HP Operations Orchestration

软件版本：10.50
Windows 和 Linux 操作系统

安全和强化指南

文档发布日期：2015 年 9 月
软件发布日期：2015 年 9 月

法律声明

担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

受限权利声明

机密计算机软件。必须拥有 HP 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

版权声明

商标声明

Adobe™ 是 Adobe Systems Incorporated 的商标。

Microsoft® 和 Windows® 是 Microsoft Corporation 在美国的注册商标。

UNIX® 是 The Open Group 的注册商标。

此产品包含“zlib”通用压缩库的接口，版权所有© 1995-2002 Jean-loup Gailly and Mark Adler。

文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，用于指示软件版本。
- 文档发布日期，该日期将在每次更新文档时更改。
- 软件发布日期，用于指示该版本软件的发布日期。

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：**https://softwaresupport.hp.com/group/softwaresupport/**

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请访问：**http://h20229.www2.hp.com/passport-registration.html**

或单击“HP Passport”登录页面上的“New users - please register”链接。

此外，如果订阅了相应的产品支持服务，则还会收到更新的版本或新版本。有关详细信息，请与您的 HP 销售代表联系。

支持

请访问 HP 软件联机支持网站：**https://softwaresupport.hp.com/**

此网站提供了联系信息，以及有关 HP 软件提供的产品、服务和支持的详细信息。

HP 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人
- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请访问：

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

要查找有关访问级别的详细信息，请访问：

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

HP Software Solutions Now 可访问 HPSW 解决方案和集成门户网站。此网站将帮助您寻找可满足您业务需求的 HP 产品解决方案，包括 HP 产品之间的集成的完整列表以及 ITIL 流程的列表。此网站的 URL 为 **http://h20230.www2.hp.com/sc/solutions/index.jsp**

目录

简介

欢迎使用《HP OO 安全和强化指南》。

本指南旨在帮助 IT 专业人员以安全的方式部署和管理 HP Operations Orchestration (HP OO) 实例。我们的目标是帮助您对 HP OO 提供的各种功能和特性做出明智的决策，以满足现代企业的安全要求。

企业的安全要求在不断变化，本指南可视为惠普为满足这些严格要求所尽的最大努力。如果本指南有未涵盖的其他安全要求，请打开 HP 支持团队提供的支持案例以记录这些要求，我们会将其包含在本指南的未来版本中。

技术系统格局

HP OO 是基于 Java 2 Enterprise Edition (J2EE) 技术的企业级应用程序。J2EE 技术为企业应用程序的设计、开发、装配和部署提供了基于组件的方法。

安全更新

从 HP OO 10.20 到 10.50 进行了以下安全更新:

- 在 Central 中选中“启用对已登录用户凭据的捕获”复选框后，此用户在 Remote Debugger 中运行流时，HP OO 将（以安全的方式）临时捕获已登录用户的凭据。此时将显示一条消息，警告可能会捕获凭据。
- 在 HP OO 10.50 中，默认设置是没有默认角色。这使管理员能够更好地控制用户授权，因为用户只能获得显式分配给他们或他们所属 LDAP 组的角色。
- HP OO 存在多个 LDAP 配置时，如果管理员将其中一个配置标记为默认配置，则属于该配置的用户在登录时无需选择域。
- HP OO 10.50 会在执行期间保护敏感数据（例如密码）。如果在 Studio 中将某个变量标记为敏感数据，则在 Scriptlet 中使用该变量时将以加密形式对其进行检索。

从 HP OO 10.10 到 10.20 进行了以下安全更新:

- 现在可以为 HP OO 中的系统帐户授予权限。这将允许管理员控制哪些用户可以查看哪些系统帐户并运行使用这些帐户的流。此功能对于拥有多个组织的客户非常有用，这些客户可能希望对部分用户隐藏某些系统帐户。

  有关详细信息，请参阅《HP OO 10.20 Release Notes》中的“Content Management Enhancements - Apply Permissions to Multiple Roles”。
- 现在可以在“编辑权限”对话框中将权限应用于多个角色。在以前的版本中，一次只能选择一个角色。

  有关详细信息，请参阅《HP OO 10.20 Release Notes》中的“Content Management Enhancements - Permissions for System Accounts”。

- 从 10.x 早期版本升级 HP OO 安装时，将更新 SSL TrustStore 使其包含 Oracle 所发布的最新受信任的根证书。此过程包括删除过期的证书和导入新证书。

  有关详细信息，请参阅《HP OO 10.20 Release Notes》中的“Installation Enhancements - Updated Trusted Root Certificates”。

- 现在 HP OO 提供了用于审核事件的选项，以便跟踪安全漏洞。审核支持您跟踪对 Central 执行的动作，如登录、触发流、创建计划、编辑配置等。

  当前只能通过 API 检索审核跟踪。有关详细信息，请参阅《HP OO API Guide》。

- 现在 HP OO 支持长度为 2048 位（及更长）的加密密钥。这使得我们的加密密钥与 FIPS 186-4 标准一致。

- 已将新的 sslEnabledProtocols 属性添加到 **server.xml** 文件（位于 **<安装文件夹>/central/tomcat/conf/server.xml**）中：

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

  此属性可确保仅允许使用 TLS v1、TLS v1.1 和 TLS v1.2，不允许使用 SSL 3.0。这样可防止受到“POODLE”攻击 (Padding Oracle On Downgraded Legacy Encryption)。

相关文档

有关 HP OO 安全强化的详细信息，请参阅以下文档：

- 《HP OO Network Architecture White Paper》

有关 HP OO 的详细信息，请参阅以下文档：

- 《HP OO 概念指南》
- 《HP OO Administrator Guide》
- 《HP OO 体系结构指南》
- 《HP OO 数据库指南》
- 《HP OO Central 用户指南》
- 《HP OO Studio 创建指南》
- 《HP OO Release Notes》
- 《HP OO 安装、升级和配置指南》
- 《HP OO 系统要求》
- 《REST Wizard User Guide》

这些文档及其他文档均可在 HPLN (https://hpln.hp.com/node/21/otherfiles#) 上找到。

安全概述

此部分提供安全模型的概述和安全实施 HP OO 的建议。包括如身份验证、授权、加密等主题。相关位置处还引用了其他 HP OO 文档，以描述如何完成安全相关任务。

安全概念

HP OO 术语表

有关 HP OO 概念的详细信息，请参阅《HP OO 概念指南》。

角色权限

权限是指执行任务的预定义授权。HP OO Central 包括一组可以分配给角色的权限。

例如，“计划”权限授予查看和创建运行计划的功能。

角色

角色是权限的集合。

例如，可以为“流管理员”角色分配“查看计划”和“管理计划”权限。

用户

用户是与人员（或应用程序标识）关联的对象，用于表示人员并定义其授权。

将角色分配给用户可定义他们在 Central 中有权执行的动作。例如，用户 Joe Smith 可以分配有“流管理员”角色。

可以配置不同种类的用户：

- **LDAP 用户**使用他们的 LDAP 用户名和密码登录到 Central。例如，使用 Active Directory 用户名和密码。
- **内部用户**使用在 Central 中本地设置的用户名和密码登录到 Central。
- **LWSSO** - HP 轻型单一登录 (SSO) 是一种机制，用户身份验证和授权的单个动作即可允许用户访问支持 LWSSO 的所有 HP 系统。例如，如果用户已登录到启用了 LWSSO 的其他 HP 产品 Web 客户端，则可以绕过 HP OO Central 登录屏幕，直接进入 HP OO Central 应用程序。

