



Red Hat Enterprise Linux 9

在 IdM 中管理证书

发布证书、配置基于证书的身份验证和控制证书的有效性

Red Hat Enterprise Linux 9 在 IdM 中管理证书

发布证书、配置基于证书的身份验证和控制证书的有效性

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

管理员使用 X.509 证书来验证用户、主机和服务，以启用数字签名和加密。在 Red Hat Identity Management (IdM) 中，您可以使用集成的或外部证书颁发机构(CA)来管理证书。您可以使用 certmonger 服务、certutil 工具或 Ansible Playbook 来请求和更新证书。要替换 IdM 服务器的 Web 服务器和 LDAP 服务器证书，您必须执行手动操作。管理员可以创建轻量级子 CA 来为特定目的发布证书，如 VPN 网关的用户证书。然后，当不再需要这个 VPN 网关时，管理员可以通过撤销子 CA 的证书来使这个服务的所有证书无效。

目录

对红帽文档提供反馈	6
第 1 章 身份管理中的公钥证书	7
1.1. IDM 中的证书颁发机构	7
1.2. 证书和 KERBEROS 的比较	8
1.3. 使用证书验证 IDM 中用户的优缺点	8
第 2 章 使用集成的 IDM CA 为用户、主机和服务管理证书	10
2.1. 使用 IDM WEB UI 为用户、主机或服务请求新证书	10
2.2. 使用 CERTUTIL 为用户、主机或服务从 IDM CA 请求新证书	11
2.3. 使用 OPENSLL 为用户、主机或服务从 IDM CA 请求新证书	12
2.4. 其他资源	13
第 3 章 使用 ANSIBLE 管理 IDM 证书	14
3.1. 使用 ANSIBLE 为 IDM 主机、服务和用户请求 SSL 证书	14
3.2. 使用 ANSIBLE 撤销 IDM 主机、服务和用户的 SSL 证书	15
3.3. 使用 ANSIBLE 恢复 IDM 用户、主机和服务的 SSL 证书	16
3.4. 使用 ANSIBLE 为 IDM 用户、主机和服务检索 SSL 证书	17
第 4 章 管理 IDM 用户、主机和服务的外部签名证书	19
4.1. 使用 IDM CLI，将外部 CA 发布的证书添加到 IDM 用户、主机或服务	19
4.2. 使用 IDM WEB UI 将外部 CA 发布的证书添加到 IDM 用户、主机或服务中	19
4.3. 使用 IDM CLI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书	20
4.4. 使用 IDM WEB UI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书	21
4.5. 其他资源	21
第 5 章 转换证书格式以和 IDM 一起工作	22
5.1. IDM 中的证书格式和编码	22
5.2. 将外部证书转换来加载到 IDM 用户帐户中	23
5.3. 准备将证书加载到浏览器	25
5.4. IDM 中与证书相关的命令和格式	26
第 6 章 在身份管理中创建和管理证书配置文件	28
6.1. 什么是证书配置文件？	28
6.2. 创建证书配置文件	29
6.3. 什么是 CA 访问控制列表？	30
6.4. 定义 CA ACL 来控制对证书配置文件的访问	30
6.5. 使用证书配置文件和 CA ACL 来发布证书	32
6.6. 修改证书配置文件	33
6.7. 证书配置文件配置参数	34
第 7 章 管理 IDM 中证书的有效性	38
7.1. 管理 IDM CA 发布的现有证书的有效性	38
7.2. 管理 IDM CA 发布的未来证书的有效性	38
7.3. 在 IDM WEBUI 中查看证书的到期日期	38
7.4. 在 CLI 中查看证书的到期日期	39
7.5. 吊销带有集成 IDM CA 的证书	39
7.6. 恢复带有集成 IDM CA 的证书	41
第 8 章 为智能卡验证配置身份管理	43
8.1. 为智能卡验证配置 IDM 服务器	43
8.2. 使用 ANSIBLE 为智能卡验证配置 IDM 服务器	45
8.3. 为智能卡验证配置 IDM 客户端	48

8.4. 使用 ANSIBLE 为智能卡验证配置 IDM 客户端	50
8.5. 在 IDM WEB UI 的用户条目中添加证书	53
8.6. 在 IDM CLI 中向用户条目中添加证书	54
8.7. 安装用来管理和使用智能卡的工具	55
8.8. 准备智能卡并将证书和密钥上传到智能卡	55
8.9. 使用智能卡登录到 IDM	57
8.10. 在 IDM 客户端中使用智能卡验证登录到 GDM	58
8.11. 在 SU 命令中使用智能卡验证	59
第 9 章 为 IDM 中智能卡验证配置 ADCS 发布的证书	60
9.1. 信任配置和证书使用量所需的 WINDOWS 服务器设置	60
9.2. 使用 SFTP 从 ACTIVE DIRECTORY 复制证书	60
9.3. 使用 ADCS 证书为智能卡身份验证配置 IDM 服务器和客户端	61
9.4. 转换 PFX 文件	63
9.5. 安装用来管理和使用智能卡的工具	63
9.6. 准备智能卡并将证书和密钥上传到智能卡	64
9.7. 在 SSSD.CONF 中配置超时	65
9.8. 为智能卡身份验证创建证书映射规则	66
第 10 章 在身份管理中配置证书映射规则	67
10.1. 用于配置身份验证的证书映射规则	67
10.2. IDM 中身份映射规则的组件	67
10.3. 从证书获取数据，以便在匹配规则中使用	68
10.4. 为存储在 IDM 中的用户配置证书映射	69
10.5. 使用 ACTIVE DIRECTORY 域信任的证书映射规则	74
10.6. 为 AD 用户条目包含整个证书的用户配置证书映射	75
10.7. 如果将 AD 配置为将用户证书映射到用户帐户，配置证书映射	77
10.8. 如果 AD 用户条目不包含证书或映射数据，配置证书映射	79
10.9. 将多个身份映射规则合并到一个中	84
10.10. 其他资源	85
第 11 章 使用存储在 IDM 客户端中的证书配置身份验证	86
11.1. 在 WEB UI 中为证书验证配置身份管理服务器	86
11.2. 请求新的用户证书并将其导出到客户端	87
11.3. 确保证书和用户已链接在一起	88
11.4. 配置浏览器以启用证书身份验证	89
11.5. 以身份管理用户的身份使用证书向身份管理 WEB UI 进行身份验证	92
11.6. 配置 IDM 客户端，以启用使用证书对 CLI 进行身份验证	93
第 12 章 使用 IDM CA 续订服务器	94
12.1. IDM CA 续订服务器说明	94
12.2. 更改和重置 IDM CA 续订服务器	95
第 13 章 管理外部签名的 CA 证书	97
13.1. 在 IDM 中从外部签名的 CA 切换到自签名 CA	97
13.2. 在 IDM 中从自签名 CA 切换到外部签名的 CA	98
13.3. 使用外部 CA 续订 IDM CA 续订服务器证书	98
第 14 章 当 IDM 离线时续订已过期的系统证书	101
14.1. 在 CA 续订服务器中续订已过期的系统证书	101
14.2. 在续订后验证 IDM 域中的其他 IDM 服务器	102
第 15 章 如果 WEB 服务器和 LDAP 服务器证书还没有在 IDM 副本上过期，请替换它们	104
第 16 章 如果 WEB 服务器和 LDAP 服务器证书已在整个 IDM 部署中过期，请替换它们	106

第 17 章 在 IDM CA 服务器中生成 CRL	110
17.1. 在 IDM 服务器中停止 CRL 生成	110
17.2. 在 IDM 副本服务器中启动 CRL 生成	110
17.3. 更改 CRL 更新间隔	111
第 18 章 停用执行 CA 续订服务器和 CRL 发布者角色的服务器	113
第 19 章 使用 CERTMONGER 为服务获取 IDM 证书	116
19.1. CERTMONGER 概述	116
19.2. 使用 CERTMONGER 为服务获取 IDM 证书	117
19.3. 请求服务证书的 CERTMONGER 的通信流	118
19.4. 查看由 CERTMONGER 跟踪的证书请求详情	121
19.5. 启动和停止证书跟踪	122
19.6. 手动续订证书	123
19.7. 使 CERTMONGER 恢复跟踪 CA 副本中的 IDM 证书	123
19.8. 使用带有 CERTMONGER 的 SCEP	125
第 20 章 在 IDM 中部署和管理 ACME 服务	129
20.1. IDM 中的 ACME 服务	129
20.2. 在 IDM 中启用 ACME 服务	129
20.3. 在 IDM 中禁用 ACME 服务	130
第 21 章 使用 RHEL 系统角色请求证书	132
21.1. CERTIFICATE RHEL 系统角色	132
21.2. 使用 CERTIFICATE RHEL 系统角色请求一个新的自签名证书	132
21.3. 使用 CERTIFICATE RHEL 系统角色从 IDM CA 请求一个新证书	133
21.4. 使用 CERTIFICATE RHEL 系统角色指定在证书颁发之前或之后要运行的命令	134
第 22 章 将应用程序限制为只信任证书子集	136
22.1. 管理轻量级子 CA	136
22.2. 从 IDM WEBUI 下载子 CA 证书	142
22.3. 为 WEB 服务器和客户端身份验证创建 CA ACL	143
22.4. 使用 CERTMONGER 为服务获取 IDM 证书	146
22.5. 请求服务证书的 CERTMONGER 的通信流	148
22.6. 设置单实例 APACHE HTTP 服务器	151
22.7. 在 APACHE HTTP 服务器中添加 TLS 加密	152
22.8. 在 APACHE HTTP 服务器中设置支持的 TLS 协议版本	154
22.9. 在 APACHE HTTP 服务器中设置支持的密码	155
22.10. 配置 TLS 客户端证书身份验证	156
22.11. 请求新的用户证书并将其导出到客户端	157
22.12. 配置浏览器以启用证书身份验证	159
第 23 章 快速地使相关证书的特定组无效	161
23.1. 在 IDM CLI 中禁用 CA ACL	161
23.2. 禁用 IDM 子 CA	162
第 24 章 使用 IDM HEALTHCHECK 验证证书	163
24.1. IDM 证书 HEALTHCHECK 测试	163
24.2. 使用 HEALTHCHECK 工具检查证书	164
第 25 章 使用 IDM HEALTHCHECK 验证系统证书	166
25.1. 系统证书健康检查测试	166
25.2. 使用 HEALTHCHECK 输出系统证书	167
第 26 章 了解 IDM 内部使用的证书	168
26.1. 关于 IDM 中的内部证书	168

26.2. IDM 内部的证书	168
26.3. IDM 内部证书续订过程	173
26.4. 其他资源	174

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 身份管理中的公钥证书

X.509 公钥证书用于验证身份管理(IdM)中的用户、主机和服务。除了身份验证外，X.509 证书还支持数字签名和加密，来提供隐私性、完整性和不可否认性。

证书包含以下信息：

- 证书验证的主题。
- 签发者，即签署证书的 CA。
- 证书有效性的开始和结束日期。
- 证书的有效使用。
- 主题的公钥。

由公钥加密的消息只能由相应的私钥解密。虽然包含的证书和公钥可以公开发布，但用户、主机或服务必须对其私钥保密。

1.1. IDM 中的证书颁发机构

证书颁发机构在信任层次结构中操作。在带有内部证书颁发机构(CA)的 IdM 环境中，所有 IdM 主机、用户和服务信任由 CA 签名的证书。除了这个根 CA 外，IdM 还支持根 CA 授予其依次签署证书能力的子 CA。通常，此类子 CA 能够签名的证书是特定类型的证书，如 VPN 证书。最后，IdM 支持使用外部 CA。下表显示了在 IdM 中使用独立 CA 的详情。

表 1.1. 在 IdM 中使用集成和外部 CA 的比较

CA 的名称	描述	使用	有用的链接
ipa CA	基于 Dogtag 上游项目的集成 CA	集成的 CA 可以为用户、主机和服务创建、吊销和发布证书。	使用 ipa CA 来请求一个新用户证书，并将其导出到客户端
IdM sub-CAs	从属于 ipa CA 的集成 CA	IdM 子 CA 是 ipa CA 对其授予了签署证书的 CA。通常，这些证书是特定类型的，如 VPN 证书。	将应用程序限制为只信任证书子集
外部 CA	外部 CA 是集成 IdM CA 或其子 CA 以外的 CA。	使用 IdM 工具，您可以将这些 CA 发布的证书添加到用户、服务或主机，以及删除它们。	管理 IdM 用户、主机和服务的外部签名证书

