

RewriteSSL

Bunch of SSL recipes here, based on frequently asked questions about this. We'll start with the favorite:

Force SSL for a certain URLs

Problem:

A certain part of the web site must always be served via SSL. So, if someone goes to that part of the site without SSL, we want to redirect them.

Recipe:

To force the entire site into SSL (regardless of the context the rule is placed in):

```
RewriteEngine On
RewriteCond %{HTTPS} !=on [NC]
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [R,L]
```

Or, if you wanted to have a particular subdirectory forced into https (again, safe to use in a per-directory context):

```
RewriteEngine On
RewriteCond %{HTTPS} !=on [NC]
RewriteCond %{REQUEST_URI} ^/secure(/.*)?
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [R,L]
```

Please note that `(/.)?` will make the presence of a `/` followed by any characters conditional.

Discussion:

While this can be done rather effectively with a Redirect, if you're not careful, it can loop. Also, the recipe provided here will work for any number of hostnames, and preserve that hostname.

The `[R]` flag means that this is a Redirect, not a rewrite. That is necessary so that the browser is requesting the content via https, not via http.

If you're running Apache httpd 2.0.x or above, you should also look at the [SSLRequireSSL](#) directive, which may, at least in part, do what you wanted.

SSL to the wrong hostname

Due to the fact that you can only have one SSL host per IP address, if you're running several names on the same IP address, you can end up with <https://faulst.example.com/> getting the wrong cert. Here's how you ensure that the "wrong" hostname doesn't go to the SSL vhost.

The following rules would go in your SSL vhost

```
RewriteCond %{HTTP_HOST} !=right.example.com [NC]
# checking for non-empty host header
RewriteCond %{HTTP_HOST} !=""
RewriteRule ^ https://right.example.com/%{REQUEST_URI} [R=301,L]
```