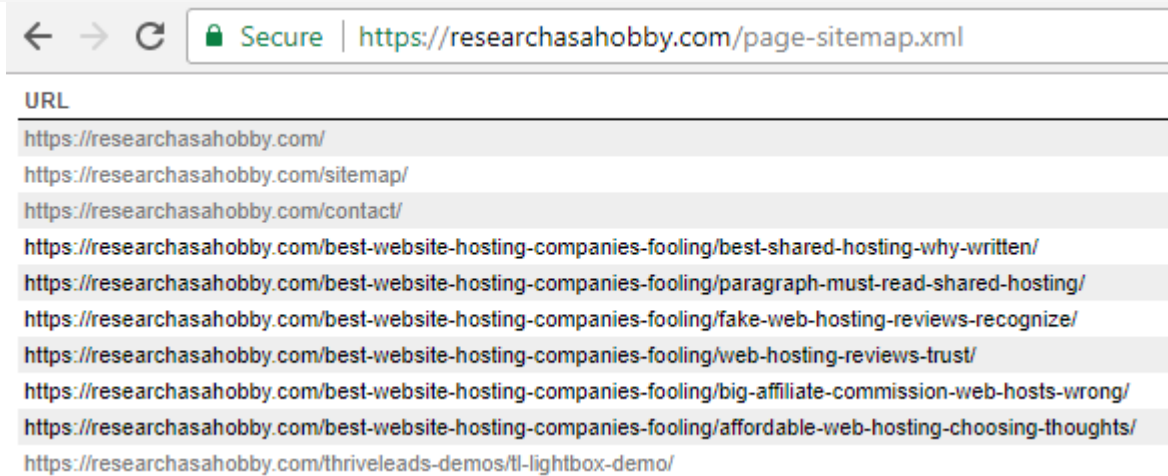
You need to know the URLs of all your website pages that you want to be secure (HTTPS with no mixed content) in the Google's eyes. It means such pages should have a green padlock and no non-secure mark in the URL field of your visitor's Google Chrome browser.

One of the easiest ways to get the list of these URLs is looking at your sitemap. The sitemap contains the list of all your website posts, pages, categories, tags, attachment pages etc. Very often the sitemap is located at *yourwebsite.com/sitemap.xml* address. If you don't know where your sitemap is, just open this address in your browser. If it works, you are good. If it does not, make sure you have sitemap specified for your website.

For example, here's the screenshot of my sitemap (I use [Yoast SEO](#) plugin that generates the sitemap for my website):



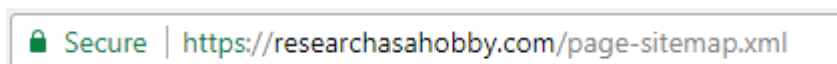
It contains the links to other sub-sitemaps which contain the pages, the posts and so on. For example, this is a fragment of the sitemap of my pages:



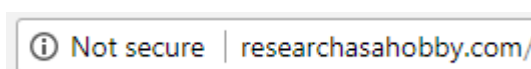
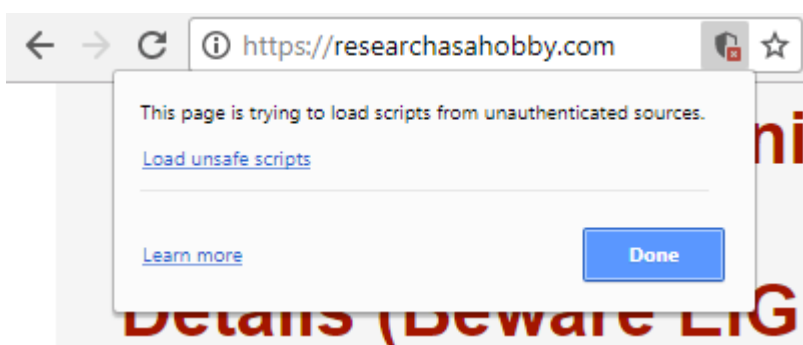
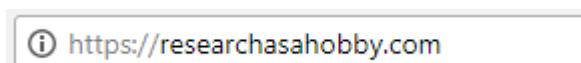
Note that the URLs are HTTPS, not HTTP. This is so because the sitemap automatically regenerates and includes your HTTPS pages as soon as you do previous steps.

Now, you can put together the list with your website posts, pages etc (let's call them pages in short).

And you can **check each page if there's a mixed content issue** on them. To do that just open each page via HTTPS one by one in Google Chrome and look at the browser URL bar. If it displays a green padlock, then there's no issues on that page:



And if there's no green padlock on the HTTPS page, and there's a non-secure notice or attention information icon like on the images below, then the page is not considered safe by Google. It means that there's a mixed content issue on that page:



By the way, on some HTTPS pages with mixed content you may notice a broken layout, wrong fonts, missing elements, broken functionality etc.

So, on this step you have found a problematic HTTPS page with mixed content.

Now you need to **locate the mixed content on that page**. Read the next section to know how to do it.

4.6. Fixing 'mixed content' issues on your HTTPS pages

4.6.1. General overview of the process and video walk-through

Here's a short overview of the process that I explain below in this pretty long section. Come back here once you are lost eventually during the process of fixing the mixed content issues.

Fixing mixed content issues on a particular website page contains **the following steps and considerations**:

1. You identify the HTTP link that causes mixed content issue on your website page(s).
2. In some cases you can easily find out where this HTTP link is located and what you need to edit in your WordPress website to fix it. If it's your case, fix it (replace it with the new HTTPS link) and you are done. Otherwise, make further steps.
3. This problematic HTTP link can technically be located only in two places: either in the website files OR in the database.
4. You download your website files and scan them searching for the HTTP link. If you find the HTTP link, you replace it with the new HTTPS link. Done. Otherwise make further steps.
5. You use [Better Search Replace](#) plugin to replace HTTP link with the HTTPS link in the database. (Note: the content of your website is stored in the database). If mixed content issue is resolved, you are done. If not, make further steps.
6. You use [Search & Replace](#) plugin to find in what database table or tables the problematic HTTP link is located. And then you use the plugin to update the database table(s) to replace the HTTP link with the HTTPS link (do it with caution, see [below](#)). Or better, use phpMyAdmin to update the table values. Now, if you have done everything correctly, the mixed content should be fixed.

You can repeat the process as many times as required to fix all mixed content issues.