从证书的角度来看，由自签名 IdM CA 签名和外部签名的证书之间没有区别。

CA 的作用包括以下目的：

- 它发布数字证书。
- 通过签署证书，它证明证书中指定的对象拥有一个公钥。主题可以是用户、主机或服务。
- 它可以吊销证书，并通过证书吊销列表(CRL)和在线证书状态协议(OCSP)提供吊销状态。

其他资源

- 请参阅 [规划您的 CA 服务](#)。

1.2. 证书和 KERBEROS 的比较

证书与 Kerberos 票据执行类似的功能。Kerberos 是一种计算机网络身份验证协议，它在票据的基础上工作，来允许节点通过非安全网络进行通信，从而以安全的方式证明它们相互的身份。下表显示了 Kerberos 和 X.509 证书的比较：

表 1.2. 证书和 Kerberos 的比较

特性	Kerberos	X.509
认证	是	是
隐私性	选填	是
完整性	选填	是
涉及的加密类型	对称	非对称
默认有效期	短 (1天)	长 (2年)

默认情况下，身份管理中的 Kerberos 仅确保通信各方的身份。

1.3. 使用证书验证 IDM 中用户的优缺点

在 IdM 中使用证书验证用户的优点包括以下几点：

- 与常规密码相比，智能卡上保护私钥的 PIN 通常不复杂、更容易记住。
- 根据设备的不同，无法导出保存在智能卡上的私钥。这提供了额外的安全性。
- 智能卡可以自动退出登录：IdM 可以配置为在用户从读卡器中移除智能卡时退出用户登录。
- 窃取私钥需要对智能卡的实际访问，这样可以防止智能卡遭受攻击。
- 智能卡验证是一双因素验证的一个示例：它要求您拥有某些东西（卡），知道某些东西(PIN)。
- 智能卡比密码更灵活，因为它们提供可用于其他用途的密钥，如加密电子邮件。
- 在作为 IdM 客户端的共享机器上使用智能卡不会给系统管理员带来额外的配置问题。事实上，智能卡验证对于共享机器来说是一个理想的选择。

在 IdM 中使用证书验证用户的缺点包括以下几点：

- 用户可能会丢失或忘记携带其智能卡或证书，并被有效锁住。
- 多次输错 PIN 可能会导致卡被锁住。
- 通常，在请求与某些安全官或批准人授权之间有一个中间步骤。在 IdM 中，安全官或管理员必须运行 `ipa cert-request` 命令。

- 智能卡和读卡器往往是特定于供应商和驱动程序的：虽然许多读卡器可用于不同的卡，但特定供应商的智能卡可能无法在另一供应商的读卡器或不是为其设计的读卡器类型中工作。
- 证书和智能卡对管理员来说有一个陡峭的学习曲线。

第 2 章 使用集成的 IDM CA 为用户、主机和服务管理证书

要了解更多有关如何使用集成的 CA、**ipa** CA 及其子 CA 管理身份管理(IdM)中证书的信息，请参阅以下部分：

- [使用 IdM Web UI 为用户、主机或服务请求新证书。](#)
- 使用 IdM CLI 为用户、主机或服务从 IdM CA 请求新证书：
 - [使用 certutil 为用户、主机或服务从 IdM CA 请求新证书](#)
 - 对于使用 **certutil** 工具从 IdM CA 请求新用户证书，并将其导出到 IdM 客户端的具体示例，请参阅 [请求新的用户证书并将其导出到客户端](#)。
 - [使用 openssl 为用户、主机或服务从 IdM CA 请求新证书](#)

您还可以使用 **certmonger** 工具为来自 IdM CA 的服务请求新证书。如需更多信息，请参阅 [使用 certmonger 为来自 IdM CA 的服务请求新证书](#)。

先决条件

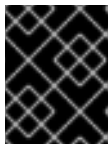
- 您的 IdM 部署包含一个集成的 CA：
 - 有关如何在 IdM 中规划您的 CA 服务的详情，请参考 [规划您的 CA 服务](#)。
 - 有关如何安装带有集成 DNS 和集成 CA 作为 root CA 的 IdM 服务器的详情，请参考 [安装 IdM 服务器：带有集成 DNS，带有集成 CA 作为根 CA](#)
 - 有关如何安装带有集成 DNS 和外部 CA 作为 root CA 的 IdM 服务器的详情，请参考 [安装 IdM 服务器：带有集成 DNS，带有外部 CA 作为根 CA](#)
 - 有关如何安装没有集成 DNS 且集成的 CA 作为根 CA 的 IdM 服务器的详情，请参考 [安装 IdM 服务器：没有集成 DNS，集成 CA 作为根 CA](#)。
 - [可选] 您的 IdM 部署支持使用证书进行用户身份验证：
 - 有关如何配置 IdM 部署以支持使用存储在 IdM 客户端文件系统中的证书进行用户身份验证的详情，请参考 [使用存储在 IdM 客户端桌面上的证书配置身份验证](#)。
 - 有关如何配置 IdM 部署以支持使用存储在插入 IdM 客户端智能卡中的证书进行用户身份验证的详情，请参考 [为智能卡身份验证配置身份管理](#)。
 - 有关如何配置 IdM 部署以支持使用活动目录证书系统发布的智能卡进行用户身份验证的详情，请参考 [为 IdM 中的智能卡身份验证配置由 AD CS 发布的证书](#)。

2.1. 使用 IDM WEB UI 为用户、主机或服务请求新证书

按照以下流程，使用身份管理(IdM) Web UI 为集成 IdM 证书颁发机构(CA)中的任何 IdM 实体请求新证书：**ipa** CA 或其任何子 CA。

IdM 实体包括：

- 用户
- 主机
- 服务



重要

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

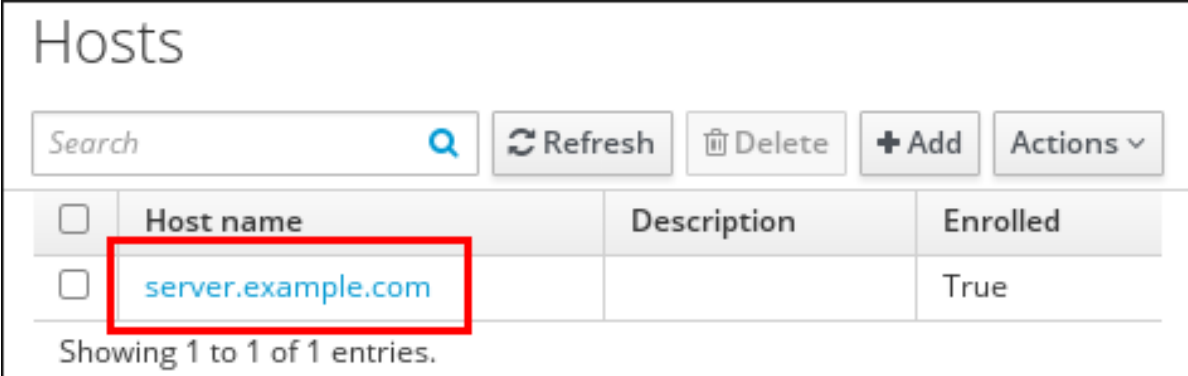
先决条件

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM Web UI。

步骤

1. 在 **Identity** 选项卡下，选择 **Users**、**Hosts** 或 **Services** 子选项卡。
2. 单击用户、主机或服务的名称，来打开其配置页面。

图 2.1. 主机列表



<input type="checkbox"/>	Host name	Description	Enrolled
<input type="checkbox"/>	server.example.com		True

Showing 1 to 1 of 1 entries.

3. 单击 **Actions** → **New Certificate**。
4. 可选：选择发布 CA 和配置文件 ID。
5. 按照屏幕上使用 **certutil** 命令行(CLI)工具的说明进行操作。
6. 单击 **Issue**。

2.2. 使用 CERTUTIL 为用户、主机或服务从 IDM CA 请求新证书

您可以使用 **certutil** 工具为标准 IdM 情况下的身份管理(IdM)用户、主机或服务请求证书。要确保主机或服务 Kerberos 别名可以使用证书，请 [使用 openssl 工具来请求证书](#)。

按照以下流程，使用 **certutil** 为来自 **ipa**、IdM 证书颁发机构(CA)的 IdM 用户、主机或服务请求证书。



重要

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

先决条件

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM 命令行界面(CLI)。

步骤

1. 为证书数据库创建一个临时目录：

```
# mkdir ~/certdb/
```

2. 创建一个新的临时证书数据库，例如：

```
# certutil -N -d ~/certdb/
```

3. 创建 CSR，并将输出重定向到文件。例如，要为 4096 位证书创建 CSR，并将主题设为 `CN=server.example.com,O=EXAMPLE.COM`：

```
# certutil -R -d ~/certdb/ -a -g 4096 -s "CN=server.example.com,O=EXAMPLE.COM" -8
server.example.com > certificate_request.csr
```

4. 将证书请求文件提交到在 IdM 服务器上运行的 CA。指定 Kerberos 主体来与新发布的证书关联：

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM 中的 `ipa cert-request` 命令使用以下默认值：

- **caIPAServiceCert** 证书配置文件
要选择自定义配置文件，请使用 `--profile-id` 选项。
- 集成的 IdM 根 CA **ipa**
要选择子 CA，请使用 `--ca` 选项。

其他资源

- 请参阅 `ipa cert-request --help` 命令的输出。
- 请参阅 [在身份管理中创建和管理证书配置文件](#)。

2.3. 使用 OPENSSSL 为用户、主机或服务从 IDM CA 请求新证书

如果要确保主机或服务的 Kerberos 别名可以使用证书，您可以使用 `openssl` 工具为身份管理(IdM)主机或服务请求证书。在标准情况下，请考虑 [使用 certutil 工具来请求一个新证书](#)。

按照以下流程，使用 `openssl` 为 IdM 主机或来自 `ipa`、IdM 证书颁发机构的服务请求证书。



重要

通常运行在存储私钥的专用服务节点上的服务。将服务的私钥复制到 IdM 服务器被视为不安全。因此，在为服务请求证书时，请在服务节点上创建证书签名请求(CSR)。

先决条件

- 您的 IdM 部署包含一个集成的 CA。
- 以 IdM 管理员身份登录到 IdM 命令行界面(CLI)。

步骤

1. 为您的 Kerberos 主体 `test/server.example.com` 创建一个或多个别名。例如，`test1/server.example.com` 和 `test2/server.example.com`。
2. 在 CSR 中，为 `dnsName(server.example.com)` 和 `otherName(test2/server.example.com)` 添加 `subjectAltName`。要做到这一点，将 **openssl.conf** 文件配置为包含以下指定 UPN `otherName` 和 `subjectAltName` 的行：

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. 使用 **openssl** 创建证书请求：

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. 将证书请求文件提交到在 IdM 服务器上运行的 CA。指定 Kerberos 主体来与新发布的证书关联：

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM 中的 **ipa cert-request** 命令使用以下默认值：

- **calPAserviceCert** 证书配置文件
要选择自定义配置文件，请使用 **--profile-id** 选项。
- 集成的 IdM 根 CA **ipa**
要选择子 CA，请使用 **--ca** 选项。

其他资源

- 请参阅 **ipa cert-request --help** 命令的输出。
- 请参阅 [在身份管理中创建和管理证书配置文件](#)。

2.4. 其他资源

- 请参阅 [撤销带有集成 IdM CA 的证书](#)。
- 请参阅 [恢复带有集成 IdM CA 的证书](#)。
- 请参阅 [将应用程序限制为只信任证书子集](#)。

第 3 章 使用 ANSIBLE 管理 IDM 证书

您可以使用 **ansible-freeipa ipacert** 模块为身份管理(IdM)用户、主机和服务请求、撤销和检索 SSL 证书。您还可以恢复已被搁置的证书。

3.1. 使用 ANSIBLE 为 IDM 主机、服务和用户请求 SSL 证书

您可以使用 **ansible-freeipa ipacert** 模块为身份管理(IdM)用户、主机和服务请求 SSL 证书。然后，它们可以使用这些证书向 IdM 进行身份验证。

