By John Spinuzzi
Published: 2007-11-16 16:24

How To Enable Multiple HTTPS Sites For One IP On Debian Etch Using TLS Extensions

This how-to is Debian specific but could be ported to other distributions since the concept is the same.In order to use TLS Extensions we have to patch and recompile apache2 and recompile OpenSSL with the enable-tlsext directive.Since TLS Extensions are relatively new, some internet browsers will not work so the apache2 server will deliver just the default site as http 1.0 does on an http 1.1 server.

This how-to assumes you have followed the **perfect_setup_debian_etch** to completion.## 1. Preparing pbuilder

Install pbuilder.

```
apt-get update

apt-get install pbuilder fakeroot sudo devscripts apt-utils
```

Next edit */etc/pbuilder/pbuilderrc* on line 11 to reflect a site closest to you. This is the site that apt will use to resolve dependencies.

```
MIRRORSITE=http://http.us.debian.org/debian
```

In the same file go to line 20 and set *DISTRIBUTION=etch*

Finally create your pbuilder image.

```
pbuilder create --distribution etch
```

2. Patching and recompiling apache2.

First we must create a directory to store the sources in and download them.

```
mkdir /usr/src/apache2

cd /usrsrc/apache2

apt-get source apache2
```

Copy and save this patch to */usr/src/apache2/apache2-2.2.3/httpd-2.2.3-sni.patch*

```
cat > /usr/src/apache2/apache2-2.2.3/httpd-2.2.3-sni.patch
```

```
httpd-2.2.3-sni.patch - server name indication support for Apache 2.2
(see RFC 4366, "Transport Layer Security (TLS) Extensions")
based on a patch from the EdelKey project
(http://www.edelweb.fr/EdelKey/files/apache-2.2.0+0.9.9+servername.patch)
Needs openssl-SNAP-20060330 / openssl-0.9.8-stable-SNAP-20070813 or later
to work properly (ftp://ftp.openssl.org/snapshot/). The 0.9.8 branch
must be configured explicitly for TLS extension support at compile time
("./config enable-tlsext").
Index: httpd-2.2.x/modules/ssl/ssl_engine_init.c
===================================================================
--- httpd-2.2.x/modules/ssl/ssl_engine_init.c (revision 423224)
+++ httpd-2.2.x/modules/ssl/ssl_engine_init.c (working copy)
@@ -156,6 +156,87 @@
     return OK;
 }

+#ifndef OPENSSL_NO_TLSEXT
+static int set_ssl_vhost(void *servername, conn_rec *c, server_rec *s)
+{
+    SSLSrvConfigRec *sc;
```

```
+   SSL *ssl;
+   BOOL found = FALSE;
+   apr_array_header_t *names;
+   int i;
+
+   /* check ServerName */
+   if (!strcasecmp(servername, s->server_hostname))
+       found = TRUE;
+
+   /* if not matched yet, check ServerAlias entries */
+   if (!found) {
+       names = s->names;
+       if (names) {
+           char **name = (char **) names->elts;
+           for (i = 0; i < names->nelts; ++i) {
+               if(!name[i]) continue;
+               if (!strcasecmp(servername, name[i])) {
+                   found = TRUE;
+                   break;
+               }
+           }
+       }
+   }
+
+   /* if still no match, check ServerAlias entries with wildcards */
+   if (!found) {
+       names = s->wild_names;
+       if (names) {
+           char **name = (char **) names->elts;
+           for (i = 0; i < names->nelts; ++i) {
+               if(!name[i]) continue;
+               if (!ap_strcasecmp_match(servername, name[i])) {
+                   found = TRUE;
```

```
+                    break;
+                }
+            }
+        }
+    }
+
+    /* set SSL_CTX (if matched) */
+    if (found) {
+        if ((ssl = ((SSLConnRec *)myConnConfig(c))->ssl) == NULL)
+            return 0;
+        if (!(sc = mySrvConfig(s)))
+            return 0;
+        SSL_set_SSL_CTX(ssl,sc->server->ssl_ctx);
+        return 1;
+    }
+    return 0;
+}
+
+int ssl_set_vhost_ctx(SSL *ssl, const char *servername)
+{
+    conn_rec *c;
+
+    if (servername == NULL)   /* should not occur. */
+        return 0;
+
+    SSL_set_SSL_CTX(ssl,NULL);
+
+    if (!(c = (conn_rec *)SSL_get_app_data(ssl)))
+        return 0;
+
+    return ap_vhost_iterate_given_conn(c,set_ssl_vhost,servername);
+}
+
```

```