After all, the point of the process is to replace the resources (images, external files) which are used insecurely (i.e. via HTTP) with the resources which are used securely (via HTTPS). Sometimes instead of doing the technically complicated procedure, you can simply re-load your resources (e.g. images) to your website to get rid of the mixed content issues which are caused by these resources (e.g. images).

Another point to keep in mind, if you use external resources (CSS files, images etc), you need to make sure that these resources are available via HTTPS. Otherwise they will be causing mixed content on your website. If you have no control over the external website, then find alternative external resources which are available via HTTPS, or move the resources to your website.

Also, mind your minification or/and [caching plugin or solutions](#). if you use any. When making changes to your website, be aware that your caching solution may cache your old version of site. That's why you may need to clear the cache in the caching solution after you change anything on your website. Or you can simply deactivate your caching plugin (and minification plugin) while you change your site.

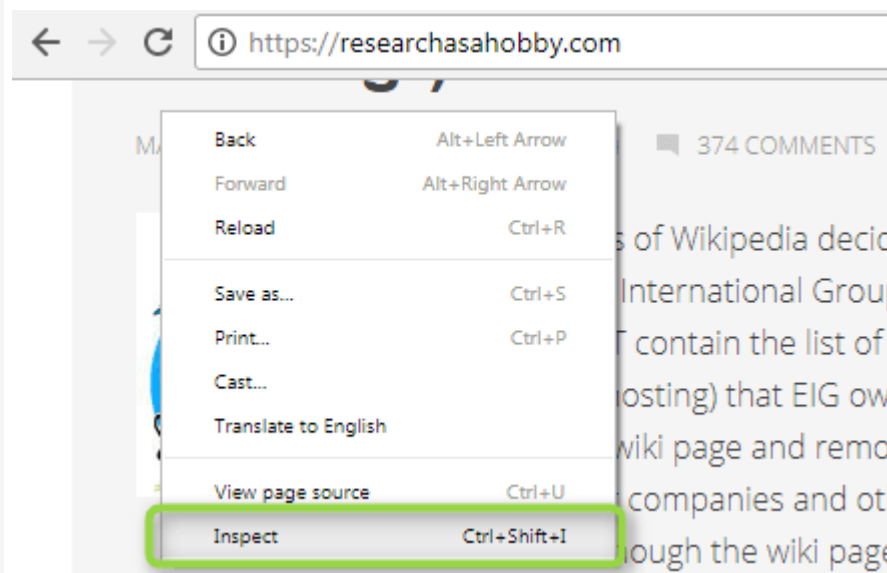
I explain this process with more details and examples right below.

And here is the video tutorial how I fix mixed content issues using different techniques when migrating website to HTTPS. You can find this video very useful in conjunction with reading the below chapters.

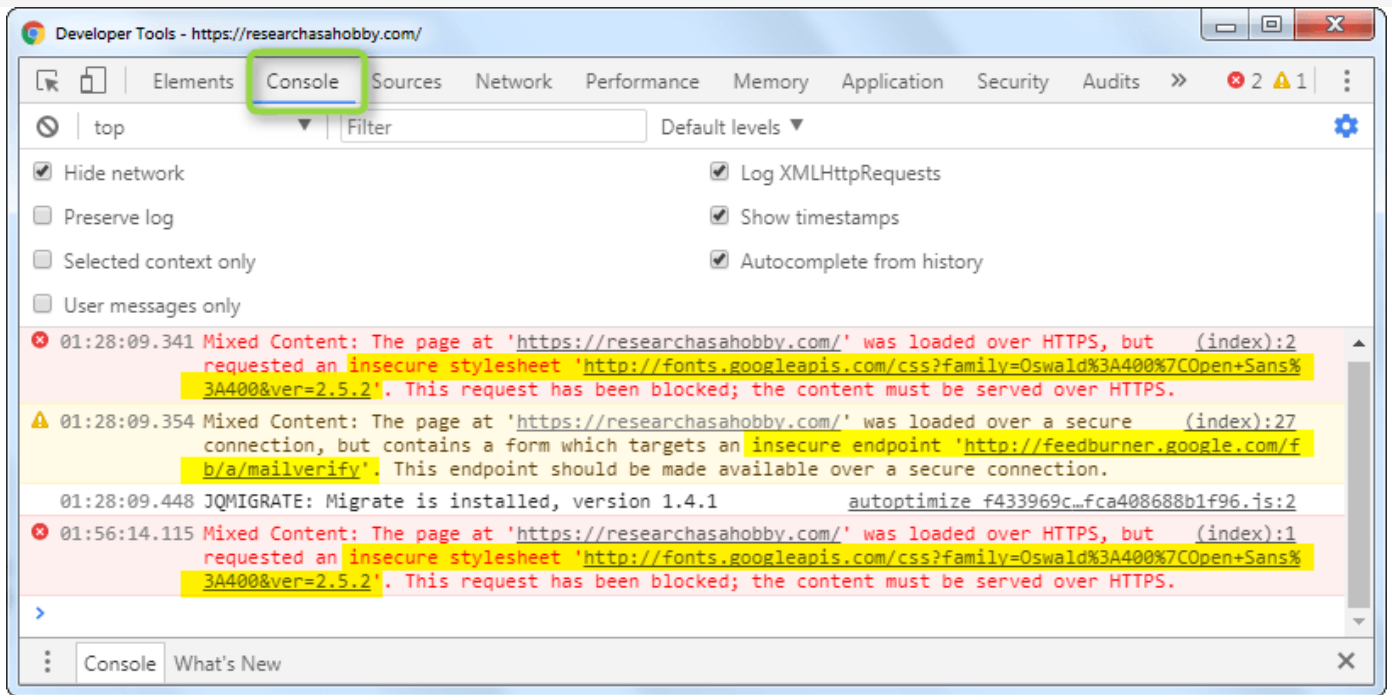
4.6.2. Using Google Developer Tools to find mixed content

Google Developer Tools / Console tab in Google Chrome is to help identify mixed content issues.

So, you have found a page on your website which contains mixed content. To locate the issues right-click with your mouse on the page and select *“Inspect”*:



Google Developer Tools panel/window will open. Go to *“Console”* tab. And you will see the list of your mixed content errors (I highlighted them in yellow on the image below):



Now you need to analyze the messages and try finding out what plugin or which element on your website cause each of the issues. Sometimes it's easy to do as you can clearly see what is causing the mixed content issue and what you need to fix. But sometimes you don't have a clue where to look at further anyway.

