# How to Provision a Linux Web Server for Intel® AES-NI


## Abstract:

This guide will review the steps to configure a server and client to use Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) when performing secure web transactions. Intel AES-NI provides significant performance improvements allowing the use of data protection not feasible before. Intel AES-NI is a set of seven new instructions in the Intel® Xeon® processor 5600 series (formerly codenamed Westmere-EP). The instructions are also available on certain desktop and mobile processors. Several newer Linux distributions have Intel AES-NI support built-in. Older distributions require the use of a patch to OpenSSL. The steps outlined in this paper ensure the software is configuration to use this new capability.


## 1.1 Background Information

A secure web transaction, like accessing one's bank account, encrypts the data before sending it over the internet. Secure Socket Layer (SSL) and the newer Transport Layer Security (TLS) are the protocols typically used to deliver secure transactions over the network. When a client machine wants to securely access a server machine over TLS or SSL a handshake occurs to choose the encryption protocol. For the new instructions to be used, the AES cipher must be selected during the handshake. The encryption cipher is chosen based on the preferred order that is configured in the software. To use AES and therefore Intel AES-NI, the AES cipher should be first on each priority list. The web server should be configured to have the AES cipher as the preferred choice, highest on the cipher list. For the client computers under your control you want to also establish AES as the default cipher. These settings will be reviewed in the steps below to ensure they use the new capabilities offered by the Intel Xeon processor 5600 series.

Several newer distributions have Intel AES-NI support built-in, such as Red Hat Enterprise Linux 6 (in beta2 at time of writing) and Fedora 13. For this paper, Fedora13, RHEL6 beta2 and Firefox 3.5.3 were used. Older distributions require the use of a patch to OpenSSL.

Since detailed step-by-step instructions are dependent on the specific distribution and configuration, some instructions may vary if a different distribution is used.

See http://www.intel.com/technology/security/ for more details on how Intel AES-NI works. Information on the performance improvement with Intel AES-NI on an Apache Web Server using VMware vSphere* is available at: http://software.intel.com/file/26724

## 1.2 Newer Distributions: Confirming AES Cipher Used

Many of the newer Linux distributions have Intel AES-NI built-in. The SSL/TLS protocol does a handshake to determine the common cipher between client and server. For the new instructions to be used, the AES cipher must be selected during the handshake. The encryption cipher is chosen based on the preferred order that is configured in the software. To use AES and therefore Intel AES-NI, the AES cipher should be first on each priority list. The web server should be configured to have the AES cipher as the preferred choice, highest on the cipher list. For the client computers under your control you want to also establish AES as the default cipher.

Apache SSL and TLS support is provided through mod_ssl (http://httpd.apache.org/docs/2.0/mod/mod_ssl.html).

To verify the proper cipher order, use the "openssl ciphers -v" command. See the list below that shows AES at the top of the list.

**Figure 1: Example of ciphers command in openssl**

```
[jeff@localhost conf.d]$ openssl ciphers -v
DHE-RSA-AES256-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
DHE-DSS-AES256-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(256)  Mac=SHA1
DHE-RSA-CAMELLIA256-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(256) Mac=SHA1
DHE-DSS-CAMELLIA256-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(256) Mac=SHA1
AES256-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
CAMELLIA256-SHA         SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(256) Mac=SHA1
PSK-AES256-CBC-SHA      SSLv3 Kx=PSK      Au=PSK  Enc=AES(256)  Mac=SHA1
EDH-RSA-DES-CBC3-SHA    SSLv3 Kx=DH       Au=RSA  Enc=3DES(168) Mac=SHA1
EDH-DSS-DES-CBC3-SHA    SSLv3 Kx=DH       Au=DSS  Enc=3DES(168) Mac=SHA1
DES-CBC3-SHA            SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
PSK-3DES-EDE-CBC-SHA    SSLv3 Kx=PSK      Au=PSK  Enc=3DES(168) Mac=SHA1
KRB5-DES-CBC3-SHA       SSLv3 Kx=KRB5     Au=KRB5 Enc=3DES(168) Mac=SHA1
KRB5-DES-CBC3-MD5       SSLv3 Kx=KRB5     Au=KRB5 Enc=3DES(168) Mac=MD5
DHE-RSA-AES128-SHA      SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
DHE-DSS-AES128-SHA      SSLv3 Kx=DH       Au=DSS  Enc=AES(128)  Mac=SHA1
DHE-RSA-SEED-SHA        SSLv3 Kx=DH       Au=RSA  Enc=SEED(128) Mac=SHA1
DHE-DSS-SEED-SHA        SSLv3 Kx=DH       Au=DSS  Enc=SEED(128) Mac=SHA1
DHE-RSA-CAMELLIA128-SHA SSLv3 Kx=DH       Au=RSA  Enc=Camellia(128) Mac=SHA1
DHE-DSS-CAMELLIA128-SHA SSLv3 Kx=DH       Au=DSS  Enc=Camellia(128) Mac=SHA1
AES128-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
SEED-SHA                SSLv3 Kx=RSA      Au=RSA  Enc=SEED(128) Mac=SHA1
CAMELLIA128-SHA         SSLv3 Kx=RSA      Au=RSA  Enc=Camellia(128) Mac=SHA1
PSK-AES128-CBC-SHA      SSLv3 Kx=PSK      Au=PSK  Enc=AES(128)  Mac=SHA1
```

