OpenJDK 17

Configuring OpenJDK 17 on RHEL with FIPS

Last Updated: 2022-05-09

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

OpenJDK is a Red Hat offering on the Red Hat Enterprise Linux platform. The Configuring OpenJDK 17 on RHEL with FIPS guide provides an overview of FIPS and explains how to enable and configure OpenJDK with FIPS.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments.

This section explains how to submit feedback.

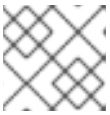
Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in **Multi-page HTML** format.

Procedure

To provide your feedback, perform the following steps:

1. Click the **Feedback** button in the top-right corner of the document to see existing feedback.



NOTE

The feedback feature is enabled only in the **Multi-page HTML** format.

2. Highlight the section of the document where you want to provide feedback.
3. Click the **Add Feedback** pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.
4. Enter your feedback in the text box and click **Submit**.
   A documentation issue is created.
5. To view the issue, click the issue tracker link in the feedback view.

CHAPTER 1. INTRODUCTION TO FEDERAL INFORMATION PROCESSING STANDARDS (FIPS)

The Federal Information Processing Standards (FIPS) provides guidelines and requirements for improving security and interoperability across computer systems and networks. The FIPS 140-2 and 140-3 series apply to cryptographic modules at both the hardware and software levels. The National Institute of Standards and Technology in the United States implements a cryptographic module validation program with searchable lists of both in-process and approved cryptographic modules.

Red Hat Enterprise Linux (RHEL) brings an integrated framework to enable FIPS 140-2 compliance system-wide. When operating under FIPS mode, software packages using cryptographic libraries are self-configured according to the global policy. Most of the packages provide a way to change the default alignment behavior for compatibility or other needs.

OpenJDK 17 is a FIPS policy-aware package.

Additional resources

- For more information about the cryptographic module validation program, see Cryptographic Module Validation Program CMVP on the *National Institute of Standards and Technology* website.
- For more information on how to install RHEL with FIPS mode enabled, see Installing a RHEL 8 system with FIPS mode enabled.
- For more information on how to enable FIPS mode after installing RHEL, see Switching the system to FIPS mode.
- For more information on how to run OpenJDK in FIPS mode on RHEL. See Running OpenJDK in FIPS mode on RHEL.
- For more information on Red Hat compliance with Government Standards, see Government Standards.

CHAPTER 2. CONFIGURE OPENJDK 17 IN FIPS MODE

OpenJDK 17 checks if the FIPS mode is enabled in the system at startup. If yes, it self-configures FIPS according to the global policy. This is the default behavior since RHEL 8.3. Previous RHEL 8 releases require the **com.redhat.fips** system property set to **true** as a JVM argument. For example, **-Dcom.redhat.fips=true**.

NOTE

If FIPS mode is enabled in the system while a JVM instance is running, the instance needs to be restarted for changes to take effect.

You can configure OpenJDK 17 to bypass the global FIPS alignment. For example, you might want to enable FIPS compliance through a Hardware Security Module (HSM) instead of the scheme provided by OpenJDK.

Following are the FIPS properties for OpenJDK 17:

- **security.useSystemPropertiesFile**
  - Security property located at **$JAVA_HOME/conf/security/java.security** or in the file directed to **java.security.properties**.
  - Privileged access is required to modify the value in the default **java.security** file.
  - Persistent configuration.
  - When set to **false**, both the global FIPS and the crypto-policies alignment are disabled. By default, it is set to **true**.
- **java.security.disableSystemPropertiesFile**
  - System property passed to the JVM as an argument. For example, **-Djava.security.disableSystemPropertiesFile=true**.
  - Non-privileged access is enough.
  - Non-persistent configuration.
  - When set to **true**, both the global FIPS and the crypto-policies alignment are disabled; generating the same effect than a **security.useSystemPropertiesFile=false** security property. If both properties are set to different behaviors, **java.security.disableSystemPropertiesFile** overrides. By default, it is set to **false**.
- **com.redhat.fips**
  - System property passed to a JVM as an argument. For example, **-Dcom.redhat.fips=false**.
  - Non-privileged access is enough.
  - Non-persistent configuration.
  - When set to **false**, disables the FIPS alignment while still applying the global crypto-policies. If any of the previous properties is set to disable the crypto-policies alignment, this property has no effect. In other words, crypto-policies is a prerequisite for FIPS alignment. By default, it is set to **true**.

Additional resources

- For more information on how to enable FIPS mode, see Switching the system to FIPS mode .

CHAPTER 3. DEFAULT FIPS CONFIGURATIONS IN OPENJDK 17

OpenJDK 17 includes Introduction to Federal Information Processing Standards (FIPS) configurations that default to FIPS-complaint settings.

Review the following OpenJDK 17 default FIPS configurations before you consider making any changes to these default configurations:

3.1. SECURITY PROVIDERS

The global java security policy file controls the OpenJDK security policy. You can locate the java security policy file at **$JRE_HOME/lib/security/java.security**.

With FIPS mode enabled, OpenJDK replaces the installed security providers with the following ones, which are listed in descending priority order:

SunPKCS11-NSS-FIPS

- Initialized with a Network Security Services (NSS) Software Token (PKCS#11 backend). The NSS Software Token contains the following configuration:
  - name = NSS-FIPS
  - nssLibraryDirectory = /usr/lib64
  - nssSecmodDirectory = /etc/pki/nssdb
  - nssDbMode = readOnly
  - nssModule = fips
- The NSS library implements a FIPS-compliant Software Token. Also, FIPS policy-aware in RHEL.

SUN

- For X.509 certificates support only. Check that your application is not using other cryptographic algorithms from this provider. Otherwise, the security provider throws a **java.security.NoSuchAlgorithmException** message.

SunEC

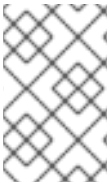
- For **SunPKCS11** auxiliary helpers only. Check that your application is not explicitly using this provider.

SunJSSE

- For TLS support, **SunJSSE** uses the **SUN** provider for X.509 certificates and the **SunPKCS11-NSS-FIPS** provider for all cryptographic primitives.

3.2. CRYPTO-POLICIES

With FIPS mode enabled, OpenJDK takes configuration values of cryptographic algorithms from global crypto-policies. You can find these values at **/etc/crypto-policies/back-ends/java.config**. You can use the **update-crypto-policies** tooling from RHEL to manage crypto-policies in a consistent way.



NOTE

A crypto-policies approved algorithm might not be usable in OpenJDK's FIPS mode. This occurs when a FIPS-compliant implementation is not available in the NSS library or when it is not supported in OpenJDK's **SunPKCS11** security provider.

3.3. TRUST ANCHOR CERTIFICATES

OpenJDK uses the global Trust Anchor certificates repository when in FIPS mode. You can locate this repository at **/etc/pki/java/cacerts**. Use the **update-ca-trust** tooling from RHEL to manage certificates in a consistent way.

3.4. KEY STORE

With FIPS mode, OpenJDK uses the NSS DB as a read-only **PKCS#11** store for keys. As a result, the **keystore.type** security property is set to **PKCS11**. You can locate the NSS DB repository at **/etc/pki/nssdb**. Use the **modutil** tooling in RHEL to manage NSS DB keys.

Revised on 2022-05-09 15:43:04 UTC