



The Handbook

Codex

[http://codex.wordpress.org/Administration Over SSL](http://codex.wordpress.org/Administration_Over_SSL)

Version Date

6 August 2005

Administration Over SSL

Sometimes, you want your whole wp-admin to run over a secure connection using the https protocol. Conceptually, the procedure works like this:

1. Set up two virtual hosts with the same url (the blog url), one secure, the other not.
2. On the secure virtual host, set up a rewrite rule that shuttles all non-wp-admin traffic to the insecure site.
3. On the insecure virtual host, set up a rewrite rule that shuttles all traffic to wp-admin to the secure host.
4. Put in a filter (via a plugin) that filters the links in wp-admin so that once activated, administrative links are rewritten to use https and that edits cookies to work only over encrypted connections.

The following guide is for WordPress 1.5 and Apache running mod_rewrite, using rewrite rules in httpd.conf (as opposed to .htaccess files) but could easily be modified to fit other hosting scenarios.

Virtual Hosts

You need a (virtual) host configured for the secure server in addition to the non-secure site. In this example, the secure virtual host uses the same `DocumentRoot` as the insecure host. Hypothetically, you could use a host with a different name, such as `wpadmin.mysite.com` and link the document root to the `wpadmin` directory.

Please ask your ISP to set up a secure virtual host for you, or if you have administrative access set up your own. Note that [you cannot use name based virtual hosting to identify different SSL servers](http://httpd.apache.org/docs-2.0/ssl/ssl_faq.html#vhosts2) (http://httpd.apache.org/docs-2.0/ssl/ssl_faq.html#vhosts2).

Rewrite Rules For The Insecure Host

In the .htaccess or virtual host stanza in httpd.conf for your insecure host, add this rewrite rule to automatically go to the secure host when you browse to `http://www.mysite.com/wp-admin/`

```
RewriteRule ^wp-admin/(.*) https://www.mysite.com/wp-admin/$1 [C]
```

If you are using permalink rewrite rules, this line must come before `RewriteRule ^.*$ - [S=40]`.

Rewrite Rules For Secure Host (Optional)

These rewrite rules are optional. They disable access to the public site over a secure connection. If you wish to remain logged in to the public portion of your site using the plugin below, you must *not* add these rules, as the plugin disables the cookie over unencrypted connections.

The secure virtual host should have two rewrite rules in an .htaccess file or in the virtual

host declaration (see [Using Permalinks](#) for more on rewriting):

```
RewriteRule !^/wp-admin/(.*) - [C]
RewriteRule ^/(.*) http://www.mysite.com/$1 [QSA,L]
```

The first rule excludes the wp-admin directory from the next rule, which shuffles traffic to the secure site over to the insecure site, to keep things nice and seamless for your audience.

Setting Wordpress URI

For some plugins to work, and for other reasons, you may wish to set your WordPress URI in options to reflect the https protocol by making this setting https://mysite.com. Your blog address should not change.

Example Config Stanzas

```
<VirtualHost nnn.nnn.nnn.nnn:443>
    ServerName www.mysite.com

    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/thissite.crt
    SSLCertificateKeyFile /etc/apache2/ssl/thissite.pem
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown

    DocumentRoot /var/www/mysite

    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteRule !^/wp-admin/(.*) - [C]
        RewriteRule ^/(.*) http://www.mysite.com/$1 [QSA,L]
    </IfModule>
    ...
</VirtualHost>

# Insecure site
<VirtualHost *>
    ServerName www.mysite.com

    DocumentRoot /var/www/ii/mysite

    <Directory /var/www/ii/mysite >
        <IfModule mod_rewrite.c>
            RewriteEngine On
            RewriteBase /
            RewriteCond %{REQUEST_FILENAME} -f [OR]
            RewriteCond %{REQUEST_FILENAME} -d
            RewriteRule ^wp-admin/(.*) https://www.mysite.com/wp-
admin/$1 [C]
            RewriteRule ^.*$ - [S=40]
            RewriteRule ^feed/(feed|rdf|rss|rss2|atom)/?$
/index.php?&feed=$1 [QSA,L]
            ...
        </IfModule>
    </Directory>
    ...
</VirtualHost>
```

Rewrite for Login and Registration

It is probably a good idea to utilize SSL for user logins and registrations. Consider the following substitute RewriteRules.

Insecure

```
RewriteRule ^wp-(admin|login|register)(.*) https://www.mysite.com/wp-$1$2 [C]
```

Secure

```
RewriteRule !^/wp-(admin|login|register)(.*) - [C]
```

The Link Filter Plugin

This is all good, but requests within the WordPress administrative system still use http. This means that requests (and request headers) first traverse the Internet unencrypted then get shuttled over to https via the rewrite rule.

The plugin is available at [Invisible Institute Secure Admin Links Plugin](http://www.invisibleinstitute.com/2005/06/wordpress-administration-over-ssl/) (<http://www.invisibleinstitute.com/2005/06/wordpress-administration-over-ssl/>), and is WordPress 1.5.1+ compatible.

This plugin processes URLs that point to files in wp-admin and replaces the http request in href tags with an https request. It also uses pluggable functions to modify wp_set_cookie() to only work with encrypted connections. This may cause confusion if your site requires registration functionality, as the user will have to visit the https:// site for their cookies to successfully authenticate.

Summary

This method does *not* fix some [inherent security risks](http://wordpress.org/support/topic/24558#post-154136) (<http://wordpress.org/support/topic/24558#post-154136>) in WordPress, nor does it protect you against man-in-the-middle attacks or other risks that can cripple secure connections.

However, this *should* make it much harder for a malicious person to steal your cookies and/or authentication headers (if using a server based [authentication mechanism](http://dev.webadmin.ufl.edu/~dwc/2005/03/10/http-authentication-plugin/) (<http://dev.webadmin.ufl.edu/~dwc/2005/03/10/http-authentication-plugin/>), which is [now possible](http://norman.rasmussen.org/77/imap-authentication-plugin/) (<http://norman.rasmussen.org/77/imap-authentication-plugin/>) starting with WordPress 1.5) and use them to impersonate you and gain access to wp-admin. It also obfuscates the ability to sniff your content, which could be important for legal blogs which may have drafts of documents that need strict protection.

Verification

On the author's server, logs indicate that both GET and POST requests are over SSL and that all traffic to wp-admin on the insecure host is being shuttled over to the secure host.

Sample POST log line:

```
[Thu Apr 28 09:34:33 2005] [info] Subsequent (No.5) HTTPS request received for  
child 6 (server foo.com:443)  
xx.xxx.xxx.xxx - - [28/Apr/2005:09:34:33 -0500] "POST /wp-admin/post.php  
HTTP/1.1" 302 - "https://foo.com/wp-admin/post.php?acti  
on=edit&post=71" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.7)  
Gecko/20050414 Firefox/1.0.3"
```

More testing, preferably with a packet sniffer and some hardcore network analysis tools, would help to confirm.

Limitations

The author assumes (but hasn't checked) that if the user has stored cookies/told their browser to remember passwords (not based on form fields but if using certain external auth mechanism) and hits <http://www.mysite.com/wp-admin/>, those packets are sent in the clear and the cookie/auth headers could be intercepted. Therefore, to ensure maximum security, the user should explicitly use the https host or always log in at the beginning of new sessions.