

Hello, and welcome to the OMNIBUS IBM Education assistance module, Configuring binaries for SSL and FIPS .

#### Introduction

Although Secure Socket Layer (SSL) security is not new to the OMNIbus product, significant changes in version 7.2.1 are available to address FIPS 140-2 cryptography requirements. This module highlights updating product binaries for preparation for SSL and FIPS usage.

Some key components for OMNIbus secure communications with FIPS 140-2 compliance are:

- The Java™ Runtime Environment (JRE) Included With OMNIbus
- √ The OMNIbus Global FIPS 140-2 Switch
- ✓ Enabling FIPS 140-2 Support Within iKeyman



Although Secure Socket Layer (SSL) security is not new to the OMNIbus product, significant changes in version 7.2.1 are available to address FIPS 140-2 cryptography requirements. This module highlights updating product binaries for preparation for SSL and FIPS usage. Some key components for OMNIbus secure communications with FIPS 140-2 compliance are:

The Java Runtime Environment (JRE) Included With OMNIbus

The OMNIbus Global FIPS 140-2 Switch

Enabling FIPS 140-2 Support Within iKeyman

### SSL and the JRE

- Secure Sockets Layer (SSL) is a protocol for transmitting sensitive information over data networks.
- SSL, a cryptographic system, uses two keys to encrypt data.
  - Public key
  - Private or secret key
- You must define elements in the java.security file
  - Element definitions are platform specific
  - AIX® and Linux® definitions are similar
  - Solaris and HP-UX definitions are similar
  - Examples of each set of definitions are presented in the following slides.



SSL, a cryptographic system, uses two keys to encrypt data. The first key is a public key the next a private or secret key. You must define different implementations of cryptographic elements in the java.security file. The java.security defines both SSL and FIPS compliance elements. Examples of each set of definitions are presented in the following slides.

# **Changing JRE elements**

 Create a, platform specific, backup copy of \$NCHOME/platform/linux2x86/jre\_1.5.6/jre/lib/s ecurity/java.security.

shell\$> cd
 \$NCHOME/platform/linux2x86/jre\_1.5.6/jre/lib/security/
shell\$> cp java.security java.security.bak

Edit the file \$NCHOME/platform/linux2x86/jre\_1.5.6/jre/lib/s ecurity/java.security.

shell\$> vi java.security



binaryconfigforssl\_fips.ppt

java.security file for editing.

# **Changing JRE elements: AIX and Linux**

# Type the list of security providers in the java.security file exactly as follows:

```
# List of providers and their preference orders:
#
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.spi.IBMCMSProvider
security.provider.4=com.ibm.crypto.provider.IBMJCE
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.security.sasl.IBMSASL
```

Note that any Java application that relies on another JRE distribution (other than the one included with OMNIbus) might require additional modification to support the FIPS-compliant SSL.



Next, for AIX and Linux platforms, modify the java.security file to include the list of cryptographic providers in exactly the order shown. Any other JRE distributions not included with the OMNIbus product distribution might require modifications.



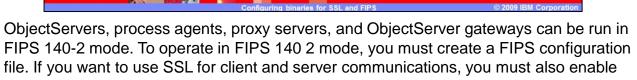
If you are configuring a Solaris or HP-UX platform, modify the java.security file to include the list of cryptographic providers in exactly the order shown.

Continue to edit the java.security file in the section under **java.net.ssl**. Type the lines for the key and trust factories. In the Socket and ServerSocket sections, add the two additional lines. After you complete those last two entries, save and close the file.

## **Enabling FIPS mode**

- ObjectServers, process agents, proxy servers, and ObjectServer gateways can be run in FIPS 140-2 mode.
- To operate in FIPS 140–2 mode, you must create a FIPS configuration file.
- If you want to use SSL for client and server communications, you must also enable FIPS 140–2 mode for the SSL communications.
- Create an OMNIbus V7.2.1 FIPS 140-2 switch with the following commands:

```
shell$> cd $NCHOME/etc/security
shell$> touch $NCHOME/etc/security/fips.conf
```



FIPS 140 2 mode for the SSL communications. You create an OMNIbus 7.2.1 FIPS 140-2 Switch by performing the following command: **touch \$NCHOME/etc/security/fips.conf**.

