Red Hat build of OpenJDK 21

FIPS を使用した RHEL での Red Hat build of OpenJDK 21 の設定

Last Updated: 2024-06-12

Red Hat build of OpenJDK 21 FIPS を使用した RHEL での Red Hat build of OpenJDK 21 の設定

法律上の通知

概要

Red Hat build of OpenJDK は、Red Hat Enterprise Linux プラットフォーム上の Red Hat 製品です。FIPS を使用した RHEL での Red Hat build of OpenJDK 21 の設定 ガイドでは、FIPS の概要と、FIPS で Red Hat build of OpenJDK を有効化および設定する方法を説明します。

目次

RED HAT BUILD OF OPENJDK ドキュメントへのフィードバック

エラーを報告したり、ドキュメントの改善を提案したりするには、Red Hat Jira アカウントにログインし、課題を送信してください。Red Hat Jira アカウントをお持ちでない場合は、アカウントを作成するように求められます。

手順

1. 次のリンクをクリックして **チケットを作成します**。
2. **Summary** に課題の簡単な説明を入力します。
3. **Description** に課題や機能拡張の詳細な説明を入力します。問題があるドキュメントのセクションへの URL も記載してください。
4. **Submit** をクリックすると、課題が作成され、適切なドキュメントチームに転送されます。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 FIPS (FEDERAL INFORMATION PROCESSING STANDARD) の概要

FIPS (Federal Information Processing Standards) は、コンピューターシステムやネットワーク間のセキュリティーおよび相互運用性を強化するためのガイドラインと要件を提供します。FIPS 140-2 および 140-3 シリーズは、ハードウェアおよびソフトウェアの両レベルで暗号化モジュールに適用されます。アメリカ国立標準技術研究所は、進行中の暗号モジュールと承認済みの暗号モジュールの両方に対して検索可能なリストとともに暗号化モジュール検証プログラムを実装しています。

Red Hat Enterprise Linux (RHEL) は、FIPS 140-2 コンプライアンスシステム全体を有効にする統合フレームワークを提供します。FIPS モードで操作する場合、暗号化ライブラリーを使用するソフトウェアパッケージはグローバルポリシーに従って自己設定されます。ほとんどのパッケージでは、互換性やその他のニーズにおいて、デフォルトの調整動作を変更する手段を提供します。

Red Hat build of OpenJDK 21 は、FIPS ポリシー対応パッケージです。

関連情報

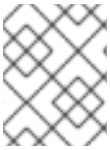
- 暗号モジュール検証プログラムの詳細は、**National Institute of Standards and Technology** Web サイトの Cryptographic Module Validation Program CMVP を参照してください。
- FIPS モードを有効にして RHEL をインストールする方法は、FIPS モードが有効になっている RHEL 8 システムのインストール を参照してください。
- RHEL をインストールした後に FIPS モードを有効にする方法は、FIPS モードへのシステムの切り替え を参照してください。
- RHEL の FIPS モードで Red Hat build of OpenJDK を実行する方法は、Running OpenJDK in FIPS mode on RHEL を参照してください。
- Government 標準による Red Hat コンプライアンスの詳細は、Goverment Standards を参照してください。

第2章 RED HAT BUILD OF OPENJDK 21 における FIPS 設定

起動時に、Red Hat build of OpenJDK 21 は、システムの FIPS ポリシーが有効化されているか確認します。このポリシーが有効化されている場合、Red Hat build of OpenJDK 21 は、Java アプリケーションが FIPS 要件に準拠できるようにするための一連の自動設定を実行します。

これらの自動設定には、次のアクションが含まれます。

- 暗号化操作用の FIPS 認定ネットワークセキュリティーサービス (NSS) ソフトウェアトークンモジュールを含むセキュリティープロバイダーの制限リストをインストールする
- 利用可能なアルゴリズムとパラメーターを制限する Java 用の Red Hat Enterprise Linux (RHEL) FIPS crypto-policies を適用する



