HP Operations Orchestration

ソフトウェアバージョン: 10.50
Windows および Linux オペレーティングシステム

セキュリティおよびハードニングガイド

ドキュメントリリース日: 2015 年 9 月 (英語版)
ソフトウェアリリース日: 2015 年 9 月

ご注意

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

商標について

Adobe™ は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。

Microsoft ® および Windows ® は、米国におけるMicrosoft Corporationの登録商標です。

UNIX ® は、The Open Group の登録商標です。

本製品には、'zlib' (汎用圧縮ライブラリ) のインタフェースが含まれています。'zlib': Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。**https://softwaresupport.hp.com/group/softwaresupport/**

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。
http://h20229.www2.hp.com/passport-registration.html

または、HP Passport のログインページの [**New users - please register**] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。**https://softwaresupport.hp.com/**

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Nowは、HPSWのソリューションと統合に関するポータルWebサイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューションを検索したり、HP製品間の統合に関する詳細なリストやITILプロセスのリストを閲覧することができます。このサイトのURLは **http://h20230.www2.hp.com/sc/solutions/index.jsp** です。

目次

概要

『HP セキュリティおよびハードニングガイド』にようこそ。

このガイドは、HP Operations Orchestration (HP OO) のインスタンスを安全な方法でデプロイおよび管理する IT の専門家を支援することを目的としています。HP OO のさまざまな機能について十分な知識を持って決定を下すことができるように支援し、企業のセキュリティに対する最新ニーズを満たすことを目的としています。

企業のセキュリティ要件は常に進化しているため、このガイドラインでは厳しい要件に対応できるように最善を尽くしています。このガイドでカバーしていないセキュリティ要件がある場合は、記録しますのでサポート事例を HP サポートチームに率直にお話ください。お話いただきましたサポート事例は、このガイドの今後の版に掲載します。

テクニカルシステムランドスケープ

HP OO は、Java 2 Enterprise Edition (J2EE) テクノロジーをベースとするエンタープライズワイドなアプリケーションです。J2EE テクノロジーは、エンタープライズアプリケーションを設計、開発、アセンブル、デプロイするためのコンポーネントベースの手法を提供します。

セキュリティ更新

HP OO 10.20 と 10.50 の間では、以下のセキュリティ更新が行われました。

- Central で [**ログインしているユーザーの資格情報のキャプチャーを有効にする**] チェックボックスが選択されている場合は、HP OO は、ログインしているユーザーがリモートデバッガーでフローを実行した時に、そのユーザーの資格情報を安全な方法で一時的にキャプチャーします。資格情報がキャプチャーされる可能性があることを警告するメッセージが表示されます。
- HP OO 10.50 では、デフォルトでは、デフォルトの役割はありません。このため、ユーザーは、ユーザーまたは LDAP グループに明示的に割り当てられている役割しか取得できず、管理者がユーザー権限をより効果的に制御できます。
- HP OO に複数の LDAP 構成があり、管理者がそれらの 1 つにデフォルトとしてフラグを立てた場合は、そこに属するユーザーはログイン時にドメインを選択する必要はありません。
- HP OO 10.50 は、実行中は機密データ (パスワードなど) をセキュリティで保護します。Studio で変数を機密とマークした場合は、スクリプトレットへの使用時に、変数が暗号化形式で取得されます。

HP OO 10.10 と 10.20 の間には、以下のセキュリティ更新が行われました:

- HP OO でシステムアカウントのアクセス許可を付与することができるようになりました。これにより、どのユーザーがどのシステムアカウントを表示可能か、またそのアカウントを使用するフローを実行可能かについて、管理者が制御できます。この機能は、複数の組織があり、一部のシステムアカウントを一部のユーザーに表示しないようにする場合便利です。

詳細については、『HP OO 10.20 リリースノート』の「コンテンツ管理の拡張 - 複数の役割へのアクセス許可の適用」を参照してください。

- [アクセス許可の編集] ダイアログボックスで、アクセス許可を複数の役割に適用できるようになりました。以前のバージョンでは、一度に 1 つの役割しか選択できませんでした。

  詳細については、『HP OO 10.20 リリースノート』の「コンテンツ管理の拡張 - システムアカウントのアクセス許可」を参照してください。
- HP OO インストールを前の 10.x バージョンからアップグレードする場合、Oracle から発行された最新の信頼されたルート証明書を含むように SSL 信頼ストアが更新されます。この処理では、期限切れの証明書の削除と、新しい証明書のインポートが行われます。

  詳細については、『HP OO 10.20 リリースノート』の「インストールの拡張 - 信頼されたルート証明書の更新」を参照してください。
- HP OO でイベントを監査するオプションが提供され、セキュリティ違反を追跡できるようになりました。監査を行うと、Central で行われるアクション (ログイン、フローの起動、スケジュールの作成、構成の編集など) を追跡できます。

  監査証跡は、現在のところ API 経由のみで取得できます。詳細については、『HP OO API Guide』を参照してください。
- HP OO が、2048 ビット長 (およびそれ以上) の暗号化キーをサポートするようになりました。これで、HP OO で使用する暗号化キーが FIPS 186-4 標準に添うようになります。
- **server.xml** (**<インストールフォルダー>/central/tomcat/conf/server.xml**) ファイルに新しく sslEnabledProtocols プロパティが追加されました。

  ```
  sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
  ```

  このプロパティにより、TLS v1、TLS v1.1、TLS v1.2 だけを許可し、SSL 3.0 は許可しないことを徹底できます。これは、 “POODLE” 攻撃 (Padding Oracle On Downgraded Legacy Encryption) に対する脆弱性を防止します。

関連ドキュメント

HP OO のセキュリティハードニングの詳細については、以下のドキュメントを参照してください。

- HP OO Network Architecture White Paper

HP OO の詳細については、以下のドキュメントを参照してください。

- HP OO コンセプトガイド
- HP OO Administrator Guide
- HP OO アーキテクチャーガイド
- HP OO データベースガイド
- HP OO Central ユーザーガイド
- HP OO Studio オーサリングガイド

- HP OO リリースノート
- HP OO インストール、アップグレード、構成ガイド
- HP OO システム要件
- REST Wizard User Guide

これらのドキュメントおよびその他のドキュメントについては、HPLN (https://hpln.hp.com/node/21/otherfiles#) を参照してください。

セキュリティの概要

このセクションでは、HP OO の安全な実装を実現するためのセキュリティモデルと推奨事項の概要を説明します。これには、認証、権限、暗号化などが含まれます。該当する場合には、他の HP OO ドキュメントへの参照もあります。ドキュメントでは、セキュリティ関連のタスクを完了する方法を説明しています。

セキュリティの概念

HP OO 用語集

HP OO の概念の詳細については、『HP OO コンセプトガイド』を参照してください。

役割のアクセス許可

アクセス許可とは、あらかじめ定義されたタスクの実行権限です。HP OO Central には、役割に割り当てられる権限のセットがあります。

たとえば、**スケジュール**権限は、実行スケジュールを表示および作成できる権限を付与します。

役割

役割は、アクセス許可の集合です。

たとえば、[**フロー管理者**] の役割は、[**スケジュールの表示**] 権限と [**スケジュールの管理**] 権限を割り当てることができます。

ユーザー

ユーザーは、個人をアラートワークシートそれらの認証を定義する個人 (またはアプリケーション ID) に関連付けられるオブジェクトです。

役割はユーザーに割り当てられ、Central での実行権限を持つ操作を定義します。たとえば、ユーザー「ジョー・スミス」には、[**フロー管理者**] の役割を割り当てることができます。

別のタイプのユーザーを構成することもできます。

- [**LDAP ユーザー**] は、LDAP ユーザー名とパスワードで Central にログオンします。たとえば、Active Directory ユーザー名とパスワードを使用します。
- [**内部ユーザー**] は、Central でローカルに設定したユーザー名とパスワードで Central にログオンします。
- **LWSSO** - HP Lightweight Single Sign On (SSO) は、1 回のユーザー認証および権限の操作で、LW SSO をサポートするすべての HP システムにユーザーがアクセスできようにするメカニズムです。たとえば、ユーザーが LW SSO が有効な別 HP 製品の Web クライアントにログオンした場合、このユー

ザーは、HP OO Central ログオン画面をバイパスして、直接 HP OO Central アプリケーションに入ることができます。

同じ役割を持つ内部ユーザーと LDAP ユーザーがログインした場合、両者のアクセス許可に違いはありません。

注: LDAP ユーザーは LDAP プロバイダーが実装したポリシーに従ってセキュリティで保護されているため、内部ユーザーより LDAP ユーザーの使用をお勧めします。

コンテンツのアクセス許可

コンテンツのアクセス許可は、個々のフローまたは特定のフォルダーのフローを表示または実行するための権限です。

特定の役割に割り当てられたユーザーは、その役割に割り当てられたコンテンツ権限に従ってフローにアクセスできます。

たとえば、[**管理者**] の役割を持つユーザーは、システム内のすべてのフローを表示および実行できますが、[**ユーザー**] の役割を持つユーザーは、特定のフローの実行と他のフローの表示のアクセス許可を付与される場合がありす。

一般的なセキュリティの概念

システムのセキュリティ

コンピューターベースの機器、情報、サービスが意図しないまたは認証されていないアクセス、変更、または損傷から保護するためのプロセスおよびメカニズム。

最小限の権限

通常の動作を許可する最小限のレベルに制限する方法。つまり、ユーザーアカウントにユーザーの作業に不可欠な権限だけを付与します。

認証

通常はユーザー名とパスワード、または証明書に基づいて個人を識別するプロセス。

権限

個人の ID に基づいたシステムオブジェクトへのアクセス許可。

暗号化

コンテンツにスクランブルをかけて、正しい暗号化キーを持っている人だけが読み取ってエンコードできるようにすることにより、メッセージやファイルのセキュリティを強化する方法。たとえば、TLS プロトコルは通信データを暗号化します。

対策

脅威リスクを低減する方法。

多層防御

保護層。1 つのセキュリティ対策だけに依存する必要はありません。

リスク

損傷の原因となる可能性があるイベント。たとえば、財務上の損失、企業イメージへのダメージなど。

脅威

脆弱性を利用したリスクイベントのトリガー。

脆弱性

セキュリティ脅威によって利用される可能性のあるターゲットの弱点。

安全な実装およびデプロイメント

デフォルトのセキュリティ設定

多くの場合、構成済みで提供されるデフォルトのセキュリティ設定は修正することをお勧めします。

- **認証** – Central で、認証はデフォルトでは有効になっていません。ユーザーのセットアップが完了したら、すぐに有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「認証の有効化」を参照してください。
- **監査** – Central で、監査はデフォルトでは有効になっていません。有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「監査の有効化」を参照してください。
- **TLS 暗号化** – HP OO は、デフォルトで 3 つの TLS プロトコル(1.0、1.1、1.2) をサポートしています。最新バージョンの使用をお勧めします。詳細については、「TLS プロトコルの構成」(58ページ)を参照してください。
- **TLS サーバー証明書** – デフォルトでは、HP OO サーバーのインストール時に、CA 証明書の提示がユーザーに求められます。
- **クライアント証明書** – クライアント証明書は、デフォルトでは有効になっていません。Central への認証には、クライアント証明書を使用することをお勧めします。詳細については、「クライアント証明書認証の構成 (Central)」(45ページ)を参照してください。
- **キーストア、信頼ストア、およびサーバー証明書のパスワード** – デフォルトでは、キーストア、信頼ストア、およびサーバー証明書用に Java パスワードが提供されています。これらのパスワードは、暗号化されたパスワードに置き換えることをお勧めします。詳細については、「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。
- **RC4 暗号** – RC4 暗号はデフォルトで有効になっています。RC4 暗号は JRE レベルで無効にすることをお勧めします。詳細については、「SSL サポート対象サイファーからの RC4 サイファーの削除」(43ページ)を参照してください。
- **セキュリティバナー** – Central で、セキュリティバナーはデフォルトでは有効になっていません。これは、カスタムメッセージを指定して、有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「セキュリティバナーのセットアップ」を参照してください。
- **データベースの Windows 認証** – Central で、Windows 認証はデフォルトでは有効になっていません。Windows および SQL サーバーの環境を使用する場合は、Windows 認証と連携するように HP OO を構成することをお勧めします。『HP OO データベースガイド』の「Windows 認証で稼働する HP OO の構成」を参照してください。
- **デフォルトのアルゴリズム** – **encryption.properties** ファイルにはデフォルトのアルゴリズムが含まれています。FIPS への準拠が必要な場合は、「HP OO での FIPS 140-2 レベル 1 互換の構成」(52ページ)を参照してください。FIPS 140-2 Level 1 のデフォルトの詳細については、「暗号化」(22ページ)の「暗号化管理」を参照してください。
- **Java ポリシー** – **java.policy** ファイルは、デフォルトではハードニングされていません。

java.policy ファイルの変更方法については、「フローが Central/RAS のローカルファイルシステムにアクセスできなくする」(58ページ)を参照してください。

