The Essential HTTPS Conversion Checklist

- **Identify the best SSL certificate for your website's needs**
  - **Domain-validated SSL** - considered the lightest weight contender. This simply verifies the domain name. Ineffective for any site with private WHOIS information. Best for internal-facing sites. Turnaround time typically is a couple of hours. Approximate cost: $10 to $30.
  - **Organizational SSL** - More effort goes into validating that these SSL holders are actually who they say they are. So, these cost a bit more, but most people agree these are the best buy for business websites. Verification paperwork is required, and that can extend turnaround to a couple of days. Approximate cost: $50 to $100.
  - **Extended Validation** - The EV is considered a good fit for eCommerce sites looking to reassure their customers that they are a legitimate business. Corporate organizational documents often are required to receive this certificate and certify that the domain is owned by a legit legal entity. As you may expect this review can take up to a few weeks to complete. Approximate cost $150 to $200.

- **Review domain infrastructure** - Consider your company’s basic growth plans and other existing subdomains that may be planned during the next year. You may need to explore a Wildcard SSL or Multiple Domain SSL if you have, for
example, a marketing subdomain (marketing.mydomain.com) or other domains under your corporate umbrella.

- **Sales Coordination**
  - Verify opportune windows for the HTTPS conversion. Remember, downtime isn't expected, but plan for the worst and expect the best.
  - Prepare Sales Team Notification
    - Sample Notice - Dear Sales Team: In order to improve our search engine rankings and maintain full website functionality, we are upgrading xyz.com to a more secure status. As with any website updates, downtime is possible. The update will occur during this window: Date/Time Range. Thank you for your cooperation.

- **Check with Your Website Host Support**
  - Are they prepared for this change?
  - Any adjustments that need to be made?
  - Any issues expected following the change?
  - When is the best time for them to provide support if you have an issue?

- **Brief your Server Admin / Web Development team**
  - Ensure essential personnel will be available during the conversion.

- **Schedule your Implementation** - Carefully consider the timing factors above, assess your schedule and set a target time and date for the implementation.
Now that your plan is in place, you’re ready for the execution phase:

- **Purchase an SSL Certificate** - If your website host does not set up the SSL Certificate for you, it will be a matter of generating keys from the seller and pasting them into the website host control panel. Be mindful of the fields and always ask Support if needed—part of your hosting costs is paying for their help in these situations.

- **Configure hosting with SSL Certificate** - HTTPS page of the site that has never visited the site before. Also, to note, if you have not configured the actual website to be HTTPS friendly, you might get redirected back to the HTTP site. Every website host is a little different—some will have an entirely separate folder for HTTPS—so keep an open mind when getting things set up.

- **Change all website links to HTTPS**
  - **Fix non-CMS generated links:**
    - Find all links on the website that are not generated by a CMS. This includes links to a CDN, links to pages, images, JavaScript, or anything that your website will use.
    - Change to relative link paths: If the link is “http://www.example.com/link” than it should be “/link”—this way, even if you are not quite ready to switch everything to HTTPS, these links will still work for an HTTP website. Be sure the links start with that first “/”, otherwise you will run into issues.

Need help? Contact us.
• Test it out: Refresh cache on your browser and the website, then go to the page the link is on and give it a click. You can test to make sure this works on an HTTP or HTTPS website, either way will work as a test.

■ **Tackle CMS generated links:** This varies from platform to platform. Here is how to do it in Magento and WordPress on regular installs. If you have any caching plugins or extensions, it is advised to check support forums for any additional tips. For other CMS platforms, you may need to consult their documentation.

• Look through your CMS pages, posts, static blocks (for Magento), template files, and anything else for improper links that need updating. Some links are generated by your CMS but may generate the wrong URL. For instance, if a URL in a Magento CMS editor is `{{unsecure_base_url}}example.html`, then it should be a relative link, such as `!/example.html`.

• The next step for Magento users: Login to the backend and go to System -> Configuration -> Web -> Secure to verify the settings are correct:
  • Base URL (ends in a slash): Your HTTPS URL, such as https://example.com/
  • Use Secure URLs in Frontend: Yes
  • Use Secure URLs in Admin: Yes

• For WordPress users: I will defer to Yoast’s instruction for this, which can be found [here](#). Essentially, you will need to change the website URL, add some code to force HTTPS in the admin...
area, and perhaps, install this plugin. Because WordPress site's vary so much in their caching plugins, look for help from your plugin's documentation.

- **Check for errors** - Are you feeling lucky...? Hopefully, all of your links and linked files are changed to HTTPS, but it is quite lucky to get them all on your first try. So, hunt for errors. One way to do this is to visit your site. Visit your site in Chrome/Safari/Firefox, right-click an element, and click Inspect Element. From there, look in the Console for errors: if there are incorrect HTTP linked files, an error will be outputted for each one. Another way to look for HTTP links is to pull up the source code for a page and search for anything with “http:” in it...hopefully, nothing is found and your work is complete. Mixed content can hurt your user experience (pesky browser warnings on unsecured pages!) and hurt your SEO rankings.

- **Setup 301 redirects from HTTP to HTTPS or consider HSTS**
  - For Apache-based websites, to redirect all incoming traffic, say from old Google links or dated links on other sites, setting up a redirect for all HTTP requests to be HTTPS can be accomplished fairly easily. Here is some code to add to the top of your .htaccess file in your root folder:
    - RewriteEngine On
    - RewriteCond %{HTTPS} off
    - RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
Once that is in there, test thoroughly that your website is still functional and that any request made to your site is redirected to an HTTPS URL.

- **Optional** Take things to the next level with **HTST** - HTTP Strict Transport Security (HSTS) is for you. A guide can be found [here](#) with steps for setup. HSTS is a way to force all connections to be HTTPS—it essentially acts in the same way as the redirect mentioned above, but in a standardized way. Unfortunately, Internet Explorer has yet to implement a solution, but most other browsers are already up to speed. In the future, HSTS will probably be standard for reputable websites.

We hope this checklist has been helpful. We covered a lot of detail, but there are always unique situations and complexities that may require additional help. Contact us for a free consultation. We’ll be happy to discuss your HTTPS conversion challenges or any of your digital marketing concerns.

*All the best from the Elevated Team!*