内部用户和具有相同角色的 LDAP 用户同时登录时，二者之间的权限不存在任何差异。

备注: 建议使用 LDAP 用户，而不使用内部用户，因为 LDAP 提供程序实施的策略确保 LDAP 用户的安全。

内容权限

内容权限是指查看或运行单个流或特定文件夹中的流的权限。

已分配指定角色的用户将可以根据分配给其角色的内容权限来访问流。

例如，具有“管理员”角色的用户有权查看和运行系统中的所有流，而具有“用户”角色的用户有运行某些流的权限，对其他流有查看权限。

常见安全概念

系统安全

保护基于计算机的设备、信息和服务免遭无意或未经授权的访问、更改或损坏的过程和机制。

最低特权

将访问权限限制为允许正常运行的最低级别的实践。这意味着仅授予用户帐户工作所需的特权。

身份验证

标识个人的过程，通常基于用户名和密码或证书。

授权

基于个人身份标识访问系统对象的权限。

加密

一种增强消息或文件安全性的方式，通过加密内容，从而使只有拥有可对内容进行编码的正确加密密钥的用户才能读取该内容。例如，TLS 协议会加密通信数据。

对策

降低威胁风险的方式。

深度防御

多层保护，使您无需依赖于单个安全措施。

风险

可能导致损坏的事件。例如，经济损失、公司形象受损等。

威胁

触发利用漏洞的风险事件。

漏洞

目标中可能会被安全威胁利用的弱点。

安全实施和部署

默认安全设置

在许多情况下，建议修改提供的开箱即用的默认安全设置。

- **身份验证** – 默认情况下，Central 中不启用身份验证。建议设置用户后立即启用。有关详细信息，请参阅《HP OO Central 用户指南》中的“启用身份验证”。
- **审核** – 默认情况下，Central 中不启用审核。建议启用。有关详细信息，请参阅《HP OO Central 用户指南》中的“启用审核”。
- **TLS 加密** – 默认情况下，HP OO 支持三种 TLS 协议：1.0、1.1、1.2。建议使用最新版本。有关详细信息，请参阅配置 TLS 协议 (第 56 页)。
- **TLS 服务器证书** – 默认情况下，安装 HP OO 服务器期间系统会要求用户提供 CA 证书。
- **客户端证书** – 默认情况下，不启用客户端证书。建议使用客户端证书对 Central 进行身份验证。有关详细信息，请参阅在 Central 中配置客户端证书身份验证 (第 44 页)。
- **KeyStore、TrustStore 和服务器证书密码** – 默认情况下，会为 keyStore、trustStore 和服务器证书提供 Java 密码。建议将这些密码替换为加密密码。有关详细信息，请参阅对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)
- **RC4 密码** – 默认情况下，启用 RC4 密码。建议在 JRE 级别禁用 RC4 密码。有关详细信息，请参阅从支持 SSL 的密码中删除 RC4 密码 (第 42 页)。
- **安全标题** – 默认情况下，Central 中不启用安全标题。建议启用，并使用自定义消息。有关详细信息，请参阅《HP OO Central 用户指南》中的“设置安全标题”。
- **数据库的 Windows 身份验证** – 默认情况下，Central 中不启用 Windows 身份验证。如果在 Windows 和 SQL Server 环境中工作，建议将 HP OO 配置为使用 Windows 身份验证。请参阅《HP OO 数据库指南》中的“将 HP OO 配置为使用 Windows 身份验证”。
- **默认算法** – **encryption.properties** 文件包含默认算法。如果要兼容 FIPS，请参阅配置 HP OO 以兼容 FIPS 140-2 1 级 (第 50 页)。有关 FIPS 140-2 1 级默认值的详细信息，请参阅加密 (第 23 页)中的“加密管理”。
- **Java 策略** – 默认情况下，不强化 **java.policy** 文件。有关如何修改 **java.policy** 文件的信息，请参阅阻止流访问 Central/RAS 本地文件系统 (第 57 页)。

HP OO 安全强化

“强化”一章提供了用于保护 HP OO 部署免遭安全风险或威胁的建议。确保应用程序安全的一些最重要的原因包括保护组织重要信息的机密性、完整性和可用性。

为了全面保护您的 HP OO 系统，确保 HP OO 和应用程序所运行的计算环境（例如，基础结构和操作系统）的安全很有必要。

“强化”一章提供了有助于在应用程序级别确保 HP OO 安全的建议，并未涵盖如何确保客户环境中的基础结构的安全。客户应自行了解其基础结构/环境以及应用相应的强化策略。

物理安全

HP 软件建议通过您组织定义的物理安全控制措施来保护 HP OO。根据最佳实践，将 HP OO 服务器组件安装在物理安全环境中。例如，服务器必须位于具有访问控制的密闭房间中。

安全安装准则

支持的操作系统

有关支持的操作系统的类型和版本，请参阅《HP OO 系统要求》。

操作系统强化建议

请与您的操作系统供应商联系，了解强化您的操作系统的建议最佳实践。

例如：

- 应安装修补程序
- 应删除或禁用不必要的服务/软件
- 应向用户分配最低权限
- 应启用审核

Tomcat 强化

安装 HP OO 时，会默认局部强化 Tomcat。如果需要额外强化，请参阅“强化”一章中的建议。

安装权限

安装和运行 HP OO 需要以下权限：

安装 HP OO	Windows/Linux：能运行 Java 进程、有权创建文件夹和服务的任何标准用户
运行 HP OO	• Windows：Windows 服务以系统用户或特定用户身份运行（该用户必须有权访问 HP OO 安装目录） • Linux：能运行 Java 进程的任何标准用户

另请参阅 CIS Apache Tomcat 7.0 文档中的建议。

网络和通信安全

《HP OO 体系结构指南》描述了基本 HP OO 拓扑、高可用性和负载平衡器安全。

《HP OO Network Architecture White Paper》描述了必需的防火墙配置，并建议了在由于策略限制无法实施必需的防火墙配置时可采用的两个解决方法：

- SSH 反向隧道
- 反向代理服务器

通信通道安全

支持的协议和配置

HP OO 支持 TLS 协议。

有关详细信息，请参阅替换 Central TLS 服务器证书 (第 33 页)。

Central 端口由管理员在安装期间定义。

通道安全

HP OO 支持以下安全通道：

通道（有向）	支持的安全协议
OOSH、浏览器、Studio Remote Debugger 或 RAS → Central	对于安全通道，请使用 TLS 通信进行加密，使用客户端证书进行身份验证。
Central → LDAP 服务器	要加密 Central 和 LDAP 之间的通信，请使用采用 TLS 协议的安全 LDAP。

管理接口安全

访问管理接口

有多种方式可控制对管理接口的访问:

- 凭据
- 客户端证书
- SAML

确保管理接口的安全 - 建议

1. 建议在 Central 中启用身份验证。

   请参阅《HP OO Central 用户指南》中的“启用身份验证”

2. 建议使用 TLS 协议确保管理接口的安全。应在客户端和 Central 接口之间设置 TLS 以进行加密。

   请参阅使用服务器和客户端证书 (第 32 页)。

3. 建议使用 LDAP 用户，而非内部用户，因为这样更加安全。

4. 建议设置通过客户端证书访问 Central 的身份验证。这比用户密码更加安全。

   请参阅使用服务器和客户端证书 (第 32 页)。

用户管理和身份验证

身份验证模型

为了在 HP OO 中轻松启动身份验证机制，该产品在启动时禁用身份验证。

强烈建议安装完成后立即启用身份验证。

有关如何启用身份验证的信息，请参阅《HP OO Central 用户指南》中的“启用身份验证”。

有多种方式可对 Central 的访问进行身份验证。

选择标识用户的方法：

- 用户名和密码
- 客户端证书
- SAML 令牌
- 单一登录 (HP LWSSO)

选择以下两种方式之一管理用户：

- LDAP 用户，作为 Active Directory 保存在 LDAP 服务器上（推荐）
- 内部用户和密码，本地保存在 Central 服务器上（不推荐）

用户类型

系统可以向不同类型的用户分配不同的权限。例如，流创建人、管理员、系统管理员等。

有关需要不同权限的不同类型用户的更多示例，请参阅《HP OO 概念指南》中的“主要角色”。

身份验证管理和配置

内部或 LDAP 用户

您可以在 Central UI 中设置内部用户和密码，或在 LDAP 服务器中定义用户并将 LDAP 组映射到 Central 角色。

> **备注:** 我们建议不要使用内部用户，而使用更安全的备用方法，如 LDAP 用户。

有关配置内部用户的信息，请参阅《HP OO Central 用户指南》中的“设置安全 - 内部用户”。

有关将 LDAP 组映射到 Central 角色的信息，请参阅《HP OO Central 用户指南》中的“设置安全 - LDAP 身份验证”和《HP OO API Guide》中的“LDAP Configuration”。

SAML/客户端证书/LW SSO

有关将 Central 配置为使用 SAML 的信息，请参阅《HP OO Central 用户指南》中的“设置安全 - SAML”。

有关将 Central 配置为使用客户端证书的信息，请参阅使用服务器和客户端证书 (第 32 页)。

有关将 Central 配置为使用 LW SSO 的信息，请参阅《HP OO Central 用户指南》中的“设置安全 - LWSSO”、《HP OO Administration Guide》中的“Configuring LWSSO Settings”以及《HP OO API Guide》中的“LW SSO”

数据库身份验证

HP OO 支持四种数据库：Oracle、MS SQL、MySQL 和 Postgres。

我们建议使用强数据库密码进行数据库身份验证以及使用强密码策略。例如，多次尝试失败后阻止登录。

使用 MS SQL 时，可以使用数据库身份验证或操作系统身份验证。我们建议尽可能使用操作系统身份验证。例如，可以使用 Windows 身份验证访问 Microsoft SQL Server 数据库。

- 有关设置操作系统身份验证的信息，请参阅《HP OO 数据库指南》中的“将 HP OO 配置为使用 Windows 身份验证”。
- 请参阅《HP OO Administration Guide》中的“Changing the Database Password”。
- 参阅数据库供应商推荐的最佳实践（如有）。

授权

授权模型

根据用户的角色和为该角色配置的权限，授权该用户对 HP OO 资源的访问权限。

请参阅：

- 《HP OO Central 用户指南》中的“设置安全 - 角色”
- 《HP OO Central 用户指南》中的“为系统帐户分配权限”

最低权限准则

建议：

- 为角色选择相应的权限。
- 创建新角色时使用最低权限。
- 授予最低权限并仅根据需要扩展权限，从而避免不必要的特权升级。例如，最开始只授予“查看”权限，然后根据需要单独添加其他权限。

授权配置

安装的 Central 随附多个开箱即用的角色，您可以配置这些角色并将其分配给用户。默认情况下，为开箱即用的角色分配了以下权限：

角色	默认权限
Administrator	全部
End_user	无
Everybody	无
Promoter	所有“内容”权限
System_admin	所有“系统”权限

默认角色

可以使用“默认角色”属性配置其中一个角色。如果执行此操作，请确保这是具有最低特权的角色。记住，为该角色授予权限时，除了与该角色显式关联的用户之外，还将影响所有 LDAP 用户。

有关详细信息，请参阅《HP OO Central 用户指南》中的“设置安全 - 角色”下的“分配要作为默认角色的角色”。

另请参阅：

- 《HP OO Central 用户指南》中的“为系统帐户分配权限”
- 《HP OO Central 用户指南》中的“设置内容的权限”

备份

为了防止数据丢失，强烈建议定期将服务器上的数据备份到安全介质上。这还有助于进行灾难恢复和保持业务连续性。

安装 HP OO 之后，请确保备份 **central\var\security** 文件夹和 **central\conf\database.properties** 文件。

数据库架构上的某些数据已加密，用于解密的密钥存储在本地 HP OO Central 服务器中。如果这些系统文件被损坏或删除，架构将变得无用，因为没有对数据解密的方法。

备注: 这些密钥将被加密，因此将其包含在备份中十分重要。密钥位于 **security** 文件夹中。

请参阅:

- 《HP OO Administration Guide》中的“Backing Up HP OO”
- 《HP OO Administration Guide》中的“Setting up Disaster Recovery”
- 《HP OO 安装指南》中的“备份和恢复 Central 安全文件”
- 《HP OO 体系结构指南》中的“在 HP OO 部署中使用负载平衡器”