HP OO のセキュリティハードニング

「ハードニング」の章には、HP OO デプロイメントをセキュリティのリスクや脅威から保護するための推奨事項が示されています。アプリケーションをセキュリティ保護する理由として最も重要なのは、組織の重要情報の機密性、整合性、可用性の保護です。

HP OO システムを包括的に保護するには、HP OO のセキュリティの保護とアプリケーションが実行されるコンピューティング環境 (インフラストラクチャーやオペレーティングシステムなど) のセキュリティ保護の両方が必要です。

「ハードニング」の章には、HP OO をアプリケーションレベルでセキュリティ保護するための推奨事項が示されています。ユーザー環境内のインフラストラクチャーをセキュリティ保護する方法はカバーしていません。使用するインフラストラクチャー/環境について理解し、それぞれのハードニングポリシーを適用するのは、もっぱらユーザーの責任です。

物理的セキュリティ

HP ソフトウェアは、組織が定義する物理的なセキュリティ管理によって HP OO を保護することをお勧めします。HP OO サーバーコンポーネントは、ベストプラクティスに従って、物理的にセキュリティ保護された環境にインストールされています。たとえば、サーバーはアクセス制御された密室に設置する必要があります。

セキュアなインストールに関するガイドライン

サポートされるオペレーティングシステム

サポートされるオペレーティングシステムのタイプおよびバージョンについては、『HP OO システム要件』を参照してください。

オペレーティングシステムのハードニングに関する推奨事項

オペレーティングシステムのハードニングの推奨されるベストプラクティスについては、オペレーティングシステムのベンダーに問い合わせてください。

例:

- パッチをインストールする必要があります。
- 不要なサービス/ソフトウェアは削除または無効にする必要があります。
- ユーザーには最小限のアクセス許可を割り当てる必要があります。
- 監査を有効にする必要があります。

Tomcat ハードニング

HP OO Central をインストールすると、デフォルトでは、Tomcat が部分的にハードニングされます。追加のハードニングが必要な場合は、「ハードニング」の章の推奨事項を参照してください。

インストール時のアクセス許可

HP OO をインストールして実行するには次のアクセス許可が必要:

HP OO のインストール	Windows/Linux:Java プロセスを実行できる、またフォルダーやサービスの作成するためのアクセス許可を持っている標準的なユーザー
HP OO の実行	• Windows:Windows サービスは、システムユーザーまたは特定のユーザーとして実行されます (ユーザーは HP OO インストールディレクトリにアクセスできる必要があります) • Linux:Java プロセスを実行できる標準的なユーザー

CIS Apache Tomcat 7.0 のドキュメントの推奨事項も参照してください。

ネットワークおよび通信のセキュリティ

『HP OO アーキテクチャーガイド』では、基本的な HP OO トポロジ、高可用性、ロードバランサーのセキュリティについて説明しています。

『HP OO Network Architecture White Paper』では、必要なファイアウォール構成を説明し、ポリシー制限によって必要なファイアウォール構成を実装できない場合に適用可能な 2 つの推奨される回避方法を提示しています:

- SSH リバーストンネリング
- リバースプロキシ

通信チャネルのセキュリティ

サポートされるプロトコルおよび構成

HP OO は TLS プロトコルをサポートしています。

詳細については、「Central TLS サーバー証明書の置き換え」(34ページ)を参照してください。

Central のポートは、インストール中に管理者によって定義されます。

チャネルのセキュリティ

HP OO は、次のセキュアなチャネルをサポートしています:

チャネル (ダイレクト)	サポートされるセキュアプロトコル
OOSH、ブラウザー、Studio リモートデバッガー、または RAS → Central	セキュアなチャネルでは、暗号化には TLS 通信を、認証にはクライアント証明書を使用します。
Central → LDAP サーバー	Central と LDAP の間の通信の暗号化には、TLS プロトコルを使用するセキュア LDAP を使用します。

管理インタフェースのセキュリティ

管理インタフェースへのアクセス

管理インタフェースへのアクセスを制御するにはいくつかの方法があります。

- **資格情報**
- **クライアント証明書**
- **SAML**

管理インタフェースのセキュリティ保護 - 推奨事項

1. Central での認証を有効にすることをお勧めします。

   『HP OO Central ユーザーガイド』の「認証の有効化」を参照してください。
2. TLS プロトコルを使用して管理インタフェースをセキュリティで保護することをお勧めします。クライアントと Central インタフェースの間の TLS を設定して暗号化する必要があります。

   「サーバーおよびクライアント証明書の使用」(33ページ)を参照してください。
3. LDAP ユーザーの方が安全なので、内部ユーザーより LDAP ユーザーを使用して作業することをお勧めします。
4. Client 証明書を使用して Central にアクセスするための認証を設定することをお勧めします。これは、ユーザーパスワードより安全です。

   「サーバーおよびクライアント証明書の使用」(33ページ)を参照してください。

ユーザーの管理および認証

認証モデル

HP OO で認証メカニズムのブートストラッピングを容易にするため、製品の認証は最初は無効になっています。

インストール後直ちに認証を有効にすることを強くお勧めします。

認証を有効にする方法については、『HP OO Central ユーザーガイド』の「認証の有効化」を参照してください。

Central へのアクセスを認証するにはいくつかの方法があります。

ユーザーの識別方法を選択します:

- ユーザー名とパスワード
- クライアント証明書
- SAML トークン
- シングルサインオン (HP LWSSO)

次のいずれかのユーザー管理方法を選択します:

- LDAP ユーザー: Active Directory として LDAP サーバーに保存 (推奨)
- 内部ユーザーおよびパスワード: Central サーバーにローカルに保存 (非推奨)

ユーザーのタイプ

ユーザーのタイプごとに異なるアクセス許可を割り当てることができます。たとえば、フロー作成者、管理者、システム管理者など。

異なるアクセス許可を必要とするこの他のタイプのユーザーの例については、『HP OO コンセプトガイド』の「主要なペルソナ」を参照してください。

認証の管理と構成

内部ユーザーまたは LDAP ユーザー

Central UI で内部ユーザーとパスワードを設定するか、LDAP サーバーでユーザーを定義して LDAP グループを Central の役割にマッピングすることができます。

注: 内部ユーザーを使用しせずに、LDAP ユーザーなど他のより安全なユーザーを使用することをお勧めします。

内部ユーザーの構成については、『HP OO Central ユーザーガイド』の「セキュリティのセットアップ – 内部ユーザー」を参照してください。

LDAP グループの Central の役割へのマッピングについては、『HP OO Central ユーザーガイド』の「セキュリティのセットアップ - LDAP 認証」および『HP OO API Guide』の「LDAP Configuration」を参照してください。

SAML/クライアント証明書/LW SSO

Central で SAML が動作するように構成するには、『HP OO Central ユーザーガイド』の「セキュリティのセットアップ - SAML」を参照してください。

Central でクライアント証明書が動作するように構成するには、「サーバーおよびクライアント証明書の使用」(33ページ)を参照してください。

Central で LW SSO が動作するように構成するには、『HP OO Central ユーザーガイド』の「セキュリティのセットアップ - LW SSO」、『HP OO Administration Guide』の「Configuring LWSSO Settings」『HP OO API Guide』の「LW SSO」を参照してください。

データベースの認証

HP OO は 4 つのデータベースをサポートしています:Oracle、MS SQL、MySQL、Postgres。

データベース認証用の強いデータベースパスワードと強いパスワードポリシーを使用することをお勧めします。たとえば、何度も試行に失敗したらブロックします。

MS SQL を使用している場合は、データベース認証か OS 認証を使用できます。可能であれば、OS 認証を使用することをお勧めします。たとえば、Microsoft SQL Server データベースへのアクセスには Windows 認証を使用できます。

- OS 認証の設定については、『HP OO データベースガイド』の「Windows 認証で稼働する HP OO の構成」を参照してください。
- 『HP OO Administration Guide』を参照してください。
- データベースベンダーが推奨するベストプラクティス (存在する場合) を参照してください。

権限

権限モデル

HP OO リソースへのユーザーアクセス権は、ユーザーの役割、およびその役割に対して設定されているアクセス許可に基づいて付与されます。

参照:

- 『HP OO Central ユーザーガイド』の「セキュリティのセットアップ – 役割」
- 『HP OO Central ユーザーガイド』の「システムアカウントへのアクセス許可の割り当て」

最小限のアクセス許可に関するガイドライン

推奨事項:

- 役割に適切なアクセス許可を選択します。
- 役割の作成時には最小限のアクセス許可を使用します。
- 最小限のアクセス許可を付与し、必要な場合にだけアクセス許可を拡大して、不必要な権限のエスカレーションを回避します。たとえば、ビューアーのアクセス許可から始め、必要に応じて個別にアクセス許可を追加します。

権限の構成

Central には多数の設定済みの役割がインストールされているので、構成してユーザーに割り当てることができます。デフォルトでは、設定済みの役割には次のアクセス許可が割り当てられています。

役割	デフォルトのアクセス許可
Administrator	すべて
End_user	なし
Everybody	なし
Promoter	すべての**コンテンツ**のアクセス許可
System_admin	すべての**システム**のアクセス許可

デフォルトの役割

デフォルト役割に関する属性を使用して、役割の 1 つを設定することができます。その場合は、最小限の権限を持つ役割にしてください。この役割にアクセス許可を付与する場合は、この役割に明示的に関連付けられているユーザーだけでなく、すべての LDAP ユーザーに影響することに留意してください。

詳細については、『HP OO Central ユーザーガイド』の「セキュリティのセットアップ - 役割」の「デフォルトの役割としての役割の割り当て」を参照してください。

以下も参照:

- 『HP OO Central ユーザーガイド』の「システムアカウントへのアクセス許可の割り当て」
- 『HP OO Central ユーザーガイド』の「コンテンツアクセス許可の設定」

バックアップ

データの損失を防ぐため、セキュアなメディアにサーバーのデータを定期的にバックアップすることを強くお勧めします。これは、ディザスターリカバリやビジネスの継続にも役立ちます。

HP OO をインストールしたら、**central\var\security** フォルダーと **central\conf\database.properties** ファイルを必ずバックアップしてください。

データベーススキーマでは、一部のデータが暗号化され、復号化キーは HP OO Central サーバーにローカルに保存されています。システムファイルが破損または削除されるとデータの復号化が不可能になるので、スキーマは使用できなくなります。

注: キーは暗号化されているので、キーをバックアップに含めることが重要です。上記のスクリプトは **bin** フォルダーにあります。

参照:

- 『HP OO Administration Guide』の「Backing Up HP OO」
- 『HP OO Administration Guide』の「Setting up Disaster Recovery」
- 『HP OO インストールガイド』の「Central セキュリティファイルのバックアップと復元」
- 『HP OO アーキテクチャーガイド』の「HP OO デプロイメントでのロードバランサーの使用」

暗号化

暗号化モデル

HP OO は、機密データを保護するために、暗号化アルゴリズムとハッシュアルゴリズムをサポートしています。暗号化は、HP OO システムのパスワードや定義などの機密データの漏洩および変更を防ぐように設計されています。

認証されていないユーザーによる復号化を防ぐためには、既知の脆弱性がないよく知られている標準的なアルゴリズムを使用することが重要です。

注: たとえば、SSL プロトコルには既知の脆弱性があるため、SSL は使用されません。

静的データ

保存されているすべてのパスワードがよく知られているアルゴリズムを使用して保護されており、クリアテキストのままのパスワードはありません。

例:

- システムアカウントのパスワードは暗号化されています。
- 内部ユーザーのパスワードはハッシュされています。
- データベースパスワードは暗号化されています。

転送中のデータ

HP OO は、トランスポートレイヤーセキュリティ (TLS) プロトコルを使用して、コンポーネント (Central や RAS など) 間のデータを暗号化します。

HTTP ポートの無効化

セキュリティ上の理由から、HTTP ポートを無効にして、TLS 上にある暗号化されたチャネルを唯一の通信チャネルにすることをお勧めします。詳細については、「HTTP/HTTPS ポートの変更または HTTP ポートの無効化」(43ページ)を参照してください。

暗号化の管理

推奨される暗号化のベストプラクティス

セキュリティレベルおよび暗号化レベルを高めるためには、HP OO を Federal Information Processing Standards (FIPS) 140-2 互換に構成することをお勧めします。HP OO を FIPS 140-2 レベル 1 互換に設定できます。

デフォルトの構成セット

- 対称キーアルゴリズム: AES (キー長: 128)
- ハッシュアルゴリズム: SHA1

詳細設定

HP OO で FIPS 140-2 互換の構成を行うと、HP OO は次のセキュリティアルゴリズムを使用します。

- 対称キーアルゴリズム: AES256
- ハッシュアルゴリズム: SHA256

「HP OO での FIPS 140-2 互換の構成」(54ページ)を参照してください。

デジタル証明書

デジタル証明書は、ユーザー、サーバー、ステーションなどの電子「パスポート」です。

- ブラウザーと Central サーバーの間で暗号化を使用するには、サーバー側にデジタル証明書をインストールする必要があります。
- Central サーバーの認証にクライアント証明書を使用するには、クライアント側 (たとえば、ブラウザー上の RAS、OOSH、Studio など) にクライアント証明書をインストールする必要があります。

HP OO では、Java Keytool ユーティリティを使用して暗号キーと信頼された証明書を管理します。このユーティリティは、HP OO のインストールフォルダー (**<インストールディレクトリ>/java/bin/keytool**) に含まれています。

証明書の場所

HP OO Central のインストールには、次の 2 つの証明書管理用ファイルが含まれています。

- **<インストールディレクトリ>/central/var/security/client.truststore**:信頼される証明書のリストが含まれています。
- **<インストールディレクトリ>/central/var/security/key.store**:HP OO プライベート証明書 (秘密キーを含む) が含まれています。

キーストアおよび信頼ストアへのアクセス制御

信頼ストアおよびキーストアの保存では、Central サービスを実行するユーザーに対してのみ読み取りアクセス許可を付与することをお勧めします。

HP OO 自己署名証明書の置き換え

HP OO を新規にインストールした場合や現在の証明書の有効期限が切れた場合は、HP OO 自己署名証明書を置き換えることをお勧めします。

証明書の置き換えプロセスの一環で、PKCS12 形式の証明書が CA を使用して作成されます。証明書プロセスの詳細については CA にお問い合わせください。または、コーポレートポリシーを参照してください。

詳細については、「Central TLS サーバー証明書の置き換え」(34ページ)を参照してください。

デジタル署名のコンテンツパックへの追加

コンテンツパックに信頼された CA のデジタル署名が付いている場合は、コンテンツは信頼できます。

デジタル署名の追加は必須ではありません。

- HP OO 設定済みのコンテンツパックには、Verisign のデジタル署名が含まれています。
- HP OO の作成者には、カスタムコンテンツパックにデジタル署名を追加することをお勧めします。
- 署名済みのコンテンツパックが破壊されている場合は、デプロイできません。
- 署名の有効期限が切れた場合は、デプロイ前に警告が表示されるので、期限切れの署名を無視することを確認するチェックボックスを選択する必要があります。

署名されていないコンテンツパックに注意してください。未署名のコンテンツパックは信頼できず、悪意のあるコンテンツが含まれている可能性があります。未署名のコンテンツパックは破壊され、署名が削除されている可能性があることにも注意してください。

コンテンツパックのデジタル証明書の詳細については、『HP OO Central ユーザーガイド』の「コンテンツパックのデプロイと管理」を参照してください。

コンテンツパックの機密情報

システムアカウントのパスワード

コンテンツパックの作成時にパスワードを含めないでください。パスワードはコンテンツパック内部で暗号化されるので、セキュアなオプションではありません。

HP OO のセキュリティに関するベストプラクティスは、Central でシステムアカウントのパスワードを設定することです。詳細については、『HP OO Central ユーザーガイド』の「コンテンツパックのシステムアカウントのセットアップ」を参照してください。

監査とログファイル

監査

監査を行うと、Central サーバーで行われるアクション (ログイン、フローの起動、スケジュールの作成、構成の編集など) を追跡できます。監査データによって、Central システム上のユーザー操作を追跡して、誰が何の操作をいつ行ったか追跡することができます。たとえば、監査によって、ユーザーによるフローの実行、構成の更新、スケジュールの削除、または認証の失敗を確認できます。

監査データはデータベースに保存されます。詳細については、『HP OO API Guide』の「Auditing」を参照してください。

ログ

ログによって、エラー、警告、情報、デバッグメッセージをトレースできます。

ログはファイルサーバーの次の場所に保存されます:

- Central - **<oo インストールフォルダー>/central/var/logs**
- Studio - **<ユーザー>/.oo/logs**
- RAS - **<oo インストールフォルダー>/ras/var/logs**

監査レコードやログファイルに保存される機密データなし

HP OO システムでは、機密データは監査レコードやログファイルに保存されません。

監査レコードの取得

監査レコードは、API 経由で、または OO_AUDIT テーブルへのクエリによって取得できます。詳細については、『HP OO API Guide』の「Auditing」を参照してください。

監査データの例:

```
[
{
“time”:1412312016740, “type”:”AuditConfigurationChange”,
“group”:”AuditManagement”, “subject”:” mydomain\myuser2”, “outcome”:”Success”,
“data”:”{"enabled":false}”
},
{
“time”:1412312016722, “type”:”InternalUserDelete”, “group”:”Authentication-
Authorization”, “subject”:”mydomain\myuser2”, “outcome”:”Success”, “data”:
{"usersNames":["admin"]}"
}
]
```

API とインタフェース

API モデルとインタフェースモデル

HP OO Central の UI ではなく、HP Operations Orchestration のパブリック Application Programming Interfaces (API) を使用して作業して、同じ操作を実行することができます。削除や監査などの一部の操作は、API 経由でのみ実行できます。パブリック API は HTTP ベースです。すべての API が RESTful で、JavaScript Object Notation を使用します。

API とインタフェースのセキュリティ構成の機能と管理

API を使用してセキュアに作業することが重要です。API を使用して作業している間は、このガイドに記載されているセキュリティメカニズム (認証、暗号化など) を使用してください。

API インタフェースは、HTTP または HTTPS 上で動作します。

注: API を使用して HTML を表示する場合は、XSS 攻撃から保護する必要があります。

詳細については、『HP OO API Guide』の以下の章を参照してください。

- 「LDAP Configuration」
- 「Users」
- 「LW SSO Configuration」
- 「Authentication」
- 「Roles」

セキュリティに関する Q&A

外部 CA による署名が可能な証明書要求の生成方法は?

証明書要求をエクスポートして外部 CA に送って署名してもらいます。手順については、「Central TLS サーバー証明書の置き換え」(34ページ)を参照してください。

HP OO を使用するのはどの TCP/UDP ポートですか?方向、ユーザー、暗号化とは?

HP OO をインストールする場合は、[HTTP/HTTPS] フィールドで、Central サーバーに使用可能な少なくとも 1 つのポートを構成する必要があります。デフォルト値は 8080 および 8443 ですが、変更可能です。Central と他のコンポーネントの間のセキュアなチャネルの詳細については、「ネットワークおよび通信のセキュリティ」(15ページ)を参照してください。

資格情報はどこにどのように保存されますか (管理者アカウント、統合ユーザー)?

「ユーザーの管理および認証」(17ページ)を参照してください。

Central/RAS/Studio の自己署名 SSL 証明書の構成方法は?

HP OO のインストール中に証明書を提示しないと、自己証明書がデフォルトで作成されます。ただし、セキュリティ上の理由から、自己署名証明書は使用しないようにしてください。HP は、カスタムルート CA または知名度の高い CA の証明書の使用をお勧めします。

HP OO の証明書の構成の詳細については、「サーバー証明書を使用した通信の暗号化」(34ページ)を参照してください。

監査を有効または無効にする方法は?

デフォルトでは、監査は有効になっていません。監査を有効にする方法の詳細については、『HP OO Central ユーザーガイド』の「監査の有効化」を参照してください。監査の詳細については、「監査とログファイル」(26ページ)を参照してください。

どの程度詳細なログか? またログ量の変更方法は?

ログはさまざまなレベルの詳細度に設定できます。デフォルトレベルは INFO ですが、調整可能です。詳細については、『HP OO Administration Guide』の「Adjusting the Logging Levels」を参照してください。

ログファイルの詳細については、「監査とログファイル」(26ページ)を参照してください。

機密情報の暗号化方法は?

「暗号化」(22ページ)を参照してください。

Central と RAS の間の通信は暗号化されていますか?

HTTPS を使用している場合は、暗号化されています。

HP OO と他の統合コンポーネント(HPNA、CSA、AD など) の間の通信は暗号化されていますか?

これは、使用している統合に依存します。HTTPS を使用している場合は、暗号化されています。

フローライブラリへのアクセスをユーザーの役割に基づいて制限する方法は?

『HP OO Central ユーザーガイド』の「セキュリティのセットアップ – 役割」を参照してください。

HP OO がサポートしている認証メカニズムは?

サポートされている認証メカニズムは、LDAP、SAML、内部ユーザーです。HP OO は、クライアント証明書と LW SSO もサポートしています。「ユーザーの管理および認証」(17ページ)を参照してください。

HP OO は FIPS 140-2 互換ですか?

はい。詳細については、「HP OO での FIPS 140-2 互換の構成」(54ページ)を参照してください。

Central と RAS の間の認証方法は?

ユーザーパスワードまたはクライアント証明書。

すべてのパスワードが暗号化されて保存またはハッシュされますか?

はい。保存されているすべてのパスワードがよく知られているアルゴリズムを使用して保護されており、クリアテキストのままのパスワードはありません。

Central ユーザーの IP アドレスを制限できますか?

いいえ。現時点ではサポートされていません。

HP OO はコモンクライテリアの認証を受けていますか?

これは進行中です。現在、「評価段階」にあります。詳細については、https://www.cse-cst.gc.ca/en/canadian-common-criteria-scheme/publication/list/evaluation-product を参照してください。

OOSH を使用した場合、機密データを Central に渡すことはできますか?