We have come to the point in this tutorial where exact steps are not enough because each website is an individual combination of theme, plugins, custom code and user content. And the mixed content errors can be quite specific in each case. That's why it's hard to say where exactly you need to look on your website to fix the mixed content issues.

However, I can give you a recommendation that will help you fix mixed content errors and make your HTTPS pages secure in most cases. Here it is right below.

The whole matter is that some links in your website code and your website content are HTTP where as they should be HTTPS. And you just need to replace these HTTP links with HTTPS. Sounds simple.

Also, keep in mind that your website is a combination of database and files. *And you have already replaced HTTP links with HTTPS in your database.*

So, there are big chances that the HTTP links which cause mixed content issues are still *somewhere in the files*. Fixing these links in the files is likely to resolve all or most of the mixed content errors.

4.6.3. Replacing insecure links in the files

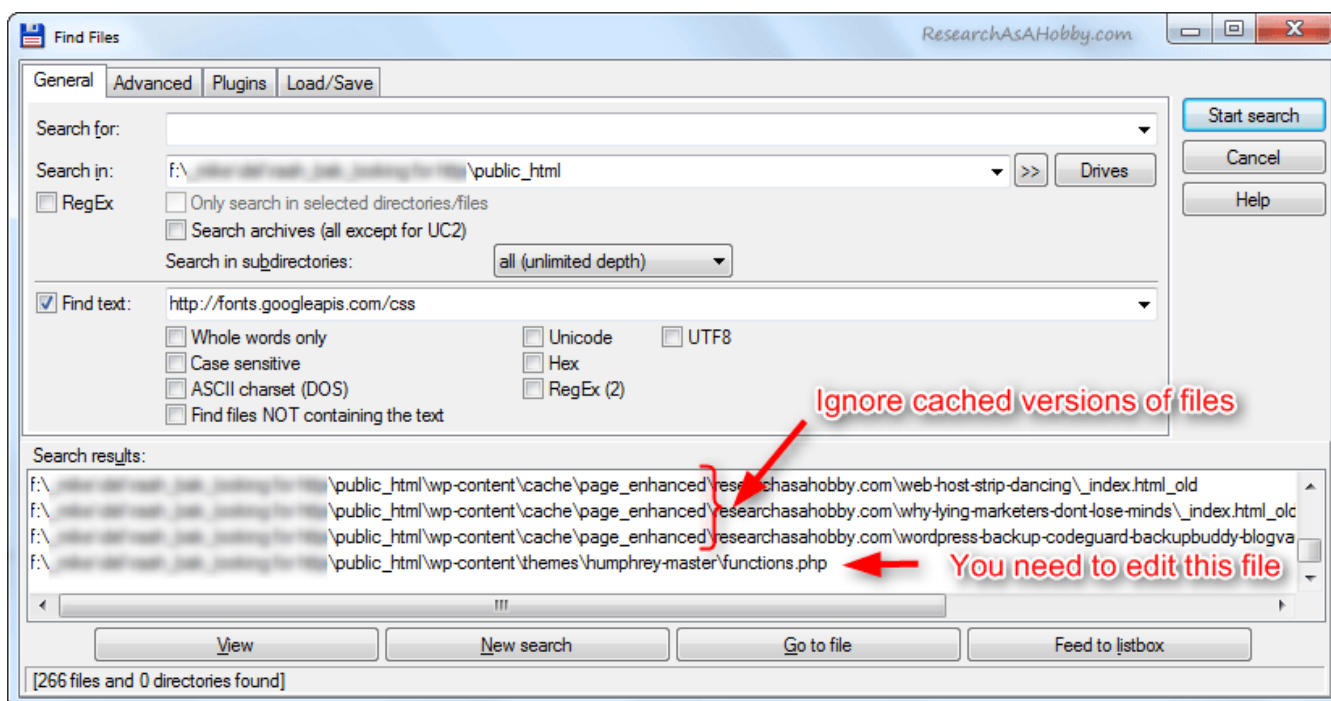
To replace HTTP links with HTTPS ones in the files, I suggest downloading your website files to your local drive. And then using your file management tools find the places in your website files where unsafe HTTP links are used.

For example, you want to fix the issue connected with an insecure style sheet file (see my example on the screenshot [above](#)). You know the name (or its part) of the style sheet file (in my case this is “<http://fonts.googleapis.com/css>”).

Before all, check that HTTPS version of the resource is available. In my example, I try opening “<https://fonts.googleapis.com/css?family=Oswald%3A400%7COpen+Sans%3A400&ver=2.5.2>” in my browser. If it opens fine, you can continue. If it does not exist, then you need to find an alternative to that resource or download it to your website to access locally via HTTPS.

So, we continue now. In my example I want to find out in what file or files the insecure resource (“<http://fonts.googleapis.com/css?family=Oswald%3A400%7COpen+Sans%3A400&ver=2.5.2>”) is used. Download to your computer your website files which are located in *public_html* folder (via FTP or using [partial backup](#)).

And then simply run your favorite file search tool that will find files by text. I use Total Commander’s file search functionality. Here’s my search result:



After you find the files, you can replace insecure HTTP resource with the secure HTTPS version of the resource. You need to use your common sense to determine which files should be edited and which should be not.

For example, it does not make sense to edit cached versions of files. Also, if you get updates of your theme (or other software like plugins), do not edit the files which belong to your theme/plugins. Otherwise with the next update of your theme/plugin your edits will be overwritten. *Contact your theme/plugin developer instead to make the theme/plugin HTTPS-ready.*

In my example, the file belongs to the theme that I support myself, so I can edit it.

```
Listner - [F:\wwwroot\wp-content\themes\humphrey-master\functions.php] ResearchAsAHobby.com 12 %
File Edit Options Encoding Help
// Load Google fonts
add_action( 'wp_enqueue_scripts', 'humphrey_load_google_font' );
function humphrey_load_google_font() {
    wp_enqueue_style( 'google-font', 'http://fonts.googleapis.com/css?family=Oswald:400|Open+Sans:400', a
}
```

After you edited the file, upload it back to your hosting.