+int ssl_servername_cb(SSL *s, int *al, modssl_ctx_t *mctx)
+{
+    const char *servername = SSL_get_servername(s,TLSEXT_NAMETYPE_host_name);
+
+    if (servername) {
+        return ssl_set_vhost_ctx(s,servername)?SSL_TLSEXT_ERR_OK:SSL_TLSEXT_ERR_ALERT_FATAL;
+    }
+    return SSL_TLSEXT_ERR_NOACK;
+}
+#endif
+
 /*
  *  Per-module initialization
  */
@@ -376,6 +457,29 @@
     }
 }

+static void ssl_init_server_extensions(server_rec *s,
+                           apr_pool_t *p,
+                           apr_pool_t *ptemp,
+                           modssl_ctx_t *mctx)
+{
+    /*
+     * Configure TLS extensions support
+     */
+
+#ifndef OPENSSL_NO_TLSEXT
+    ap_log_error(APLOG_MARK, APLOG_DEBUG, 0, s,
+                 "Configuring TLS extensions facility");
+
+    if (!SSL_CTX_set_tlsext_servername_callback(mctx->ssl_ctx, ssl_servername_cb) ||
+        !SSL_CTX_set_tlsext_servername_arg(mctx->ssl_ctx, mctx)) {
```

```
+       ap_log_error(APLOG_MARK, APLOG_ERR, 0, s,
+                    "Unable to initialize servername callback, bad openssl version.");
+       ssl_log_ssl_error(APLOG_MARK, APLOG_ERR, s);
+       ssl_die();
+   }
+#endif
+}
+
 static void ssl_init_ctx_protocol(server_rec *s,
                                   apr_pool_t *p,
                                   apr_pool_t *ptemp,
@@ -709,6 +810,8 @@
         /* XXX: proxy support? */
         ssl_init_ctx_cert_chain(s, p, ptemp, mctx);
     }
+
+    ssl_init_server_extensions(s, p, ptemp, mctx);
 }

 static int ssl_server_import_cert(server_rec *s,
@@ -1035,6 +1138,7 @@
         }
     }

+#ifdef OPENSSL_NO_TLSEXT
     /*
      * Give out warnings when more than one SSL-aware virtual server uses the
      * same IP:port. This doesn't work because mod_ssl then will always use
@@ -1079,6 +1183,7 @@
                     "Init: You should not use name-based "
                     "virtual hosts in conjunction with SSL!!");
     }
+#endif
```

```
 }

 #ifdef SSLC_VERSION_NUMBER
Index: httpd-2.2.x/modules/ssl/ssl_engine_kernel.c
===================================================================
--- httpd-2.2.x/modules/ssl/ssl_engine_kernel.c (revision 423224)
+++ httpd-2.2.x/modules/ssl/ssl_engine_kernel.c (working copy)
@@ -231,7 +231,20 @@
     * the currently active one.
     */

+#ifndef OPENSSL_NO_TLSEXT
    /*
+     * We will switch to another virtualhost and to its ssl_ctx
+     * if changed, we will force a renegotiation.
+     */
+    if (r->hostname && !SSL_get_servername(ssl, TLSEXT_NAMETYPE_host_name)) {
+        SSL_CTX *ctx = SSL_get_SSL_CTX(ssl);
+        if (ssl_set_vhost_ctx(ssl,(char *)r->hostname) &&
+            ctx != SSL_get_SSL_CTX(ssl))
+            renegotiate = TRUE;
+    }
+#endif
+
+    /*
     * Override of SSLCipherSuite
     *
     * We provide two options here:
@@ -997,6 +1010,9 @@
    SSLDirConfigRec *dc = myDirConfig(r);
    apr_table_t *env = r->subprocess_env;
    char *var, *val = "";
+#ifndef OPENSSL_NO_TLSEXT
```

```
+    const char* servername;
+#endif
     STACK_OF(X509) *peer_certs;
     SSL *ssl;
     int i;
@@ -1018,6 +1034,12 @@
     /* the always present HTTPS (=HTTP over SSL) flag! */
     apr_table_setn(env, "HTTPS", "on");

+#ifndef OPENSSL_NO_TLSEXT
+    /* add content of SNI TLS extension (if supplied with ClientHello) */
+    if (servername = SSL_get_servername(ssl, TLSEXT_NAMETYPE_host_name))
+ apr_table_set(env, "TLS_SNI", servername);
+#endif
+
     /* standard SSL environment variables */
     if (dc->nOptions & SSL_OPT_STDENVVARS) {
         for (i = 0; ssl_hook_Fixup_vars[i]; i++) {
Index: httpd-2.2.x/modules/ssl/ssl_toolkit_compat.h
===================================================================
--- httpd-2.2.x/modules/ssl/ssl_toolkit_compat.h (revision 423224)
+++ httpd-2.2.x/modules/ssl/ssl_toolkit_compat.h (working copy)
@@ -258,6 +258,12 @@
 #define SSL_SESS_CACHE_NO_INTERNAL  SSL_SESS_CACHE_NO_INTERNAL_LOOKUP
 #endif

+#ifndef OPENSSL_NO_TLSEXT
+#ifndef SSL_CTRL_SET_TLSEXT_HOSTNAME
+#define OPENSSL_NO_TLSEXT
+#endif
+#endif
+
 #endif /* SSL_TOOLKIT_COMPAT_H */
```

```
/** @} */
```

