-
-

SSL-enabled Name-based Apache Virtual Hosts with mod_gnutls

August 10th, 2007 by George Notaras

This article describes how to implement SSL-enabled name-based vhosts - that is secure virtual hosts which share the same IP address and port - with the **SNI**-capable mod_gnutls module for Apache's httpd web server.

Server Name Indication (**SNI**), as described in section 3.1 of the RFC3546, is a TLS extension which makes the configuration of SSL-enabled name-based virtual hosts possible. This extention eliminates the need for the assignment of one IP address per secure virtual host, therefore the cost for secure web hosting is greatly reduced, as all secure virtual hosts can share the same IP address and port combination. SNI is a huge step forward as it promotes security by making secure web services easier and cheaper to implement. The current version of OpenSSL - 0.98 at the time of writing - does not support SNI yet, but this is planned for the upcoming 0.99 release. On the other hand, mod_gnutls, an experimental module for Apache's httpd which has been around for 2+ years, includes support for **SNI**.

Introduction

Searching the web for mod_gnutls binary distribution packages or information on how to set it up returned very few relevant results. This was a surprise, as, at this moment, the only implementation that supports SNI is mod_gnutls. So, I decided to write a tutorial on how to set things up for a test. I hope you find it useful.

The test that is described in this guide includes:

1. The compilation of the mod_gnutls module.
2. The generation of SSL certificates.
3. The configuration of the SSL-enabled name-based virtual hosts.

This test was performed on a server that runs Fedora 7.

Installation

In order to compile mod_gnutls, you will need the development tools for Fedora:

```
# yum groupinstall "Development Tools"
```

Install the mod_gnutls dependencies:

```
# yum install httpd-devel gnutls-devel
```

As an unprivileged user, download the mod_gnutls distribution and compile it.

```
$ wget http://www.outoforder.cc/downloads/mod_gnutls/mod_gnutls-0.2.0.tar.bz2
$ tar -xjvf mod_gnutls-0.2.0.tar.bz2
```

```
$ cd mod_gnutls-0.2.0
$ ./configure --prefix=/usr
$ make
```

Do not use the '`make install`' script, but perform the installation manually - it is only one library.

As root, copy **libmod_gnutls.so** to the directory that holds the Apache modules (usually `/usr/lib/httpd/modules`) and rename it to **mod_gnutls.so** for consistency:

```
# cp mod_gnutls-0.2.0/src/.libs/libmod_gnutls.so /usr/lib/httpd/modules/mod_gnutls.so
```

During the compilation, two keys, `dhfile` and `rsafile`, have been generated in the `mod_gnutls-0.2.0/data/` directory. It is absolutely important to copy these files in httpd's configuration directory (usually `/etc/httpd/conf/`), otherwise mod_gnutls will never work. This is undocumented, and I found out about it after some trial&error.

As root:

```
# cp mod_gnutls-0.2.0/data/{dh,rsa}file /etc/httpd/conf/
```

Installation is complete.

SSL certificates

In this test installation, two virtual hosts will be used. Thus, two SSL certificates will be required. Please read my article on how to generate SSL certificates for your servers, as this information is beyond the scope of this document. Alternatively, you may use a ready-made **script** which will create those certificates for you quickly. Such scripts are shipped will almost all Linux distributions. Please consult your distribution's documentation for more information.

HTTPd Configuration

The configuration of the Apache web server includes two phases:

1. The configuration of the main server.
2. The configuration of the virtual hosts.

In the following instructions, some brief notes about what each directive does is included. For more detailed information, please consult the mod_gnutls documentation.

Main Server Configuration

This includes setting some general mod_gnutls options, which will be inherited by all virtual hosts.

But, first of all, httpd needs to be set to listen on port 443 (in addition to port 80). Instead of specifying the SSL port only (`Listen 443`) which will lead httpd to listen to all the available network interfaces, you may specify the exact network interface on which the server will listen. For example:

```
Listen 192.168.0.1:443
```

Next, load mod_gnutls:

```
LoadModule gnutls_module modules/mod_gnutls.so
```

Add some MIME-types for downloading Certificates and CRLs from your web sites

(taken from the `mod_ssl` configuration):

```
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
```

It is suggested that you use a session cache for mod_gnutls. This will increase its performance. In this example, the **dbm** cache type is used. This cache type requires a directory where mod_gnutls will actually save SSL session data. So, creating a directory for this purpose and giving ownership to the user that runs Apache (usually `apache` or `www-data`) is needed. Assuming that the Apache user is `apache`, as root issue the commands:

```
# mkdir -m 0700 /var/cache/mod_gnutls_cache
# chown apache:apache /var/cache/mod_gnutls_cache
```

Now, back to the Apache configuration. The following directive sets the **dbm** SSL Session Cache for mod_gnutls:

```
GnuTLSCache dbm "/var/cache/mod_gnutls_cache"
```

Set a timeout for the SSL Session Cache entries. Usually, this is set to 300 seconds:

```
GnuTLSCacheTimeout 300
```

Finally, specify that on the `192.168.0.1:443` interface and port there will be name-based virtual hosts; that is vhosts that share the specified interface and port:

```
NameVirtualHost 192.168.0.1:443
```

Virtual Host Configuration

The example virtual hosts are: `v1.example.org` and `v2.example.org`. It is assumed that two SSL certificates with the canonical name (CN) correctly set to each of the forementioned vhost domains have been generated.