完成此流程，使用 Ansible playbook 为 HTTP 服务器请求来自 IdM 证书颁发机构(CA)的证书。

先决条件

- 在控制节点上：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已安装 **ansible-freeipa** 软件包。
 - 您已在 `~/MyPlaybooks/` 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 您已将 **ipaadmin_password** 存储在 `secret.yml` Ansible vault 中。
- 您的 IdM 部署有一个集成的 CA。

流程

1. 为您的用户、主机或服务生成一个证书签名请求(CSR)。例如，要使用 **openssl** 工具为运行 `client.idm.example.com` 上的 **HTTP** 服务生成一个 CSR，请输入：

```
# openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -
subj '/CN=client.idm.example.com,O=IDM.EXAMPLE.COM'
```

因此，CSR 存储在 `new.csr` 中。

2. 使用以下内容创建 Ansible playbook 文件 `request-certificate.yml`：

```
---
- name: Playbook to request a certificate
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Request a certificate for a web server
    ipacert:
      ipadmin_password: "{{ ipadmin_password }}"
      state: requested
      csr: |
        -----BEGIN CERTIFICATE REQUEST-----

MIGYMEwCAQAwGTEXMBUGA1UEAwwOZnJlZWlwYXNlYm9keWxlcjEwKjAFBgMrZXADIQBs
```

```

HlqIr4b/XNK+K8QLJKIzfvuNK0buBhLz3LAzY7QDEqAAMAUGAytIcANBAF4oSCbA
5aIPukCidnZJdr491G4LBE+URecYXsPknwYb+V+ONnf5ycZHyaFv+jkUBFGFeDgU
SYaXm/gF8cDYjQI=
-----END CERTIFICATE REQUEST-----
principal: HTTP/client.idm.example.com
register: cert

```

将证书请求替换为 `new.csr` 中的 CSR。

3. 请求证书：

```

$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/request-
certificate.yml

```

其他资源

- [ansible-freeipa 上游文档中的 cert 模块](#)

3.2. 使用 ANSIBLE 撤销 IDM 主机、服务和用户的 SSL 证书

您可以使用 `ansible-freeipa ipacert` 模块撤销身份管理(IdM)用户、主机和服务向 IdM 进行身份验证说使用的 SSL 证书。

完成此流程，以使用 Ansible playbook 撤销 HTTP 服务器的证书。吊销证书的原因是 "keyCompromise"。

先决条件

- 在控制节点上：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已安装 `ansible-freeipa` 软件包。
 - 您已在 `~/MyPlaybooks/` 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 您已将 `ipadmin_password` 存储在 `secret.yml` Ansible vault 中。
 - 您已获得证书的序列号，例如通过输入 `openssl x509 -noout -text -in <path_to_certificate>` 命令得到。在本例中，证书的序列号为 123456789。
- 您的 IdM 部署有一个集成的 CA。

流程

1. 使用以下内容创建 Ansible playbook 文件 `revoke-certificate.yml`：

```

---
- name: Playbook to revoke a certificate
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

```

```

tasks:
- name: Revoke a certificate for a web server
  ipacert:
    ipadmin_password: "{{ ipadmin_password }}"
    serial_number: 123456789
    revocation_reason: "keyCompromise"
    state: revoked

```

2. 吊销证书：

```

$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/revoke-
certificate.yml

```

其他资源

- [ansible-freeipa](#) 上游文档中的 `cert` 模块
- RFC 5280 中的 [原因代码](#)

3.3. 使用 ANSIBLE 恢复 IDM 用户、主机和服务的 SSL 证书

您可以使用 `ansible-freeipa ipacert` 模块恢复之前撤销的身份管理(IdM)用户、主机或服务用来向 IdM 进行身份验证的 SSL 证书。



注意

您只能恢复被搁置的证书。您可能搁置它，例如，您不确定私钥是否已丢失。但是，现在您已恢复了密钥，并且同时您确定没有人访问它，您希望重新恢复证书。

完成此流程，以使用 Ansible playbook 为注册到 IdM 中的搁置的服务发布证书。这个示例描述了如何为搁置的 HTTP 服务发布证书。

先决条件

- 在控制节点上：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已安装 `ansible-freeipa` 软件包。
 - 您已在 `~/MyPlaybooks/` 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 您已将 `ipadmin_password` 存储在 `secret.yml` Ansible vault 中。
- 您的 IdM 部署有一个集成的 CA。
- 您已获得证书的序列号，例如通过输入 `openssl x509 -noout -text -in path/to/certificate` 命令获得。在本例中，证书序列号为 123456789。

流程

1. 使用以下内容创建 Ansible playbook 文件 `restore-certificate.yml`：

```

---
- name: Playbook to restore a certificate
  hosts: ipaserver
  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

  tasks:
  - name: Restore a certificate for a web service
    ipacert:
      ipadmin_password: "{{ ipadmin_password }}"
      serial_number: 123456789
      state: released

```

2. 运行 playbook:

```

$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/restore-
certificate.yml

```

其他资源

- [ansible-freeipa](#) 上游文档中的 [cert](#) 模块

3.4. 使用 ANSIBLE 为 IDM 用户、主机和服务检索 SSL 证书

您可以使用 **ansible-freeipa ipacert** 模块来检索为身份管理(IdM)用户、主机或服务发布的 SSL 证书，并将其存储在受管节点上的一个文件中。

先决条件

- 在控制节点上：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已安装 [ansible-freeipa](#) 软件包。
 - 您已在 `~/MyPlaybooks/` 目录中创建了一个带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 您已将 **ipadmin_password** 存储在 `secret.yml` Ansible vault 中。
- 您已获得证书的序列号，例如通过输入 **openssl x509 -noout -text -in <path_to_certificate>** 命令得到。在本例中，证书的序列号为 123456789，存储检索到的证书的文件是 `cert.pem`。

流程

1. 使用以下内容创建 Ansible playbook 文件 `retrieve-certificate.yml`：

```

---
- name: Playbook to retrieve a certificate and store it locally on the managed node
  hosts: ipaserver

  vars_files:
  - /home/user_name/MyPlaybooks/secret.yml

```

```
tasks:
- name: Retrieve a certificate and save it to file 'cert.pem'
  ipacert:
    ipadmin_password: "{{ ipadmin_password }}"
    serial_number: 123456789
    certificate_out: cert.pem
    state: retrieved
```

2. 检索证书：

```
$ ansible-playbook --vault-password-file=password_file -v -i
<path_to_inventory_directory>/hosts <path_to_playbooks_directory>/retrieve-
certificate.yml
```

其他资源

- [ansible-freeipa](#) 上游文档中的 cert 模块

第 4 章 管理 IDM 用户、主机和服务的外部签名证书

本章描述了如何使用身份管理(IdM)命令行界面(CLI)和 IdM Web UI 来添加或删除用户、主机，以及由外部证书颁发机构(CA)发布的服务证书。

4.1. 使用 IDM CLI，将外部 CA 发布的证书添加到 IDM 用户、主机或服务

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)CLI 向 IdM 用户的帐户、主机或服务添加外部签名的证书。

先决条件

- 您已获得管理员用户的票据授予票据。

流程

- 要为 IdM 用户添加证书，请输入：

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```

该命令要求您指定以下信息：

- 用户名
- Base64 编码的 DER 证书



注意

您可以将证书转换为 DER 格式，然后将其重新编码为 Base64，而不是将证书内容复制并粘贴到命令行。例如，要将 `user_cert.pem` 证书添加给 `user`，请输入：

```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

您可以在不添加任何选项的情况下，以交互方式运行 `ipa user-add-cert` 命令。

要在 IdM 主机中添加证书，请输入：

- `ipa host-add-cert`

要在 IdM 服务中添加证书，请输入：

- `ipa service-add-cert`

其他资源

- [使用集成的 IdM CA 为用户、主机和服务管理证书](#)

4.2. 使用 IDM WEB UI 将外部 CA 发布的证书添加到 IDM 用户、主机或服务中

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)Web UI 将外部签名的证书添加到 IdM 用户的帐户、主机或服务中。

先决条件

- 您以管理用户的身份登录到身份管理(IdM)Web UI。

流程

1. 打开 **Identity** 选项卡，然后选择 **Users**、**Hosts** 或 **Services** 子选项卡。
2. 点用户、主机或服务的名称，来打开其配置页面。
3. 点 **Certificates** 条目旁边的 **Add**。

图 4.1. 在用户帐户中添加证书

The screenshot shows the configuration page for a user named 'demouser'. At the top, there are tabs for 'Settings', 'User Groups', 'Netgroups', 'Roles', 'HBAC Rules', and 'Sudo Rules'. Below the tabs are buttons for 'Refresh', 'Revert', 'Save', and 'Actions'. The main content is divided into two columns: 'Identity Settings' and 'Account Settings'. 'Identity Settings' includes fields for Job Title, First name (Demo), Last name (User), Full name (Demo User), Display name (Demo User), Initials (DU), GECOS (Demo User), and Class. 'Account Settings' includes fields for User login (demouser), Password (*****), Password expiration (2016-07-14 10:14:41Z), UID (373000005), GID (373000005), Principal alias (demouser@IDM.EXAMPLE.COM), Kerberos principal expiration (YYYY-MM-DD hh:mn UTC), Login shell (/bin/sh), Home directory (/home/demouser), SSH public keys (Add), and Certificates (Add). The 'Add' button for Certificates is highlighted with a red box.

4. 将 Base64 或 PEM 编码格式的证书粘贴到文本字段中，然后点 **Add**。
5. 点 **Save** 以保存更改。

4.3. 使用 IDM CLI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)CLI 从 IdM 用户的帐户、主机或服务中删除外部签名的证书。

先决条件

- 您已获得管理员用户的票据授予票据。

流程

- 要从 IdM 用户中删除证书，请输入：


```
$ ipa user-remove-cert user --certificate=MIQTPrajQAwg...
```

该命令要求您指定以下信息：

- 用户名
- Base64 编码的 DER 证书



注意

您可以将证书转换为 DER 格式，然后将其重新编码为 Base64，而不是将证书内容复制并粘贴到命令行。例如，要从 `user` 中删除 `user_cert.pem` 证书，请输入：

```
$ ipa user-remove-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

您可以在不添加任何选项的情况下，以交互方式运行 `ipa user-remove-cert` 命令。

要从 IdM 主机中删除证书，请输入：

- `ipa host-remove-cert`

要从 IdM 服务中删除证书，请输入：

- `ipa service-remove-cert`

其他资源

- [使用集成的 IdM CA 为用户、主机和服务管理证书](#)

4.4. 使用 IDM WEB UI 从 IDM 用户、主机或服务帐户中删除外部 CA 发布的证书

作为身份管理(IdM)管理员，您可以使用身份管理(IdM)Web UI 从 IdM 用户的帐户、主机或服务中删除外部签名的证书。

先决条件

- 您以管理用户的身份登录到身份管理(IdM)Web UI。

流程

1. 打开 **Identity** 选项卡，然后选择 **Users**、**Hosts** 或 **Services** 子选项卡。
2. 点用户、主机或服务的名称，来打开其配置页面。
3. 单击要删除的证书旁边的 **Actions**，然后选择 **Delete**。
4. 点 **Save** 以保存更改。

4.5. 其他资源

- [使用 Ansible playbook 确保 IdM 服务条目中存在外部签名的证书](#)

第 5 章 转换证书格式以和 IDM 一起工作

这个用户故事描述了如何确保您作为 IdM 系统管理员使用正确的带有特定 IdM 命令的证书的格式。例如，这在以下情况下非常有用：

- 您将外部证书加载到用户配置文件中。详情请参阅 [转换外部证书以加载到 IdM 用户帐户中](#)。
- 在 [为智能卡验证配置 IdM 服务器](#) 或 [为智能卡验证配置 IdM 客户端时](#)，您在使用外部 CA 证书，以使用户可以使用其上带有由外部证书颁发机构签发的证书的智能卡向 IdM 进行身份验证。
- 您从 NSS 数据库将证书导出为 pkcs #12 格式，其中包括证书和私钥。详情请查看 [将 NSS 数据库中的证书和私钥导出到 PKCS #12 文件中](#)。