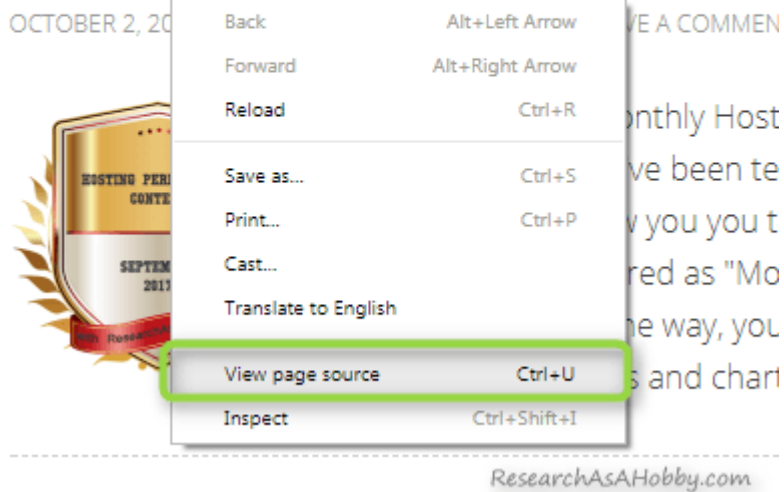
4.6.4. Viewing page source code to locate mixed content

Another trick that will help you locate a particular mixed content issue is looking at your page source code.

By looking at the surroundings of the mixed content code string in the page source code you can identify where the mixed content element is located.

To see the HTML source code of your website page, right click on the page and select “View page source”:

Hosting Performance Contest Roundup (15 Hosts Tested: Th



And in the opened window with the HTML code, find (Ctrl-F) the resource which causes mixed content.

For example, the other mixed issue that I found on my page is caused by this resource: “<http://feedburner.google.com/fb/a/mailverify>”. So, I open the source code of the page and find the problematic resource in the code. The code beside it tell me that I need to go to my sidebar widget in my WordPress dashboard to edit the resource link from HTTP to HTTPS:

This helps to understand that the content of the sidebar widget should be edited

```
>Next Page &#x000BBB</li></ul></div></div><div id="sidebar"
class="sidebar widget-area"><div id="text-4" class="widget
widget_text"><div class="widget-wrap"><h4 class="widget-title
widgettitle">Enter your E-mail:</h4><div class="textwidget"><form
style="padding:0px;text-align:left;"
action="http://feedburner.google.com/fb/a/mailverify" method="post"
target="popupwindow"
onsubmit="window.open('http://feedburner.google.com/fb/a/mailverify?
uri=ResearchAsaHobby', 'popupwindow',
'scrollbars=yes,width=550,height=520');return true"><p><p><input
```

The above techniques (Better Search Replace plugin, manual editing your files and WordPress content) are comparatively easy and safe to use. And if you are far from being a technical person, I'd recommend using them rather than using Search & Replace plugin that I talk about right below.

4.6.5. Replacing insecure links in the content (advanced)

Sometimes the HTTP link which causes mixed content issue is located not in your website files, but somewhere in the content. And you can't find this place in your website content easily.

And even using *Better Search Replace* plugin like I described [above](#) does not help to replace the link. This happens if the link is technically stored in the *serialized* data in the database. *Better Search Replace* plugin does not "see" such content.

In this case [Search & Replace](#) plugin can do a great job. It's free.

Here's how the plugin generally works. it can scan your database tables for some text (e.g. URL) and tell you in which tables and in what table fields this URL is found.. Also, the plugin can generate a SQL script and run it to replace text in the database tables.

However, here's **a word of warning**. Please note that this plugin may seem to work very long on shared hosting (or even sort of endlessly) if you use it for *importing* the whole database or large tables (more details on it below). That's why you better to use it only for updating not all, but only selected database tables which are not large (less than a pair of megabytes is fine on a shared hosting).

And that's why I recommended trying to use first *Better Search Replace* plugin instead which works fast and may fix all your mixed content issues located in the database.

Before showing you how to use *Search & Replace* plugin, let me tell you a bit technical note to explain you **a good trick**. As I've mentioned, this plugin can scan your database for a specific text (HTTP link in your case). And then it can create a script for the new database that you can import to replace your existing database. This way this plugin helps you replace an old link with a new

link. But this approach (replacing a whole database table) is too heavy in case of large database tables.

That's why I recommend using this plugin not for importing large database tables, but for *scanning your database* (even a large one) for the HTTP link you need to replace. This way you can find which database tables you need to replace. Then you can tell the plugin to create a script not for the whole database, but just for the selected tables. Although the tables should not be very large (a script of tens of megabytes is likely to fail on a shared hosting).

By the way, instead of running SQL script in Search & Replace plugin, use *phpMyAdmin* to edit the database table values instead. I show how to do it in my [video](#).

This trick helps you to use the power of *Search & Replace* plugin without risks to fail database import execution.

Also, sometimes after just scanning the tables with *Search & Replace* plugin, it gets clear where you should go in your WordPress and change the content the usual way.

Also sometimes, especially if there are not many changes to do, it's even much safer and easier to change the values in the database table(s) using phpMyAdmin, than using import functionality in *Search & Replace* plugin.

One more warning is avoid replacing values in [GUID](#) fields. One reason is that it has nothing to do with mixed content issues. Another reason is that it can make your script too large. Besides, it can [affect](#) your RSS feed.

And of course, **don't forget to [make database backups](#)**, especially before running the scripts generated by *Search & Replace* plugin. If anything goes wrong, you can [restore the database backup](#) easily. It will let you start over again right from the failure point.

When editing or updating database values, it's most likely that you will need to update just *Content* fields.

After all, *Search & Replace* plugin allows you to see what you are going to change in your database. And you need to review the changes before you run the script if you decide to do so.

Alright, the theoretical part about *Search & Replace* plugin is over. Now let's get to the practice.

4.6.6. Using Search & Replace plugin

After you read the previous section, you can go on.


Install and activate [Search & Replace](#) plugin:

ResearchAsAHobby.com

Search Results Featured Popular Recommended Favorites

Keyword search replace

1,170 items 1 of 39



Better Search Replace

Active

More Details

A simple plugin to update URLs or other text in a database.


By Delicious Brains

★★★★☆ (207)

200,000+ Active Installs

Last Updated: 2 weeks ago

✓ Compatible with your version of WordPress



Search & Replace

Install Now

More Details

Search & Replace data in your database with WordPress admin, replace domains/URLs of your WordPress installation.