In the following vhost configs, only the absolutely required directives have been used. The rest of the options are inherited from the main server.

```
<VirtualHost 192.168.0.1:443>
    ServerName v1.example.org:443
    GnuTLSEnable on
    GnuTLSCertificateFile /etc/pki_custom/certs/v1.example.org.crt
    GnuTLSKeyFile /etc/pki_custom/private/v1.example.org.key
    DocumentRoot "/var/www/v1/public_html"
</VirtualHost>

<VirtualHost 192.168.0.1:443>
    ServerName v2.example.org:443
    GnuTLSEnable on
    GnuTLSCertificateFile /etc/pki_custom/certs/v2.example.org.crt
    GnuTLSKeyFile /etc/pki_custom/private/v2.example.org.key
    DocumentRoot "/var/www/v2/public_html"
</VirtualHost>
```

Testing the setup

Having finished with the configuration, **review** the changes, **restart** the server and **check** the error logs for any errors.

Use a web browser to visit each of the virtual hosts by using the HTTPS protocol:

```
https://v1.example.org/
https://v2.example.org/
```

Until now, the web server did not support the SNI TLS extension. Therefore, when

visiting the `v2.example.org` virtual host, you would see **two** warnings in your browser. The first one would be because the vhost's certificate has not been issued by a trusted Certificate Authority - this is normal as it was you who issued that certificate - and the other one because on a server without SNI support it is actually the V1 vhost's certificate that is used when visiting V2 vhost over https. Remember the limitation with SSL and name-based virtual hosts?

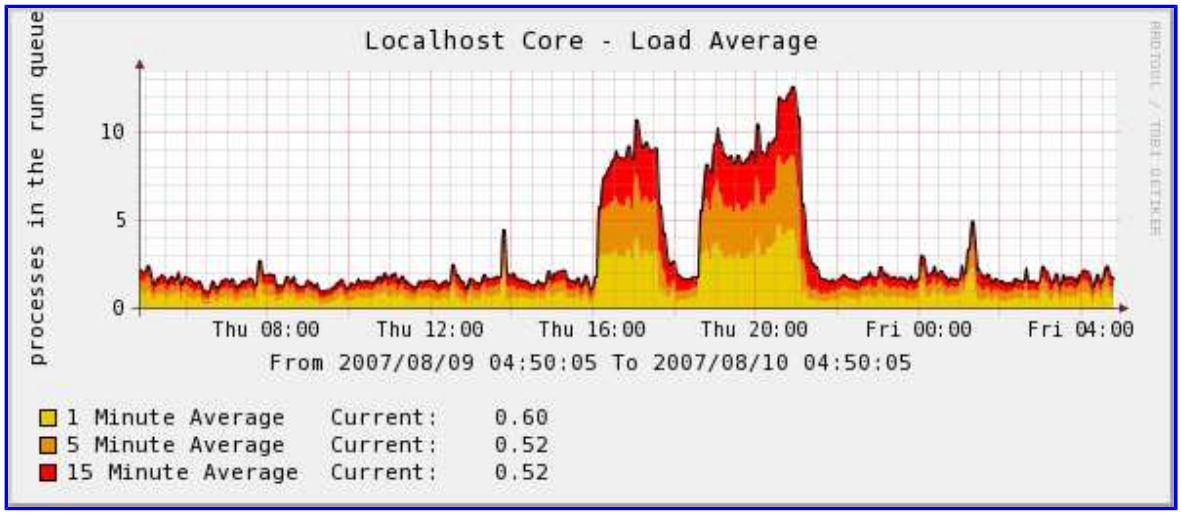
With mod_gnutls, the server supports the SNI TLS extension. Although the virtual hosts are name-based, no matter which one you visit, the relevant certificate for each vhost is used and the only warning you see is the one about the certificates being self-signed. You can get rid of these by purchasing a certificate that is issued by a trusted Certificate Authority.

Conclusion

mod_gnutls works. Actually, it was a real pleasure to see SNI work!

It is important to note though that mod_gnutls is still in experimental phase. Therefore, performance issues should be considered as normal when using it.

At the moment of writing, my server uses Fedora 7 as an operating system. As I haven't upgraded my desktop to F7 yet and my server does not have any development tools installed, I compiled mod_gnutls on a Fedora 6 system and used it on Fedora 7. I do not know if that was the reason - and I did not have the necessary free time to investigate - or anything else, but, during the use of mod_gnutls, my server's load average increased significantly.