## 1.3 Older Distributions: Applying AES-NI Patch to OpenSSL

The OpenSSL libraries distributed with older versions Linux, such as RHEL5, do not support Intel AES-NI. To add this capability the patch should be downloaded from openssl.org , apply the patch to OpenSSL and then recompile the Apache Web server. The general instructions for doing this are given below.

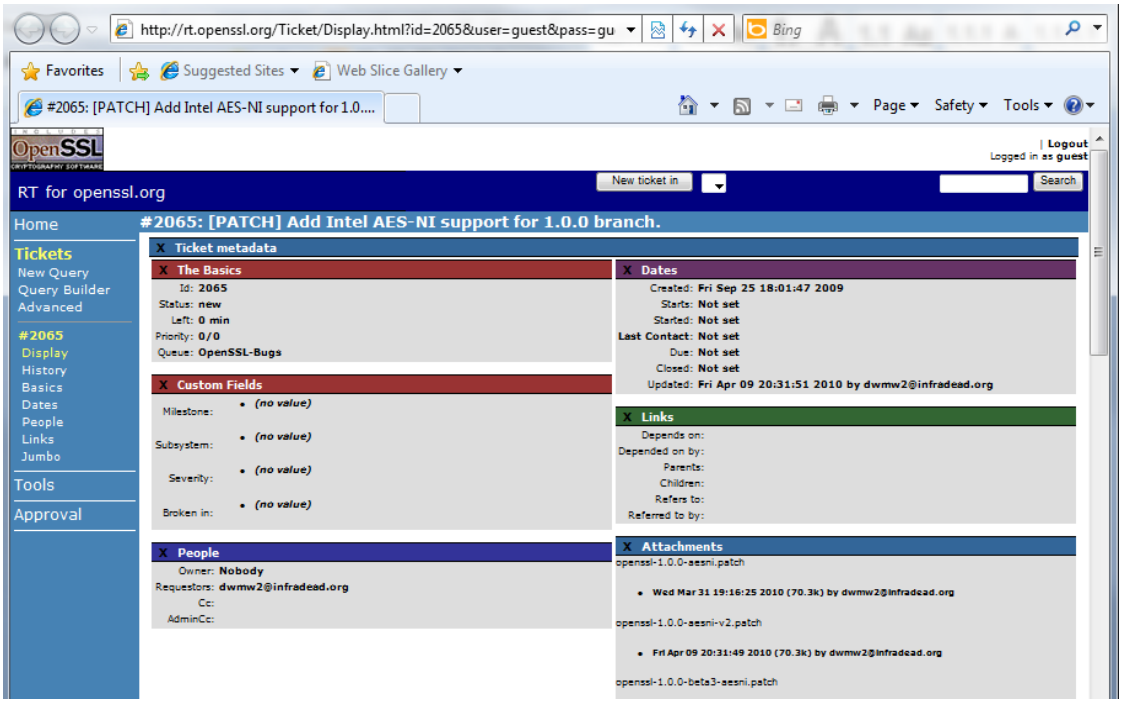
1) Fetch the OpenSSL 1.0.0 tarball from:
   http://www.openssl.org/source/openssl-1.0.0.tar.gz

