SettingUpModSSL

Following Eric Covener's advise from

http://www.mail-archive.com/docs@httpd.apache.org/msg07081.html

I'm using this space to kick-off an SSL How-to, which we will use to replace the current, highly out-dated one.

In the first step I'll just fill in the titles. I really hope for some participation here 🙂 Please also take into consideration features from 2.4, such as OSCP

Also consider that recommendations should include security at a high enough level to make sense. For generating certs, that means 2048, for picking cipher suites, that means strong encryption, etc.

create a (self-signed) certificate or certificate request

add a listen directive

make sure Listen 443 is *before* Listen 80..

Why before??? --DRuggeri

Because http://wiki.apache.org/httpd/InternalDummyConnection

add a VH *:443

elaborate on the option of SNI.

add cert-related directives

These are two. Not 3409.

Required:

- SSLEngine On
- SSLCertificateFile /path/to/cert.pem
- SSLCertificateKeyFile /path/to/key.pem
- SSLCertificateCAFile or SSLCertificateCAPath /path/to/target

Smart to have:

- SSLCipherSuite <Insert cipher suite that is common, but strong>
- SSLProtocol SSLv3 TLSv1

We should probably follow: http://journal.paul.querna.org/articles/2010/07/10/overclocking-mod_ssl/
for some good guidelines on SSLCipherSuite, or mod_ssl in general 😉

Required for client auth

- SSLCACertificateFile
- SSLVerifyClient require

Enabling OCSP for client auth

- directive1
- directive2

Handling the passphrase

When you encrypt a private key with a passphrase, httpd will need access to that passphrase. There are three ways to handle this situation:

- Do nothing - In this case, httpd will prompt the server administrator for the passphrase when starting the instance. This is often an undesired effect in the event you have configured httpd to start via init scripts or are using some other method for automated startup.
- Use SSLPassPhraseDialog "exec:/path/to/command" - there are currently only two requirements for using this method. First, /path/to/command must be a file (TBD: will Daniel ever submit the patch to remove this restriction?) and should be executable as the root user. Second, the STDOUT output to /path/to/command must deliver the full passphrase. Using this method allows you to take any security mechanisms you have in mind to any extent you can imagine before delivering the passphrase via the /path/to/command.
- Decrypt the key - To do this, you would need to use openssl to remove the encryption on the key like so: openssl rsa -in key.pem -out newkey.pem
- See also RemoveSSLCertPassPhrase

WARNING: Although you may implement elaborate methods to protect your passphrase, the httpd image in memory contains the passphrase as a string which can be obtained by triggering a core dump (ie. in the event the root process is compromised).