Central への接続時にはセキュアなチャネルの使用をお勧めします。「ネットワークおよび通信のセキュリティ」(15ページ)を参照してください。

HP Operations Orchestration のハードニング

このセクションでは、HP Operations Orchestration のセキュリティハードニングの構成方法について説明します。

注: 管理作業については、『HP OO インストール、アップグレード、構成ガイド』を参照してください。

セキュリティハードニングの推奨事項

1. 最新バージョンの HP OO をインストールします。詳細については、『HP OO インストール、アップグレード、構成ガイド』を参照してください。
2. (オプション) HP OO に FIPS 140-2 互換を構成します。これを行う場合は、Central サーバーを起動する前に構成する必要があります。「HP OO での FIPS 140-2 レベル 1 互換の構成」(52ページ)を参照してください。
3. Central サーバー証明書で TLS 暗号化を構成し、クライアント証明書で強い認証 (相互) を構成します。

   注: これは、インストール時に実行できます。

   RAS、デバッガー、および OOSH について、(サーバー証明書に) 必要であれば、証明書認証を提供し、Central に対する認証でクライアント証明書を使用します。「サーバーおよびクライアント証明書の使用」(33ページ)を参照してください。
4. HTTP ポートを削除し、キーストアと信頼ストアのパスワードを強いパスワードに置き換えて、HP OO Central サーバーをハードニングします。「HTTP/HTTPS ポートの変更または HTTP ポートの無効化」(43ページ)および「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。
5. キーストアと信頼ストアのパスワードを強いパスワードに置き換えて、HP OO Studio をハードニングし、構成ファイルのパスワードを暗号化または難読化します。「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。
6. SSL サポート対象サイファーから RC4 サイファーを削除します。「SSL サポート対象サイファーからの RC4 サイファーの削除」(43ページ)を参照してください。
7. (オプション) TLS プロトコルのバージョンを設定します。「TLS プロトコルの構成」(58ページ)を参照してください。
8. Central での認証を有効にします。『HP OO Central ユーザーガイド』の「認証の有効化」を参照してください。

   内部ユーザーはセキュリティで保護されていないため、セキュアな LDAP と強いパスワードポリシーを使用してください。『HP OO Central ユーザーガイド』の「セキュリティのセットアップ - LDAP 認証」を参照してください。

9. オペレーティングシステムとデータベースのハードニング/セキュリティ保護を行います。
10. わかりやすいメッセージのセキュリティバナーを追加します。たとえば、「実稼働環境にログオンしようとしています。当システムの管理ルールを理解していないユーザーはログオンする前に必要なトレーニングを受けてください」というバナーを作成することができます。『HP OO Central ユーザーガイド』の「セキュリティバナーのセットアップ」を参照してください。
11. Windows および SQL サーバーの環境で、HP OO が Windows 認証と連携するように構成します。『HP OO データベースガイド』の「Windows 認証で稼働する HP OO の構成」を参照してください。
12. Central で監査が有効なことを確認します。詳細については、『HP OO Central ユーザーガイド』の「監査の有効化」を参照してください。

デフォルトのセキュリティ設定

多くの場合、構成済みで提供されるデフォルトのセキュリティ設定は修正することをお勧めします。

- **認証** – Central で、認証はデフォルトでは有効になっていません。ユーザーのセットアップが完了したら、すぐに有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「認証の有効化」を参照してください。
- **監査** – Central で、監査はデフォルトでは有効になっていません。有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「監査の有効化」を参照してください。
- **TLS 暗号化** – HP OO は、デフォルトで 3 つの TLS プロトコル(1.0、1.1、1.2) をサポートしています。最新バージョンの使用をお勧めします。詳細については、「TLS プロトコルの構成」(58ページ)を参照してください。
- **TLS サーバー証明書** – デフォルトでは、HP OO サーバーのインストール時に、CA 証明書の提示がユーザーに求められます。
- **クライアント証明書** – クライアント証明書は、デフォルトでは有効になっていません。Central への認証には、クライアント証明書を使用することをお勧めします。詳細については、「クライアント証明書認証の構成 (Central)」(45ページ)を参照してください。
- **キーストア、信頼ストア、およびサーバー証明書のパスワード** – デフォルトでは、キーストア、信頼ストア、およびサーバー証明書用に Java パスワードが提供されています。これらのパスワードは、暗号化されたパスワードに置き換えることをお勧めします。詳細については、「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。
- **RC4 暗号** – RC4 暗号はデフォルトで有効になっています。RC4 暗号は JRE レベルで無効にすることをお勧めします。詳細については、「SSL サポート対象サイファーからの RC4 サイファーの削除」(43ページ)を参照してください。
- **セキュリティバナー** – Central で、セキュリティバナーはデフォルトでは有効になっていません。これは、カスタムメッセージを指定して、有効にすることをお勧めします。詳細については、『HP OO Central ユーザーガイド』の「セキュリティバナーのセットアップ」を参照してください。

- **データベースの Windows 認証** – Central で、Windows 認証はデフォルトでは有効になっていません。Windows および SQL サーバーの環境を使用する場合は、Windows 認証と連携するように HP OO を構成することをお勧めします。『HP OO データベースガイド』の「Windows 認証で稼働する HP OO の構成」を参照してください。
- **デフォルトのアルゴリズム** – **encryption.properties** ファイルにはデフォルトのアルゴリズムが含まれています。FIPS への準拠が必要な場合は、「HP OO での FIPS 140-2 レベル 1 互換の構成」(52 ページ)を参照してください。FIPS 140-2 Level 1 のデフォルトの詳細については、「暗号化」(22 ページ)の「暗号化管理」を参照してください。
- **Java ポリシー** – **java.policy** ファイルは、デフォルトではハードニングされていません。**java.policy** ファイルの変更方法については、「フローが Central/RAS のローカルファイルシステムにアクセスできなくする」(58ページ)を参照してください。

サーバーおよびクライアント証明書の使用

トランスポートレイヤーセキュリティ (TLS) 証明書は、暗号キーを組織の詳細にデジタル的に結び付けます。これにより、Web サーバーからブラウザーへの暗号化されたセキュアな接続が可能になります。

HP OO では、Keytool ユーティリティを使用して暗号キーと信頼された証明書を管理します。このユーティリティは、HP OO のインストールフォルダー (**<インストールディレクトリ>/java/bin/keytool**) に含まれています。Keytool ユーティリティの詳細については、http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html を参照してください。

注: Keytool はオープンソースのユーティリティです。

HP OO Central のインストールには、次の 2 つの証明書管理用ファイルが含まれています。

- **<インストールディレクトリ>/central/var/security/client.truststore**: 信頼される証明書のリストが含まれています。
- **<インストールディレクトリ>/central/var/security/key.store**: HP OO 証明書 (秘密キー) が含まれています。

推奨事項:

- HP OO を新規にインストールした場合や現在の証明書の有効期限が切れた場合は、HP OO 自己署名証明書を置き換えることをお勧めします。
- 信頼ストアとキーストアは、Central サービスを実行するユーザーのみに対する読み取り権限で格納することをお勧めします。
- Keytool の使用後はコンソールをクリアするか、パスワード入力のプロンプトを使用することをお勧めします。

サーバー証明書を使用した通信の暗号化

Central TLS サーバー証明書の置き換え

よく知られている証明機関によって署名された証明書か、ローカル証明機関のカスタムサーバー証明書を使用することができます。

key.store ファイルやコンピューターの設定に合わせて、<黄色> でハイライトされているパラメーターを置換します。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. Central を停止し、**<インストールディレクトリ>/central/var/security/key.store** にある **key.store** ファイルをバックアップします。
2. **<インストールディレクトリ>/central/var/security** でコマンドラインを開きます。
3. 次のコマンドを使用して、Central の **key.store** ファイルから既存のサーバー証明書を削除します。

   ```
   keytool -delete -alias tomcat -keystore key.store -storepass changeit
   ```

4. 拡張子が **.pfx** または **.p12** の証明書がすでに存在する場合は、次の手順に進みます。存在しない場合は、秘密キー付きの証明書を PKCS12 形式 (.pfx,.p12) にエクスポートします。たとえば、証明書の形式が PM の場合、次のようになります。

   ```
   >openssl pkcs12 -export -in <cert.pem> -inkey <.key> -out <証明書名>.p12 -name <名前>
   ```

   証明書の形式が DER の場合、次のように、`-inform DER` パラメーターを pkcs12 の後に追加します。

   ```
   >openssl pkcs12 -inform DER -export -in <cert.pem> -inkey <.key> -out <証明書名>.p12 -name <名前>
   ```

注:

PKCS12 形式の証明書を生成するには CA を使用する必要があります。その手順は CA ベンダーとポリシーによって異なる可能性があるため、CA に連絡し、証明書の生成プロセスに関する詳細な説明を求める必要があります。

注: パスワードを記録しておいてください。この秘密キーのパスワードは、後の手順でキーストアのパスフレーズ入力で使用します。

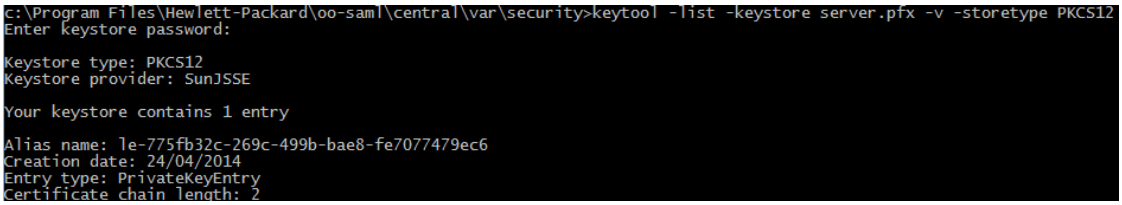
必ず、強いパスワードを選択してください。

5. 次のコマンドを使用して証明書のエイリアスをリストします。

   ```
   keytool -list -keystore <証明書名> -v -storetype PKCS12
   ```

   証明書のエイリアスが表示されます。このエイリアスは、この次のコマンドで入力します。

   次の例では、下から 4 番目の行です。

6. 次のコマンドを使用して、PKCS12 形式のサーバー証明書を Central の **key.store** ファイルにインポートします。

   ```
   keytool -importkeystore -srckeystore <PKCS12 形式の証明書のパス> -destkeystore key.store -srcstoretype pkcs12 -deststoretype JKS -alias <証明書のエイリアス> -destalias tomcat
   ```

7. インポートしたサーバー証明書のパスワードが元のサーバー証明書と異なる場合は、keyPass パスワードを変更することが重要です。「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)の手順を実行してください。

   Central サーバーの自動生成されたキーストア内のデフォルトの "changeit" パスワードを変更することをお勧めします。「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。

8. Central を起動します。

Central の信頼ストアへの CA ルート証明書のインポート

Central でカスタムルート証明書を使用する場合、信頼されたルート証明機関 (CA) を **client.truststore** にインポートする必要があります。よく知られているルート CA (Verisign など) を使用する場合、証明書はすでに **client.truststore** ファイルに登録されているので、以下の手順を実行する必要はありません。

デフォルトで、HP OO はすべての自己署名証明書をサポートします。ただし、実稼働環境では、セキュリティ上の理由から、このデフォルトをカスタム CA またはよく知られている CA に変更することをお勧めします。

<黄色> でマークされているパラメーターを置き換えます。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. Central を停止し、**<インストールディレクトリ>/central/var/security/client.truststore** にある **client.truststore** ファイルをバックアップします。
2. 信頼されたルート証明機関 (CA) が CA リスト内にまだない場合は、Central の **client.truststore** ファイルにインポートします (デフォルトでは、よく知られているすべての CA がリストにあります)。

   ```
   keytool -importcert -alias <任意のエイリアス> -keystore <path to the client.truststore> -file <証明書名.cer> -storepass <changeit>
   ```

3. Central を起動します。

RAS 信頼ストアへの CA ルート証明書のインポート

RAS のインストール後、Central でカスタムルート証明書を使用し、RAS のインストール時にこのルート証明書を提示しなかった場合、信頼されたルート証明機関 (CA) を RAS **client.truststore** にインポートする必要があります。よく知られているルート CA (Verisign など) を使用する場合、証明書はすでに **client.truststore** ファイルに登録されているので、以下の手順を実行する必要はありません。

デフォルトで、HP OO はすべての自己署名証明書をサポートします。ただし、実稼働環境では、セキュリティ上の理由から、このデフォルトをカスタム CA またはよく知られている CA に変更することをお勧めします。

<黄色> でマークされているパラメーターを置き換えます。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. RAS を停止し、**<インストールディレクトリ>/ras/var/security/client.truststore** にある元の **client.truststore** ファイルをバックアップします。
2. **<インストールディレクトリ>/ras/var/security** でコマンドラインを開きます。
3. **<インストールディレクトリ> ras/conf/ras-wrapper.conf** ファイルを開き、`-Dssl.support-self-signed` の値が **false** に設定されていることを確認します。これにより、信頼されたルート証明機関 (CA) が有効になります。

   例:

   ```
   wrapper.java.additional.<x>=-Dssl.support-self-signed=false
   ```

4. **<インストールディレクトリ> ras/conf/ras-wrapper.conf** ファイルを開き、-

Dssl.verifyHostName の値が **true** に設定されていることを確認します。 これにより、証明書内の FQDN が、要求の FQDN に一致することが検証されます。