2) Fetch the latest Intel AES-NI patch from OpenSSL (see Figure 2)
   http://rt.openssl.org/Ticket/Display.html?id=2065&user=guest&pass=guest

3) Extract the tarball and apply the patch:
   a) tar xvfz openssl-1.0.0.tar.gz
   b) cd openssl-1.0.0
   c) patch -p0 < /where/you/put/openssl-1.0.0-aesni-v2.patch
4) After applying the patch compile OpenSSL using the following steps:
   a) /config
   b) make clean
   c) make
   d) make test
   e) make install

5) The final step is to compile the Apache Web service with the patched OpenSSL

**Figure 2: Intel® AES-NI patch from OpenSSL**




## 1.4 Speed Test

OpenSSL includes an easy to run SSL performance test. It will test the different ciphers and different block sizes. Specific algorithm-specific subtest can be run directly. To use the Intel AES-NI technology the –engine aesni and –evp options should be used. The test also allows for multiple tests to be run simultaneous. The example below runs one test using just the AES 256-bit tests with and without Intel AES-NI. The example was run on an Intel SR2625URT "Urbanna" system with one Intel Xeon processor X5670 2.93GHz

Note that a BIOS update (Rev. 50) was required enable the Intel AES-NI instructions on this particular platform.

**Figure 3: Example of OpenSSL Speed Test without Intel® AES-NI**

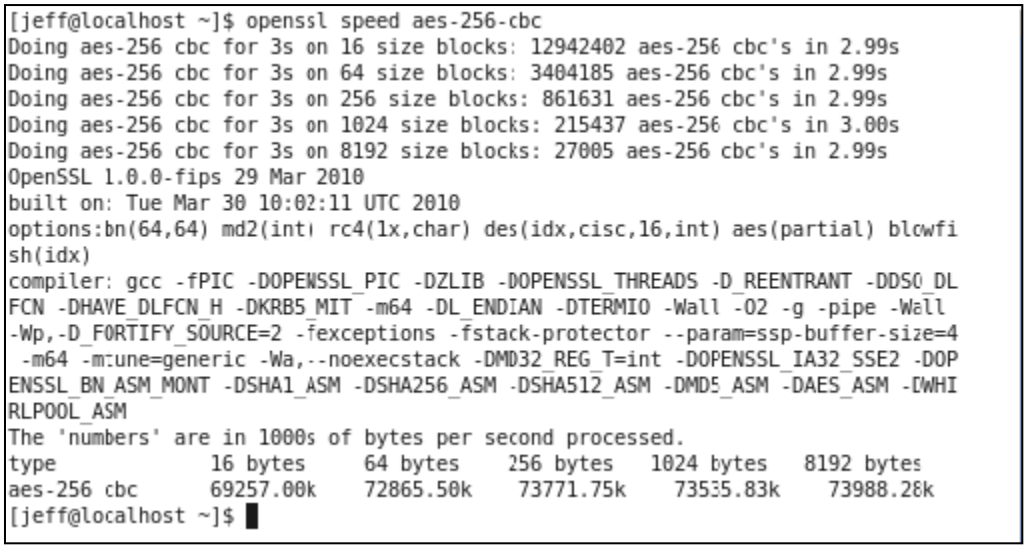
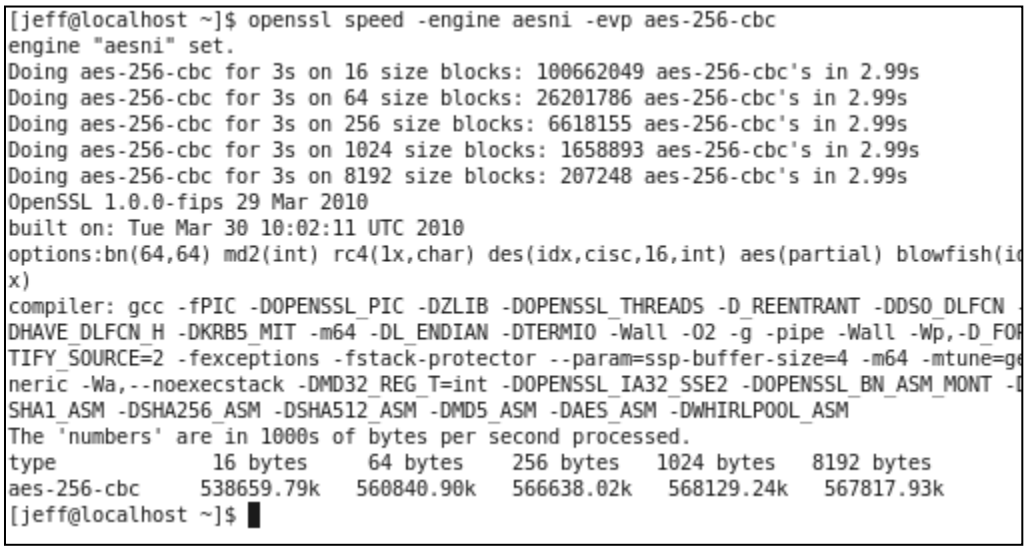