加密

加密模型

HP OO 支持使用加密和哈希算法来保护敏感数据。加密旨在防止暴露和篡改 HP OO 系统中的敏感数据，如密码、定义等。

为了防止敏感数据被未经授权的人员解密，使用没有已知漏洞且众所周知的标准算法很重要。

备注: 例如，不使用 SSL，因为 SSL 协议中存在已知漏洞。

静态数据

所有已保存的密码都将使用众所周知的算法进行保护，而不会采用纯文本形式。

例如:

- 对系统帐户密码进行加密。
- 对内部用户密码进行哈希处理。
- 对数据库密码进行加密。

传输中的数据

HP OO 使用传输层安全性 (TLS) 协议对组件（如 Central 和 RAS）之间的数据进行加密。

禁用 HTTP 端口

出于安全原因，建议禁用 HTTP 端口，以便唯一的通信通道将使用 TLS 并进行加密。有关详细信息，请参阅更改 HTTP/HTTPS 端口或禁用 HTTP 端口 (第 42 页)。

加密管理

建议的加密最佳实践

为达到更高的安全和加密级别，建议配置 HP OO 以兼容联邦信息处理标准 (FIPS) 140-2。可将 HP OO 设置为兼容 FIPS 140-2 1 级。

默认配置集

- 对称密钥算法：密钥大小为 128 的 AES
- 哈希算法：SHA1

高级设置

在配置 HP OO 以兼容 FIPS 140-2 后，HP OO 使用以下安全算法：

- 对称密钥算法：AES256
- 哈希算法：SHA256

请参阅配置 HP OO 以兼容 FIPS 140-2 (第 52 页)。

数字证书

数字证书是人员、服务器、工作站等的数字“通行证”。

- 要在浏览器和 Central 服务器之间使用加密，需要在服务器端安装数字证书。
- 要使用客户端证书对 Central 服务器进行身份验证，您需要在客户端上（例如在浏览器、RAS、OOSH、Studio 上）安装客户端证书。

HP OO 使用 Java Keytool 实用程序管理加密密钥和受信任的证书。HP OO 安装文件夹中包含此实用程序，它位于 **<安装目录>/java/bin/keytool** 中。

证书位置

HP OO Central 的安装包含两个使用 Keytool 的管理证书文件：

- **<安装目录>/central/var/security/client.truststore**：包含受信任证书的列表
- **<安装目录>/central/var/security/key.store**：包含 HP OO 私有证书（包含私钥）

对 KeyStore 和 TrustStore 的访问控制

建议只有运行 Central 服务的用户才能使用读取权限存储 TrustStore 和 KeyStore。

替换 HP OO 自签名证书

在安装新的 HP OO 后或者如果当前证书已过期，建议替换 HP OO 自签名证书。

替换证书的部分流程是使用您的 CA 生成 PKCS12 格式的证书。请与您的 CA 联系以了解有关证书流程的具体详细信息或参考您的公司政策。

有关详细信息，请参阅替换 Central TLS 服务器证书 (第 33 页)。

将数字签名添加到内容包

如果内容包具有来自受信任 CA 的数字签名，就可以保证内容受到信任。

添加数字签名并非强制要求。

- HP OO 开箱即用的内容包包含来自 Verisign 的数字签名。
- 建议 HP OO 创建人将数字签名添加到其自定义内容包。

- 如果签名的内容包被损坏，则无法部署该内容包。
- 如果签名已过期，则部署前会显示警告，用户必须选中复选框以确认他们将忽略已过期的签名。

注意是否存在未签名的内容包。未签名的内容包不受信任，且可能包含恶意内容。另请注意，未签名的内容包可能被损坏，并且签名被删除。

有关内容包的数字证书的详细信息，请参阅《HP OO Central 用户指南》中的“部署和管理内容包”。

内容包中的敏感信息

系统帐户密码

创建内容包时请勿包含密码。将在内容包中对密码采用模糊处理，这不是安全的选择。

HP OO 安全最佳实践是在 Central 中配置系统帐户密码。有关详细信息，请参阅《HP OO Central 用户指南》中的“为内容包设置系统帐户”。

审核和日志文件

审核

审核支持您跟踪 Central 服务器上执行的动作，如登录、触发流、创建计划、编辑配置等。通过审核数据，您可以跟踪 Central 系统上用户的活动，跟踪谁在什么时候执行了什么动作。例如，审核将显示用户运行了流、更新了配置、删除了计划或身份验证失败。

审核数据保存在数据库中。有关详细信息，请参阅《HP OO API Guide》中的“Auditing”。

日志

通过日志，您可以跟踪错误、警告、信息和调试消息。

日志保存在文件服务器的以下位置：

- Central - **<oo 安装文件夹>/central/var/logs**
- Studio - **<用户>/.oo/logs**
- RAS - **<oo 安装文件夹>/ras/var/logs**。