例:

```
wrapper.java.additional.<x>=-Dssl.verifyHostName=true
```

注: このプロパティは、デフォルトで **true** に設定されています。

5. 信頼されたルート証明機関 (CA) が CA リスト内にまだない場合は、RAS の **client.truststore** ファイルにインポートします (デフォルトでは、よく知られているすべての CA がリストにあります)。

```
keytool -importcert -alias <任意のエイリアス> -keystore <client.truststore へのパス>
-file <証明書名.cer> -storepass <changeit>
```

6. RAS を起動します。

OOSH 信頼ストアへの CA ルート証明書のインポート

Central でカスタムルート証明書を使用する場合、信頼されたルート証明機関 (CA) を OOSH **client.truststore** にインポートする必要があります。よく知られているルート CA (Verisign など) を使用する場合、証明書はすでに **client.truststore** ファイルに登録されているので、以下の手順を実行する必要はありません。

デフォルトで、HP OO はすべての自己署名証明書をサポートします。ただし、実稼働環境では、セキュリティ上の理由から、このデフォルトをカスタム CA またはよく知られている CA に変更することをお勧めします。

<黄色> でマークされているパラメーターを置き換えます。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. Central を停止し、**<インストールディレクトリ>/central/var/security/client.truststore** にある **client.truststore** ファイルをバックアップします。
2. **<インストールディレクトリ>/central/bin** にある **oosh.bat** を編集します。
3. -Dssl.support-self-signed の値が **false** に設定されていることを確認します。これにより、信頼されたルート証明機関 (CA) が有効になります。

   例:

   ```
   -Dssl.support-self-signed=false
   ```

4. -Dssl.verifyHostName が **true** に設定されていることを確認します。 これにより、証明書内の FQDN が、要求の FQDN に一致することが検証されます。

   例:

   ```
   -Dssl.verifyHostName=true
   ```

注: このプロパティは、デフォルトで **true** に設定されています。

5. 信頼されたルート証明機関 (CA) が CA リスト内にまだない場合は、Central の **client.truststore** ファイルにインポートします (デフォルトでは、よく知られているすべての CA がリストにあります)。

   ```
   keytool -importcert -alias <任意のエイリアス> -keystore <client.truststore へのパス>
   -file <証明書名.cer> -storepass <changeit>
   ```

6. OOSH を実行します。
7. Central を起動します。

Studio デバッガー信頼ストアへの CA ルート証明書のインポート

Studio のインストール後、Studio でカスタムルート証明書を使用する場合、信頼されたルート証明機関 (CA) を Studio **client.truststore** にインポートする必要があります。よく知られているルート CA (Verisign など) を使用する場合、証明書はすでに **client.truststore** ファイルに登録されているので、以下の手順を実行する必要はありません。

デフォルトで、HP OO はすべての自己署名証明書をサポートします。ただし、実稼働環境では、セキュリティ上の理由から、このデフォルトをカスタム CA またはよく知られている CA に変更することをお勧めします。

新規の **.oo** フォルダーの場合、Studio は **<インストールディレクトリ>/studio/var/security** の **client.truststore** ファイルを **<ユーザー>/.oo** フォルダーにコピーします。これは、Studio で (たとえば、Studio リモートデバッガーの) 証明書を自動的にインポートできるようにするために、一度だけ行われる操作です。このファイルが存在する場合は、それが **client.truststore** として使用され、存在しない場合は Studio インストールのファイル (**<インストールディレクトリ>/studio/var/security/client.truststore**) が使用されます。

証明書を手動でインポートする場合は、**.oo/client.truststore** または Studio インストールフォルダーの **client.truststore** のいずれかにコピーできます。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. Studio を閉じます。インストールフォルダーの **client.trustore** ファイルにインポートする場合は、元のファイルをバックアップします。
2. **<インストールディレクトリ>/studio** にある **Studio.l4j.ini** ファイルを編集します。
3. `-Dssl.support-self-signed` の値が **false** に設定されていることを確認します。これにより、信頼されたルート証明機関 (CA) が有効になります。

   例:

   ```
   -Dssl.support-self-signed=false
   ```

4. `-Dssl.verifyHostName` が **true** に設定されていることを確認します。 これにより、証明書内の FQDN が、要求の FQDN に一致することが検証されます。

   例:

   ```
   -Dssl.verifyHostName=true
   ```

5. 信頼されたルート証明機関 (CA) が CA リスト内にまだない場合は、Studio の **client.truststore** ファイルにインポートします (デフォルトでは、よく知られているすべての CA がリストにあります)。<黄色> でマークされているパラメーターを置き換えます。

   ```
   keytool -importcert -alias <任意のエイリアス> -keystore <client.truststore へのパス>
   -file <証明書名.cer> -storepass <changeit>
   ```

6. Studio を起動します。

詳細については、『Studio オーサリングガイド』の「リモート Central の Studio でのデバッグ」を参照してください。

キーストア/信頼ストアのパスワードの変更と暗号化/難読化

Central 構成のキーストア、信頼ストア、およびサーバー証明書のパスワードの変更

1. Central が実行中であることを確認します。

   注: このステップを実行する前に、暗号化されたパスワードが存在することを確認します。パスワードを暗号化する方法については、『HP OO インストール、アップグレード、構成ガイド』の「Encrypting Passwords」を参照してください。

   OOSH から、次のコマンドを実行します。

   ```
   set-sys-config --key <キー名> --value <暗号化されたパスワード>
   ```

   ここで、<キー名> は、次の表のいずれかの値です。

構成アイテム	操作
`key.store.password`	**key.store** へのアクセスに使用するパスワードを設定します。デフォルト値は `"changeit"` です。 これは、下の手順で設定する `keystorePass` の値に対応している必要があります。

key.store.private.key.alias.password	**key.store** からサーバー証明書 (プライベートキー) にアクセスするために使用するパスワードを設定します。デフォルト値は "changeit" です。 これは、下の手順で設定する keyPass の値に対応している必要があります。

2. Central サービスを停止します。
3. Keytool を使用して、キーストア、信頼ストア、およびサーバー証明書のパスワードを変更します。

   キーストアのパスワードを変更するには、次の keytool コマンドを使用します。

   ```
   keytool -storepasswd -keystore <インストールフォルダー
   >/central/var/security/key.store
   ```

   サーバー証明書の秘密キーエントリパスワードを変更するには、次の keytool コマンドを使用します。

   ```
   keytool –keypasswd -alias tomcat -keystore <インストールフォルダー
   >/central/var/security/key.store
   ```

   信頼ストアのパスワードを変更するには、次の keytool コマンドを使用します。

   ```
   keytool -storepasswd -keystore <インストールフォルダー
   >/central/var/security/client.truststore
   ```

4. **<インストールディレクトリ>/central/tomcat/conf/** にある **server.xml** ファイルでもパスワードを編集します。
   a. HTTPS コネクターを検索します。例:

   ```
   keyPass="changeit" keystoreFile="C:/Program Files/Hewlett-Packard/HP
   Operations Orchestration/central/var/security/key.store"
   keystorePass="changeit" keystoreType="JKS" maxThreads="200" port="8443"
   protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
   secure="true" sslProtocol="TLSv1.2"
   sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2" truststoreFile="C:/Program
   Files/Hewlett-Packard/HP Operations
   Orchestration/central/var/security/client.truststore"
   truststorePass="changeit" truststoreType="JKS"/>
   ```

   パスワードを変更します。

   - keyPass - 指定する key.store ファイルのサーバー証明書の秘密キーにアクセスする際に使用するパスワード。デフォルト値は "changeit" です。
   - keystorePass - 指定する key.store ファイルへのアクセスに使用するパスワード。デフォルト値は keyPass 属性の値です。

     注: keyPass と同じパスワードを使用すること、および強いパスワードを使用することをお勧めします。

- truststorePass - (信頼されているすべての CA を含む) 信頼ストアにアクセスするためのパスワード。デフォルト値は **javax.net.ssl.trustStorePassword** システムプロパティの値です。このプロパティが null の場合、信頼ストアのパスワードは設定されません。信頼ストアのパスワードに無効な値が指定されると、警告がログに記録され、パスワードなしで信頼ストアにアクセスします。信頼ストアの内容の検証は省略されます。

b. ファイルを保存します。

5. **<インストールディレクトリ> central\conf\central** にある **central-wrapper.conf** ファイルを編集して、信頼ストアのパスワードを、暗号化または難読化した形式の新しいパスワードに置き換えます。例:

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.trustStorePassword={ENCRYPTED}
   <encrypted_password>
   ```

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.trustStorePassword={OBFUSCATED}
   <obfuscated_password>
   ```

   パスワードを暗号化する方法については、「パスワードの暗号化と難読化」(41ページ)を参照してください。

6. Central サービスを起動します。

RAS、OOSH、および Studio の信頼ストアのパスワードの変更

注: 以下の手順を実行する前に、Keytool を使用して、キーストア、信頼ストア、およびサーバー証明書のパスワードを変更してください。

- **スタンドアロンの RAS 信頼ストアのパスワードを変更するには、次の手順を実行します。ras-wrapper.conf** ファイルを編集し、信頼ストアの password パラメーターを変更します。
- **OOSH 信頼ストアのパスワードを変更するには、次の手順を実行します。oosh.bat** ファイルを編集し、信頼ストアの password パラメーターを変更します。
- **Studio 信頼ストアのパスワードを変更するには、次の手順を実行します。**暗号化した形式のパスワードを指定したプロパティ **client.truststore.password** を **<ユーザー>/.oo** フォルダーの **Studio.properties** ファイルに追加します。

  ```
  client.truststore.password=={OBFUSCATED}6L9+NqBjKYp5heuvMEzg0g==
  ```

  このプロパティが定義されていない場合、Studio はシステムプロパティ **javax.net.ssl.trustStorePassword** にフォールバックして、信頼ストアのパスワードを取得します。

パスワードを暗号化する方法については、「パスワードの暗号化と難読化」(41ページ)を参照してください。

パスワードの暗号化と難読化

パスワードは encrypt-password スクリプトを使用して暗号化または難読化できます。このスクリプトは **<インストールフォルダー>/central/bin** に保存されています。

暗号化を使用することを推奨します。

重要: encrypt-password スクリプトを使用した後で、コマンド履歴をクリアしてください。

これは、Linux OS の場合、パスワードパラメーターはクリアテキストで **/$USER/.bash_history** に保存され、history コマンドでアクセスできるためです。

パスワードの暗号化

1. encrypt-password スクリプトを **<インストールフォルダー>/central/bin** から探します。
2. -e -p <パスワード> オプションを指定して、スクリプトを実行します。ここでパスワードには暗号化するパスワードを指定します。

   注: パスワードを暗号化するためのフラグとしての -p、または --password のいずれかを使用できます。

   暗号化したパスワードは次のように表示されます。

   ```
   {ENCRYPTED}<文字列>
   ```

パスワードの難解化

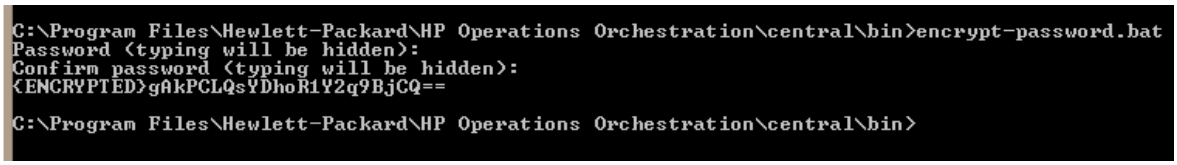
1. encrypt-password スクリプトを **<インストールフォルダー>/central/bin** から探します。
2. -o <パスワード> オプションを指定してスクリプトを実行します。ここでパスワードには難解化するパスワードを指定します。

   難解化したパスワードは次のように表示されます。

   ```
   {OBFUSCATED}<文字列>
   ```

パスワード入力のためのプロンプトの作成

-p 引数を指定しないで encrypt-password スクリプトを実行することをお勧めします。例:



これにより、非表示パスワード入力のためのプロンプトが作成されます。

SSL サポート対象サイファーからの RC4 サイファーの削除

リモートホストは、RC4 暗号の使用をサポートしています。この暗号は、バイトの擬似乱数ストリームの生成処理に欠陥があるため、ストリームに多様で軽微な偏りが生じ、そのランダム性が低下します。