5.1. IDM 中的证书格式和编码

包括 IdM 中的智能卡身份验证的证书验证通过比较用户提供的证书或证书数据（保存在用户的 IdM 配置文件中）来进行。

系统配置

IdM 配置文件中存储的内容只是证书，而不是相应的私钥。在身份验证期间，用户还必须显示其拥有相应的私钥。用户通过显示包含证书和私钥的 PKCS #12 文件，或提供两个文件：一个包含证书，另一个包含私钥，来执行此操作。

因此，将证书加载到用户配置文件的进程等只接受不包含私钥的证书文件。

同样，当系统管理员为您提供外部 CA 证书时，他将仅提供公共数据：不带私钥的证书。为 IdM 客户端的智能卡验证配置 IdM 服务器的 `ipa-advise` 工具需要输入文件包含外部 CA 的证书，而不是私钥。

证书编码

有两种常见的证书编码：隐私增强的电子邮件(PEM)和区分的编码规则(DER)。`base64` 格式与 PEM 格式几乎一样，但它不包含 `-----BEGIN CERTIFICATE-----/-----END CERTIFICATE-----` 标头和页脚。

已使用 DER 编码的证书是二进制 X509 数字证书文件。作为二进制文件，证书不可读。DER 文件有时使用 `.der` 文件扩展名，但带有 `.crt` 和 `.cer` 文件扩展名的文件有时也会包含 DER 证书。包含密钥的 DER 文件可以命名为 `.key`。

使用 PEM Base64 编码的证书是一个人类可读的文件。该文件包含前缀为 `"-----BEGIN ..."` 的 ASCII(Base64)保护的数据行。PEM 文件有时使用 `.pem` 文件扩展名，但带有 `.crt` 和 `.cer` 文件扩展名的文件有时也包含 PEM 证书。包含密钥的 PEM 文件可以命名为 `.key`。

不同的 `ipa` 命令对其接受的证书类型有不同的限制。例如，`ipa user-add-cert` 命令只接受以 `base64` 格式编码的证书，但 `ipa-server-certinstall` 接受 PEM、DER、PKCS #7、PKCS #8 和 PKCS #12 证书。

表 5.1. 证书编码

编码格式	人类可读	常用的文件扩展名	接受编码格式的 IdM 命令示例
PEM/base64	是	.pem, .crt, .cer	ipa user-add-cert, ipa-server-certinstall, ...
DER	否	.der, .crt, .cer	ipa-server-certinstall, ...

IdM 中与证书相关的命令和格式 列出了其它 ipa 命令以及这些命令可以接受的证书格式。

用户身份验证

在使用 Web UI 访问 IdM 时，用户证明自己通过将两者都存储在浏览器的数据库中，证明自己拥有与证书对应的私钥。

当使用 CLI 访问 IdM 时，用户通过以下方法之一证明自己拥有与证书对应的私钥：

- 用户添加连接到包含证书和密钥的智能卡模块的路径，作为 `kinit -X` 命令的 `X509_user_identity` 参数的值：

```
$ kinit -X X509_user_identity='PKCS11:opencsc-pkcs11.so' idm_user
```

- 用户添加两个文件作为 `kinit -X` 命令的 `X509_user_identity` 参数的值，一个包含证书，另一个包含私钥：

```
$ kinit -X X509_user_identity='FILE:/path/to/cert.pem,/path/to/cert.key' idm_user
```

有用的证书命令

查看证书数据，如主题和签发者：

```
$ openssl x509 -noout -text -in ca.pem
```

要比较两个证书在哪些行不同：

```
$ diff cert1.crt cert2.crt
```

要通过两列中显示的输出来比较两个证书在哪些行不同：

```
$ diff cert1.crt cert2.crt -y
```

5.2. 将外部证书转换来加载到 IDM 用户帐户中

本节描述了如何确保在将外部证书添加到用户条目之前正确对其进行编码和格式化。

5.2.1. 先决条件

- 如果您的证书是由活动目录证书认证机构签发，并使用 PEM 编码的，请确保 PEM 文件已转换为 UNIX 格式。要转换文件，请使用 `eponymous` 软件包提供的 `dos2unix` 工具。

5.2.2. 在 IdM CLI 中转换外部证书，并将其加载到 IdM 用户帐户中

IdM CLI 只接受 PEM 证书，从中删除了第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）。

按照以下流程将外部证书转换为 PEM 格式，并使用 IdM CLI 将其添加到 IdM 用户帐户中。

步骤

1. 将证书转换为 PEM 格式：

- 如果您的证书为 **DER** 格式：

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

- 如果您的文件为 **PKCS #12** 格式，其常用文件扩展名为 **.pfx** 和 **.p12**，并且包含证书、私钥和其他数据，请使用 **openssl pkcs12** 工具提取证书。提示时，输入保护存储在文件中的私钥的密码：

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. 获取管理员凭证：

```
$ kinit admin
```

3. 使用 **IdM CLI** 将证书添加到用户帐户中，按照以下方法之一：

- 在将字符串添加到 **ipa user-add-cert**前，使用 **sed** 工具删除 **PEM** 文件的第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）：

```
$ ipa user-add-cert some_user --certificate="$(sed -e '/BEGIN CERTIFICATE/d;/END CERTIFICATE/d' cert.pem)"
```

- 将没有第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----）的证书文件的内容复制并粘贴到 **ipa user-add-cert** 命令中：

```
$ ipa user-add-cert some_user --
certificate=MIIDlzCCAn+gAwIBAgIBATANBgkqhki...
```



注意

如果不首先删除第一行和最后一行（-----BEGIN CERTIFICATE----- 和 -----END CERTIFICATE-----），您无法直接将包含证书的 **PEM** 文件作为输入传给 **ipa user-add-cert** 命令：

```
$ ipa user-add-cert some_user --cert=some_user_cert.pem
```

此命令会导致产生 "ipa: ERROR: Base64 decoding failed: Incorrect padding" 错误消息。

4. （可选）检查证书是否被系统接受：

```
[idm_user@r8server]$ ipa user-show some_user
```

5.2.3. 在 IdM web UI 中转换外部证书，以便将其加载到 IdM 用户帐户中

按照以下流程将外部证书转换为 **PEM** 格式，并将其添加到 IdM Web UI 中的 IdM 用户帐户中。

步骤

1. 使用 **CLI**，将证书转换为 **PEM** 格式：

- 如果您的证书为 **DER** 格式：

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

- 如果您的文件为 **PKCS #12** 格式，其常用文件扩展名为 **.pfx** 和 **.p12**，并且包含证书、私钥和其他数据，请使用 **openssl pkcs12** 工具提取证书。提示时，输入保护存储在文件中的私钥的密码：

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. 在编辑器中打开证书，并复制内容。您可以包含 "-----BEGIN CERTIFICATE-----" 和 "-----END CERTIFICATE-----" 标头和页脚行，但您不必这样做，因为 IdM Web UI 接受 **PEM** 和 **base64** 格式。
3. 在 IdM Web UI 中，以安全官身份登录。
4. 前往 **Identity** → **Users** → **some_user**。
5. 单击 **Certificates** 旁边的 **Add**。
6. 将证书的 PEM 格式内容粘贴到打开的窗口中。
7. 单击 **Add**。

如果证书被系统接受，您可以在用户配置文件中看到它列在 **Certificates** 中。

5.3. 准备将证书加载到浏览器

在将用户证书导入到浏览器前，请确保证书和相应的私钥为 **PKCS #12** 格式。通常有两种情况需要额外的准备工作：

- 证书位于 NSS 数据库中。有关在这种情况下如何处理的详情，请参考 [将 NSS 数据库中的证书和私钥导出到 PKCS #12 文件中](#)。
- 证书和私钥位于两个单独的 **PEM** 文件中。有关在这种情况下如何处理的详情，请参考 [将证书和私钥 PEM 文件合并到 PKCS #12 文件中](#)。

之后，要将 **PEM** 格式的 CA 证书以及 **PKCS #12** 格式的用户证书导入到浏览器中，请按照 [配置浏览器以启用证书身份验证](#) 和 [以身份管理用户的身份使用证书验证身份管理 Web UI](#) 中的流程。

5.3.1. 将证书和私钥从 NSS 数据库导出到 PKCS #12 文件中

步骤

1. 使用 **pk12util** 命令将证书从 NSS 数据库导出为 **PKCS12** 格式。例如，要将昵称为 **some_user** 的证书从存储在 **~/certdb** 目录中的 NSS 数据库导出到 **~/some_user.p12** 文件中：

```
$ pk12util -d ~/certdb -o ~/some_user.p12 -n some_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

2. 为 **.p12** 文件设置合适的权限：

```
# chmod 600 ~/some_user.p12
```

由于 **PKCS #12** 文件也包含私钥，因此必须对其进行保护，以防止其他用户使用该文件。否则，他们可以模拟用户。

5.3.2. 将证书和私钥 PEM 文件合并到 PKCS #12 文件中

按照以下流程将证书和存储在单独的 **PEM** 文件中的相应密钥合并到 **PKCS #12** 文件中。

步骤

- 将存储在 **certfile.cer** 中的证书和存储在 **certfile.key** 中的密钥合并到包含证书和密钥的 **certfile.p12** 文件中：

```
$ openssl pkcs12 -export -in certfile.cer -inkey certfile.key -out certfile.p12
```

5.4. IDM 中与证书相关的命令和格式

下表显示了 IdM 中与证书相关的具有可接受格式的命令。

表 5.2. IdM 证书命令和格式

命令	可接受的格式	备注
ipa user-add-cert some_user --certificate	base64 PEM 证书	
ipa-server-certinstall	PEM 和 DER 证书；PKCS#7 证书链；PKCS#8 和原始私钥；PKCS#12 证书和私钥	
ipa-cacert-manage install	DER; PEM; PKCS#7	
ipa-cacert-manage renewal --external-cert-file	PEM 和 DER 证书; PKCS#7 证书链	
ipa-ca-install --external-cert-file	PEM 和 DER 证书; PKCS#7 证书链	
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	N/A	创建具有 <cert_serial> 序列号证书的 PEM 编码的 file.pem 文件。
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	N/A	创建具有 <cert_serial> 序列号证书的 PEM 编码的 file.pem 文件。如果使用 --chain 选项，PEM 文件将含有包含证书链的证书。

命令	可接受的格式	备注
<code>ipa cert-request --certificate-out=FILE /path/to/req.csr</code>	N/A	使用新证书创建 PEM 格式的 req.csr 文件。
<code>ipa cert-request --certificate-out=FILE /path/to/req.csr</code>	N/A	使用新证书创建 PEM 格式的 req.csr 文件。如果使用 --chain 选项，PEM 文件将含有包含证书链的证书。

第 6 章 在身份管理中创建和管理证书配置文件

证书授权机构(CA)在签名证书时使用证书配置文件，来确定证书签名请求(CSR)是否可以接受，如果可以接受，证书上有哪些功能和扩展。证书配置文件与发布特定类型的证书相关联。通过组合证书配置文件和 CA 访问控制列表(ACL)，您可以定义和控制对自定义证书配置文件的访问。

在描述如何创建证书配置集时，流程使用 S/MIME 证书作为示例。某些电子邮件程序支持使用安全多用途互联网邮件扩展(S/MIME)协议进行数字签名和加密的电子邮件。使用 S/MIME 签名或加密电子邮件消息，要求消息的发送方具有 S/MIME 证书。

- [什么是证书配置文件](#)
- [创建证书配置文件](#)
- [什么是 CA 访问控制列表](#)
- [定义 CA ACL 来控制对证书配置文件的访问](#)
- [使用证书配置文件和 CA ACL 来发布证书](#)
- [修改证书配置文件](#)
- [证书配置文件配置参数](#)

6.1. 什么是证书配置文件？

您可以使用证书配置文件来确定证书的内容，以及发布证书的限制，如下所示：

- 用于隔离证书签名请求的签名算法。
- 证书的默认有效期。
- 用于吊销证书的吊销原因。
- 如果主体的通用名称被复制到主题备用名称字段。
- 应该出现在证书中的功能和扩展。

单个证书配置文件与签发特定类型的证书相关联。您可以在 IdM 中为用户、服务和主机定义不同的证书配置文件。IdM 默认包括以下证书配置文件：

- **calPAserviceCert**
- **IECUserRoles**
- **KDCs_PKINIT_Certs**（内部使用）

另外，您可以创建和导入自定义配置文件，其允许您为特定目的发布证书。例如，您可以将特定配置文件的使用限制给一个用户或一个组，防止其他用户和组使用该配置文件发布用于身份验证的证书。要创建自定义证书配置文件，请使用 **ipa certprofile** 命令。

其他资源

- 请参阅 **ipa help certprofile** 命令。

6.2. 创建证书配置文件

按照以下流程，通过为请求 S/MIME 证书创建一个证书配置文件，通过命令行来创建一个配置文件。

步骤

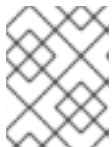
1. 通过复制现有的默认配置文件来创建自定义配置文件：