By Inpsyde GmbH

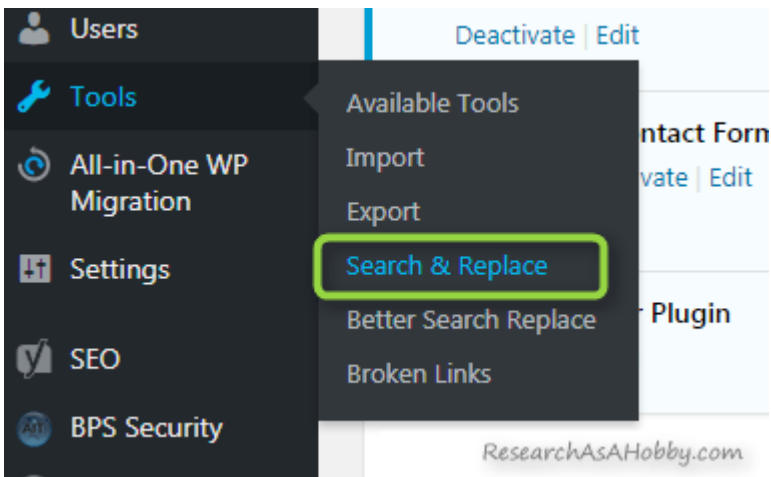
★★★★☆ (167)

100,000+ Active Installs

Last Updated: 9 months ago

Untested with your version of WordPress

Go to the plugin settings in your WordPress dashboard / Tools / Search & Replace:



Users Deactivate | Edit

Tools Available Tools Contact Form

All-in-One WP Migration Import vate | Edit

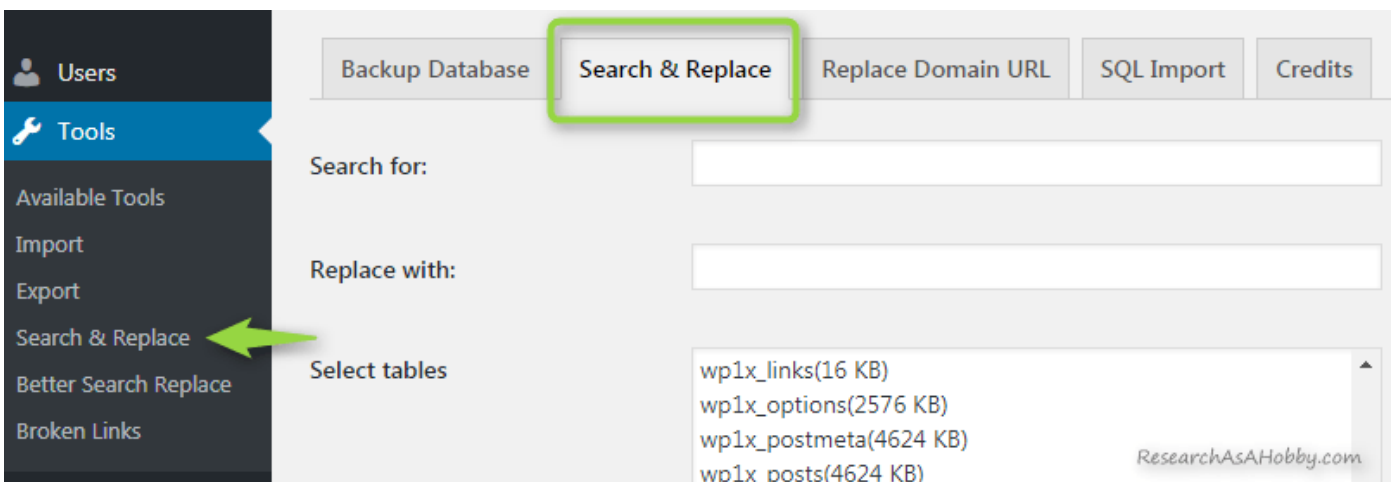
Settings Export Search & Replace

SEO Better Search Replace Plugin

BPS Security Broken Links

ResearchAsAHobby.com

On the plugin setting's page go to "Search & Replace" tab:



Users Backup Database Search & Replace Replace Domain URL SQL Import Credits

Tools

Available Tools

Import

Export

Search & Replace

Better Search Replace

Broken Links

Search for:

Replace with:

Select tables

wp1x_links(16 KB)

wp1x_options(2576 KB)

wp1x_postmeta(4624 KB)

wp1x_posts(4624 KB)

ResearchAsAHobby.com

So, you have a problematic HTTP link that causes mixed content issue. You need it to be replaced with the secure HTTPS link.

In my example, here's the HTTP link that I want to replace with the HTTPS link: "<http://feedburner.google.com/fb/a/mailverify>" (see the screenshot with mixed content report [above](#)). This HTTP link is connected with my email subscription form that is hidden somewhere in the content on my website (probably in my [ThriveLeads](#) opt-in forms). I could not quickly find all the places where this URL is used on my website, so using *Search & Replace* plugin will help me in fixing it.

Now find out in which database tables your problematic HTTP link is stored.

To do that do the following:

1. Input your problematic HTTP link in the field "*Search for*".
2. In the field "*Replace with*" you can actually enter whatever you want (it does not affect anything on this step). But you can input the HTTPS link there.
3. Click "*Select all tables*" (you want to search in all database tables).
4. Leave "*Dry Run*" option checked (it's a safe option which helps to see the required database changes in a safe mode which works fast).
5. And click "*Do Search & Replace*" button (it's safe to click it now, nothing will be changed in your database actually).

Here's the screenshot of these settings:

Backup Database Search & Replace Replace Domain URL SQL Import Credits

Input the problematic not-secure URL →

Search for:

Replace with: →

No matter what you input here on this step, but you can input the new secure URL

Select tables

- wp _bfc_filters(0 KB)
- wp _bfc_instances(827.79 KB)
- wp _bfc_links(3064.82 KB)
- wp _bfc_synch(43.3 KB)
- wp _bpspro_db_backup(0.41 KB)
- wp _bpspro_login_security(0.17 KB)
- wp _bpspro_mscan(0 KB)
- wp _bpspro_seclog_ignore(0 KB)
- wp _commentmeta(16 KB)
- wp _comments(4624 KB)
- wp _links(16 KB)
- wp _options(2576 KB)
- wp _postmeta(4624 KB)
- wp _posts(4624 KB)
- wp _subscribe_reloaded_subscribers(64 KB)
- wp _tcb_api_error_log(0 KB)
- wp _term_relationships(16 KB)
- wp _term_taxonomy(16 KB)
- wp _termmeta(0.05 KB)
- wp _terms(16 KB)

Select all tables

Dry Run will quickly find out the tables where your problematic URL is found

Dry Run

Export SQL file or write changes to DB?