This command is all that is required to enable FIPS within OMNIbus.

# **Enable FIPS in IBM Key Manager (iKeyman)**

- You must set the relevant iKeyman properties before creating the key database.
- The ikminit.properties file location:
   \$NCHOME/platform/arch/classes/ikminit.properties
  - Where arch represents your operating system directory, such as linux2x86.
- Values changes in the file
  - DEFAULT FIPS MODE PROCESSING=ON
  - DEFAULT\_CRYPTOGRAPHIC\_BASE\_LIBRARY=ICC



To enable FIPS in iKeyman you must set iKeyman properties before creating the key database. The iKeyman.properties file is located under the operating system specific directory. There are two values you must change.

## **Enable FIPS in IBM Key Manager (iKeyman)**

To perform the iKeyman changes, use the following commands:

shell\$> cd \$NCHOME/platform/linux2x86/classes
shell\$> cp
 \$NCHOME/linux2x86/arch/classes/ikminit.properties
 \$NCHOME/platform/arch/classes/ikminit.properties.bak
shell\$> vi
 \$NCHOME/linux2x86/arch/classes/ikminit.properties.

 After opening the file, modify the following properties (uncomment) using an editor:

DEFAULT\_FIPS\_MODE\_PROCESSING=ON
DEFAULT\_CRYPTOGRAPHIC\_BASE\_LIBRARY=ICC



first, change directories to \$NCHOME/platform/linux2x86/classes.

Next, copy ikminit.properties to ikminit.properties.bak then use vi ikminit.properties.

After opening the file, modify the following properties by uncomment them using the editor:

DEFAULT\_FIPS\_MODE\_PROCESSING and DEFAULT\_CRYPTOGRAPHIC\_BASE\_LIBRARY.

# **Encryption**

- Your OMNIbus binaries are now configured to use FIPS compliant encryption within a SSL-secured environment.
- You must use the Advanced Encryption Standard (AES) to encrypt SSL password values while in FIPS mode.
- To maintain FIPS compliance, you might also encrypt string values in files.
- You can use the AES encryption mechanism in ObjectServer, proxy server, probe, and gateway properties files and process agent configuration files.
- To use the Advanced Encryption Standard (AES), use the nco\_aes\_crypt command in place of other encryption commands used in version 7.2.0 and earlier of the OMNIbus product.
- You can find additional information regarding AES property value encryption online or in the IBM Education Assistance Tivoli Netcool/OMNIbus V7.2.1 Security AES property value encryption.



You must use the Advanced Encryption Standard (AES) to encrypt SSL password values while in FIPS mode.

To maintain FIPS compliance, you might also encrypt string values in files.

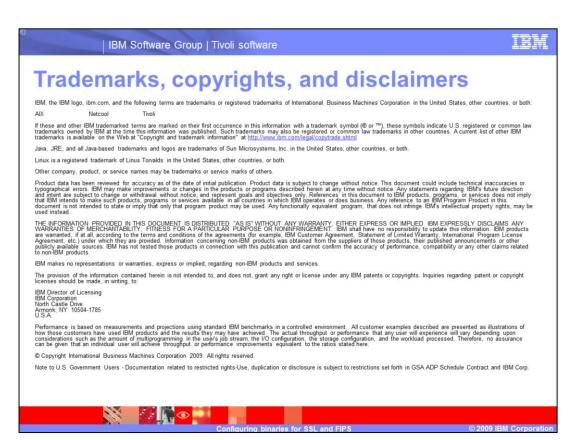
You can use the AES encryption mechanism in ObjectServer, proxy server, probe, and gateway properties files and process agent configuration files.

To use the Advanced Encryption Standard (AES), use the **nco\_aes\_crypt** command in place of other encryption commands used in version 7.2.0 and earlier of the OMNIbus product.

Additional information regarding AES property value encryption can be referenced on-line or in the IBM Education Assistance Tivoli Netcool/OMNIbus V7.2.1 Security AES property value encryption.



For further training, refer to the link, <a href="http://www.ibm.com/software/tivoli/education/edu\_prd.html">http://www.ibm.com/software/tivoli/education/edu\_prd.html</a>.



This concludes this module.