**Figure 4: Example of OpenSSL Speed Test with Intel® AES-NI**

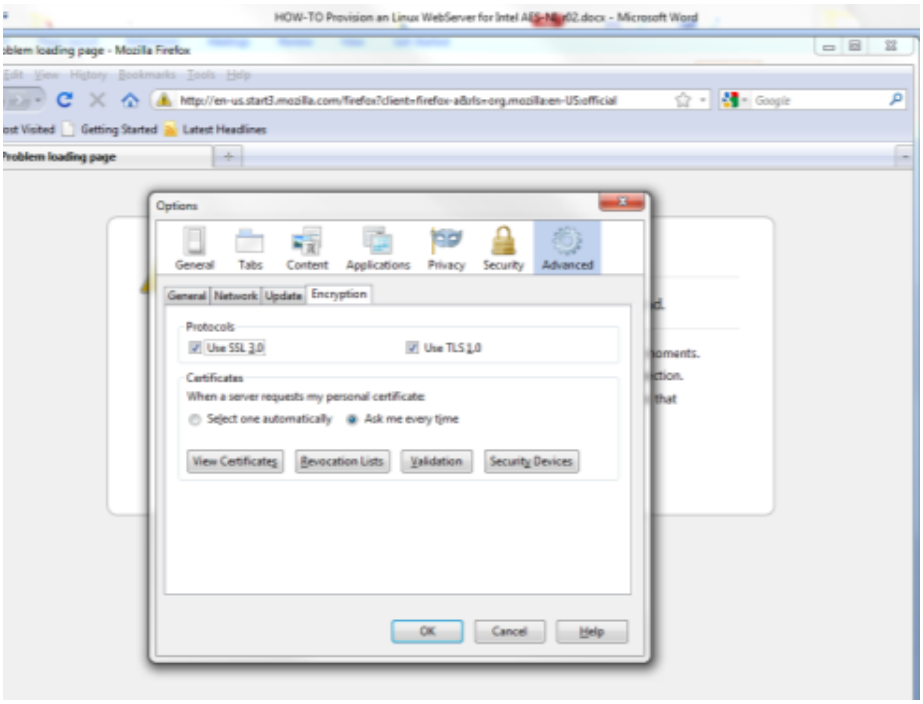



## 1.5 Client Configuration

Since the handshake picks the highest common cipher supported by both server and client, for the clients systems under your control establish AES as the default cipher.