プレーンテキストを繰り返し暗号化するときに (たとえば、HTTP Cookie など)、攻撃者が数多く (数千万) の暗号化テキストを入手できる場合、攻撃者はプレーンテキストを推測できることがあります。

JRE レベルで RC4 暗号を無効にします (Java 7 以降)。

1. **$JRE_HOME/lib/security/java.security** ファイルを開きます。
2. 次の例に従ってコメントを削除し、パラメーターを変更して、RC4 暗号を無効にします。

```
jdk.certpath.disabledAlgorithms=RC4, MD2, RSA keySize < 1024

jdk.tls.disabledAlgorithms=RC4, MD5, DSA, RSA keySize < 1024
```

3. HP OO Central サーバーを再起動します。

詳細については、http://stackoverflow.com/questions/18589761/restict-cipher-suites-on-jre-level を参照してください。

注: 前のバージョンの HP OO 10.x からアップグレードしたら、この手順を繰り返します。

HTTP/HTTPS ポートの変更または HTTP ポートの無効化

[OO_HOME]/central/tomcat/conf の下の **server.xml** ファイルには、**<Service>** 要素の下に **<Connector>** という名前の要素が 2 つあります。これらのコネクターでは、サーバーがリッスンしているポートを定義または有効にします。

各コネクターの構成は、それぞれの属性を使用して定義します。最初のコネクターでは通常の HTTP コネクターを定義し、2 番目のコネクターでは HTTPS コネクターを定義します。

デフォルトで、これらのコネクターは次のようになります。

HTTP コネクター:

```
<Connector URIEncoding="UTF-8" compression="on" connectionTimeout="20000"
port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"
redirectPort="8443"/>
```

HTTPS コネクター:

```
<Connector SSLEnabled="true" URIEncoding="UTF-8" clientAuth="false"
compression="on" keyAlias="tomcat" keyPass="changeit" keystoreFile="C:/Program
```

```
Files/Hewlett-Packard/HP Operations
Orchestration/central/var/security/key.store" keystorePass="changeit"
keystoreType="JKS" maxThreads="200" port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
secure="true" sslProtocol="TLSv1.2" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
truststoreFile="C:/Program Files/Hewlett-Packard/HP Operations
Orchestration/central/var/security/client.truststore" truststorePass="changeit"
truststoreType="JKS"/>
```

デフォルトでは、両方とも有効です。

重要: Central ポートのいずれかを **server.xml** ファイルで変更または無効化する場合は、**central-wrapper.conf** ファイルおよび各 **RAS-wrapper.conf** ファイル更新し、Central URL を更新したポートで指すようにする必要もあります。そうしない場合、Central から実行するすべてのフローが失敗します。さらに、ロードバランサーの構成も必ずチェックしてください。

ポートの値の変更

いずれかのポートの値を変更するには、次の手順を実行します。

1. **<インストールディレクトリ>/central/tomcat/conf/server.xml** にある **server.xml** ファイルを編集します。
2. HTTP または HTTPS コネクターを探し、**port** の値を変更します。

   注: HTTP と HTTPS を両方使用する場合に HTTPS ポートを変更するには、HTTP コネクターの **redirectPort** 値および HTTPS コネクターの **port** 値を変更する必要があります。
3. ファイルを保存します。
4. Central を再起動します。

HTTP ポートの無効化

セキュリティ上の理由から、HTTP ポートを無効にして、TLS 上にある暗号化されたチャネルを唯一の通信チャネルにしなければならないことがあります。

1. **<インストールディレクトリ>/central/tomcat/conf/server.xml** にある **server.xml** ファイルを編集します。
2. HTTP コネクターを探し、その行を削除またはコメント行にします。
3. 信頼されたルート証明機関 (CA) が CA リスト内にまだない場合は、Central の **client.truststore** ファイルにインポートします。

   ```
   keytool -importcert -alias <任意のエイリアス> -keystore <client.truststore へのパス
   > -file <証明書名.cer> -storepass <changeit>
   ```

注: よく知られているルート CA (Verisign など) を使用する場合、証明書はすでに **client.truststore** ファイルに登録されているので、この手順を実行する必要はありません。

4. ファイルを保存します。
5. Central を再起動します。

注: インストール時に HTTP ポートを無効にすることもできます。

トラブルシューティング

サーバーが起動しない場合は、**wrapper.log** ファイルを開いて、`ProtocolHandler ["http-nio-8443"]` でエラーを確認します。

これは Tomcat でコネクターを初期化または起動する際に発生します。さまざまなバリエーションがありますが、エラーメッセージから情報を得ることができます。

HTTPS コネクターのパラメーターはすべて **C:\HP\oo\central\tomcat\conf\server.xml** にある Tomcat 構成ファイル内にあります。

ファイルを開いて下にスクロールし、HTTPS コネクターを確認します。

```
<Connector SSLEnabled="true" clientAuth="false" keyAlias="tomcat"
keystoreFile="C:/HP/oo/central/var/security/keystore.p12" keystorePass="tomcat-
keystore-password" keystoreType="PKCS12" maxThreads="200" port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https" secure="true"
sslProtocol="TLSv1.2" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"/>
```

前のステップで入力したパラメーターと比較して、一致しないパラメーターがないかどうかを確認します。

クライアント証明書の認証 (相互認証)

X.509 証明書認証は、TLS を使用するサーバーの ID 検証によく使用され、特にブラウザーで HTTPS を使用する場合です。ブラウザーは、サーバーが提示する証明書が、信頼される証明機関リストに含まれる証明機関が発行したものかどうかを自動的にチェックします。

TLS を相互認証で使用することもできます。サーバーは、TLS ハンドシェイクにおいて、クライアントに有効な証明書を要求します。サーバーは、証明書が適切な証明機関によって署名されていることをチェックし、クライアントを認証します。有効な証明書が提供されている場合には、アプリケーション内のサーブレット API を使用して取得できます。

クライアント証明書認証の構成 (Central)

Central でクライアント証明書認証を構成する前に、「サーバーおよびクライアント証明書の使用」(33ページ)の手順に従って TLS サーバー証明書を構必要があります。

接続を確立する前に、TLS スタックがクライアントに有効な証明書チェーンを要求する場合は、clientAuth 属性を true に設定します。TLS スタックはクライアント証明書を要求するが、提示されなくてもエラーにしない場合は、want に設定します。false (デフォルト) に設定すると、CLIENT-CERT 認成しておく証を使用するセキュリティ制限で保護されているリソースをクライアントが要求した場合を除き、証明書チェーンは要求されなくなります。詳細については、『Apache Tomcat Configuration Reference』を参照してください。

証明書失効リスト (CRL) ファイルを設定します。CRL は複数存在することがあります。暗号化システムでは一般的に公開キーインフラストラクチャー (PKI) が使用され、証明書失効リスト (CRL) には無効な証明書のリスト (具体的には、証明書のシリアル番号) が格納されています。したがって、ここに含まれる証明書を提示したエンティティは信頼できないエンティティということになります。

注: 次の手順は、Keytool ユーティリティ (**<インストールディレクトリ>/java/bin/keytool**) で実行されます。

1. Central サーバーを停止します。
2. 適切なルート証明書 (CA) が CA リスト内にまだない場合は、Central の**client.truststore**: **<インストールディレクトリ>/central/var/security/client.truststore** にインポートします (CA リストには、よく知られているすべての CA がデフォルトで登録されています)。例:

   ```
   keytool -importcert -alias <任意のエイリアス>  -keystore <パス>/client.truststore -file <証明書のパス>  -storepass <changeit>
   ```

3. **<インストールディレクトリ>/central/tomcat/conf/server.xml** にある **server.xml** ファイルを編集します。
4. Connector タグの clientAuth 属性を want または true に変更します。デフォルトは false です。

   例:

   ```
   <Connector SSLEnabled="true" URIEncoding="UTF-8" clientAuth="false"
   compression="on" keyAlias="tomcat" keyPass="changeit"
   keystoreFile="C:/Program Files/Hewlett-Packard/HP Operations
   Orchestration/central/var/security/key.store" keystorePass="changeit"
   keystoreType="JKS" maxThreads="200" port="8443"
   protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
   secure="true" server="OO" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
   sslProtocol="TLSv1.2" truststoreFile="C:/Program Files/Hewlett-Packard/HP
   Operations Orchestration/central/var/security/client.truststore"
   truststorePass="changeit" truststoreType="JKS"/>
   ```

   注: この手順が終わってからサーバーを起動することをお勧めしますが、この時点でサーバーを起動することもできます。

5. (オプション) crlFile 属性を追加し、TLS 証明書の検証に使用する CRL を定義します。次に例を示します。

   ```
   crlFile="<パス>/crlname.<crl/pem>"
   ```

ファイルの拡張子が .crl の場合は CRL が 1 つ、.pem (PEM CRL 形式) の場合は CRL が複数含まれています。PEM CRL 形式では、次のようなヘッダー行とフッター行を使用します。

```
-----BEGIN X509 CRL-----
-----END X509 CRL-----
```

CRL を 1 つ含む .pem ファイルの例を示します (複数の場合、CRL ブロックを連結していきます)。

```
-----BEGIN X509 CRL-----
MIIBbzCB2QIBATANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJVUzEYMBYGA1UE
ChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxEDAOBgNVBAsTB1Rlc3Rp
bmcxFTATBgNVBAMTDFRydXN0IEFuY2hvchcNOTkwMTAxMTIwMTAwWhcNNDgwMTAx
MTIwMTAwWjAiMCACASScXDTk5MDEwMTEyMDAwMFowDDAKBgNVHRUEAwoBAaAjMCEw
CgYDVR0UBAMCAQEwEwYDVR0jBAwwCoAIq5rr+cLnVI8wDQYJKoZIhvcNAQEFBQAD
gYEAC7lqZwejJRW7QvzH11/7cYcL3racgMxH3PSU/ufvyLk7ahR++RtHary/WeCv
RdyznLiIOA8ZBiguWtVPqsNysNn7WLofQIVa+/TD3T+lece4e1NwGQvj5Q+e2wRt
GXg+gCuTjTKUFfKRnWz7O7RyiJKKim0jtAF4RkCpLebNChY=
-----END X509 CRL-----
```

6. **<インストールディレクトリ> central\conf\central** にある **central-wrapper.conf** ファイルを編集します。

   以下のプロパティをコメント解除し、クライアント証明書の場所とパスワードを管理者ユーザーとともにクライアント証明書に設定します。

   ```
   #wrapper.java.additional.23=-Djavax.net.ssl.keyStore="%CENTRAL_
   HOME%/var/security/certificate.p12"

   #wrapper.java.additional.24.stripquotes=TRUE

   #wrapper.java.additional.25=-Djavax.net.ssl.keyStorePassword={OBFUSCATED}
   ZUoMreNLw6qIOyzX7g5YKw==

   #wrapper.java.additional.26=-Djavax.net.ssl.keyStoreType=PKCS12
   ```

   パスワードを暗号化する方法については、「パスワードの暗号化と難読化」(41ページ)を参照してください。

7. Central サーバーを起動します。

注: クライアント証明書ごとに、ユーザー (内部ユーザーまたは LDAP ユーザー) を定義します。ユーザー名は、証明書属性で定義する必要があります。デフォルトは、CN 属性の値です。詳細については、「証明書のプリンシパルの処理」を参照してください。

HP OO で LDAP 構成を複数設定しても、ユーザー認証に使用できるのは、デフォルト LDAP のクライアント証明書属性のみです。

クライアント証明書の構成の更新 (RAS)

クライアント証明書は、RAS のインストール時に構成されます。ただし、クライアント証明書の更新が必要な場合は、**ras-wrapper.conf** ファイルを手動で編集します。

事前確認: Central の CA ルート証明書を RAS 信頼ストアにインポートする必要があります。「RAS 信頼ストアへの CA ルート証明書のインポート」(36ページ)を参照してください。

外部 RAS でクライアント証明書を更新するには、次の手順を実行します。

1. RAS サーバーを停止します。
2. **<インストールディレクトリ>ras/conf/ras-wrapper.conf** の **ras-wrapper.conf** ファイルを開きます。
3. クライアント証明書に基づいて次の変更を行います。