```
$ ipa certprofile-show --out smime.cfg caIPAServiceCert
-----
Profile configuration stored in file 'smime.cfg'
-----
Profile ID: caIPAServiceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE
```

2. 在文本编辑器中打开新创建的配置文件。

```
$ vi smime.cfg
```

3. 将 **Profile ID** 更改为反映配置文件用法的名称，如 **smime**。



注意

当您导入新创建的配置文件时，如果有 **profileid** 字段，则其必须与命令行中指定的 ID 匹配。

4. 更新扩展的密钥用法配置。默认的扩展的密钥用法扩展配置用于 TLS 服务器和客户端身份验证。例如，对于 S/MIME，必须为电子邮件保护配置扩展的密钥用法：

```
policyset.serverCertSet.7.default.params.exKeyUsageOIDs=1.3.6.1.5.5.7.3.4
```

5. 导入新配置文件：

```
$ ipa certprofile-import smime --file smime.cfg \
--desc "S/MIME certificates" --store TRUE
-----
Imported profile "smime"
-----
Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
```

验证步骤

- 验证新证书配置文件已被导入：

```
$ ipa certprofile-find
-----
4 profiles matched
-----
```

```

Profile ID: caIPAserviceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE

Profile ID: IECUserRoles
Profile description: User profile that includes IECUserRoles extension from request
Store issued certificates: TRUE

Profile ID: KDCs_PKINIT_Certs
Profile description: Profile for PKINIT support by KDCs
Store issued certificates: TRUE

Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
-----
Number of entries returned 4
-----

```

其他资源

- 请参阅 `ipa help certprofile`。
- 请参阅 [RFC 5280](#)，4.2.1.12 部分。

6.3. 什么是 CA 访问控制列表？

证书颁发机构访问控制列表(CA ACL)规则定义哪些配置文件可用于向哪些主体发布证书。您可以使用 CA ACL 来执行此操作，例如：

- 确定可以使用特定配置文件向哪些用户、主机或服务发布证书
- 确定允许哪个 IdM 证书颁发机构或子 CA 发布证书

例如，使用 CA ACL，您可以将只用于伦敦办事处工作的员工的配置文件限制为与伦敦办事处相关的 IdM 用户组的成员。

用于管理 CA ACL 规则的 `ipa caacl` 工具允许特权用户添加、显示、修改或删除指定的 CA ACL。

其他资源

- 请参阅 `ipa help caacl`。

6.4. 定义 CA ACL 来控制对证书配置文件的访问

按照以下流程，使用 `caacl` 工具定义一个 CA 访问控制列表(ACL)规则，以允许组中的用户访问自定义证书配置文件。在这种情况下，流程描述了如何创建 S/MIME 用户的组以及 CA ACL，以允许该组中的用户访问 `smime` 证书配置文件。

先决条件

- 确保您已获取 IdM 管理员的凭据。

步骤

1. 为证书配置文件的用户创建一个新组：

```
$ ipa group-add smime_users_group
-----
Added group "smime users group"
-----
Group name: smime_users_group
GID: 75400001
```

2. 创建一个新用户来添加到 **smime_user_group** 组中：

```
$ ipa user-add smime_user
First name: smime
Last name: user
-----
Added user "smime_user"
-----
User login: smime_user
First name: smime
Last name: user
Full name: smime user
Display name: smime user
Initials: TU
Home directory: /home/smime_user
GECOS: smime user
Login shell: /bin/sh
Principal name: smime_user@IDM.EXAMPLE.COM
Principal alias: smime_user@IDM.EXAMPLE.COM
Email address: smime_user@idm.example.com
UID: 1505000004
GID: 1505000004
Password: False
Member of groups: ipausers
Kerberos keys available: False
```

3. 将 **smime_user** 添加到 **smime_users_group** 组中：

```
$ ipa group-add-member smime_users_group --users=smime_user
Group name: smime_users_group
GID: 1505000003
Member users: smime_user
-----
Number of members added 1
-----
```

4. 创建 CA ACL 以允许组中的用户访问证书配置文件：

```
$ ipa caacl-add smime_acl
-----
Added CA ACL "smime_acl"
-----
ACL name: smime_acl
Enabled: TRUE
```

5. 将用户组添加到 CA ACL 中：

```
$ ipa caacl-add-user smime_acl --group smime_users_group
ACL name: smime_acl
Enabled: TRUE
User Groups: smime_users_group
-----
Number of members added 1
-----
```

6. 将证书配置文件添加到 CA ACL 中：

```
$ ipa caacl-add-profile smime_acl --certprofile smime
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
-----
Number of members added 1
-----
```

验证步骤

- 查看您创建的 CA ACL 的详情：

```
$ ipa caacl-show smime_acl
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
...
```

其他资源

- 请参阅 [ipa 手册页](#)。
- 请参阅 [ipa help caacl](#)。

6.5. 使用证书配置文件和 CA ACL 来发布证书

当证书颁发机构访问控制列表(CA ACL)允许时，您可以使用证书配置文件来请求证书。按照以下流程，使用已通过 CA ACL 授予了访问权限的自定义证书配置文件来为用户请求 S/MIME 证书。

先决条件

- 您的证书配置文件已创建。
- 允许用户使用所需证书配置文件请求证书的 CA ACL 已创建。



注意

您可以绕过 CA ACL 检查用户是否执行了 **cert-request** 命令：

- 是 **admin** 用户。
- 具有 **请求证书忽略 CA ACL** 权限。

步骤

1. 为用户生成证书请求。例如，使用 OpenSSL：

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout private.key -out cert.csr -
subj '/CN=smime_user'
```

2. 为用户从 IdM CA 请求新证书：

```
$ ipa cert-request cert.csr --principal=smime_user --profile-id=smime
```

(可选) 将 `--ca sub-CA_name` 选项传给命令，以从子 CA，而不是根 CA 请求证书。

验证步骤

- 验证新发布的证书是否已分配给用户：

```
$ ipa user-show user
User login: user
...
Certificate: MIICfzCCAWcCAQA...
...
```

其他资源

- 请参阅 **ipa(a)** 手册页。
- 请参阅 **ipa help user-show** 命令。
- 请参阅 **ipa help cert-request** 命令。
- 请参阅 **openssl(1ssl)** 手册页。

6.6. 修改证书配置文件

按照以下流程，使用 **ipa certprofile-mod** 命令直接通过命令行修改证书配置文件。

步骤

1. 确定您要修改的证书配置文件的证书配置文件 ID。显示当前存储在 IdM 中的所有证书配置文件：

```
# ipa certprofile-find
-----
4 profiles matched
-----
```

```

Profile ID: caIPAServiceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE

Profile ID: IECUserRoles
...

Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
-----
Number of entries returned
-----

```

2. 修改证书配置文件描述。例如，如果您使用现有的配置文件为 S/MIME 证书创建了自定义证书配置文件，请按照新用法更改描述：

```

# ipa certprofile-mod smime --desc "New certificate profile description"
-----
Modified Certificate Profile "smime"
-----
Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE

```

3. 在文本编辑器中打开您的客户证书配置文件，并进行修改以满足您的要求：

```
# vi smime.cfg
```

有关可以在证书配置文件中配置哪些选项的详情，请查看 [证书配置文件配置参数](#)。

4. 更新现有证书配置文件：

```
# ipa certprofile-mod _profile_ID_ --file=smime.cfg
```

验证步骤

- 验证证书配置文件是否已更新：

```

$ ipa certprofile-show smime
Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE

```

其他资源

- 请参阅 [ipa\(a\)](#) 手册页。
- 请参阅 [ipa help certprofile-mod](#)。

6.7. 证书配置文件配置参数

证书配置文件配置参数存储在 CA 配置文件目录 `/var/lib/pki/pki-tomcat/ca/profiles/ca` 中的

`profile_name.cfg` 文件中。配置文件的所有参数 - 默认值、输入、输出和约束 - 都在单个策略集中配置。为证书配置集设置的策略具有名称 **policyset.policyName.policyNumber**。例如，对于策略设置 **serverCertSet**：

```

policyset.list=serverCertSet
policyset.serverCertSet.list=1,2,3,4,5,6,7,8
policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl
policyset.serverCertSet.1.constraint.name=Subject Name Constraint
policyset.serverCertSet.1.constraint.params.pattern=CN=[^,]+.+
policyset.serverCertSet.1.constraint.params.accept=true
policyset.serverCertSet.1.default.class_id=subjectNameDefaultImpl
policyset.serverCertSet.1.default.name=Subject Name Default
policyset.serverCertSet.1.default.params.name=CN=$request.req_subject_name.cn$, OU=pki-ipa, O=IPA
policyset.serverCertSet.2.constraint.class_id=validityConstraintImpl
policyset.serverCertSet.2.constraint.name=Validity Constraint
policyset.serverCertSet.2.constraint.params.range=740
policyset.serverCertSet.2.constraint.params.notBeforeCheck=false
policyset.serverCertSet.2.constraint.params.notAfterCheck=false
policyset.serverCertSet.2.default.class_id=validityDefaultImpl
policyset.serverCertSet.2.default.name=Validity Default
policyset.serverCertSet.2.default.params.range=731
policyset.serverCertSet.2.default.params.startTime=0

```

每个策略集都包含按照策略 ID 号为证书配置文件配置的策略列表，以它们的评估顺序排列。服务器为其收到的每个请求评估每个策略集。收到单个证书请求时，将评估一个集合，并忽略配置文件中的任何其他集合。发布双密钥对后，对第一个证书请求评估第一个策略集，对第二个证书请求评估第二个策略集。在发布双密钥对时，在发布单个证书或多个集合时，您不需要多个策略集。

表 6.1. 证书配置文件参数

参数	描述
desc	证书配置文件的自由文本描述，显示在终端实体页面上。例如， desc=This certificate profile 用于使用代理身份验证注册服务器证书。
enable	启用配置文件，使它可通过终端实体页面访问。例如： enable=true 。
auth.instance_id	设置身份验证管理者插件，用来验证证书请求。要进行自动注册，如果身份验证成功，CA 会立即发布证书。如果身份验证失败或者没有指定身份验证插件，则会将请求排队，来由代理手动批准。例如， auth.instance_id=AgentCertAuth 。

参数	描述
authz.acl	<p>指定授权约束。这主要用于设置组评估访问控制列表 (ACL)。例如，caCMCUserCert 参数要求 CMC 请求的签名者属于证书管理者代理组：</p> <p>authz.acl=group="Certificate Manager Agents</p> <p>在基于目录的用户证书续订中，此选项用于确保原始请求者和当前验证的用户是同一个。在评估授权前，实体必须验证（绑定或登录到系统）。</p>
名称	<p>证书配置文件的名称。例如，name=Agent-Authenticated Server Certificate Enrollment。此名称显示在最终用户注册或续订页面上。</p>
input.list	<p>按名称列出证书配置文件允许的输入。例如，input.list=i1,i2。</p>
input.input_id.class_id	<p>按输入 ID（在 input.list 中列出的输入名称）表示输入的 java 类名称。例如，input.i1.class_id=certReqInputImpl。</p>
output.list	<p>按名称列出证书配置文件可能的输出格式。例如 output.list=o1。</p>
output.output_id.class_id	<p>为在 output.list 中命名的输出格式指定 java 类名称。例如：output.o1.class_id=certOutputImpl。</p>
policyset.list	<p>列出配置的证书配置文件规则。对于双证书，一组规则适用于签名密钥，另一组规则适用于加密密钥。单个证书仅使用一组证书配置文件规则。例如，policyset.list=serverCertSet。</p>
policyset.policyset_id.list	<p>按照策略 ID 号，按评估的顺序，列出为证书配置文件配置的策略集中的策略。例如：policyset.serverCertSet.list=1,2,3,4,5,6,7,8。</p>
policyset.policyset_id.policy_number.constraint.class_id	<p>表示配置文件规则中配置的默认约束插件集的 java 类名称。例如，policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl。</p>
policyset.policyset_id.policy_number.constraint.name	<p>提供用户定义的约束名称。例如，policyset.serverCertSet.1.constraint.name=Subject Name Constraint。</p>