注記

JVM インスタンスの実行中にシステムで FIPS モードが有効になっている場合は、変更を有効にするために JVM インスタンスを再起動する必要があります。

前述した FIPS 自動化をバイパスするように、Red Hat build of OpenJDK 21 を設定できます。たとえば、NSS ソフトウェアトークンモジュールではなく、Hardware Security Module (HSM) を使用して FIPS 準拠を実現することを推奨します。

システム または **セキュリティー** プロパティーを使用して FIPS 設定を指定できます。

FIPS プロパティーをより深く理解するには、次の JDK プロパティークラスを理解する必要があります。

- システムプロパティーは、**-D** という接頭辞が付いた JVM 引数であり、通常は **-Dproperty.name=property.value** という形式になります。これらの値を渡すための特権アクセスは必要ありません。起動された JVM のみが設定の影響を受け、永続性はランチャースクリプトの存在に依存します。UTF-8 でエンコードされた値はシステムプロパティーに有効です。
- セキュリティープロパティーは 、**$JRE_HOME/conf/security/java.security** または **java.security.properties** システムプロパティーが指すファイルで使用できます。**$JRE_HOME/conf/security/java.security** ファイルの値を変更するには、特権アクセスが必要です。このファイルへの変更は保持され、同じ Red Hat build of OpenJDK 21 のすべてのインスタンスに影響します。基本ラテン文字以外の Unicode 文字は **\uXXXX** でエンコードする必要があります。

システムプロパティーとセキュリティープロパティーの名前が同じで、異なる値に設定されている場合、システムプロパティーが優先されます。設定によっては、プロパティーが異なる名前を持つ他のプロパティーに影響を与える場合があります。

セキュリティープロパティーとそのデフォルト値の詳細は、**java.security** ファイルを参照してください。

次のリストは、Red Hat build of OpenJDK 21 の FIPS 設定に影響するプロパティーの詳細を示しています。

プロパティー	型	デフォルト値	説明
security.useSystemPropertiesFile	セキュリティー	**true**	このプロパティーを **false** に設定すると、グローバル crypto-policies の調整を含む FIPS 自動化が無効になります。
java.security.disableSystemPropertiesFile	システム	**false**	**true** に設定すると、このプロパティーはグローバル crypto-policies の調整を含む FIPS 自動化を無効にします。これは、**security.useSystemPropertiesFile=false** セキュリティープロパティーと同じ効果があります。両方のプロパティーが異なる動作に設定されている場合、**java.security.disableSystemPropertiesFile** が優先されます。
com.redhat.fips	システム	**true**	このプロパティーを **false** に設定すると、FIPS crypto-policies は引き続き適用されますが、FIPS 自動化は無効になります。上記のいずれかのプロパティーが FIPS 自動化を無効にするように設定されている場合、このプロパティーは効果がありません。暗号化ポリシーは FIPS 自動化の前提条件です。
fips.keystore.type	セキュリティー	**PKCS12**	このプロパティーは、Red Hat build of OpenJDK 21 が FIPS モードの場合のデフォルトのキーストアタイプを設定します。サポートされる値は **PKCS12** と **PKCS11** です。

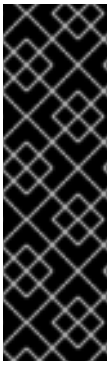
前述の設定に加えて、FIPS モードで NSS DB キーストアを使用するために特定の設定を適用できます。これらのキーストアは、**SunPKCS11** セキュリティープロバイダーと、セキュリティープロバイダーの **PKCS#11** バックエンドである NSS ソフトウェアトークンによって処理されます。

次のリストは、Red Hat build of OpenJDK 21 の NSS DB FIPS プロパティーの詳細を示しています。

プロパティー	型	デフォルト値	説明

プロパティー	型	デフォルト値	説明
fips.nssdb.path	システムまたはセキュリティー	**sql:/etc/pki/nssdb**	NSS DB の場所を指すファイルシステムパス。 このプロパティーの構文は、**SunPKCS11** NSS 設定ファイルで使用可能な **nssSecmodDirectory** 属性と同じです。このプロパティーでは、参照される NSS DB が SQLite タイプであることを示す **sql:** 接頭辞を使用できます。