审核记录和日志文件不保留任何敏感数据

HP OO 系统中的审核记录或日志文件中不保留任何敏感数据。

获取审核记录

您可以通过 API 或通过查询 OO_AUDIT 表获取审核记录。有关详细信息，请参阅《HP OO API Guide》中的“Auditing”。

审核数据示例：

```
[
{
“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 和接口

API 和接口模型

您可以使用 HP Operations Orchestration 公共应用程序编程接口 (API) 而非通过 HP OO Central UI 执行相同的动作。某些动作只能通过 API 执行，如清除和审核。公共 API 基于 HTTP。所有 API 均为 RESTful 并使用 JavaScript 对象表示法。

API 的功能和管理以及接口安全配置

安全使用 API 非常重要。使用 API 时，请使用本指南中提到的安全机制（身份验证、加密等）。

API 接口可以使用 HTTP 或 HTTPS。

> **备注:** 使用 API 显示 HTML 时，您有责任保护它免遭 XSS 攻击。

有关详细信息，请参阅《HP OO API Guide》中的以下章节:

- “LDAP Configuration”
- “Users”
- “LW SSO Configuration”
- “Authentication”
- “Roles”

安全问题和答案

我如何才能生成可由外部 CA 签名的证书请求？

导出证书请求，然后将其发送给外部 CA 进行签名。有关说明，请参阅替换 Central TLS 服务器证书 (第 33 页)。

HP OO 使用哪个 TCP/UDP 端口？什么是方向、用户和加密？

安装 HP OO 时，需要在 HTTP/HTTPS 字段中为 Central 服务器配置至少一个可用端口。默认提供的值为 8080 和 8443，但您可以更改它们。有关 Central 和其他组件之间的安全通道的详细信息，请参阅网络和通信安全 (第 16 页)

管理员帐户和集成用户的凭据存储在何处以及如何存储？

请参阅用户管理和身份验证 (第 18 页)。

如何为 Central/RAS/Studio 配置自签名 SSL 证书？

在 HP OO 安装期间，如果未提供证书，则将默认创建自签名证书。但出于安全原因，不建议使用自签名证书。HP 建议使用来自自定义根 CA 或来自众所周知的 CA 的证书。

有关为 HP OO 配置证书的详细信息，请参阅使用服务器证书对通信进行加密 (第 33 页)。

如何启用或禁用任何一种审核？

默认情况下，不启用审核。有关如何启用审核的详细信息，请参阅《HP OO Central 用户指南》中的“启用审核”。有关审核的详细信息，请参阅审核和日志文件 (第 27 页)。

日志中的详细信息有多少以及如何更改日志记录的数量？

可将日志设置为不同的粒度级别。默认级别为 INFO，但您可以进行调整。有关详细信息，请参阅《HP OO Admistration Guide》中的“Adjusting the Logging Levels”。

有关日志文件的详细信息，请参阅审核和日志文件 (第 27 页)。

如何加密敏感信息？

请参阅加密 (第 23 页)。

是否已对 Central 和 RAS 之间的通信进行加密？

如果使用 HTTPS，则会加密。

是否已对 HP OO 和其他集成组件（HPNA、CSA、AD 等）之间的通信进行加密？

这取决于您要使用的集成。如果使用 HTTPS，则会加密。

如何基于用户角色限制对“流库”的访问？

请参阅《HP OO Central 用户指南》中的“设置安全 - 角色”。

HP OO 支持哪种身份验证机制？

支持的身份验证机制包括 LDAP、SAML 和内部用户。HP OO 还支持客户端证书和 LWSSO。请参阅用户管理和身份验证 (第 18 页)。

HP OO 是否兼容 FIPS 140-2？

是。有关详细信息，请参阅配置 HP OO 以兼容 FIPS 140-2 (第 52 页)。

Central 和 RAS 之间的身份验证方法是什么？

用户密码或客户端证书。

所有密码是否都采用加密存储或经过哈希处理？

是。所有已保存的密码都将使用众所周知的算法进行保护，而不会采用纯文本形式。

我是否可以限制 Central 用户 IP 地址？

不行，目前还不支持此功能。

HP OO 是否已通过共同准则认证？

此认证正在进行中。我们当前处于“评估中”。有关详细信息，请访问 https://www.cse-cst.gc.ca/en/canadian-common-criteria-scheme/publication/list/evaluation-product。

使用 OOSH 时，能否将敏感数据传递给 Central？

我们建议在连接到 Central 时使用安全通道。请参阅网络和通信安全 (第 16 页)。

强化 HP Operations Orchestration

此部分描述如何配置对 HP Operations Orchestration 的安全强化。

备注: 有关其他管理任务的信息，请参阅《HP OO 安装、升级和配置指南》。

安全强化建议

1. 安装最新版本的 HP OO。有关详细信息，请参阅《HP OO 安装、升级和配置指南》。
2. （可选）配置 HP OO 以兼容 FIPS 140-2。如果选择执行此操作，则必须在启动 Central 服务器之前先对其进行配置。请参阅配置 HP OO 以兼容 FIPS 140-2 1 级 (第 50 页)。
3. 配置用于 TLS 加密的 Central 服务器证书和用于加强身份验证（相互）的客户端证书。

   备注: 可在安装期间完成此操作。

   对于 RAS、Debugger 和 OOSH，请根据需要提供证书身份验证（针对服务器证书），并对 Central 使用客户端证书进行身份验证。请参阅使用服务器和客户端证书 (第 32 页)。
4. 通过删除 HTTP 端口并使用强密码替换 KeyStore 和 TrustStore 的密码来强化 HP OO Central 服务器。请参阅更改 HTTP/HTTPS 端口或禁用 HTTP 端口 (第 42 页)和对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)。
5. 通过将 KeyStore 和 TrustStore 密码替换为强密码以及对配置文件中的密码进行加密或模糊处理来强化 HP OO Studio。请参阅对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)。
6. 从支持 SSL 的密码中删除 RC4 密码。请参阅从支持 SSL 的密码中删除 RC4 密码 (第 42 页)。
7. （可选）配置 TLS 协议版本。请参阅配置 TLS 协议 (第 56 页)。
8. 在 Central 中启用身份验证。请参阅《HP OO Central 用户指南》中的“启用身份验证”。

   无法确保内部用户的安全，因此请使用具有强密码策略的安全 LDAP。请参阅《HP OO Central 用户指南》中的“设置安全 – LDAP 身份验证”。
9. 强化/确保操作系统和数据库的安全。
10. 添加具有有意义的消息的安全标题。例如，“您正在登录生产环境！如果您不熟悉此系统的管理规则并且未接受必要的培训，请勿继续执行操作。”请参阅《HP OO Central 用户指南》中的“设置安全标题”。
11. 在 Windows 和 SQL Server 环境中，将 HP OO 配置为使用 Windows 身份验证。请参阅《HP OO 数据库指南》中的“将 HP OO 配置为使用 Windows 身份验证”。

12. 确保在 Central 中已启用审核。有关详细信息，请参阅《HP OO Central 用户指南》中的“启用审核”。

默认安全设置

在许多情况下，建议修改提供的开箱即用的默认安全设置。

- **身份验证** – 默认情况下，Central 中不启用身份验证。建议设置用户后立即启用。有关详细信息，请参阅《HP OO Central 用户指南》中的“启用身份验证”。
- **审核** – 默认情况下，Central 中不启用审核。建议启用。有关详细信息，请参阅《HP OO Central 用户指南》中的“启用审核”。
- **TLS 加密** – 默认情况下，HP OO 支持三种 TLS 协议：1.0、1.1、1.2。建议使用最新版本。有关详细信息，请参阅配置 TLS 协议 (第 56 页)。
- **TLS 服务器证书** – 默认情况下，安装 HP OO 服务器期间系统会要求用户提供 CA 证书。
- **客户端证书** – 默认情况下，不启用客户端证书。建议使用客户端证书对 Central 进行身份验证。有关详细信息，请参阅在 Central 中配置客户端证书身份验证 (第 44 页)。
- **KeyStore、TrustStore 和服务器证书密码** – 默认情况下，会为 keyStore、trustStore 和服务器证书提供 Java 密码。建议将这些密码替换为加密密码。有关详细信息，请参阅对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)
- **RC4 密码** – 默认情况下，启用 RC4 密码。建议在 JRE 级别禁用 RC4 密码。有关详细信息，请参阅从支持 SSL 的密码中删除 RC4 密码 (第 42 页)。
- **安全标题** – 默认情况下，Central 中不启用安全标题。建议启用，并使用自定义消息。有关详细信息，请参阅《HP OO Central 用户指南》中的“设置安全标题”。
- **数据库的 Windows 身份验证** – 默认情况下，Central 中不启用 Windows 身份验证。如果在 Windows 和 SQL Server 环境中工作，建议将 HP OO 配置为使用 Windows 身份验证。请参阅《HP OO 数据库指南》中的“将 HP OO 配置为使用 Windows 身份验证”。
- **默认算法** – **encryption.properties** 文件包含默认算法。如果要兼容 FIPS，请参阅配置 HP OO 以兼容 FIPS 140-2 1 级 (第 50 页)。有关 FIPS 140-2 1 级默认值的详细信息，请参阅加密 (第 23 页)中的“加密管理”。
- **Java 策略** – 默认情况下，不强化 **java.policy** 文件。有关如何修改 **java.policy** 文件的信息，请参阅阻止流访问 Central/RAS 本地文件系统 (第 57 页)。

使用服务器和客户端证书

传输层安全性 (TLS) 证书采用数字绑定方式将加密密钥绑定到组织的详细信息，从而确保 Web 服务器到浏览器的连接安全并被加密。

HP OO 使用 Keytool 实用程序管理加密密钥和受信任的证书。HP OO 安装文件夹中包含此实用程序，它位于 **<安装目录>/java/bin/keytool** 中。有关 Keytool 实用程序的详细信息，请访问 http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html。

备注: Keytool 是开源实用程序。

HP OO Central 的安装包含两个证书管理文件：

- **<安装目录>/central/var/security/client.truststore**：包含受信任证书的列表。
- **<安装目录>/central/var/security/key.store**：包含 HP OO 证书（私钥）。

建议：

- 在安装新的 HP OO 后或者如果当前证书已过期，建议您替换 HP OO 自签名证书。
- 建议存储 TrustStore 和 KeyStore 时设置运行 Central 服务的用户仅具有读取权限。
- 建议使用 Keytool 之后将控制台清空或使用密码输入提示。

使用服务器证书对通信进行加密

替换 Central TLS 服务器证书

您可以使用由知名证书颁发机构签名的证书，也可以使用来自本地证书颁发机构的自定义服务器证书。

替换用 <黄色> 突出显示的参数以在计算机上匹配 **key.store** 文件的位置和其他详细信息。

备注: 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/key.store** 中的原始 **key.store** 文件。
2. 在 **<安装目录>/central/var/security** 中打开命令行。
3. 使用以下命令从 Central **key.store** 文件中删除现有服务器证书：

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

4. 如果您的证书已经带有 **.pfx** 或 **.p12** 扩展名，则转至下一步。如果没有，则需要将证书与私钥导出为 PKCS12 格式 (.pfx,.p12)。例如，如果证书格式为 PEM：

   ```
   >openssl pkcs12 -export -in <cert.pem> -inkey <.key> -out <证书名称>.p12 -name <名称>
   ```

   如果证书格式为 DER，请在 pkcs12 后添加 `-inform DER` 参数。例如：

   ```
   >openssl pkcs12 -inform DER -export -in <cert.pem> -inkey <.key> -out <证书名称>.p12 -name <名称>
   ```

   > **备注:**
   >
   > 要生成 PKCS12 格式的证书，需要使用您的 CA。由于此步骤可能因 CA 供应商和策略而异，您应当咨询 CA 以获取证书生成过程的详细说明。

   > **备注:** 记下您提供的密码。稍后在此过程中输入 KeyStore 密码时将需要此私钥的密码。
   >
   > 确保选择强密码。

5. 使用以下命令列出证书的别名：

   ```
   keytool -list -keystore <证书名称> -v -storetype PKCS12
   ```

   将显示证书别名，且应在下一个命令中提供此证书别名。

   在下例中，别名位于倒数第四行。

   ```
   c:\Program Files\Hewlett-Packard\oo-saml\central\var\security>keytool -list -keystore server.pfx -v -storetype PKCS12
   Enter keystore password:

   Keystore type: PKCS12
   Keystore provider: SunJSSE

   Your keystore contains 1 entry

   Alias name: le-775fb32c-269c-499b-bae8-fe7077479ec6
   Creation date: 24/04/2014
   Entry type: PrivateKeyEntry
   Certificate chain length: 2
   ```

6. 使用以下命令将 PKCS12 格式的服务器证书导入 Central **key.store** 文件：