参数	描述
policyset.policyset_id.policy_number.constraint.params.attribute	为约束的允许的属性指定值。可能的属性因约束类型而异。例如， policyset.serverCertSet.1.constraint.params.pattern=CN=.*。
policyset.policyset_id.policy_number.default.class_id	给出配置文件规则中默认集的 java 类名称。例如， policyset.serverCertSet.1.default.class_id=userSubjectNameDefaultImpl
policyset.policyset_id.policy_number.default.name	给出用户定义的默认值的名称。例如： policyset.serverCertSet.1.default.name=Subject Name Default
policyset.policyset_id.policy_number.default.params.attribute	为默认值的允许的属性指定值。可能的属性因默认类型而异。例如： policyset.serverCertSet.1.default.params.name=CN=(Name)\$request.requestor_name\$。

第 7 章 管理 IDM 中证书的有效性

在身份管理(IdM)中，您可以管理现有证书和未来要发布的证书的有效性，但方法有所不同。

7.1. 管理 IDM CA 发布的现有证书的有效性

在 IdM 中，可以使用以下方法查看证书的到期日期：

- [在 IdM WebUI 中查看到期日](#)。
- [在 CLI 中查看到期日](#)。

您可以使用以下方法管理 IdM CA 发布的现有证书的有效性：

- 通过使用原始证书签名请求(CSR)或私钥生成的新 CSR 请求新的证书来续订证书。您可以使用以下工具请求新证书：

certmonger

您可以使用 **certmonger** 请求服务证书。证书到期之前，**certmonger** 将自动续订证书，从而确保服务证书持续有效。详情请参阅 [使用 certmonger 为服务获取 IdM 证书](#)。

certutil

您可以使用 **certutil** 续订用户、主机和服务证书。有关请求用户证书的详情，请参阅 [请求新的用户证书并将其导出到客户端](#)；

openssl

您可以使用 **openssl** 续订用户、主机和服务证书。

- 吊销证书。详情请查看：
 - [使用 IdM Web UI 吊销带有集成 IdM CA 的证书](#)；
 - [使用 IdM CLI 吊销带有集成 IdM CA 的证书](#)；
- 如果证书已被临时吊销，则恢复证书。详情请查看：
 - [使用 IdM WebUI 恢复带有集成 IdM CA 的证书](#)；
 - [使用 IdM CLI 恢复带有集成 IdM CA 的证书](#)。

7.2. 管理 IDM CA 发布的未来证书的有效性

要管理 IdM CA 发布的未来证书的有效性，请修改、导入或创建证书配置文件。详情请参阅在 [在身份管理中创建和管理证书配置文件](#)。

7.3. 在 IDM WEBUI 中查看证书的到期日期

您可以使用 IdM WebUI 来查看 IdM CA 发布的所有证书的到期日期。

先决条件

- 确保您已获取管理员的凭证。

步骤

1. 在 **Authentication** 菜单中，点击 **Certificates > Certificates**。
2. 单击证书的序列号，来打开证书信息页面。

图 7.1. 证书列表

Certificates		
Subject ▼		Search 🔍
		Refresh ↻
		+ Issue
<input type="checkbox"/>	Serial Number	Subject
<input type="checkbox"/>	1	CN=Certificate Authority,O=EXAMPLE.COM
<input type="checkbox"/>	2	CN=OCSP Subsystem,O=EXAMPLE.COM
<input type="checkbox"/>	3	CN=server.example.com,O=EXAMPLE.COM
<input type="checkbox"/>	4	CN=CA Subsystem O=EXAMPLE.COM

3. 在证书信息页面中，找到 **Expires On** 信息。

7.4. 在 CLI 中查看证书的到期日期

您可以使用命令行界面(CLI)查看证书的到期日期。

步骤

- 使用 **openssl** 工具以人类可读的格式打开文件：

```
$ openssl x509 -noout -text -in ca.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O = IDM.EXAMPLE.COM, CN = Certificate Authority
    Validity
      Not Before: Oct 30 19:39:14 2017 GMT
      Not After : Oct 30 19:39:14 2037 GMT
```

7.5. 吊销带有集成 IDM CA 的证书

7.5.1. 证书吊销原因

已吊销的证书是无效的，不能用于身份验证。所有取消都是永久的，除了原因 6：证书冻结。

默认的吊销原因为 0：未指定。

表 7.1. 吊销原因

ID	原因	解释
0	未指定	

ID	原因	解释
1	密钥泄露	签发证书的密钥不再被信任。 可能的原因是：丢失令牌，非正常访问文件。
2	CA 泄露	签发证书的 CA 不再被信任。
3	隶属关系更改了	可能的原因： * 本人已离开公司或转到另一个部门。 * 主机或服务将被停用。
4	被取代	较新的证书替换了当前的证书。
5	停止操作	主机或服务将被停用。
6	证书冻结	证书被临时吊销。您可稍后恢复证书。
8	从 CRL 中删除	证书不再包含在证书吊销列表(CRL)中。
9	特权收回	用户、主机或服务不再被允许使用证书。
10	属性授权(AA)泄露	AA 证书不再被信任。

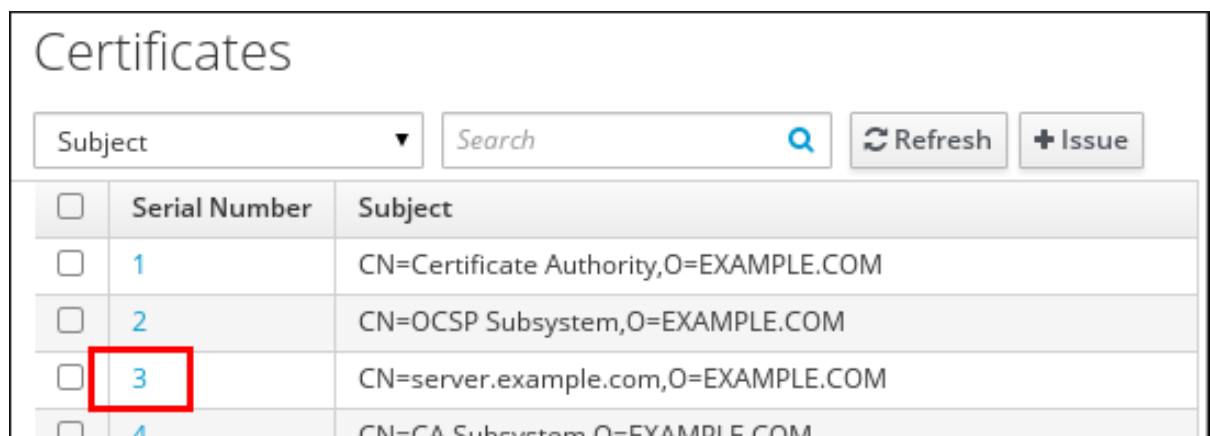
7.5.2. 使用 IdM Web UI 吊销带有集成 IdM CA 的证书

如果您知道您已丢失证书的私钥，则您必须吊销证书以防止其被滥用。完成此流程，以使用 IdM WebUI 吊销 IdM CA 发布的证书。

步骤

1. 点击 **Authentication > Certificates > Certificates**。
2. 单击证书的序列号，来打开证书信息页面。

图 7.2. 证书列表



3. 在证书信息页面中，单击 **Actions → Revoke Certificate**。

4. 选择吊销的原因，然后单击 **Revoke**。详情请参阅 [证书吊销原因](#)。

7.5.3. 使用 IdM CLI 吊销带有集成 IdM CA 的证书

如果您知道您已丢失证书的私钥，则您必须吊销证书以防止其被滥用。完成此流程，以使用 IdM CLI 吊销 IdM CA 发布的证书。

步骤

- 使用 **ipa cert-revoke** 命令，并指定：
 - 证书序列号
 - 吊销原因的 ID 号；有关详细信息，请参阅 [证书吊销原因](#)

例如，因为原因 1：**密钥泄露**，要吊销序列号为 **1032** 的证书，请输入：

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

有关请求新证书的详情，请查看以下文档：

- [请求新的用户证书并将其导出到客户端](#)
- [使用 certmonger 为服务获取 IdM 证书](#)。

7.6. 恢复带有集成 IdM CA 的证书

如果您因为原因 6：**证书冻结** 吊销了证书，如果证书的私钥未泄露，您可以恢复它。要恢复证书，请使用以下流程之一：

- [使用 IdM WebUI 恢复带有集成 IdM CA 的证书](#) ；
- [使用 IdM CLI 恢复带有集成 IdM CA 的证书](#)。

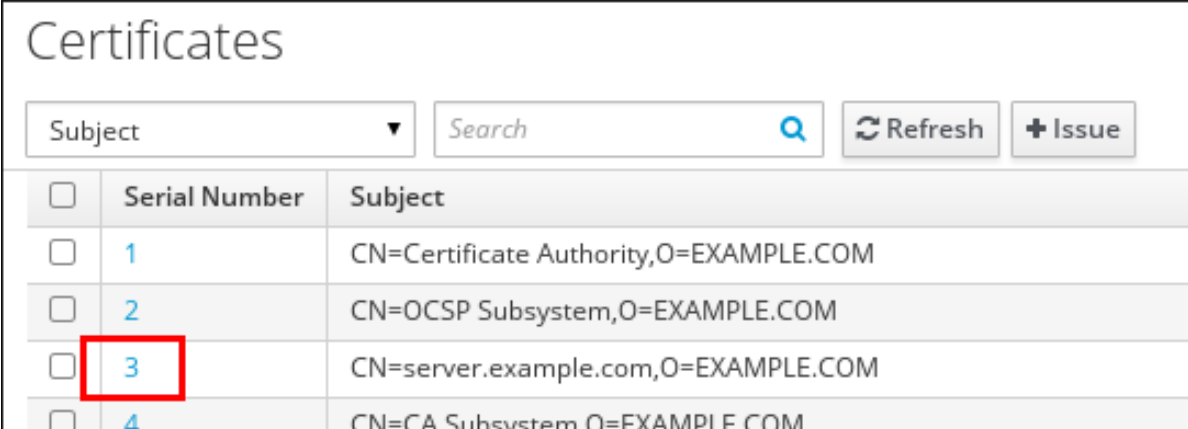
7.6.1. 使用 IdM WebUI 恢复带有集成 IdM CA 的证书

完成这个流程，来使用 IdM WebUI 恢复因为原因 6：**凭证冻结** 而吊销的 IdM 证书。

流程

1. 在 **Authentication** 菜单中，点击 **Certificates > Certificates**。
2. 单击证书的序列号，来打开证书信息页面。

图 7.3. 证书列表



<input type="checkbox"/>	Serial Number	Subject
<input type="checkbox"/>	1	CN=Certificate Authority,O=EXAMPLE.COM
<input type="checkbox"/>	2	CN=OCSP Subsystem,O=EXAMPLE.COM
<input type="checkbox"/>	3	CN=server.example.com,O=EXAMPLE.COM
<input type="checkbox"/>	4	CN=CA Subsystem,O=EXAMPLE.COM

3. 在证书信息页面中，单击 **Actions** → **Restore Certificate**。

7.6.2. 使用 IdM CLI 恢复带有集成 IdM CA 的证书

完成此流程，以使用 IdM CLI 恢复因为原因 6：证书冻结而撤销的 IdM 证书。

流程

- 使用 **ipa cert-remove-hold** 命令并指定证书序列号。例如：

```
$ ipa cert-remove-hold 1032
```

第 8 章 为智能卡验证配置身份管理

身份管理(IdM)支持使用如下方式的智能卡身份验证：

- IdM 证书颁发机构发布的用户证书
- 外部证书颁发机构发布的用户证书

您可以在 IdM 中为两种类型的证书配置智能卡验证。在这种情况下，**rootca.pem** CA 证书是包含可信外部证书颁发机构证书的文件。

有关 IdM 中智能卡验证的详情，请参考 [了解智能卡验证](#)。

有关配置智能卡验证的详情：

- [为智能卡验证配置 IdM 服务器](#)
- [为智能卡验证配置 IdM 客户端](#)
- [在 IdM Web UI 的用户条目中添加证书](#)
- [在 IdM CLI 中向用户条目中添加证书](#)
- [安装用来管理和使用智能卡的工具](#)
- [在智能卡中存储证书](#)
- [使用智能卡登录到 IdM](#)
- [使用智能卡身份验证配置 GDM 访问](#)
- [使用智能卡验证配置 su 访问](#)