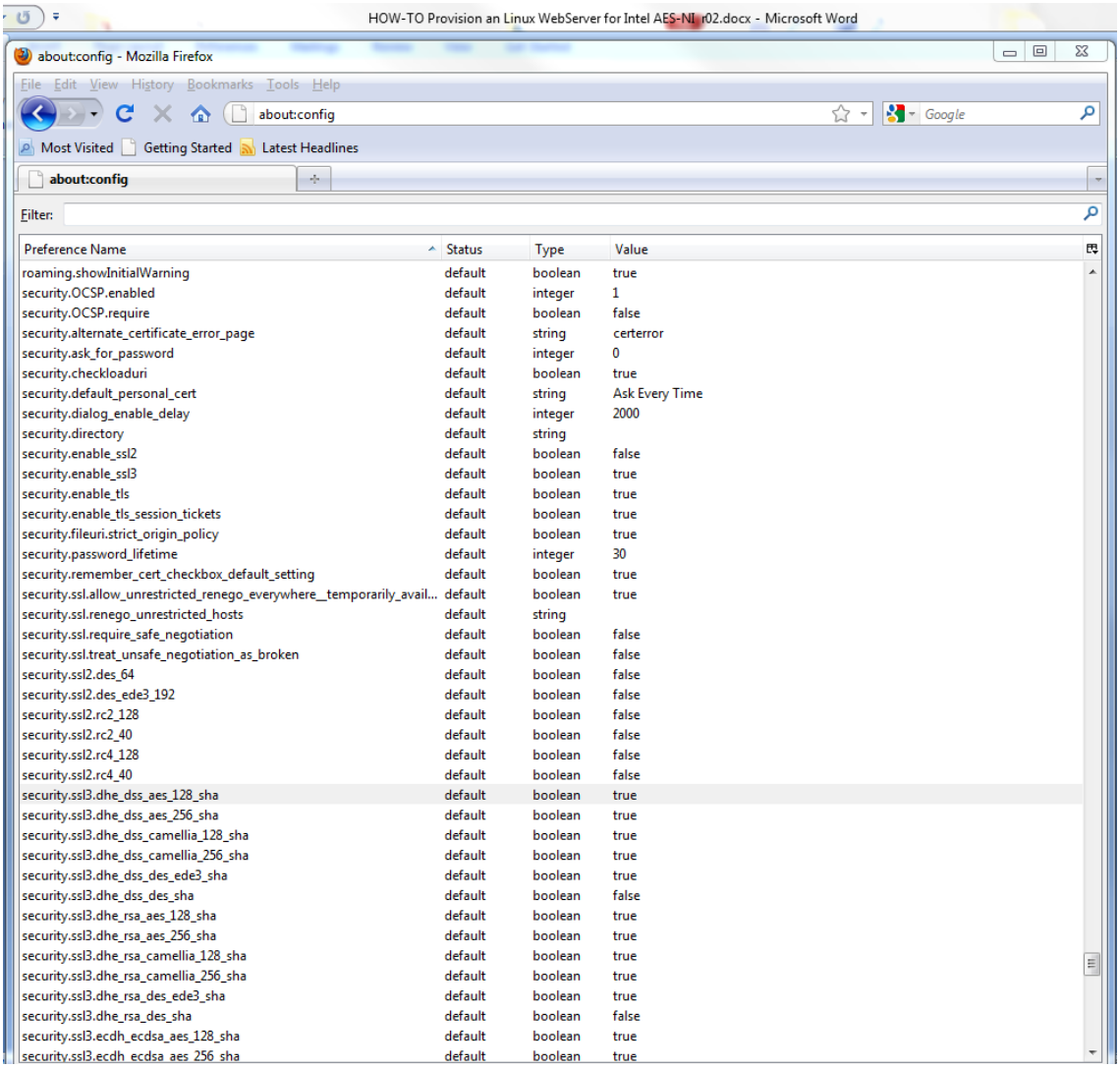
1. First ensure that SSL and TLS are enabled by default. This is found under the menus “Tools” “Options” “Advanced”

**Figure 5: Ensure SSL & TLS are enabled by Default**




2. Next ensure the AES cipher is enabled. In the address of the Firefox browser type "about:config". From the resulting list, ensure AES is enabled as is shown in Figure 6.
3. In our testing (using https://www.fortify.net/sslcheck.html and also using the S_Server capability of Apache http://www.madboa.com/geek/openssl/). An additional step to disable camellia was required in order for the AES cipher to be chosen during the handshake. Double-clicking on the camellia cipher in the "about:config" window will disable it and allow AES to be selected.

**Figure 6: Ensure AES Cipher is enabled and preferred**




## 1.6 Summary

The system is now provisioned for Intel AES-NI which can greatly accelerate the AES encryption algorithm in SSL.