```
keytool -importkeystore -srckeystore <PKCS12 格式证书路径> -destkeystore
key.store -srcstoretype pkcs12 -deststoretype JKS -alias <证书别名> -
destalias tomcat
```

7. 如果导入的服务器证书的密码与原始服务器证书不同，则更改 keyPass 密码非常重要。请按照对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)中的说明执行操作。

   此外，还建议更改 Central 服务器中自动生成的 KeyStore 的默认“changeit”密码。请参阅对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)。

8. 启动 Central。

将 CA 根证书导入 Central TrustStore

如果针对 Central 使用自定义根证书，则需要将受信任的根证书颁发机构 (CA) 导入 **client.truststore** 中。如果使用知名根 CA（如 Verisign），则不需要执行以下过程，因为证书已位于 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全原因，建议将此默认值更改为自定义 CA 或知名 CA。

替换用 **<黄色>** 突出显示的参数。

> **备注:** 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 如果受信任的根证书颁发机构 (CA) 在 CA 列表中（默认情况下，所有知名 CA 均位于此处）不存在，请将其导入到 Central **client.truststore** 文件中。

   ```
   keytool -importcert -alias <任何别名> -keystore <client.truststore 的路径> -
   file <证书名称.cer> -storepass <changeit>
   ```

3. 启动 Central。

将 CA 根证书导入 RAS TrustStore

在安装 RAS 后，如果针对 Central 使用自定义根证书，但是在安装 RAS 期间没有提供此根证书，则需要将受信任的根证书颁发机构 (CA) 导入到 RAS **client.truststore**。如果使用知名根 CA（如 Verisign），则不需要执行以下过程，因为证书已位于 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全原因，建议将此默认值更改为自定义 CA 或知名 CA。

替换用 **<黄色>** 突出显示的参数。

> **备注:** 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 RAS 并备份位于 **<安装目录>/ras/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 在 **<安装目录>/ras/var/security** 中打开命令行。
3. 打开 **<安装目录> ras/conf/ras-wrapper.conf** 文件并确保 `-Dssl.support-self-signed` 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 打开 **<安装目录> ras/conf/ras-wrapper.conf** 文件并确保 `-Dssl.verifyHostName` 设置为 **true**。 这样可验证证书中的 FQDN 是否与请求的 FQDN 匹配。

   例如：

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

   备注: 默认情况下此属性设置为 **true**。

5. 如果受信任的根证书颁发机构 (CA) 在 CA 列表中（默认情况下，所有知名 CA 均位于此处）不存在，请将其导入到 RAS **client.truststore** 文件中。

   ```
   keytool -importcert -alias <任何别名> -keystore <client.truststore 的路径> -file <证书名称.cer> -storepass <changeit>
   ```

6. 启动 RAS。

将 CA 根证书导入 OOSH TrustStore

如果针对 Central 使用自定义根证书，则需要将受信任的根证书颁发机构 (CA) 导入 OOSH **client.truststore** 中。如果使用知名根 CA（如 Verisign），则不需要执行以下过程，因为证书已位于 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全原因，建议将此默认值更改为自定义 CA 或知名 CA。

替换用 **<黄色>** 突出显示的参数。

备注: 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 并备份位于 **<安装目录>/central/var/security/client.truststore** 中的原始 **client.truststore** 文件。
2. 编辑 **<安装目录>/central/bin** 中的 **oosh.bat**。

3. 确保 -Dssl.support-self-signed 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 确保 -Dssl.verifyHostName 设置为 **true**。 这样可验证证书中的 FQDN 是否与请求的 FQDN 匹配。

   例如：

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

   > **备注:** 默认情况下此属性设置为 **true**。

5. 如果受信任的根证书颁发机构 (CA) 在 CA 列表中（默认情况下，所有知名 CA 均位于此处）不存在，请将其导入到 Central **client.truststore** 文件中。

   ```
   keytool -importcert -alias <任何别名> -keystore <client.truststore 的路径> -file <证书名称.cer> -storepass <changeit>
   ```

6. 运行 OOSH。

7. 启动 Central。

将 CA 根证书导入 Studio Debugger TrustStore

在安装 Studio 后，如果针对 Studio 使用自定义根证书，则需要将受信任的根证书颁发机构 (CA) 导入 Studio **client.truststore** 中。如果使用知名根 CA（如 Verisign），则不需要执行以下过程，因为证书已位于 **client.truststore** 文件中。

默认情况下，HP OO 支持所有自签名证书。但是，在生产环境中，出于安全原因，建议将此默认值更改为自定义 CA 或知名 CA。

对于新 **.oo** 文件夹，Studio 将 **client.truststore** 文件从 **<安装目录>/studio/var/security** 复制到 **<用户>/.oo** 文件夹。这是一次性动作，目的是确保 Studio 可以自动导入证书（例如针对 Studio Remote Debugger）。如果此文件存在，Studio 会将此文件用作 **client.truststore**；否则，它将使用 Studio 安装（**<安装目录>/studio/var/security/client.truststore**）中的文件。

如果您希望手动导入证书，则可以将其导入到 **.oo/client.truststore** 或 Studio 安装文件夹中的 **client.truststore**。

> **备注:** 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 关闭 Studio。如果准备导入到安装文件夹中的 **client.trustore** 文件，请备份原始文件。

2. 编辑 **<安装目录>/studio** 中的 **Studio.l4j.ini** 文件。

3. 确保 -Dssl.support-self-signed 值设置为 **false**。这将启用受信任的根证书颁发机构 (CA)。

   例如：

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

4. 确保 -Dssl.verifyHostName 设置为 **true**。 这样可验证证书中的 FQDN 是否与请求的 FQDN 匹配。

   例如：

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

5. 如果受信任的根证书颁发机构 (CA) 不在 CA 列表中（默认情况下，所有知名 CA 均位于此处），则将其导入到 Studio **client.truststore** 文件中。替换用 **<黄色>** 突出显示的参数：

   ```
   keytool -importcert -alias <任何别名> -keystore <client.truststore 的路径> -file <证书名称.cer> -storepass <changeit>
   ```

6. 启动 Studio。

有关详细信息，请参阅《Studio 创建指南》中的“使用 Studio 调试远程 Central”。

对 KeyStore/TrustStore 密码进行更改和加密/模糊处理

更改 Central 配置中的 KeyStore、TrustStore 和服务器证书密码

1. 确保 Central 正在运行。

   > **备注:** 执行此步骤之前，请确保密码已加密。有关如何对密码进行加密的信息，请参阅《HP OO 安装、升级和配置指南》中的“对密码进行加密”。

   从 OOSH，运行以下命令：

   ```
   set-sys-config --key <键名称> --value <已加密的密码>
   ```

   其中 <键名称> 是下表中的值之一：

配置项	动作

key.store.password	可设置用于访问 **key.store** 的密码。默认值为“changeit”。 此值需要与以下步骤中设置的 keystorePass 的值对应。
key.store.private.key.alias.password	可设置用于从 **key.store** 访问服务器证书（私钥）的密码。默认值为“changeit”。 此值需要与以下步骤中设置的 keyPass 的值对应。

2. 停止 Central 服务。
3. 使用 Keytool 更改 KeyStore、TrustStore 和服务器证书密码。

   使用以下 keytool 命令更改 KeyStore 密码：

   ```
   keytool -storepasswd -keystore <安装文件夹>/central/var/security/key.store
   ```

   使用以下 keytool 命令更改服务器证书私钥输入密码：

   ```
   keytool –keypasswd -alias tomcat -keystore <安装文件夹
   >/central/var/security/key.store
   ```

   使用以下 keytool 命令更改 TrustStore 密码：

   ```
   keytool -storepasswd -keystore <安装文件夹
   >/central/var/security/client.truststore
   ```

4. 此外，还可以在位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件中更改这些密码。

   a. 找到 HTTPS 连接器。例如：

   ```
   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"/>
   ```

   更改所需的密码。

- keyPass - 用于从指定的 key.store 文件访问服务器证书私钥的密码。默认值为“changeit”。
- keystorePass - 用于访问指定的 key.store 文件的密码。默认值是 keyPass 属性的值。

  备注: 建议不要使用与 **keyPass** 相同的密码，而是使用强密码。

- truststorePass - 用于访问 TrustStore（包含所有受信任的 CA）的密码。默认值是 **javax.net.ssl.trustStorePassword** 系统属性的值。如果该属性为 null，则不配置 TrustStore 密码。如果指定了无效的 TrustStore 密码，则将记录警告并尝试不用密码访问 TrustStore，这样将跳过对 TrustStore 内容的验证。

b. 保存文件。

5. 编辑位于 **<安装目录> central\conf\central** 中的 **central-wrapper.conf** 文件，将 TrustStore 的密码替换为已加密或模糊处理形式的新密码。示例：

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.trustStorePassword={ENCRYPTED}<已加密的密码>
   ```

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.trustStorePassword={OBFUSCATED}<已模糊处理的密码>
   ```

   有关如何对密码进行加密或模糊处理的信息，请参阅对密码进行加密和模糊处理 (第 41 页)。

6. 启动 Central 服务。

更改 RAS、OOSH 和 Studio TrustStore 密码

备注: 完成以下步骤之前，应先使用 Keytool 更改 KeyStore、TrustStore 和服务器证书密码。

- **要更改独立 RAS TrustStore 密码，请执行以下操作**：编辑 **ras-wrapper.conf** 文件，并更改 TrustStore 的密码。
- **要更改 OOSH TrustStore 密码，请执行以下操作**：编辑 **oosh.bat** 文件，并更改 TrustStore 的密码。
- **要更改 Studio TrustStore 密码，请执行以下操作**：将带有已模糊处理的密码的属性 **client.truststore.password** 添加到 **<用户>/.oo** 文件夹中的 **Studio.properties** 文件。

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

  如果未定义此属性，Studio 将回退到系统属性 **javax.net.ssl.trustStorePassword** 以获取 TrustStore 密码。

有关如何对密码进行模糊处理的信息，请参阅对密码进行加密和模糊处理 (第 41 页)。

对密码进行加密和模糊处理

您可以使用 encrypt-password 脚本（位于 **<安装文件夹>/central/bin** ）对密码进行加密或模糊处理。

我们建议使用加密。

重要事项！使用 encrypt-password 脚本后，请清除命令历史记录。

这是因为在 Linux OS 中，密码参数将以纯文本形式存储在 **/$USER/.bash_history** 下，并可通过 history 命令进行访问。

对密码进行加密

1. 在 **<安装文件夹>/central/bin** 中找到 encrypt-password 脚本。
2. 使用 -e -p <密码> 选项运行该脚本，其中密码是要加密的密码。

   备注: 可以使用 -p 作为对密码加密的标志或使用 --password。

   已加密的密码应如下所示：

   ```
   {ENCRYPTED}<some_chars>。
   ```

对密码进行模糊处理

1. 在 **<安装文件夹>/central/bin** 中找到 encrypt-password 脚本。
2. 使用 -o <密码> 选项运行该脚本，其中密码是要进行模糊处理的密码。

   已模糊处理的密码应如下所示：

   ```
   {OBFUSCATED}<some_chars>。
   ```

创建密码的提示

建议在不提供 –p 参数的情况下运行 encrypt-password 脚本。例如：

```
C:\Program Files\Hewlett-Packard\HP Operations Orchestration\central\bin>encrypt-password.bat
Password (typing will be hidden):
Confirm password (typing will be hidden):
{ENCRYPTED}gAkPCLQsYDhoR1Y2q9BjCQ==