Use ctrl+d to exit

```
cd apache2-2.2.3/

patch â€“p1 < httpd-2.2.3-sni.patch
```

Change the version.

```
cd debian/

dch â€“i
```

And modify lines 1-5 so it looks like this:

```
apache2 (2.2.3-4a+etch) stable; urgency=low

  * Enabled TLS Extensions

 -- John Doe <john@domain.tld>  Tue,  5 Nov 2007 06:29:54 -0600
```

Recompile the source package.

```
cd ../..

dpkg-source â€“b apache2-2.2.3/ apache2_2.2.3.orig.tar.gz
```

Compile apache2 with pbuilder.

```
pbuilder build apache2_2.2.3-4a+etch.dsc
```

3. Compiling OpenSSL-0.9.8g

Edit /etc/apt/sources.lst with your favorite editor and add a new line using sid for the distrobution

```
deb-src http://ftp.debian.org/debian/ sid main
```

```
apt-get update

mkdir /usr/src/openssl

cd /usr/src/openssl/

apt-get source openssl
```

Edit the file /usr/src/openssl/openssl-0.9.8g/debian/rules and add enable-tlsext on line 22 so it looks like this:

```
CONFARGS  = --prefix=/usr --openssldir=/usr/lib/ssl no-idea no-mdc2 no-rc5 zlib enable-tlsext
```

```
cd openssl-0.9.8g/debian/

dch -i
```

Change the version on lines 1-5 so it look like this:

```
openssl (0.9.8g-1) unstable; urgency=low

  * Enabled TLS Extensions

 -- John Doe <john@domain.tld>  Mon,  5 Nov 2007 22:40:05 -0600
```