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.keyStore=<インストールディレクトリ>/var/security/certificate.p12"

   wrapper.java.additional.<x>=-Djavax.net.ssl.keyStorePassword={OBFUSCATED}<obfuscated_password>

   wrapper.java.additional.<x>=-Djavax.net.ssl.keyStoreType=PKCS12
   ```

4. RAS サーバーを起動します。

重要X.509 クライアント証明書には、RAS のプリンシパル名が必要です。これは、RAS ID です(「証明書のプリンシパルの処理」を参照してください)。

RAS ID は、Central の [**トポロジ**] タブで確認できます。『HP OO Central ユーザーガイド』の「トポロジのセットアップ – ワーカー」を参照してください。

HP OO 10.20 以降では、パスワードがデフォルトのままだった場合に、keyStorePassword パラメーターがデフォルトで暗号化されます。このパラメーターは変更し、クリアテキストまたは暗号化して保存できます。「パスワードの暗号化と難読化」(41ページ)を参照してください。

Studio リモートデバッガーでのクライアント証明書の構成

事前確認: Central の CA ルート証明書を Studio Debugger 信頼ストアにインポートする必要があります。「Studio デバッガー信頼ストアへの CA ルート証明書のインポート」(38ページ)を参照してください。

Studio リモートデバッガーでクライアント証明書を構成するには、次の手順を実行します。

1. Studio を閉じます。
2. **<インストールディレクトリ>/studio** にある **Studio.l4j.ini** ファイルを編集します。
3. クライアント証明書に基づいて次の変更を行います。

```
-Djavax.net.ssl.keyStore="<インストールディレクトリ
>/studio/var/security/certificate.p12"

-Djavax.net.ssl.keyStorePassword={OBFUSCATED}<obfuscated_password>

-Djavax.net.ssl.keyStoreType=PKCS12
```

4. Studio を起動します。

注:

- HP OO 10.20 以降では、パスワードがデフォルトのままだった場合に、keyStorePassword パラメーターがデフォルトで暗号化されます。このパラメーターは変更し、クリアテキストまたは暗号化して保存できます。「パスワードの暗号化と難読化」(41ページ)を参照してください。
- クライアント証明書で使用するユーザー (内部ユーザーまたは LDAP ユーザー) を定義します。ユーザー名は、証明書属性で定義する必要があります。デフォルトは、CN 属性の値です。詳細については、「証明書のプリンシパルの処理」を参照してください。
- HP OO で LDAP 構成を複数設定しても、ユーザー認証に使用できるのは、デフォルト LDAP のクライアント証明書属性のみです。Central は、まずデフォルトの LDAP でユーザー認証を行い、失敗すると、HP OO 内部ドメインで認証を行います。

OOSH でのクライアント証明書の構成

事前確認: Central の CA ルート証明書を OOSH 信頼ストアにインポートする必要があります。「OOSH 信頼ストアへの CA ルート証明書のインポート」(37ページ)を参照してください。

1. OOSH を停止します。
2. **<インストールディレクトリ>/central/bin** にある **oosh.bat** を編集します。
3. クライアント証明書に基づいて次の変更を行います。

```
-Djavax.net.ssl.keyStore="<インストールディレクトリ>/var/security/certificate.p12"