8.1. 为智能卡验证配置 IDM 服务器

如果要为其证书是由身份管理(IdM)CA 信任的 <EXAMPLE.ORG> 域的证书颁发机构发布的用户启用智能卡验证，您必须获取以下证书，以便在运行配置 IdM 服务器的 **ipa-advise** 脚本时添加它们：

- 为 <EXAMPLE.ORG> CA 直接发布证书的根 CA 的证书，或者通过一个或多个其子 CA 签发证书。您可以从认证机构发布的证书的网页下载证书链。详情请查看 [配置浏览器来启用证书身份验证](#) 中的步骤 1 - 4a。
- IdM CA 证书。您可以从在其上运行 IdM CA 实例的 IdM 服务器上的 **/etc/ipa/ca.crt** 文件获取 CA 证书。
- 所有中间 CA 的证书，即介于 <EXAMPLE.ORG> CA 和 IdM CA 之间。

要为智能卡验证配置 IdM 服务器：

1. 获取 PEM 格式的 CA 证书文件。
2. 运行内置的 **ipa-advise** 脚本。
3. 重新加载系统配置。

先决条件

- 有到 IdM 服务器的 root 访问权限。
- 您有 root CA 证书和所有中间 CA 证书。

流程

1. 创建要进行配置的目录：

```
[root@server]# mkdir ~/SmartCard/
```

2. 进入该目录：

```
[root@server]# cd ~/SmartCard/
```

3. 获取存储在 PEM 格式文件中的相关 CA 证书。如果您的 CA 证书存储在再不同格式的文件中，如 DER，请将其转换为 PEM 格式。IdM 证书颁发机构证书采用 PEM 格式，位于 `/etc/ipa/ca.crt` 文件中。

将 DER 文件转换为 PEM 文件：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4. 为方便起见，将证书复制到您要配置进行配置的目录中：

```
[root@server SmartCard]# cp /tmp/rootca.pem ~/SmartCard/  
[root@server SmartCard]# cp /tmp/subca.pem ~/SmartCard/  
[root@server SmartCard]# cp /tmp/issuingca.pem ~/SmartCard/
```

5. 另外，如果您使用外部证书颁发机构的证书，请使用 `openssl x509` 工具查看 PEM 格式的文件内容，来检查 **Issuer** 和 **Subject** 值是否正确：

```
[root@server SmartCard]# openssl x509 -noout -text -in rootca.pem | more
```

6. 使用管理员特权，通过内置的 `ipa-advise` 工具生成配置脚本：

```
[root@server SmartCard]# kinit admin  
[root@server SmartCard]# ipa-advise config-server-for-smart-card-auth > config-server-for-smart-card-auth.sh
```

`config-server-for-smart-card-auth.sh` 脚本执行以下操作：

- 它配置 IdM Apache HTTP 服务器。
 - 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
 - 它将 IdM Web UI 配置为接受智能卡授权请求。
7. 执行脚本，将包含根 CA 和子 CA 证书的 PEM 文件添加为参数：

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh  
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh rootca.pem subca.pem  
issuingca.pem  
Ticket cache:KEYRING:persistent:0:0
```



```
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



注意

在任何子 CA 证书前，确保将根 CA 的证书添加为参数，并且 CA 或子 CA 证书还没有过期。

8. 另外，如果发布用户证书的证书颁发机构不提供任何在线证书状态协议(OCSP)响应程序，您可能需要禁用对 IdM Web UI 身份验证的 OCSP 检查：
 - a. 在 `/etc/httpd/conf.d/ssl.conf` 文件中将 **SSLOCSPEnable** 参数设为 **off**：

```
SSLOCSPEnable off
```

- b. 重启 Apache 守护进程(httpd)使更改立即生效：

```
[root@server SmartCard]# systemctl restart httpd
```



警告

如果您只使用 IdM CA 发出的用户证书，不要禁用 OCSP 检查。OCSP 响应器是 IdM 的一部分。

有关如何保持 OCSP 检查处于启用状态，同时防止 IdM 服务器拒绝用户证书（如果 IdM 服务器不包含有关颁发用户证书的 CA 侦听 OCSP 服务请求的位置的信息）的说明，请参阅 [Apache mod_ssl 配置选项](#) 中的 **SSLOCSPEnable** 指令。

该服务器现在被配置为智能卡验证。



注意

要在整个拓扑中启用智能卡验证，请在每个 IdM 服务器中运行操作过程。

8.2. 使用 ANSIBLE 为智能卡验证配置 IDM 服务器

您可以使用 Ansible 为其证书是由身份管理(IdM) CA 信任的 <EXAMPLE.ORG> 域的证书颁发机构发布的用户启用智能卡验证。要做到这一点，您必须获取以下证书，以便在使用 `ipasmartcard_server ansible-freeipa` 角色脚本运行 Ansible playbook 时使用它们：

- 为 <EXAMPLE.ORG> CA 直接发布证书的根 CA 的证书，或者通过一个或多个其子 CA 签发证书。您可以从认证机构发布的证书的网页下载证书链。详情请参阅 [配置浏览器以启用证书验证](#) 中的步骤 4。
- IdM CA 证书。您可以从任何 IdM CA 服务器上的 `/etc/ipa/ca.crt` 文件获取 CA 证书。

- 介于 <EXAMPLE.ORG> CA 和 IdM CA 之间的所有 CA 的证书。

先决条件

- 您有访问 IdM 服务器的 **root** 权限。
- 您需要知道 IdM **admin** 密码。
- 您有 root CA 证书、IdM CA 证书以及所有中间 CA 证书。
- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 **Ansible 清单文件**。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点,也就是在其上执行 **ansible-freeipa** 模块的节点，是 IdM 域的一部分，作为 IdM 客户端、服务器或副本。

步骤

1. 如果您的 CA 证书存储在不同格式（如 **DER**）的文件中，请将其转换为 **PEM** 格式：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

IdM 证书颁发机构证书采用 **PEM** 格式，位于 `/etc/ipa/ca.crt` 文件中。

2. （可选）使用 **openssl x509** 工具查看 **PEM** 格式的文件的内容，以检查 **Issuer** 和 **Subject** 值是否正确：

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. 进入您的 `~/MyPlaybooks/` 目录：

```
$ cd ~/MyPlaybooks/
```

4. 创建一个专用于 CA 证书的子目录：

```
$ mkdir SmartCard/
```

5. 为方便起见，将所有必需的证书复制到 `~/MyPlaybooks/SmartCard/` 目录中：

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/  
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/  
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. 在 Ansible 清单文件中指定以下内容：

- 要为智能卡验证进行配置的 IdM 服务器。

- IdM 管理员密码。
- CA 证书的路径按以下顺序：
 - root CA 证书文件
 - 中间 CA 证书文件
 - IdM CA 证书文件

文件类似如下：

```
[ipaserver]
ipaserver.idm.example.com

[ipareplicas]
ipareplica1.idm.example.com
ipareplica2.idm.example.com

[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password= "{{ ipaadmin_password }}"
ipasmartcard_server_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. 使用以下内容创建 **install-smartcard-server.yml** playbook：

```
---
- name: Playbook to set up smart card authentication for an IdM server
  hosts: ipaserver
  become: true

  roles:
  - role: ipasmartcard_server
    state: present
```

8. 保存该文件。
9. 运行 Ansible playbook。指定 playbook 文件、存储保护 **secret.yml** 文件的密码，以及清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-
smartcard-server.yml
```

ipasmartcard_server Ansible 角色执行以下操作：

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

10. 另外，如果发布用户证书的证书颁发机构不提供任何在线证书状态协议(OCSP)响应程序，您可能需要禁用对 IdM Web UI 身份验证的 OCSP 检查：
- 以 **root** 用户身份连接到 IdM 服务器：

```
ssh root@ipaserver.idm.example.com
```

- 在 `/etc/httpd/conf.d/ssl.conf` 文件中将 **SSLOCSPEnable** 参数设为 **off**：

```
SSLOCSPEnable off
```

- 重启 Apache 守护进程(httpd)使更改立即生效：

```
# systemctl restart httpd
```



警告

如果您只使用 IdM CA 发出的用户证书，不要禁用 OCSP 检查。OCSP 响应器是 IdM 的一部分。

有关如何保持 OCSP 检查处于启用状态，同时防止 IdM 服务器拒绝用户证书（如果 IdM 服务器不包含有关颁发用户证书的 CA 侦听 OCSP 服务请求的位置的信息）的说明，请参阅 [Apache mod_ssl 配置选项](#) 中的 **SSLOCSPEnable** 指令。

清单文件中列出的服务器现在已配置为进行智能卡验证。



注意

要在整个拓扑中启用智能卡验证，请将 Ansible playbook 中的 **hosts** 变量设为 **ipacluster**：

```
---
- name: Playbook to setup smartcard for IPA server and replicas
  hosts: ipacluster
  [...]
```

其他资源

- 在 `/usr/share/doc/ansible-freeipa/playbooks/` 目录中使用 **ipasmartcard_server** 角色的 playbook 示例

8.3. 为智能卡验证配置 IDM 客户端

按照以下流程为智能卡验证配置 IdM 客户端。这个过程需要运行在每个 IdM 系统、客户端或服务器上，您希望在使用智能卡进行身份验证时连接到这些系统。例如，若要启用从主机 A 到主机 B 的 **ssh** 连接，需要在主机 B 上运行脚本。

作为管理员，运行这个流程来使用如下方法启用智能卡身份验证

- **ssh** 协议
详情请查看 [使用智能卡验证配置 SSH 访问](#)。
- 控制台登录
- GNOME 显示管理器(GDM)
- **su** 命令

对于向 IdM Web UI 进行身份验证，不需要此流程。向 IdM Web UI 进行身份验证涉及两个主机，它们都不必是 IdM 客户端：

- 在其上运行浏览器的机器。机器可以在 IdM 域之外。
- 在其上运行 **httpd** 的 IdM 服务器。

以下流程假设您在 IdM 客户端，而不是 IdM 服务器上配置智能卡身份验证。因此，您需要两台计算机：生成配置脚本的 IdM 服务器，以及运行脚本的 IdM 客户端。

先决条件

- 为智能卡验证配置了您的 IdM 服务器，如 [为智能卡验证配置 IdM 服务器](#) 所述。
- 有对 IdM 服务器和 IdM 客户端的 root 访问权限。
- 您有 root CA 证书和所有中间 CA 证书。
- 您使用 **--mkhomedir** 选项安装了 IdM 客户端，以确保远程用户可以成功登录。如果您没有创建主目录，则默认登录位置为目录结构的根目录 `/`。

步骤

1. 在 IdM 服务器上，使用管理员权限通过 **ipa-advise** 生成配置脚本：

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

config-client-for-smart-card-auth.sh 脚本执行以下操作：

- 它配置智能卡守护进程。
 - 它设置系统范围的信任存储。
 - 它配置系统安全服务守护进程 (SSSD)，允许用户使用其用户名和密码或其智能卡进行验证。有关智能卡验证的 SSSD 配置集选项的详情，请参考 [RHEL 中的智能卡验证选项](#)。
2. 从 IdM 服务器中，将脚本复制到 IdM 客户端机器中选择的目录中：

```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh
root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh    100% 2419    3.5MB/s  00:00
```

3. 为了方便起见，将 IdM 服务器上的 PEM 格式的 CA 证书文件复制到 IdM 客户端机器上与在上一步中所使用的相同的目录中：

```
[root@server SmartCard]# scp {rootca.pem,subca.pem,issuingca.pem}
root@client.idm.example.com:/root/SmartCard/
Password:
rootca.pem          100% 1237  9.6KB/s 00:00
subca.pem           100% 2514 19.6KB/s 00:00
issuingca.pem       100% 2514 19.6KB/s 00:00
```

4. 在客户端机器上执行脚本，将包含 CA 证书的 PEM 文件添加为参数：

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh rootca.pem subca.pem
issuingca.pem
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