C:\Program Files\Hewlett-Packard\HP Operations Orchestration\central\bin>
```

这将创建隐藏密码输入的提示。

从支持 SSL 的密码中删除 RC4 密码

远程主机支持使用 RC4 密码。此密码在生成伪随机字节流时有缺陷，因此流中包含各种小偏差，从而降低了随机性。

如果纯文本重复加密（例如 HTTP cookie），并且攻击者能够获得许多（即数千万）密码文本，则攻击者可能会派生纯文本。

禁用 JRE 级别 RC4 密码（从 Java 7 开始）：

1. 打开 **$JRE_HOME/lib/security/java.security** 文件。
2. 根据以下示例，通过删除注释和更改参数来禁用 RC4 密码：

```
jdk.certpath.disabledAlgorithms=RC4, MD2, RSA keySize < 1024
jdk.tls.disabledAlgorithms=RC4, MD5, DSA, RSA keySize < 1024
```

3. 重新启动 HP OO Central 服务器。

有关详细信息，请访问 http://stackoverflow.com/questions/18589761/restict-cipher-suites-on-jre-level。

备注: 从 HP OO 10.x 早期版本升级后，请重复执行以上步骤。

更改 HTTP/HTTPS 端口或禁用 HTTP 端口

在 **[OO 主目录]/central/tomcat/conf** 下的 **server.xml** 文件中，**<Service>** 元素下包含两个名为 **<Connector>** 的元素。这些连接器定义或启用服务器侦听的端口。

通过连接器属性定义每个连接器的配置。第一个连接器定义常规 HTTP 连接器，第二个连接器则定义 HTTPS 连接器。

默认情况下，连接器类似于如下内容。

HTTP 连接器：

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

HTTPS 连接器：

```
<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"/>
```

默认情况下，同时启用这两种连接器。

重要事项！如果在 **server.xml** 文件中更改或禁用其中一个 Central 端口，则还将需要更新 **central-wrapper.conf** 文件和每个 **RAS-wrapper.conf** 文件，以指向端口已更新的 Central URL。否则，从 Central 运行所有流时均将失败。此外，还要确保检查负载平衡器配置。

更改端口值

要更改其中一个端口的值，请执行以下操作：

1. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
2. 找到 HTTP 或 HTTPS 连接器，并调整行中的 **port** 值。

   备注: 如果将 HTTP 和 HTTPS 保留为活动状态，并且想更改 HTTPS 端口，则将需要更改 HTTP 连接器的 **redirectPort** 值和 HTTPS 连接器的 **port** 值。
3. 保存文件。
4. 重新启动 Central。

禁用 HTTP 端口

出于安全原因，您可能需要禁用 HTTP 端口，以便唯一的通信通道将使用 TLS 并进行加密。

1. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
2. 找到 HTTP 连接器，并删除或注释掉行。
3. 如果受信任的根证书颁发机构 (CA) 在 CA 列表中不存在，则将其导入到 Central **client.truststore** 文件中：

   ```
   keytool -importcert -alias <任何别名> -keystore <client.truststore 的路径> -
   file <证书名称.cer> -storepass <changeit>
   ```

   备注: 如果使用知名根 CA（如 Verisign），则不需要执行此步骤，因为证书已位于

client.truststore 文件中。

4. 保存文件。
5. 重新启动 Central。

备注: 还可以在安装期间禁用 HTTP 端口。

疑难解答

如果服务器没有启动，请打开 **wrapper.log** 文件并在 `ProtocolHandler ["http-nio-8443"]` 中查找错误。

当 Tomcat 正在初始化或正在启动连接器时，会发生这种问题。存在很多变体，但错误消息可提供相关信息。

所有 HTTPS 连接器参数均位于 **C:\HP\oo\central\tomcat\conf\server.xml** 的 Tomcat 配置文件中。

打开此文件并滚动到最后，直到看到 HTTPS 连接器：

```
<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"/>
```

通过将这些参数与在先前步骤中输入的参数进行比较，查看参数是否存在任何不匹配的情况。

客户端证书身份验证（相互身份验证）

X.509 证书身份验证的最常见用途是在使用 TLS 时验证服务器的标识，通常是在浏览器中使用 HTTPS 时验证服务器的标识。浏览器会自动检查其维护的受信任证书颁发机构列表中的机构是否已经颁发了服务器显示的证书。

您也可以使用 TLS 进行相互身份验证。服务器将请求来自客户端的有效证书以用作 TLS 握手协议的一部分。服务器通过检查客户端证书是否由可接受的颁发机构签名来对客户端进行身份验证。如果提供了有效证书，则可以通过应用程序中的 servlet API 获得该证书。

在 Central 中配置客户端证书身份验证

在 Central 中配置客户端证书身份验证之前，请确保按使用服务器和客户端证书 (第 32 页)中所述配置了 TLS 服务器证书。

如果希望 TLS 堆栈在接受连接前要求从客户端获得有效证书链，请将 `clientAuth` 属性设置为 `true`。如果希望 TLS 堆栈请求客户端证书但在未提供证书时不失败，则设置为 `want`。`false` 值

（默认值）不要求证书链，除非客户端请求了受使用 CLIENT-CERT 身份验证的安全约束保护的资源。（有关详细信息，请参阅《Apache Tomcat Configuration Reference》。）

设置“证书吊销列表 (CRL)”文件。这包含多个 CRL。在部分加密系统（通常为公钥基础结构 (PKI)）的操作中，证书吊销列表 (CRL) 是已吊销的证书列表（更具体地说，是证书序列号的列表），因此应当不再信任表示这些（已吊销）证书的实体。

备注: 以下过程使用位于 **<安装目录>/java/bin/keytool** 中的 Keytool 实用程序。

1. 停止 Central 服务器。
2. 如果相应的根证书 (CA) 在 CA 列表中（默认情况下，所有知名 CA 均位于此处）不存在，请将其导入到 Central **client.truststore**：**<安装目录>/central/var/security/client.truststore**。例如：

   ```
   keytool -importcert -alias <任何别名> -keystore <path>/client.truststore -
   file <证书路径> -storepass <changeit>
   ```

3. 编辑位于 **<安装目录>/central/tomcat/conf/server.xml** 中的 **server.xml** 文件。
4. 将 `Connector` 标签中的 `clientAuth` 属性设置为 `want` 或 `true`。默认值为 `false`。

   例如：

   ```
   <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"/>
   ```

   备注: 建议在此过程结束时启动服务器，但是请注意，还可以在此时启动服务器。

5. （可选）添加 `crlFile` 属性以定义用于 TLS 证书验证的证书吊销列表文件，例如：

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

   文件可以带 `.crl` 扩展名以表示单个证书吊销列表，也可以带 `.pem`（PEM CRL 格式）扩展名以表示一个或多个证书吊销列表。PEM CRL 格式使用以下页眉和页脚行：

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

一个 CRL 的 .pem 文件结构示例（至于多个 CRL，需连接其他 CRL 块）：

```
-----BEGIN X509 CRL-----
MIIBbzCB2QIBATANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJVUzEYMBYGA1UE
ChMPVS5TLiBHb3Zlcm5tZW50MQwwCgYDVQQLEwNEb0QxEDAOBgNVBAsTB1Rlc3Rp
bmcxFTATBgNVBAMTDFRydXN0IEFuY2hvchcNOTkwMTAxMTIwMTAwWhcNNDgwMTAx
MTIwMTAwWjAiMCACAScXDTk5MDEwMTEyMDAwMFowDDAKBgNVHRUEAwoBAaAjMCEw
CgYDVR0UBAMCAQEwEwYDVR0jBAwwCoAIq5rr+cLnVI8wDQYJKoZIhvcNAQEFBQAD
gYEAC7lqZwejJRW7QvzH11/7cYcL3racgMxH3PSU/ufvyLk7ahR++RtHary/WeCv
RdyznLiIOA8ZBiguWtVPqsNysNn7WLofQIVa+/TD3T+lece4e1NwGQvj5Q+e2wRt
GXg+gCuTjTKUFfKRnWz7O7RyiJKKim0jtAF4RkCpLebNChY=
-----END X509 CRL-----
```

6. 编辑位于 **<安装目录> central\conf\central** 中的 **central-wrapper.conf** 文件。

   以管理员用户身份取消注释以下属性，并设置客户端证书位置和客户端证书密码。

```
#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
```

   有关如何对密码进行加密或模糊处理的信息，请参阅对密码进行加密和模糊处理 (第 41 页)。

7. 启动 Central 服务器。

备注: 对于每个客户端证书，您需要定义用户（内部用户或 LDAP 用户）。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。

请注意，即使已将 HP OO 设置为使用多个 LDAP 配置，仍然只能使用默认 LDAP 通过客户端证书属性对用户进行身份验证。

更新 RAS 中客户端证书的配置

客户端证书是在 RAS 安装期间配置的。但是，如果需要更新客户端证书，则可以在 **ras-wrapper.conf** 文件中手动执行此操作。

先决条件：必须将 Central 的 CA 根证书导入到 RAS TrustStore。请参阅将 CA 根证书导入 RAS TrustStore (第 35 页)。

要在外部 RAS 中更新客户端证书的配置，请执行以下操作：

1. 停止 RAS 服务器。
2. 在 **<安装目录>ras/conf/ras-wrapper.conf** 中打开 **ras-wrapper.conf** 文件。
3. 根据您的客户端证书更改以下内容：

   ```
   wrapper.java.additional.<x>=-Djavax.net.ssl.keyStore=<安装目录
   >/var/security/certificate.p12"

   wrapper.java.additional.<x>=-Djavax.net.ssl.keyStorePassword={OBFUSCATED}<已
   模糊处理的密码>

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

4. 启动 RAS 服务器。

重要说明！ X.509 客户端证书需要具有 RAS 的主体名称，即 RAS ID（请参阅处理证书主体）。

可以在 Central 的“拓扑”选项卡下找到 RAS ID。请参阅《HP OO Central 用户指南》中的“设置拓扑 – 工作程序”。

在 HP OO 10.20 及更高版本中，如果将密码保留为默认值，则默认情况下将对 `keyStorePassword` 参数进行模糊处理。您可以更改此参数并以纯文本格式或以模糊处理的方式进行存储。请参阅对密码进行加密和模糊处理 (第 41 页)。