Export SQL file with changes
 Save changes to Database

Use GZ compression

ResearchAsAHobby.com

Search & Replace plugin settings to find out where the insecure URL is found

When the plugins finishes working, click the “View details” link:

Dry run is selected. No changes were made to the database and no SQL file was written .

39 tables were processed. 2 cells need to be updated.

[View details](#)

Backup Database

Search & Replace

Replace Domain URL

SQL Import

Credits

Search for:

http://feedburner.google.com/fb/a/mailverify

ResearchAsAHobby.com

A window will appear with the database tables entries where the problematic URL was found. What you need is the database tables where the URL was found:

✕

Table [wp_tve_leads_form_variations](#) Changes: 2

row	2	column	content	Old value:	New value:
row	24	column	content	Old value:	New value:

ResearchAsAHobby.com

Note the database tables where the insecure URL was found

Remember (or put down somewhere) the found table names. On the next step you will re-create these tables where the insecure HTTP link was found. The new tables will contain the new HTTPS link.

So, in my case I need to re-create just one table.

Now get back to the plugin settings:

- Select only the tables which you have just found (the tables which contain the insecure HTTP link).
- Unselect "Dry Run" option (because you need to generate a script now).

- Set option “Export SQL file with changes” (with this option the plugin will generate the script).

The screenshot shows the 'Search & Replace' tab of a WordPress plugin. The 'Search for:' field contains 'http://feedburner.google.com/fb/a/mailverify' with a red arrow pointing to it and the text 'Input here the unsecure HTTP URL'. The 'Replace with:' field contains 'https://feedburner.google.com/fb/a/mailverify' with a red arrow pointing to it and the text 'Input here the secure HTTPS URL'. A list of tables is shown, with 'wp_tve_leads_form_variations . (423.11 KB)' selected and a red arrow pointing to it from the text 'Select here the tables that you found on the previous step'. Below the table list is a checkbox for 'Select all tables'. The 'Dry Run' checkbox is unchecked, with a green box around it and the text 'Uncheck this option'. The 'Export SQL file or write changes to DB?' section has two radio buttons: 'Export SQL file with changes' (selected) and 'Save changes to Database'. A green box is around the selected radio button with the text 'Select this option'. The 'Use GZ compression' checkbox is unchecked. At the bottom left is a blue button labeled 'Do Search & Replace'. At the bottom right is the text 'ResearchAsAHobby.com'.

Search & Replace plugin settings to generate SQL script

“Use GZ compression” option helps if you get a large script.

Then click “Search & Replace” button. And the plugin will generate SQL script that will replace database tables with the new ones containing the new HTTPS link.

Since you migrate your website from HTTP to HTTPS, you need to force HTTPS. It means that all visitors of your website including search engine bots should consider that your new version of your website is available from now on via HTTPS, not HTTP. So, anyone trying to access the HTTP version of your website will be redirected to the secure HTTPS version.

This also helps to avoid duplicate content issues and is very good for your SEO.

You can easily do it by adding the following lines of code to your `.htaccess` file:

```
1# Redirecting from HTTP to HTTPS
2RewriteEngine On
3RewriteCond %{HTTPS} off
4RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
```

Technical note. By the way, other examples of the code blocks you can find [here](#). And if case of redirect issues, [this](#) might help (e.g. redirect to HTTPS should go [before](#) rules for WordPress).

You can place this code in the beginning of your `.htaccess` file (above the rules for WordPress). By the way, it's advised to put the code inside the block "... " (see an example on the screenshot [below](#)).

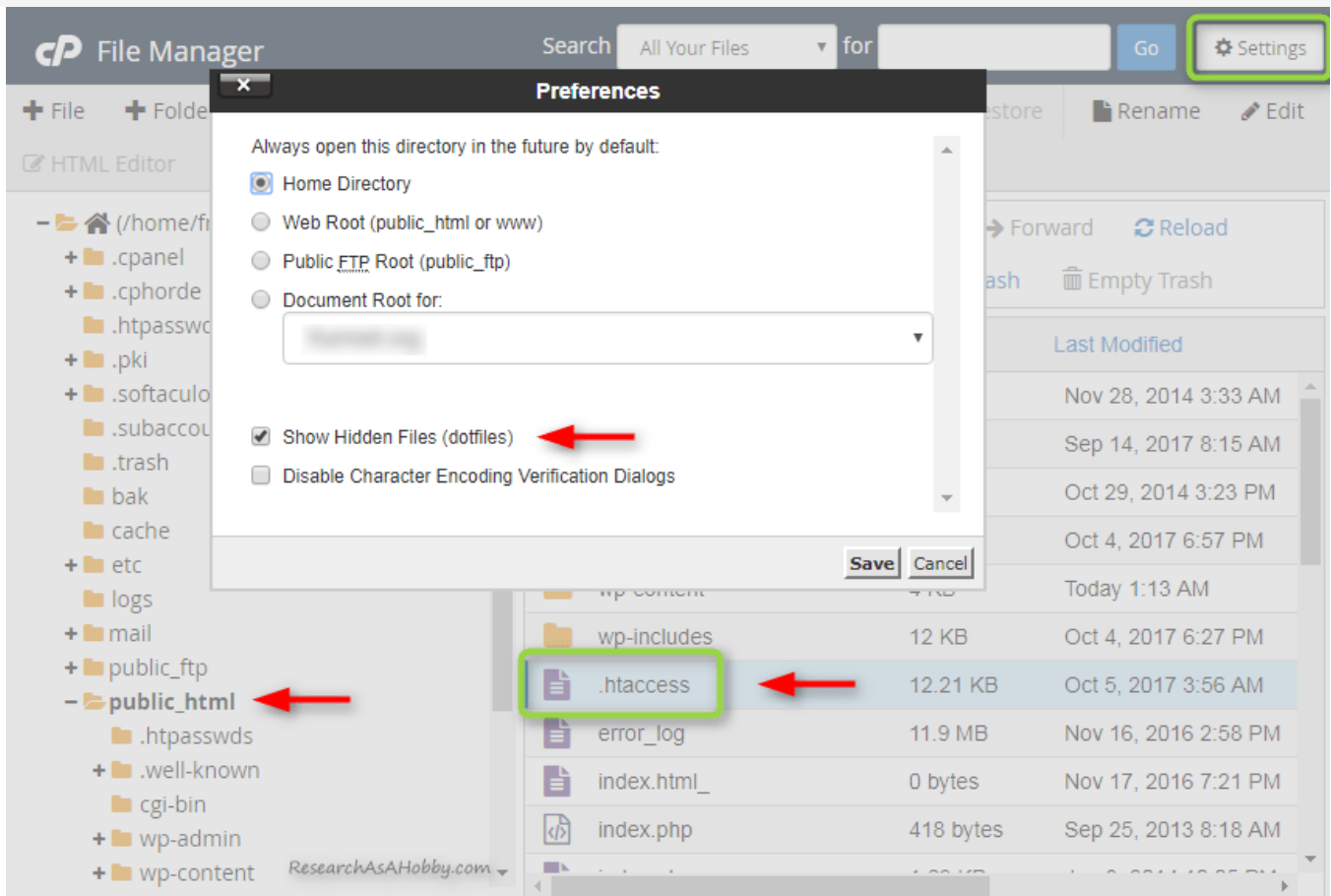
Also, it's advised to have just one line "`RewriteEngine On`" which goes above the `rewrite` rules.