プロパティー	型	デフォルト値	説明
fips.nssdb.pin	システムまたはセキュリティー	**pin:** (空の PIN)	**fips.nssdb.path** が指す NSS DB の PIN (パスワード)。 このプロパティーを使用して、NSS DB PIN を次のいずれかの形式で渡すことができます。 ● **pin:<value>** この場合、**<value>** はクリアテキストの PIN 値 (例: **pin:1234abc**) です。 ● **env:<value>** この場合、**<value>** は PIN 値を含む環境変数です (例: **env:NSSDB_PIN_VAR**)。 ● **file:<value>** この場合、**<value>** は、最初の行に PIN 値が含まれる UTF-8 でエンコードされたファイルへのパスです (例: **file:/path/to/pin.txt**)。 **pin:<value>** オプションは、PIN 値が JVM 引数として渡されるか、システムプロパティーを通じてプログラムで渡されるかの両方のケースに対応します。PIN 値をプログラムで設定すると、アプリケーションが PIN を取得する方法を柔軟に決定できるようになります。 **file:<value>** オプションは、NSS DB PIN の変更に使用される NSS **modutil -pwfile** および **-newpwfile** 引数と互換性があります。  **注記** 暗号化操作に NSS DB 認証が必要で、ステータスが認証されていない場合、Red Hat build of OpenJDK 21 はこの PIN 値を使用して暗黙的なログインを実行します。アプリケーションは、暗号化操作の前に **KeyStore::load** を呼び出すことで、明示的なログインを実行できます。



重要

セキュリティー評価を実行して、保存されたキーと証明書のインテグリティーと機密性を保護する設定を決定できるようにします。この評価では、脅威、コンテキスト情報、およびオペレーティングシステムのユーザー分離やファイルシステムの権限など、実施されているその他のセキュリティー対策を考慮する必要があります。たとえば、キーを保存し、マルチユーザー環境で実行されるアプリケーションには、デフォルトの設定値が適切でない場合があります。RHEL の **modutil** ツールを使用して NSS DB キーストアを作成および管理し、**certutil** または **keytool** を使用して証明書とキーをインポートします。

関連情報

- FIPS モードを有効にする方法の詳細は、システムを FIPS モードに切り替える を参照してください。

第3章 RED HAT BUILD OF OPENJDK 21 における FIPS 自動化

この章では、Red Hat build of OpenJDK 21 で FIPS 自動化がどのように実装されているか、また FIPS 自動化がアプリケーションにどのような影響を与えるかについて説明します。

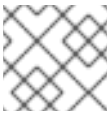
3.1. セキュリティープロバイダー

FIPS モードが有効になっている場合、Red Hat build of OpenJDK 21 は、インストールされているセキュリティープロバイダーを制約リストに置き換えます。FIPS 認定モジュールのみが暗号化操作を実行するように、一部のセキュリティーサービスとアルゴリズムが削除される場合があります。次のリストは、インストールされているセキュリティープロバイダー、サービス、アルゴリズム、および有効な設定を示しています。

SunPKCS11-NSS-FIPS

$JRE_HOME/conf/security/nss.fips.cfg にある設定に従って、サービスプロバイダーの **PKCS#11** バックエンドである NSS ソフトウェアトークンで初期化されます。

- **name = NSS-FIPS**
- **nssLibraryDirectory = /usr/lib64**
- **nssSecmodDirectory = ${fips.nssdb.path}**
- **nssDbMode = readWrite**
- **nssModule = fips**
- **attributes(*,CKO_SECRET_KEY,CKK_GENERIC_SECRET)={ CKA_SIGN=true }**



注記

この設定を変更することは推奨されません。

すべての暗号化サービスが有効になっています。これらには、**AlgorithmParameters**、**Cipher**、**KeyAgreement**、**KeyFactory**、**KeyGenerator**、**KeyPairGenerator**、**KeyStore**、**Mac**、**MessageDigest**、**SecretKeyFactory**、**SecureRandom**、**Signature** が含まれます。