Recompile the source package.

```
cd ../..

dpkg-source â€“b openssl-0.9.8g/ openssl_0.9.8g.orig.tar.gz
```

compile OpenSSL with pbuilder.

```
pbuilder build openssl_0.9.8g-1.dsc
```

Installing the newly built packages.

```
cd /var/cache/pbuilder/result

dpkg â€“i apache2_2.2.3-4a+etch_all.deb

dpkg â€“i apache2.2-common_2.2.3-4a+etch_i386.deb

dpkg â€“i apache2-mpm-prefork_2.2.3-4a+etch_i386.deb

dpkg â€“i libssl0.9.8_0.9.8g-1_i386.deb

dpkg â€“i openssl_0.9.8g-1_i386.deb
```

Run this command to fix any dependencies.

```
apt-get install â€“f
```

4. Configure ISPConfig to allow multiple secure websites on one IP address

Edit */home/admispconfig/ispconfig/lib/classes/ispconfig_isp_web.lib.php* and search for the following:

```
//////////////////////////////////////////////////////
// Check ob bereits ein SSL Cert auf der IP Existiert
//////////////////////////////////////////////////////
$ssl_count = $go_api->db->queryOneRecord("SELECT count(doc_id) as ssl_co
if($ssl_count["ssl_count"] > 1) {
  // Es existiert bereits ein SSL Web mit dieser IP
  $status = "NOTIFY";
  $errorMessage .= $go_api->lng("error_web_ssl_exist");
  $go_api->db->query("UPDATE isp_isp_web set web_ssl = 0 where doc_id =
}
```

Comment it out so it looks like this:

```
//////////////////////////////////////////////////////
// Check ob bereits ein SSL Cert auf der IP Existiert
//////////////////////////////////////////////////////
// $ssl_count = $go_api->db->queryOneRecord("SELECT count(doc_id) as ssl_co
// if($ssl_count["ssl_count"] > 1) {
//   // Es existiert bereits ein SSL Web mit dieser IP
//   $status = "NOTIFY";
//   $errorMessage .= $go_api->lng("error_web_ssl_exist");
//   $go_api->db->query("UPDATE isp_isp_web set web_ssl = 0 where doc_id =
// }
```

Search again in the same file and comment those out.

Create a default secure site that users will see if they are using a non RFC 4366 compliant browser.

```
mkdir /var/www/sharedip/ssl

cd /var/www/sharedip/ssl

openssl genrsa -des3 -passout pass:yourpassword -out 192.168.1.2.key2 1024

openssl req -new -passin pass:yourpassword -passout pass:yourpassword -key 192.168.1.2.key2 -out 192.168.1.2.csr -days 365

openssl req -x509 -passin pass:yourpassword -passout pass:yourpassword -key 192.168.1.2.key2 -in 192.168.1.2.csr -out 192.168.1.2.crt -days 365

openssl rsa -passin pass:yourpassword -in 192.168.1.2.key2 -out 192.168.1.2.key

chmod 400 192.168.1.2.key
```

Be sure to enter your own password.
Also, I would use an asteric "*" for the common name.

Edit `/etc/apache2/apache2.conf` and place this above `Include /etc/apache2/vhosts/Vhosts_ispconfig.conf`

```
NameVirtualHost 192.168.1.2:443
<VirtualHost 192.168.1.2:443>
  ServerName localhost
  ServerAdmin root@localhost
  DocumentRoot /var/www/sharedip
  SSLEngine on
  SSLCertificateFile /var/www/sharedip/ssl/192.168.1.2.crt
  SSLCertificateKeyFile /var/www/sharedip/ssl/192.168.1.2.key
```

```
</VirtualHost>
```

Test creating multiple sites with SSL enabled.
You must have an RFC 4366 compliant browser to be able to view the sites correctly.
To test your browser go to **https://dave.sni.velox.ch/** and check if your browser works.

Links:

http://edseek.com/~jasonb/articles/pbuilder_backports/index.html
https://dave.sni.velox.ch/
http://www.edelweb.fr/EdelKey/