The `.htaccess` file is located in the root directory in the files on your hosting.

A technical side-note: the code above works for Apache web server only. If you don't know what it means, then just ignore this side-note. Anyway, it's very likely to work for you if you use a shared hosting.

You can edit your `.htaccess` file via File Manager in your hosting cPanel. Alternatively, if you use a [security plugin Bulletproof Security](#) or [Bulletproof Security Pro](#) you can edit your `.htaccess` file right from your WordPress dashboard.

Here's the screenshot below to show you where my `.htaccess` file is located in File Manager through cPanel (your `.htaccess` file is located in a very similar place, e.g. in your `public_html` folder). If you don't see your `.htaccess` file there, make sure that "**show hidden files**" setting is selected:



And here's an example of a standard WordPress `.htaccess` file with the **HTTP-to-HTTPS redirect block**:

```

1 <IfModule mod_rewrite.c>
2
3 # Redirecting from HTTP to HTTPS
4 RewriteEngine On
5 RewriteCond %{HTTPS} off
6 RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
7
8 # BEGIN WordPress
9 RewriteBase /
10 RewriteRule ^index\.php$ - [L]
11 RewriteCond %{REQUEST_FILENAME} !-f
12 RewriteCond %{REQUEST_FILENAME} !-d
13 RewriteRule . /index.php [L]
14 # END WordPress
15
16 </IfModule>
17
ResearchAsAHobby.com

```

If you have done everything correctly, the redirect from HTTP to HTTPS should work fine. To check it, open some page on your website using HTTP (e.g. <http://yourwebsite.com/somepage/> or <http://www.yourwebsite.com/somepage/>) in your browser. And it should redirect you to the HTTPS page (e.g. <https://yourwebsite.com/somepage/> or <https://www.yourwebsite.com/somepage/>).

6. Check out robots.txt – whether HTTPS pages are blocked

It's likely that you will not need to do anything regarding robots.txt file. But I'm including this section just in case.

The file *robots.txt* may contain [rules](#) for search engine crawlers. The file is located in the root directory (usually this is *public_html* folder, see the screenshot [above](#)).

You just need to make sure that HTTPS pages of your site are not blocked. Open this file and see its code if there's anything suspicious (e.g. you see HTTPS or SSL there).

If *robots.txt* does not exist, just skip this section and move on. You are fine.

Usually, if you or your website developers did not block the HTTPS version of your website using *robots.txt* on purpose, then you should be fine. Otherwise you need to allow search engines crawl your HTTPS website. You may need to contact a technical person to edit your *robots.txt*. This goes out of scope of this tutorial.

7. Make sure your site map now contains HTTPS pages

The sitemap is the file that contains a list of all your posts, pages etc of your website. I've already mentioned where you can find your sitemap (see [above](#)).

You just need to make sure that all the entries in your sitemap are HTTPS, not HTTP.

If it's so, move on to the next section of this tutorial.

If it's not so, something is wrong with one of the previous steps you made. Check out again [this](#) (and particularly [this](#)) section.

8. Let Webmaster Tools and Google Analytics know about your migration to HTTPS

There's nothing special in this section. A few simple steps and you are done. I just refer you to the short tutorial [here](#).

Also, a couple of practical short key notes about Webmaster Tools:

- Add your both HTTPS versions of your website (<https://yoursite.com> and <https://www.youriste.com>).

- Don't forget to submit *sitemap* ("*Crawl / Sitemaps*") for the main HTTPS version of your site for faster re-indexing of your site.
- It's advised to initiate "*Crawl / Fetch as Google*" for your home page of HTTPS version of your site just to make sure Google can see it.
- After everything is done, monitor the indexing statistics ("*Google Index / Index Status*") of your new site. It may take several days or so (depends on your site size) for your new HTTPS version of your site to get fully re-indexed by Google. Meanwhile, some of your pages continue stay indexed as HTTP. Google re-indexes your site page by page, not your website altogether.
- Your website ranking in Google may fluctuate a little during this re-indexing period. But Google promises to give you more SEO love after the migration of your site to HTTPS is complete. (More on SEO concerns see [below](#)).
- Monitor Crawl errors "*Crawl / Crawl Errors*" for all your versions (http://, https://, http://www, https://www). By the way, it's a good idea to monitor the errors regularly, not just after the migration to HTTPS. This is so especially for your main HTTPS version of the site.

9. Setting up your CDN and/or cloud firewall for HTTPS

If you use a CDN (or a cloud proxy/firewall), make sure if you need to do anything to let your website be serviced via HTTPS. Simply contact your CDN/Firewall provider for details if you are not sure.

For example, I use [Sucuri WAF](#) (which I reviewed [here](#), by the way) and when I migrated my website to HTTPS everything worked fine without doing anything from my side. Sucuri Firewall's Basic plan [allows](#) encrypting the connection via Let's Encrypt certificate.

10. SEO concerns connected with migrating to HTTPS

Perhaps you doubt whether your website SEO suffers after migrating to HTTPS. My short answer is that the influence is not significant.

If you have a [301 redirect from HTTP to HTTPS](#) version of your site, then there's nothing much to be concerned about.

However, it's believed that backlinks going to your exact website location (HTTPS) are a bit more precious than links going to a former location (HTTP) which is redirected. So, it may seem that migrating to HTTPS is not good for SEO.

On the other hand, Google [promises](#) to favor websites on HTTPS.

Thus, I think the general SEO effect of migrating to HTTPS is somewhat neutral in short term and positive in a long term.