SUN

X.509 証明書関連 (**CertificateFactory**、**CertPathBuilder**、**CertPathValidator**、**CertStore**)、**AlgorithmParameterGenerator**、**AlgorithmParameters**、および **KeyStore** (**JKS**、**PKCS12**) サービスのみが有効になっています。

SunEC

AlgorithmParameters および **KeyFactory** サービスのみが有効化されます。

SunJSSE

TLS 関連サービス (**KeyManagerFactory**、**SSLContext**、**TrustManagerFactory**) と **KeyStore** (**PKCS12**) のみが有効になります。

SunJCE

AlgorithmParameterGenerator、**AlgorithmParameters**、**KeyFactory**、および **SecretKeyFactory** (**BKDF2** アルゴリズムを除く) サービスのみが有効になっています。

SunRsaSign

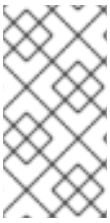
AlgorithmParameters および **KeyFactory** サービスのみが有効化されます。

XMLDSig

すべてのサービスが有効化されています。これらには、**TransformService**、**KeyInfoFactory**、**XMLSignatureFactory** が含まれます。

3.2. CRYPTO-POLICIES

FIPS モードでは、Red Hat build of OpenJDK 21 は、RHEL のグローバル FIPS crypto-policies から無効な暗号化アルゴリズムとその他の設定のリストを取得します。これらの値は **/etc/crypto-policies/back-ends/java.config** にあります。RHEL の **update-crypto-policies** ツールを使用して、crypto-policies を一貫して管理できます。



注記

Red Hat build of OpenJDK が FIPS モードで設定されている場合、crypto-policies 承認アルゴリズムが使用できない可能性があります。これは、FIPS 認定の実装が NSS ソフトウェアトークンで利用できない場合、または **SunPKCS11** セキュリティープロバイダーでサポートされていない場合に発生します。

3.3. TRUST ANCHOR 証明書

FIPS モードでは、Red Hat build of OpenJDK 21 はデフォルトでグローバルトラストアンカー証明書リポジトリーを使用します。この動作は非 FIPS モードと同じです。このリポジトリーは **/etc/pki/java/cacerts** にあります。証明書を一貫して管理するには、RHEL の **update-ca-trust** ツールを使用します。オプションで、トラストアンカー証明書を独自の **PKCS12** および **PKCS11** キーストアに保存し、TLS 通信に使用することもできます。詳細は、**TrustManagerFactory::init** のドキュメントを参照してください。

javax.net.ssl.trustStoreType システムプロパティーが設定されておらず、FIPS モードが有効になっている場合、Red Hat build of OpenJDK 21 は、このシステムプロパティーを **keystore.type** セキュリティープロパティーの値に自動的に設定します。この動作は非 FIPS モードと同じです。

3.4. キーストア

FIPS モードでは、Red Hat build of OpenJDK 21 により、**PKCS12** および **PKCS11** キーストアタイプの使用が有効化されます。デフォルトでは **PKCS12** が使用されます。**fips.keystore.type** セキュリティープロパティーを使用して、デフォルトのキーストアタイプを変更できます。アプリケーションは、**KeyStore.getInstance(<type>)** を呼び出すときに使用するキーストアタイプを選択することもできます。

PKCS11 キーストアを開くとき、Red Hat build of OpenJDK 21 は **/etc/pki/nssdb** にある SQLite NSS DB を使用します。この NSS DB はキーを保存するのに適していない可能性があります。**fips.nssdb.path** プロパティーに値を設定することで、別のデータベースを指定できます。詳細情報とセキュリティー上の考慮事項については、Red Hat build of OpenJDK 21 の FIPS 設定 を参照してください。

fips.keystore.type セキュリティープロパティーを **PKCS11** に設定し、FIPS モードを有効にすると、Red Hat build of OpenJDK 21 は **javax.net.ssl.keyStore** システムプロパティーに **NONE** の値を自動的に割り当てます。この動作により、手動の設定手順が省略され、**PKCS#11** キーストアの使用が容易になります。詳細は、JDK-8238264 を参照してください。

改訂日時: 2024-05-29