-Djavax.net.ssl.keyStorePassword={OBFUSCATED}<obfuscated_password>

-Djavax.net.ssl.keyStoreType=PKCS12
```

4. OOSH を起動します。

注:

HP OO 10.20 以降では、パスワードがデフォルトのままだった場合に、keyStorePassword パラメーターがデフォルトで暗号化されます。このパラメーターは変更し、クリアテキストまたは暗号化して保存できます。「パスワードの暗号化と難読化」(41ページ)を参照してください。

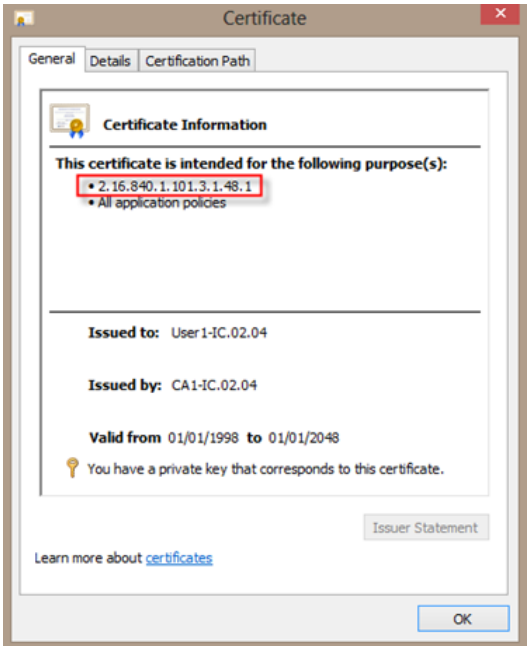
クライアント証明書で使用するユーザー (内部ユーザーまたは LDAP ユーザー) を定義します。ユーザー名は、証明書属性で定義する必要があります。デフォルトは、CN 属性の値です。詳細については、「証明書のプリンシパルの処理」を参照してください。

HP OO で LDAP 構成を複数設定しても、ユーザー認証に使用できるのは、デフォルト LDAP のクライアント証明書属性のみです。Central は、まずデフォルトの LDAP でユーザー認証を行い、失敗すると、HP OO 内部ドメインで認証を行います。

証明書ポリシーの処理

HP OO は、エンドポイントの証明書に適用する証明書ポリシーを処理します。

- 証明書では、使用目的を示す文字列を設定できます。
- HP OO では、ポリシー文字列を構成アイテムとして追加し、エンドポイントの証明書ごとにポリシー文字列をチェックすることができます。一致しないと、証明書は却下されます。
- 証明書ポリシーの検証を有効または無効にするには、次の構成アイテムを追加します。
  `x509.certificate.policy.enabled=true/false` (デフォルトは `false`)
- 次の構成アイテムを追加して、ポリシーリストを定義します。
  `x509.certificate.policy.list=<カンマ区 切りのリスト>` (デフォルトは空のリスト)。



HP OO システムプロパティを変更する方法の詳細については、『HP OO Shell Guide』を参照してください。

証明書のプリンシパルの処理

`Subject` に対する正規表現を使用して、証明書からプリンシパルを取得する方法を定義できます。正規表現には、単一のグループを指定します。デフォルトの式は `CN=(.?)` であり、一般的な名前フィールドに一致します。たとえば `CN=Jimi Hendrix, OU=` は、`Jimi Hendrix` というユーザー名に一致します。

- 一致の比較では、大文字と小文字を区別します。
- 証明書のプリンシパルは、HP OO のユーザー名です (LDAP または内部ユーザー)。
- 正規表現を変更するには、次の構成アイテムを変更します。`x509.subject.principal.regex`.

HP OO システムプロパティを変更する方法の詳細については、『HP OOO Shell (OOSH) User Guide』を参照してください。

HP OO での FIPS 140-2 レベル 1 互換の構成

以下のセクションでは、HP Operations Orchestration を Federal Information Processing Standards (FIPS) 140-2 レベル 1 互換に構成する手順を説明します。

FIPS 140-2 は、暗号化モジュールに適用されるセキュリティ要件の標準であり、National Institute of Standards Technology (NIST) によって規定されています。標準の規定の内容は、次で参照できます。csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

HP OO で FIPS 140-2 互換の構成を行うと、HP OO は次のセキュリティアルゴリズムを使用します。

- 対称キーアルゴリズム: AES256
- ハッシュアルゴリズム: SHA256

HP OO が使用するセキュリティプロバイダーは、RSA BSAFE Crypto ソフトウェアバージョン 6.1 です。これは、FIPS 140-2 でサポートされる唯一のセキュリティプロバイダーです。

注: HP OO で FIPS 140-2 互換構成が完了すると、標準構成に戻すことはできません。戻すには、HP OO の再インストールが必要です。

前提条件

アップグレードプログラムのメモ:

FIPS ですでに構成された HP OO 10.10 (以降) のインストールからアップグレードする場合は、「アップグレードプログラムの前提手順」を参照してください。

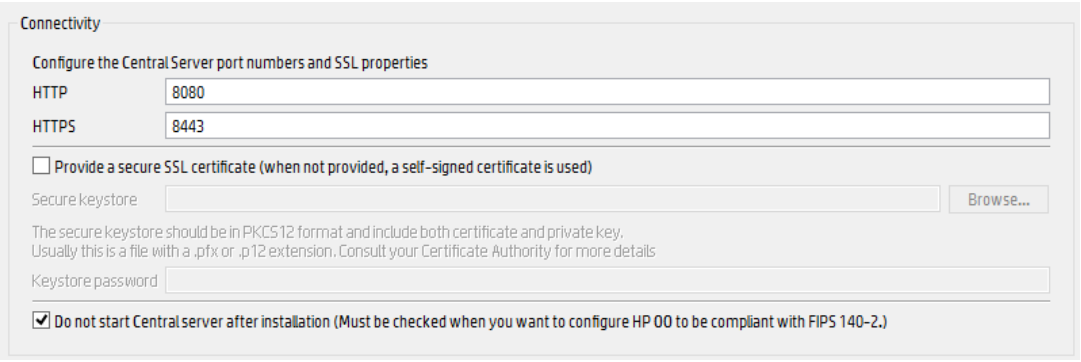
HP OO で FIPS 140-2 互換構成を行う前は、次の手順を実行します。

注: FIPS140-2 互換の構成には、LW SSO を無効にする必要があります。

1. FIPS 140-2 互換構成には、HP OO バージョン 10.10 以降の新規インストールが必要です。

   インストール済みの HP OO (バージョン 9.x または 10.x を問わず) は使用できません。
2. HP OO のインストール時に、インストール後に Central サーバーを起動しないように設定されていることを確認します。
   - サイレントインストールでは、`should.start.central` パラメーターは [**No**] に設定されます。
   - ウィザードの [**Connectivity**] 手順で、[**Do not start Central server after installation**] チェッ

クボックスを選択します。



3. 次のディレクトリをバックアップします。
   - **<インストールディレクトリ>\central\tomcat\webapps\oo.war**
   - **<インストールディレクトリ>\central\tomcat\webapps\PAS.war**
   - **<インストールディレクトリ>\central\conf**
   - **<インストールディレクトリ>\java (java** フォルダー全体のバックアップが必要)
4. http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html から **Server Oracle JRE 8** をダウンロードし、**OpenJDK (Zulu) JRE** を **Server Oracle JRE** に置き換えます。
   a. **<インストールディレクトリ>\JAVA** フォルダーの内容をすべて削除します。
   b. ダウンロードしたアーカイブを展開します。
   c. **JRE** フォルダーの内容を **<インストールディレクトリ>\JAVA** にコピーします。
5. Java Cryptographic Extension (JCE) 無制限強度管轄ポリシーファイルを次のサイトからダウンロードおよびインストールします。

   http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html

   注: ファイルのデプロイと HP OO で使用する JRE のアップグレードの手順は、ダウンロードした **ReadMe.txt** ファイルを参照してください。
6. RSA BSAFE Crypto ソフトウェアファイルをインストールします。HP OO がインストールされているシステムで、次のファイルを **<oo_jre>\lib\ext** (**<oo_jre>** は、HP OO が使用する JRE のインストール先。デフォルトディレクトリは **<インストールディレクトリ>\java**) にコピーします。
   - **<インストールディレクトリ>\central\lib\cryptojce-6.1.jar**
   - **<インストールディレクトリ>\central\lib\cryptojcommon-6.1.jar**
   - **<インストールディレクトリ>\central\lib\jcmFIPS-6.1.jar**

アップグレードプログラムの前提手順

1. Server Oracle JRE 8 をダウンロードし、OpenJDK (Zulu) JRE を Server Oracle JRE に置き換えます。
   a. **<アップグレードディレクトリ>\JAVA** フォルダーの内容をすべて削除します。
   b. ダウンロードしたアーカイブを展開します。
   c. **JRE** フォルダーの内容を **<アップグレードディレクトリ>\JAVA** にコピーします。

   http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html
2. Java Cryptographic Extension (JCE) 無制限強度管轄ポリシーファイルを次のサイトからダウンロードおよびインストールします。

   http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

   ファイルのデプロイと HP OO で使用する JRE のアップグレードの手順は、ダウンロードした **ReadMe.txt** ファイルを参照してください。
3. RSA BSAFE Crypto ソフトウェアファイルをインストールします。HP OO がインストールされているシステムで、次のファイルを **<oo_jre>\lib\ext** にコピーします。

   (ここで、**<oo_jre>** は、HP OO アップグレードプログラムによって使用される JRE がインストールされているディレクトリです。これは、デフォルトでは **<アップグレードディレクトリ>\java** です)。
   - <インストールディレクトリ>\central\lib\cryptojce-6.1.jar
   - <インストールディレクトリ>\central\lib\cryptojcommon-6.1.jar
   - <インストールディレクトリ>\central\lib\jcmFIPS-6.1.jar

次に、「HP OO での FIPS 140-2 互換の構成」(54ページ) の「Java セキュリティファイルのプロパティの構成」セクションの手順を実行します。

HP OO での FIPS 140-2 互換の構成

FIPS 140-2 との互換性を維持するために HP OO で必要な構成手順を示します。

1. Java セキュリティファイルのプロパティ構成。
2. encryption.properties ファイルの構成と FIPS モードの有効化。
3. FIPS 互換の HP OO 暗号化の作成。
4. 新しい暗号化によるデータベースパスワードの再暗号化。
5. HP OO の起動。

Java セキュリティファイルのプロパティ構成

JRE で使用する Java セキュリティファイルを編集してセキュリティプロバイダーを追加し、FIPS 140-2 互換のプロパティを構成します。

注: HP OO 10.x にアップグレードすると、インストール済みの JRE ファイルは完全に置換されます。したがって、10.x にアップグレードする場合は、次の手順を実行する必要があります。

注: FIPS で構成済みの HP OO 10.10 以降のインストールからアップグレードする場合は、「HP OO での FIPS 140-2 レベル 1 互換の構成」(52ページ)の「アップグレードプログラムの前提手順」セクションを実行してから、ここの手順を実行する必要があります。ここで、**<oo_jre>** は (場所 **<アップグレードディレクトリ>\JAVA** にある) アップグレードに含まれる JRE です。

抽出された **upgrade** フォルダー内の **java** フォルダーで、すべての変更を行ってください。

エディターで **<oo_jre>\lib\security\java.security** ファイルを開き、次の手順を実行します。

1. プロバイダーごとに (**security.provider.<nn>=<プロバイダー名>** という形式)、プリファレンス順序の数値 <nn> を 2 つずつ増やします。

   たとえば、次のようなプロバイダーエントリがある場合、次のように変更します。

   ```
   security.provider.1=sun.security.provider.Sun
   ```

   変更後

   ```
   security.provider.3=sun.security.provider.Sun
   ```

2. 新しいデフォルトプロバイダー (RSA JCE) を追加します。次のプロバイダーをリストの一番上に追加します。

   ```
   security.provider.1=com.rsa.jsafe.provider.JsafeJCE
   ```

3. RSA BSAFE SSL-J Java Secure Sockets Extension (JSSE) Provider を追加します。

   ```
   security.provider.2=com.rsa.jsse.JsseProvider
   ```

4. 次の行を **java.security** ファイルに貼り付けます。これにより、**RSA BSAFE** が FIPS 140-2 互換モードで使用されます。

   ```
   com.rsa.cryptoj.fips140initialmode=FIPS140_SSL_MODE
   ```

   この行は、**java.security** ファイル内の任意の場所に貼り付けることができます。

5. デフォルトの DRBG アルゴリズム ECDRBG128 は安全性が低いので (NIST の報告)、セキュリティプロパティ **com.rsa.crypto.default** を **HMACDRBG** に設定します。設定には、次の行を **java.security** ファイルにコピーしてください。

   ```
   com.rsa.crypto.default.random=HMACDRBG
   ```

   この行は、**java.security** ファイル内の任意の場所に貼り付けることができます。

6. **java.security** ファイルを保存してから閉じます。

encryption.properties ファイルの構成と FIPS モードの有効化

HP OO 暗号化プロパティファイルで、FIPS 140-2 互換の設定を行います。

1. **encryption.properties** ファイルをバックアップします。このファイルは **<インストールディレクトリ>\central\var\security** にあります。
2. **encryption.properties** ファイルをテキストエディターで開きます。たとえば、次の行を編集します。

   C:\Program Files\Hewlett-Packard\HP Operations Orchestration\central\var\security\encryption.properties.
3. `keySize=128` を探して、`keySize=256` に変更します。
4. `secureHashAlgorithm=SHA1` を探して、`secureHashAlgorithm=SHA256` に変更します。
5. `FIPS140ModeEnabled=false` を探して、`FIPS140ModeEnabled=true` に変更します。

   注: `FIPS140ModeEnabled=false` が存在しない場合、`FIPS140ModeEnabled=true` を新しくファイルの末尾に追加します。
6. ファイルを保存してから閉じます。

FIPS 互換の HP OO 暗号化の作成

FIPS 互換の設定には、HP OO 暗号化ストアファイルの作成または置換が必要です。手順は、「FIPS 暗号化の置き換え」(57ページ)を参照してください。

注: AES では、NIST SP800-131A パブリケーションによる 128/192/256 の 3 つのキー長が認められています。

FIPS では、安全なハッシュアルゴリズムとして、SHA1、SHA256、SHA384、SHA512 がサポートされています。

注: key.store (およびその秘密キーエントリ) と信頼ストアのパスワードを変更することをお勧めします。「キーストア/信頼ストアのパスワードの変更と暗号化/難読化」(39ページ)を参照してください。

注: 使用していないデフォルトの CA ルート証明書は、HP OO 信頼ストアからすべて削除することをお勧めします (**client.truststore** は **<インストール>/central/var/security** にあります)。

注: クライアント証明書を使用する場合、その証明書は、FIPS 準拠の RSA JCE プロバイダーと、上記リストに示す FIPS でサポートされるセキュアなハッシュアルゴリズムで生成されている必要があります。

新しい暗号化によるデータベースパスワードの再暗号化

データベースパスワードを、『HP OO Administration Guide』の「データベースパスワードの変更」の説明に従って、再暗号化します。

HP OO の起動

FIPS 暗号化の置き換え

HP OO Central および RAS は、機密データや重要データを保護するための暗号ベースのセキュリティシステムを指定する際に、連邦機関で使用する技術要件を定めた Federal Information Processing Standard 140-2 (FIPS 140-2) に準拠しています。

HP OO を新規にインストールした場合、FIPS 暗号化キーを変更することができます。

注: この手順は、新規インストール専用です。アップグレードで実行することはできません。

Central での FIPS 暗号化キーの変更

generate-keys.bat/sh ファイルを使用して、暗号化リポジトリの FIPS 暗号化キーを置き換えます。

注: このプロセスでは **encryption_repository** ファイルがバックアップされます。そのため、適切な書き込み権限が必要です。

1. **<Central インストールフォルダー>/var/security** に移動します。
2. **encryption_repository** ファイルをバックアップし、**<Central インストールフォルダー>/var/security** フォルダーからそのファイルを削除します。
3. **<Central インストールフォルダー>/bin** に移動します。
4. **generate-keys** スクリプトを実行します。
5. **Y** キーを押して、続行します。

   新しいマスターキーが、**<Central インストールフォルダー>/var/security/encryption_repository** に生成されます。

注: ユーザーが **Y** または **N** を入力するための一時停止を行わずに、**generate-keys** スクリプトを実行する場合は、スクリプトを実行するときにサイレントモードフラグ **–s** を使用します。

RAS 暗号化プロパティの変更

RAS を新しい場所にインストールする場合、次の手順を実行します。

注: 以下の変更内容が有効になるのは、Central 暗号化プロパティの変更後に新しく RAS インストールを行う場合のみです。

RAS 暗号化プロパティを変更するには、次の手順を実行します。

1. 「HP OO での FIPS 140-2 レベル 1 互換の構成」(52ページ).の「前提条件」の手順をすべて実行します。
2. 「HP OO での FIPS 140-2 互換の構成」(54ページ) の「Java セキュリティファイルのプロパティの構成」の手順をすべて実行します。
3. 現在の **encryption.properties** ファイルを、**<インストールディレクトリ>\ras\var\security** フォルダーから **<インストールディレクトリ>\ras\bin** フォルダーにコピーします。
4. テキストエディターで **encryption.properties** ファイルを開き、必要な変更を行います。

   詳細は、「HP OO での FIPS 140-2 互換の構成」(54ページ)の「encryption.properties ファイルの構成と FIPS モードの有効化」を参照してください。
5. 変更内容を保存します。
6. **<インストールディレクトリ>\ras\bin** フォルダーでコマンドラインプロンプトを開きます。
7. **oosh.bat** を実行します。
8. 次の OOShell コマンドを実行します。`replace-encryption --file encryption.properties`

   注: encryption.properties ファイルを別のフォルダーにコピーした場合は、OOShell コマンドの場所を正しく指定してください。
9. RAS サービスを再起動します。

TLS プロトコルの構成

HP OO は、サポートされる TLS プロトコルバージョンを定義するように構成できます。HP OO は、デフォルトでは TLS v1、TLS v1.1、TLS v1.2 を使用できますが、これは制限することができます。

注: SSLv3 などの SSL バージョンはサポートされていません。

1. **<installation_folder>/central/tomcat/conf/server.xml** ファイルを開きます。
2. SSL コネクターを探します (ファイルの最後にあります)。
3. `sslEnabledProtocols` のデフォルト値を編集します。たとえば、`sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"` を `sslEnabledProtocols="TLSv1.2"` に変更します。
4. サーバーを再起動します。

フローが Central/RAS のローカルファイルシステムにアクセスできなくする

フローが Central または RAS のローカルファイルシステムにアクセスできなくしたり、機密リソースにアクセスできるようにしたりするためには、Central または RAS のラッパー構成ファイルと

java.policy ファイルを変更する必要があります。

注: このシナリオを利用するには、フローでの権限またはフローに権限を付与する権限に加え、デプロイメントとトリガー権限の両方が必要です。このような権限を持つユーザーは、信頼できるユーザーである可能性が高いです。

このシナリオから保護するには、以下を実行します。

1. Central または RAS のラッパー構成ファイル (**<インストールフォルダー>/<ras/central>/conf/<central/ras>-wrapper.conf**) で、次のように `wrapper.java.additional.<nn>` パラメーターを追加します。

   ```
   wrapper.java.additional.<nn>=-Djava.security.manager
   ```

   <nn> は最後の番号の次の番号で置き換えます。

2. **java.policy** ファイル (**<インストールフォルダー>/java/lib/security/java.policy** にある) に、以下を追加します。これにより、HP OO が必要とする最小リソースへのアクセスを可能にしたり、機密データが含まれている Central/RAS のローカルファイルシステムにアクセスできなくしたりすることができます。

   ```
   grant codebase "file:${oo.home}/bin/-" {
                   permission java.security.AllPermission;
   };

   grant codebase "file:${oo.home}/lib/-" {
                   permission java.security.AllPermission;
   };

   grant codebase "file:${oo.home}/tomcat/-" {
                   permission java.security.AllPermission;
   };

   grant codebase "file:${oo.home}/var/cache/-" {
                   permission java.io.FilePermission "${oo.home}/var/logs",
   "read, write";
   };
   ```

   フローが Central/RAS のローカルファイルシステム内のリソースにアクセスできるようにするには、上記を java.policy に指定します。例:

   ```
   grant codebase "file:${oo.home}/var/cache/-" {
                   permission java.io.FilePermission
   "C:\\users\\cathy\\foo.bat", "read, write, execute, delete";
                   permission java.io.FilePermission "C:\\users\\cathy\\-",
   "read,write,execute,delete"; // Recursive Example
                   permission java.io.FilePermission "C:\\users\\cathy\\*",
   "read,write,execute,delete"; // Flat Example
   ```

```
                    .....
};
```


Java™
COMPATIBLE