I will test mod_gnutls again soon and post the new results, if they are different than the ones I present in this article. I highly recommend that you try it, as it is currently the only way to easily achieve SSL-enabled name-based virtual hosts using the SNI TLS extension. Note, that this extension will be supported by openssl 0.99, so the moment that SNI goes mainstream and such a setup becomes easy and cheap to implement with any Linux distribution is close.

One last thing that has not been mentioned at all is about SNI support in **web browsers**. Currently, with the exception of Safari (this is unconfirmed, please correct me if I am wrong), the latest versions of all major web browsers, Firefox and other Mozilla-based browsers, Internet Explorer, Opera, support SNI.

Tags: Web, Review, Compiling, Apache, Administration, Encryption, Servers, HOWTO, Security, Fedora

The SSL-enabled Name-based Apache Virtual Hosts with mod_gnutls by

19 Responses to “SSL-enabled Name-based Apache Virtual Hosts with mod_gnutls”

1. «» Pingback from TuxFeed › SSL-enabled Name-based Apache Virtual Hosts with mod_gnutls
2. «» Trackback from zymee.com
3. «» Trackback from HackITLinux
4. «» Pingback from the debian user » Blog Archive » Finally - named-based SSL
5. «» Pingback from carlo beccaria - blog / links for 2007-08-12
6. Ignacio Says :
   August 13th, 2007 at 4:06 am

   Have you considered packaging mod_gnutls for Fedora?

7. Vasili Sviridov Says :
   August 13th, 2007 at 9:21 pm

   check out https://sni.velox.ch/ for a mod_ssl (0.9.9 snapshot) with SNI support

8. What is Apache? Says :
   August 14th, 2007 at 1:01 pm

   This is the reason that Apache is so widespread. IIS isn’t this configurable, at least not without an M$ support contract. I love the Apache community. Thanks for the help. Bookmarked!

9. George Notaras Says :
   August 15th, 2007 at 6:00 pm

   I’ve been away for some days. I would like to thank you all for your feedback. :-)

   @Ignacio: Indeed, mod_gnutls is a very good piece of software to get started with RPM packaging for the Fedora project. I have already written a SPEC file including a mod_gnutls.conf configuration file (based on mod_ssl.conf) for the in-house needs. Time permitting, I’ll soon start following each one of the 20 guidelines in order to become a packager. :)

10. «» Trackback from Financial Cryptography
11. Adam Says :
    August 17th, 2007 at 6:22 pm

    Actually IIS has had a feature not as powerful but close since IIS6 came out. It only works with sharing a single wildcard certificate for all sites so they have to be under the same domain (site1.example.com, site2.example.com, etc). It also requires some edits to the metabase for each site. We have been using that for quite a few years now with great results and it works with all browsers.

    This feature with apache is a big step forward though because it allows different certificates for each site. Not supporting some older browsers could be an issue though. I know I wouldn’t want to deploy it just yet.

12. «» Pingback from teh geekosphere.org » » August 2007 in der Geekosphere

13. Nikos Says :
August 20th, 2007 at 11:31 pm

    If you increase the GnuTLSCacheTimeout to 900 or 1500 does the load decrease?

14. George Notaras Says :
August 21st, 2007 at 3:22 am

    @Nikos: This is a very interesting suggestion that I hadn't thought about when trying mod_gnutls. Next time (when time permits) I will try setting the timeout to different values as suggested and check if it has any impact on the server load.

    Thanks for your feedback.

15. Tony Boylan Says :
August 23rd, 2007 at 12:44 pm

    I would like to try out the mod_gnutls module. However I cannot compile it because the following definitions cant be found: "gnutls_certificate_credentials_t, cert_x509, privkey_x509, mgs_srvconf_rec ……. ".

    Can you tell me how to get these definitions ?

    This is the first line of the reported errors when I run make:

    In file included from mod_gnutls.c:18:
    ../include/mod_gnutls.h:79: error: syntax error before
    'gnutls_certificate_credentials_t'

    Thank you. I found the articale very interesting.

16. Tony Boylan Says :
August 24th, 2007 at 10:19 am

    Sorry. RTFM. To Answer my own question. http://www.gnutls.org/download.html

17. George Notaras Says :
August 24th, 2007 at 6:15 pm

    I write this for other readers who run into issues:

    mod_gnutls requires the **httpd** and **gnutls** development libraries in order to compile.

18. Jeremy Says :
September 2nd, 2007 at 5:32 am

    Unfortunately SNI support in browsers is limited to these:

    Firefox 2, IE 7 on Vista, Opera 7.6+ and other modern browsers.

    (http://weblogs.mozillazine.org/gerv/archives/2007/08/virtual_hosting_ssl_and_sni.l

    IE 6, lynx, safari and the like are not supported.

    (http://wiki.cacert.org/wiki/VhostTaskForce#head-7236c4e2c9932ef42056b3ff6d3(

19. «» Pingback from GCU-Squad! » george met ssl à gnou

Leave a Reply