NameBasedSSLVHosts

Name-Based VirtualHosts and SSL

Also see SSL with Virtual Hosts Using SNI

As a rule, it is impossible to host more than one SSL virtual host on the same IP address and port. This is because Apache needs to know the name of the host in order to choose the correct certificate to setup the encryption layer. But the name of the host being requested is contained only in the HTTP request headers, which are part of the encrypted content. It is therefore not available until after the encryption is already negotiated. This means that the correct certificate cannot be selected, and clients will receive certificate mismatch warnings and be vulnerable to man-in-the-middle attacks.

In reality, Apache will allow you to configure name-based SSL virtual hosts, but it will always use the configuration from the first-listed virtual host (on the selected IP address and port) to setup the encryption layer. In certain specific circumstances, it is acceptable to use a single SSL configuration for several virtual hosts. In particular, this will work if the SSL certificate applies to all the virtual hosts. For example, this will work if:

1. All the VirtualHosts are within the same domain, eg: one.example.com and two.example.com. 2. You have a wildcard SSL certificate for that domain (one where the Common Name begins with an asterix: i.e `*.example.com`)

Here is the config snippet for two SSL NameVirtualHost's, using a single wildcard SSL certificate. Remember that the SSL directives from the second virtual host will be ignored when setting up the initial SSL connection.

```
Listen 192.168.1.1:443

LoadModule ssl_module   modules/mod_ssl.so

SSLPassPhraseDialog     builtin
AcceptMutex             flock
SSLSessionCache         shmcb:/var/cache/httpd/mod_ssl/ssl_scache(512000)
SSLSessionCacheTimeout  300
SSLMutex                default
SSLRandomSeed           startup /dev/urandom  256
SSLRandomSeed           connect builtin

NameVirtualHost 192.168.1.1:443

<VirtualHost 192.168.1.1:443>
        SSLEngine on
        SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP

        SSLCertificateFile      /etc/ssl/star.example.com.crt
        SSLCertificateKeyFile   /etc/ssl/star.example.com.key

        ServerName      "one.example.com"
        DocumentRoot    "/var/www/html/one"

        CustomLog       "/var/log/httpd/one-access.log" combined
        ErrorLog        "/var/log/httpd/one-error.log"

        <Directory /var/www/html>
                AllowOverride none

                Order Allow,Deny
                Allow from all
        </Directory>
</VirtualHost>

<VirtualHost 192.168.1.1:443>
        SSLEngine on
        SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP

        SSLCertificateFile      /etc/ssl/star.example.com.crt
        SSLCertificateKeyFile   /etc/ssl/star.example.com.key

        ServerName      "two.example.com"
        DocumentRoot    "/var/www/html/two"

        CustomLog       "/var/log/httpd/two-access.log" combined
        ErrorLog        "/var/log/httpd/two-error.log"

        <Directory /var/www/html>
                AllowOverride none

                Order Allow,Deny
                Allow from all
        </Directory>
</VirtualHost>
```

In addition to this configuration, you should still be able to do the following:

1. SSL VirtualHost for a different domain (example2.com), as long as you use a different IP Address or port (i.e. one that is not used by the wildcard sites):

```
<VirtualHost 192.168.1.2:443>
ServerName www.example2.com
...
</VirtualHost>
```

2.#2 `NameVirtualHost <IP>:443` for a different domain (*.example2.com), where `<IP>` is different from the IP Address used for *.example.com

```
NameVirtualHost 192.168.1.2:443
<VirtualHost 192.168.1.2:443>
ServerName one.example2.com
...
</VirtualHost>

<VirtualHost 192.168.1.2:443>
ServerName two.example2.com
...
</VirtualHost>
```

However you cannot do the following:

SSL VirtualHost for a different ServerName (three.example2.com), where the IP address is the same as that used for *.example.com.

```
<VirtualHost 192.168.1.1:443>
ServerName three.example2.com
...
</VirtualHost>
```