After all, the sooner you migrate to HTTPS the better since from now on new backlinks will go to the HTTPS version of your site.

By the way, if you can, **update the external links to your site to make them HTTPS**.

This is not critically, but if you can why not doing it?

Of course, you can't replace most of the links. But at least you can do so with the links which you can control (e.g. links from your social media accounts, email templates, ads etc.)

The most important thing regarding SEO is to monitor your crawling errors and indexing status in Google Webmaster Tools as I mentioned [above](#). As Google re-crawls and re-indexes your HTTPS version of your site, it may find errors that you need to fix (e.g. broken URLs).

11. Loosing social media shares after migrating to HTTPS

As migrating to HTTPS is technically considered as migrating to another domain, your social media shares like Facebook, G+, Twitter, Pinterest, LinkedIn etc likes/shares will be lost. To be precise, the social media counts will stay on your previous pages (i.e. HTTP). And you will need to start getting social shares for new HTTPS website afresh.

Probably the most pleasant approach to this issue used to be using a premium plugin that allowed sort of recovery of your social media counts. But due to API changes of some social media networks this functionality has become not guaranteed, unfortunately.

12. Speed concerns after migrating to HTTPS

If you have migrated your site to HTTPS correctly, there should not be any significant speed issues on your site. The negotiation with HTTPS is quite fast (about 150 ms) so the slowdown should not be noticeable.

However, the real life can be more complicated sometimes. And some website owners experience significant slowdowns. In this case the issues causing it should be located and eliminated.

Fixing speed issues goes out of scope of this tutorial. However I can give you some tips regarding it.

In short, some possible reasons **why the HTTPS version of your website may slow down**:

- redirect issues (you may want to check you [this section](#) again);

- loading external resources which are slow via HTTPS (in this case it may make sense to find an alternative external resources or download the resources to your server);
- caching issues (in this case disable temporarily your caching/minification plugins to locate the issue; contact your CDN provider if you use it);
- misconfigured SSL on your server (in this case check out your SSL [here](#) or/and contact your hosting);
- SSL certificate with too heavy encryption (this is the case if you chose SSL certificate with higher security metrics. You don't need RSA keys stronger than 2,048 bits and ECDSA keys stronger than 256 bits).

By the way, there's a strategic solution to **make your HTTPS site faster even compared to your HTTP version**. This is taking advantage of using another protocol ([HTTP/2 \(SPDY\)](#)) which is available only via a secure (HTTPS) connection.

For example, [A2Hosting](#) offers HTTP/2 on [Turbo plan](#).

Another way to improve your HTTPS website performance is to use HSTS. This is a mechanism that allows connecting to your site using only secure connections. This is a good performance improvement for any HTTPS site. Also, it helps to [protect your website](#) better. And it's good for faster [re-crawling of your site](#). Contact you host to make sure HSTS is enabled on your account. And ask them what you need to do to enable HSTS on your website. Usually adding this line of code to your `.htaccess` file is enough:

```
1# Enabling HTTP Strict Transport Security (HSTS)
2Header set Strict-Transport-Security "max-age=31536000" env=HTTPS
```

Some hosts allow HSTS. For example, [A2Hosting enables HSTS](#) for HTTPS websites.

I highly recommend enabling HSTS after you make sure that you don't have mixed content issues on your site and set up redirect from HTTP to HTTPS.

One more efficient way to make your site faster over HTTPS is using [OCSP stapling](#). This is an optimization technique that makes HTTPS requests much faster. However, most shared hosts do not support it, at least for now.

13. Additional steps and other information

In this section I include notes that complete this tutorial.

Just in case: if you disabled caching/minification plugin before migrating your website to HTTPS, don't forget to enable it again. You can do it now.

If you have affiliate links on your website, it's a good idea to *make your affiliate links secure* (i.e. HTTPS) if it's possible. The point is that referrer information is [lost](#) if a visitor goes from a secure (HTTPS) web page to an insecure (HTTP) web page. This might be a reason that your affiliate sales may be not registered.

Here is some more additional information:

- [Useful information from Google for webmasters and FAQ about HTTP to HTTPS moving](#) from Google's guy John Muller.
- Watch [this video](#) for more general information from Google guys about encryption and particularly about moving your site from HTTP to HTTPS (from 19m30s).
- Partly a bit technical but useful overview of best practices implementing HTTPS is [here](#).

Conclusion

Migrating your site to HTTPS it's [not really difficult](#) whether you want to use [a free or paid](#) SSL certificate. You just need you understand the matter of what it means to migrate your website to HTTPS. This article helps you to do it.

However, this tutorial may look big and partly technically complicated. This is so because I wanted to make this tutorial comprehensive. And I wanted to focus on a real life process and issues connected with migrating your site to HTTPS.

Basically, you have [several options](#) how to migrate your website to HTTPS. In some cases migrating to HTTPS is just few clicks away.

In many cases using a plugin helps to migrate your site to HTTPS with minimum efforts. [Really Simple SSL pro](#) plugin is the best option in this respect.

If you want to do everything manually without having to use a plugin or in case you will need to fix mixed content issues on your site you will need to do some easy actions, but do them carefully.

The last but not least. It's advantageous to have a [professional hosting](#) which will not be a cause of your possible issues on a server side.

That's it. Good job! You've done it!

ORIGINAL BLOG POST URL: [HTTPS://RESEARCHASAHOBBY.COM/MOVE-WORDPRESS-WEBSITE-HTTPS-RIGHT-WAY-EASILY/](https://researchasahobby.com/move-wordpress-website-https-right-way-easily/)



I hope you enjoyed the article!

You can read my free researches on resources and tools for bloggers and small business owners on this website.

By the way, if we haven't met before - my name is Michael Bely.

If you have any questions, visit my website and ask any questions in the comments or privately via the Contact Form. Don't be shy!

Do you know that...

[More expensive hosts do NOT always mean better hosts?](#)

My Best Materials:

- [As full as possible list of EIG companies and brands with details \(beware EIG hosting!\)](#)
- [Non-stop hosting monitoring reports](#)
- [One best security plugin or combination of plugins?](#)
- [Protect your website from hacking step-by-step – easy, free and very effective](#)
- [How to migrate WordPress website to HTTPS the right way for free](#)
- [How to copy, clone, migrate big WordPress site easily and for free](#)
- [The best email opt-in plugin I could find for my use](#)
- [Other useful articles...](#)