在 Studio Remote Debugger 中配置客户端证书

先决条件：必须将 Central 的 CA 根证书导入到 Studio Debugger TrustStore。请参阅将 CA 根证书导入 Studio Debugger TrustStore (第 37 页)。

要在 Studio Remote Debugger 中配置客户端证书，请执行以下操作：

1. 关闭 Studio。
2. 编辑 **<安装目录>/studio** 中的 **Studio.l4j.ini** 文件。
3. 根据您的客户端证书更改以下内容：

   ```
   -Djavax.net.ssl.keyStore="<安装目录>/studio/var/security/certificate.p12"

   -Djavax.net.ssl.keyStorePassword={OBFUSCATED}<已模糊处理的密码>

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

4. 启动 Studio。

注意：

- 在 HP OO 10.20 及更高版本中，如果将密码保留为默认值，则默认情况下将对 keyStorePassword 参数进行模糊处理。您可以更改此参数并以纯文本格式或以模糊处理的方式进行存储。请参阅对密码进行加密和模糊处理 (第 41 页)。
- 对于客户端证书，您需要定义用户（内部用户或 LDAP 用户）。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。
- 请注意，即使已将 HP OO 设置为使用多个 LDAP 配置，仍然只能使用默认 LDAP 通过客户端证书属性对用户进行身份验证。Central 将首先尝试使用默认的 LDAP 对用户进行身份验证，如果失败，则将尝试在 HP OO 内部域中进行身份验证。

在 OOSH 中配置客户端证书

先决条件：必须将 Central 的 CA 根证书导入到 OOSH TrustStore。请参阅将 CA 根证书导入 OOSH TrustStore (第 36 页)。

1. 停止 OOSH。
2. 编辑 **<安装目录>/central/bin** 中的 **oosh.bat**。
3. 根据您的客户端证书更改以下内容：

```
-Djavax.net.ssl.keyStore="<安装目录>/var/security/certificate.p12"
-Djavax.net.ssl.keyStorePassword={OBFUSCATED}<已模糊处理的密码>
-Djavax.net.ssl.keyStoreType=PKCS12
```

4. 启动 OOSH。

备注:

在 HP OO 10.20 及更高版本中，如果将密码保留为默认值，则默认情况下将对 keyStorePassword 参数进行模糊处理。您可以更改此参数并以纯文本格式或以模糊处理的方式进行存储。请参阅对密码进行加密和模糊处理 (第 41 页)。

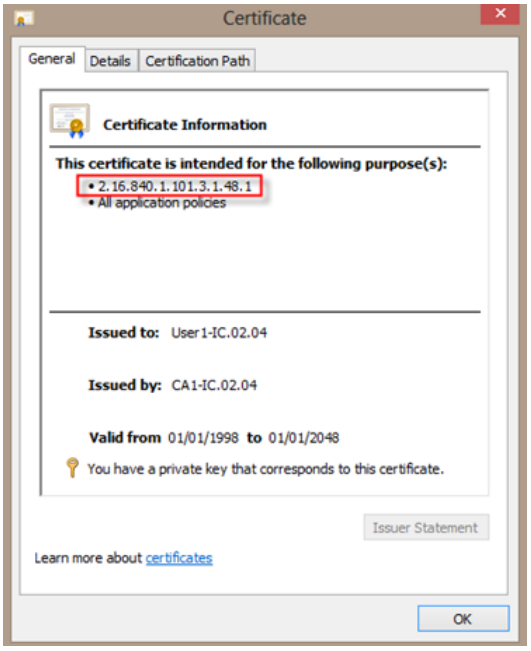
对于客户端证书，您需要定义用户（内部用户或 LDAP 用户）。用户的名称应该在证书属性中定义。默认为 CN 属性的值。有关更多详细信息，请参阅处理证书主体。

请注意，即使已将 HP OO 设置为使用多个 LDAP 配置，仍然只能使用默认 LDAP 通过客户端证书属性对用户进行身份验证。Central 将首先尝试使用默认的 LDAP 对用户进行身份验证，如果失败，则将尝试在 HP OO 内部域中进行身份验证。

处理证书策略

HP OO 将处理对端点证书的证书策略所进行的处理。

- 可以在证书中设置目的字符串。
- HP OO 允许您将策略字符串添加为配置项，并检查每个端点证书的策略字符串。如果不匹配，则拒绝证书。
- 通过添加以下配置项启用或禁用证书策略验证：`x509.certificate.policy.enabled=true/false`（默认值为 `false`）。
- 通过添加以下配置项定义策略列表：`x509.certificate.policy.list=<逗号分隔的列表`（默认为空列表）。



有关如何更改 HP OO 系统属性的详细信息，请参阅《HP OO Shell User Guide》。

处理证书主体

您可以定义如何使用针对 `Subject` 的正则表达式匹配来从证书中获得主体。正则表达式应包含单个组。默认表达式 `CN=(.?)` 与常用名字段匹配。例如，`CN=Jimi Hendrix, OU=` 分配用户名 `Jimi Hendrix`。

- 匹配不区分大小写。
- 证书的主体是 HP OO 中的用户名（LDAP 或内部用户）。
- 要更改正则表达式，请更改配置项：`x509.subject.principal.regex`。

有关如何更改 HP OO 系统属性的详细信息，请参阅《HP OO Shell (OOSH) User Guide》。

配置 HP OO 以兼容 FIPS 140-2 1 级

以下部分说明如何配置 HP Operations Orchestration 以兼容联邦信息处理标准 (FIPS) 140-2 1 级。

FIPS 140-2 是由国家标准和技术研究所 (NIST) 定义的用于加密模块的安全要求标准。要查看此标准的出版物，请转至：csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf。

在配置 HP OO 以兼容 FIPS 140-2 后，HP OO 使用以下安全算法：

- 对称密钥算法：AES256
- 哈希算法：SHA256

HP OO 使用此安全提供程序：RSA BSAFE Crypto 6.1 版软件。该软件是唯一支持的 FIPS 140-2 安全提供程序。

备注: 在配置 HP OO 以兼容 FIPS 140-2 后，除非重新安装 HP OO，否则无法还原到标准配置。

先决条件

升级者注意事项:

如果从已配置使用 FIPS 的 HP OO 10.10（及更高版本）的安装进行升级，请参阅升级者的先决条件步骤。

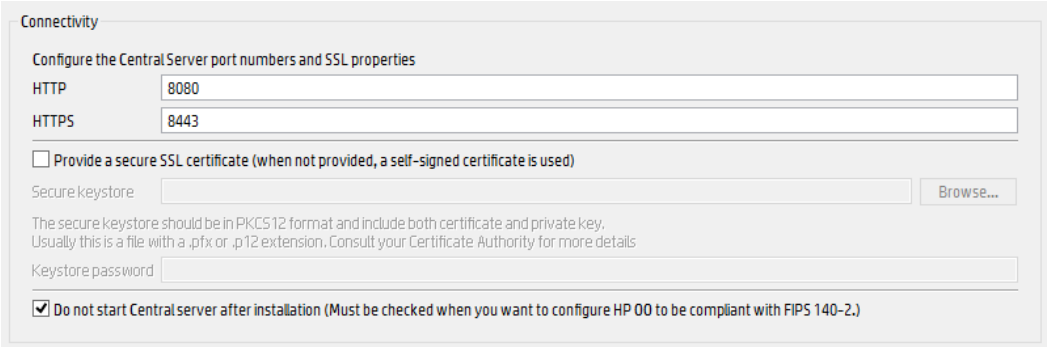
在配置 HP OO 以兼容 FIPS 140-2 之前，请执行以下步骤：

备注: 为了兼容 FIPS140-2，您需要关闭 LWSSO。

1. 验证是否正在配置新安装的 HP OO 版本 10.10 或更高版本以与 FIPS 140-2 兼容，以及该版本是否未在使用中。

   您不能配置正在使用的 HP OO 安装（不管是版本 9.x 还是 10.x）。

2. 验证在安装 HP OO 时是否未将其配置成安装后启动 Central 服务器：
   - 在静默安装中，`should.start.central` 参数设置为 **no**。
   - 在向导安装中的“Connectivity”步骤中，选中了“Do not start Central server after installation”复选框。



3. 备份以下目录：

   - **<安装目录>\central\tomcat\webapps\oo.war**
   - **<安装目录>\central\tomcat\webapps\PAS.war**
   - **<安装目录>\central\conf**
   - **<安装目录>\java**（应备份整个 **java** 文件夹）

4. 从 http://www.oracle.com/technetwork/java/javase/downloads/server-jre8-downloads-2133154.html 下载 **Server Oracle JRE 8**，将 **OpenJDK (Zulu) JRE** 替换为 **Server Oracle JRE**。

   a. 删除 **<安装目录>\JAVA** 文件夹中的所有内容。

   b. 解压缩下载的存档。

   c. 将 **JRE** 文件夹内容复制到 **<安装目录>\JAVA**。

5. 在以下站点上下载并安装 Java Cryptographic Extension (JCE) Unlimited Strength Jurisdiction 策略文件：

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

   备注: 有关如何部署文件和升级 HP OO 所使用的 JRE 的信息，请参阅下载内容中的 **ReadMe.txt** 文件。

6. 安装 RSA BSAFE Crypto 软件文件。在安装了 HP OO 的系统上，将以下内容复制到 **<oo_jre>\lib\ext** 中（其中 **<oo_jre>** 是安装 HP OO 时使用的 JRE 所在的目录。默认为 **<安装目录>\java**）。

   - **<安装目录>\central\lib\cryptojce-6.1.jar**
   - **<安装目录>\central\lib\cryptojcommon-6.1.jar**

- **<安装目录>\central\lib\jcmFIPS-6.1.jar**

升级者的先决条件步骤

1. 下载 Server Oracle JRE 8，将 OpenJDK (Zulu) JRE 替换为 Server Oracle JRE。

   a. 删除 **<升级目录>\JAVA** 文件夹中的所有内容。

   b. 解压缩下载的存档。

   c. 将 **JRE** 文件夹内容复制到 **<升级目录>\JAVA**。

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

2. 在以下站点上下载并安装 Java Cryptographic Extension (JCE) Unlimited Strength Jurisdiction 策略文件：

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

   有关如何部署文件和升级 HP OO 所使用的 JRE 的信息，请参阅下载内容中的 **ReadMe.txt** 文件。

3. 安装 RSA BSAFE Crypto 软件文件。在安装 HP OO 的系统上，将以下文件复制到 **<oo_jre>\lib\ext**：

   （其中 **<oo_jre>** 是 HP OO 升级者使用的 JRE 所在的目录。默认为 **<升级目录>\java**。）

   - <安装目录>\central\lib\cryptojce-6.1.jar
   - <安装目录>\central\lib\cryptojcommon-6.1.jar
   - <安装目录>\central\lib\jcmFIPS-6.1.jar

接下来，按照配置 HP OO 以兼容 FIPS 140-2 (第 52 页)的“配置 Java 安全文件中的属性”中的步骤进行操作。

配置 HP OO 以兼容 FIPS 140-2

以下列表显示为了配置 HP OO 以兼容 FIPS 140-2 而需要执行的过程：

1. 配置 Java 安全文件中的属性。
2. 配置 encryption.properties 文件并启用 FIPS 模式。
3. 创建兼容 FIPS 的 HP OO 加密。

4. 使用新加密对数据库密码进行重新加密。
5. 启动 HP OO。

配置 Java 安全文件中的属性

编辑 JRE 的 Java 安全文件以添加其他安全提供程序，并为兼容 FIPS 140-2 配置属性。

备注: 升级到 HP OO 10.x 会完全替换已安装的 JRE 文件。因此，如果您要升级到 10.x，则必须完成以下步骤。

备注: 如果从已配置使用 FIPS 的 HP OO 10.10 及更高版本的安装进行升级，则必须按照配置 HP OO 以兼容 FIPS 140-2 1 级 (第 50 页)中的“升级者先决条件步骤”部分操作，然后按此处的步骤操作，其中 **<oo_jre>** 是升级中包括的 JRE （位于 **<升级目录>\JAVA** 中）。

确保在解压缩的 **upgrade** 文件夹内的 **java** 文件夹中进行所有更改。

在编辑器中打开 **<oo_jre>\lib\security\java.security** 文件并执行以下步骤：

1. 对于每个列出的提供程序，以 **security.provider.<nn>=<提供程序名称>** 格式递增首选项顺序编号 <nn>，每次递增 2。

   例如，将提供程序条目从

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

   更改为

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

2. 添加新的默认提供程序 (RSA JCE)。在提供程序列表顶部添加以下提供程序：

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

3. 添加 RSA BSAFE SSL-J Java 安全套接字扩展 (JSSE) 提供程序。

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

4. 将以下行复制并粘贴到 **java.security** 文件中以确保在兼容 FIPS 140-2 模式下使用 **RSA BSAFE**：

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

   可以在 **java.security** 文件中的任意位置粘贴此行。

5. 由于默认的 DRBG 算法 ECDRBG128 并不安全（根据 NIST），可通过将以下行复制到 **java.security** 文件中，将安全属性 **com.rsa.crypto.default** 设置成 **HMACDRBG**：

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

可以在 **java.security** 文件中的任意位置粘贴此行。

6. 保存并退出 **java.security** 文件。

配置 encryption.properties 文件并启用 FIPS 模式

必须更新 HP OO 加密属性文件以便兼容 FIPS 140-2。

1. 备份位于 **<安装目录>\central\var\security** 中的 **encryption.properties** 文件。
2. 在文本编辑器中打开 **encryption.properties** 文件。例如，编辑以下文件：

   C:\Program Files\Hewlett-Packard\HP Operations Orchestration\central\var\security\encryption.properties。

3. 找到 `keySize=128` 并将其替换为 `keySize=256`。
4. 找到 `secureHashAlgorithm=SHA1` 并将其替换为 `secureHashAlgorithm=SHA256`。
5. 找到 `FIPS140ModeEnabled=false` 并将其替换为 `FIPS140ModeEnabled=true`。

   备注: 如果 `FIPS140ModeEnabled=false` 不存在，请将 `FIPS140ModeEnabled=true` 作为新行添加到文件末尾。

6. 保存并退出文件。

创建兼容 FIPS 的 HP OO 加密

要创建或替换 HP OO 加密存储文件以便兼容 FIPS，请参阅替换 FIPS 加密 (第 55 页)。

备注: AES 有三种批准的密钥长度：128/192/256（根据 NIST SP800-131A 出版物）。

FIPS 支持以下安全哈希算法：SHA1、SHA256、SHA384、SHA512。

备注: 建议更改 key.store（及其私钥条目）和 TrustStore 的密码。请参阅对 KeyStore/TrustStore 密码进行更改和加密/模糊处理 (第 38 页)

备注: 建议删除 HP OO TrustStore 中所有未使用的默认 CA 根证书。（**client.truststore** 位于 **<安装>/central/var/security** 中。）

备注: 如果使用客户端证书，该证书应当用兼容 FIPS 的 RSA JCE 提供程序和 FIPS 中支持的安全哈希算法（如上所述）生成。

使用新加密对数据库密码进行重新加密

按照《HP OO Administration Guide》的“Changing the Database Password”中所述，对数据库密码进行重新加密。

启动 HP OO

替换 FIPS 加密

HP OO Central 和 RAS 符合联邦信息处理标准 140-2 (FIPS 140-2)，此标准定义了联邦机构为保护敏感数据或有价值数据而指定基于加密的安全系统时要使用的技术要求。

进行 HP OO 的全新安装后，可选择更改 FIPS 加密密钥。

备注: 此过程仅适用于全新安装。升级后无法执行此过程。

更改 Central 上的 FIPS 加密密钥

使用 **generate-keys.bat/sh** 文件替换加密存储库中的 FIPS 加密密钥。

备注: 此过程将备份 **encryption_repository** 文件，因此您必须具有相关写权限。

1. 转至 **<Central 安装文件夹>/var/security**。
2. 备份 **encryption_repository** 文件，然后将其从 **<Central 安装文件夹>/var/security** 文件夹中删除。
3. 转至 **<Central 安装文件夹>/bin/**。
4. 运行 **generate-keys** 脚本。
5. 按 **Y** 键继续。

   此时将在 **<Central 安装文件夹>/var/security/encryption_repository** 中生成新的主密钥。

备注: 如果您运行 **generate-keys** 脚本时不希望暂停来让用户输入 **Y** 或 **N**，请在运行该脚本时使用静默模式标志 **– s**。

更改 RAS 加密属性

如果在新位置安装 RAS，则需要完成以下所有步骤。

备注: 在更改 Central 加密属性后，只有当在新 RAS 安装上工作时，这些更改才有效。

要更改 RAS 加密属性，请执行以下操作:

1. 完成配置 HP OO 以兼容 FIPS 140-2 1 级 (第 50 页)的“先决条件”部分中的所有步骤。
2. 完成配置 HP OO 以兼容 FIPS 140-2 (第 52 页)的“配置 Java 安全文件中的属性”中的所有步骤。
3. 将当前 **encryption.properties** 文件从 **<安装目录>\ras\var\security** 复制到 **<安装目录>\ras\bin** 文件夹。
4. 使用任意文本编辑器，根据需要编辑和更改 **encryption.properties** 文件。

   有关详细信息，请参阅配置 HP OO 以兼容 FIPS 140-2 (第 52 页)中的“配置 encryption.properties 文件并启用 FIPS 模式”。
5. 保存变更。
6. 在文件夹 **<安装目录>\ras\bin** 中打开命令行提示符。
7. 运行 **oosh.bat**。
8. 运行 OOShell 命令: `replace-encryption --file encryption.properties`

   备注: 如果将 **encryption.properties** 文件复制到其他文件夹，请确保在 OOShell 命令中输入正确的位置。
9. 重新启动 RAS 服务。

配置 TLS 协议

可配置 HP OO 以定义支持的 TLS 协议版本。默认情况下，HP OO 允许使用 TLS v1、TLS v1.1 和 TLS v1.2，但您可以缩小此范围。

备注: SSLv3 和 SSL 的其他版本不受支持。

1. 打开 **<安装文件夹>/central/tomcat/conf/server.xml** 文件。
2. 找到 SSL 连接器（在文件末尾）。
3. 编辑 `sslEnabledProtocols` 的默认值。例如，

   将 `sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"` 更改为

   `sslEnabledProtocols="TLSv1.2"`
4. 重新启动服务器。

阻止流访问 Central/RAS 本地文件系统

您应当修改 Central 或 RAS 的包装配置和 java.policy 文件，以阻止流访问 Central 或 RAS 本地文件系统和获取敏感资源的访问权。

备注: 若要利用此场景，用户除了流中的授权或能够对流授权之外，还需要具有部署和触发权限。具有此类权限的用户可能是可信用户。

要防止此场景出现，请执行以下操作:

1. 在 Central 或 RAS 的包装配置文件（**<安装文件夹>/<ras/central>/conf/<central/ras>-wrapper.conf**）中，按如下所示添加 `wrapper.java.additional.<nn>` 参数:

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

   将 <nn> 替换为最后一个数字之后的数字。

2. 在 **java.policy** 文件（位于 **<安装文件夹>/java/lib/security/java.policy**）中，添加以下内容：这将允许访问 HP OO 所需的最少资源，并阻止访问包含敏感数据的 Central/RAS 本地文件系统。

   ```
   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";
   };
   ```

   要允许流访问 Central/RAS 本地文件系统中的资源，您应当在 java.policy 中指定此内容。例如:

   ```
   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