注意

在任何子 CA 证书前，确保将根 CA 的证书添加为参数，并且 CA 或子 CA 证书还没有过期。

现在为智能卡验证配置了客户端。

8.4. 使用 ANSIBLE 为智能卡验证配置 IDM 客户端

按照以下流程，使用 `ansible-freeipa ipasmartcard_client` 模块配置特定的身份管理(IdM)客户端，以允许 IdM 用户使用智能卡进行身份验证。运行这个流程为使用以下任一方法访问 IdM 的用户启用智能卡验证：

- **ssh** 协议
详情请查看 [使用智能卡验证配置 SSH 访问](#)。
- 控制台登录
- GNOME 显示管理器(GDM)
- **su** 命令



注意

对于向 IdM Web UI 进行身份验证，不需要此流程。向 IdM Web UI 进行身份验证涉及两个主机，它们都不必是 IdM 客户端：

- 在其上运行浏览器的机器。机器可以在 IdM 域之外。
- 在其上运行 **httpd** 的 IdM 服务器。

先决条件

- 为智能卡验证配置了您的 IdM 服务器，如 [使用 Ansible 配置 IdM 服务器以进行智能卡验证](#) 中所述。
- 有对 IdM 服务器和 IdM 客户端的 root 访问权限。
- 您有 root CA 证书、IdM CA 证书以及所有中间 CA 证书。
- 您已配置了 Ansible 控制节点以满足以下要求：
 - 您使用 Ansible 版本 2.14 或更高版本。
 - 您已在 Ansible 控制器上安装了 **ansible-freeipa** 软件包。
 - 示例假定在 `~/MyPlaybooks/` 目录中，您已创建了带有 IdM 服务器的完全限定域名(FQDN)的 [Ansible 清单文件](#)。
 - 示例假定 `secret.yml` Ansible vault 存储了 `ipadmin_password`。
- 目标节点,也就是在其上执行 **ansible-freeipa** 模块的节点，是 IdM 域的一部分，作为 IdM 客户端、服务器或副本。

流程

1. 如果您的 CA 证书存储在不同格式（如 **DER**）的文件中，请将其转换为 **PEM** 格式：

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

IdM CA 证书采用 **PEM** 格式，位于 `/etc/ipa/ca.crt` 文件中。

2. （可选）使用 **openssl x509** 工具查看 **PEM** 格式的文件的内容，以检查 **Issuer** 和 **Subject** 值是否正确：

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. 在 Ansible 控制节点上，导航到 `~/MyPlaybooks/` 目录：

```
$ cd ~/MyPlaybooks/
```

4. 创建一个专用于 CA 证书的子目录：

```
$ mkdir SmartCard/
```

5. 为方便起见，将所有必需的证书复制到 `~/MyPlaybooks/SmartCard/` 目录中，例如：

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. 在 Ansible 清单文件中指定以下内容：

- 要为智能卡验证配置的 IdM 客户端。
- IdM 管理员密码。
- CA 证书的路径按以下顺序：

- root CA 证书文件
- 中间 CA 证书文件
- IdM CA 证书文件

文件类似如下：

```
[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
ipasmartcard_client_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-
ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. 使用以下内容创建 `install-smartcard-clients.yml` playbook：

```
---
- name: Playbook to set up smart card authentication for an IdM client
  hosts: ipaclients
  become: true

  roles:
  - role: ipasmartcard_client
    state: present
```

8. 保存该文件。
9. 运行 Ansible playbook。指定 playbook 和清单文件：

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-
smartcard-clients.yml
```

`ipasmartcard_client` Ansible 角色执行以下操作：

- 它配置智能卡守护进程。
- 它设置系统范围的信任存储。
- 它配置系统安全服务守护进程(SSSD)，以允许用户使用其用户名和密码或者智能卡进行身份验证。有关智能卡验证的 SSSD 配置集选项的详情，请参考 [RHEL 中的智能卡验证选项](#)。

现在为智能卡验证配置了清单文件的 `ipaclients` 部分中列出的客户端。



注意

如果您使用 `--mkhomedir` 选项安装了 IdM 客户端，远程用户将能够登录到其主目录。否则，默认登录位置是目录结构 `/` 的根。

其他资源

- 在 `/usr/share/doc/ansible-freeipa/playbooks/` 目录中使用 `ipasmartcard_server` 角色的 playbook 示例

8.5. 在 IDM WEB UI 的用户条目中添加证书

按照以下流程，在 IdM Web UI 中向用户条目添加一个外部证书。



注意

也可以将证书映射数据上传到 IdM 中的用户条目，而不必上传整个证书。包含完整证书或证书映射数据的用户条目可以和相应的证书映射规则一起使用，以便于系统管理员配置智能卡身份验证。详情请查看

[配置身份验证的证书映射规则。](#)



注意

如果用户的证书由 IdM 证书颁发机构发布，则证书已存储在用户条目中，您不需要按照此流程操作。

先决条件

- 您有要添加到用户条目的证书。

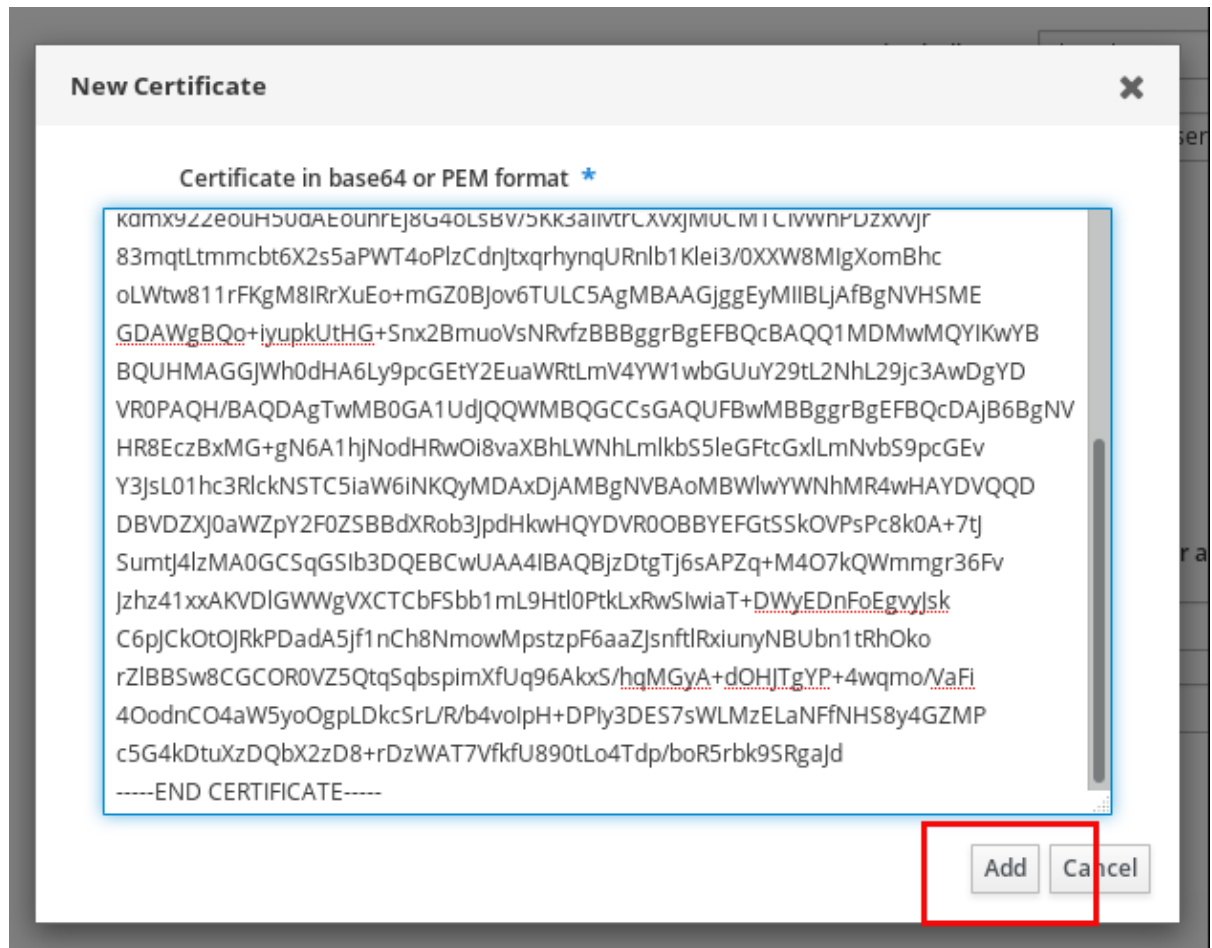
流程

1. 如果要给另一个用户添加证书，请以管理员身份登录到 IdM Web UI。要在您自己的配置文件中添加证书，您不需要管理员的凭证。
2. 导航到 **Users** → **Active users** → **sc_user**。
3. 找到 **Certificate** 选项，并单击 **Add**。
4. 在命令行界面中，使用 **cat** 工具或文本编辑器以 **PEM** 格式显示证书：

```
[user@client SmartCard]$ cat testuser.crt
```

5. 将证书从 CLI 复制并粘贴到 Web UI 中打开的窗口中。
6. 单击 **Add**。

图 8.1. 在 IdM Web UI 中添加新证书



`sc_user` 条目现在包含一个外部证书。

8.6. 在 IDM CLI 中向用户条目中添加证书

按照以下流程，在 IdM CLI 中将外部证书添加到用户条目。



注意

也可以将证书映射数据上传到 IdM 中的用户条目，而不必上传整个证书。包含完整证书或证书映射数据的用户条目可以和相应的证书映射规则一起使用，以便于系统管理员配置智能卡身份验证。详情请参阅 [配置身份验证的证书映射规则](#)。



注意

如果用户的证书由 IdM 证书颁发机构发布，则证书已存储在用户条目中，您不需要按照此流程操作。

先决条件

- 您有要添加到用户条目的证书。

流程

1. 如果要给另一个用户添加证书，请以管理员身份登录到 IdM CLI：

```
[user@client SmartCard]$ kinit admin
```

要在您自己的配置文件中添加证书，您不需要管理员的凭证：

```
[user@client SmartCard]$ kinit sc_user
```

2. 创建一个包含证书的环境变量，该变量移除了标头和页脚，并串联成一行，这是 `ipa user-add-cert` 命令期望的格式：

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt |
base64 -w0 -`
```

请注意，`testuser.crt` 文件中的证书必须是 **PEM** 格式。

3. 使用 `ipa user-add-cert` 命令将证书添加到 `sc_user` 的配置文件：

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

`sc_user` 条目现在包含一个外部证书。

8.7. 安装用来管理和使用智能卡的工具

先决条件

- **gnutls-utils** 软件包已安装。
- **opensc** 软件包已安装。
- **pcscd** 服务正在运行。

在配置智能卡前，您必须安装相应的工具，该工具可以生成证书，并启动 **pcscd** 服务。

步骤

1. 安装 **opensc** 和 **gnutls-utils** 软件包：

```
# dnf -y install opensc gnutls-utils
```

2. 启动 **pcscd** 服务。

```
# systemctl start pcscd
```

验证步骤

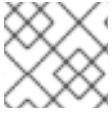
- 验证 **pcscd** 服务是否已启动并运行

```
# systemctl status pcscd
```

8.8. 准备智能卡并将证书和密钥上传到智能卡

按照以下流程，使用 **pkcs15-init** 工具配置智能卡，该工具可帮助您进行配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys (PUKs)
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 如果需要，请锁定智能卡设置，因为某些智能卡需要这个类型的最终化



注意

pkcs15-init 工具可能无法使用所有智能卡。您必须使用您使用智能卡的工具。

先决条件

- 已安装 **opensc** 软件包，其中包括 **pkcs15-init** 工具。
如需了解更多详细信息，请参阅 [安装用于管理和使用智能卡的工具](#)。
- 该卡插入读卡器并连接到计算机。
- 您有一个要存储在智能卡上的私钥、公钥和证书。在此流程中，**testuser.key**、**testuserpublic.key** 和 **testuser.crt** 是用于私钥、公钥和证书的名称。
- 您有当前的智能卡用户 PIN 和 Security Officer PIN (SO-PIN)。

流程

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
--pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pkcs15-init 工具在智能卡上创建一个新插槽。

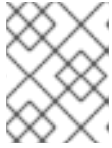
3. 为插槽设置标签和验证 ID：

```
$ pkcs15-init --store-pin --label testuser \
--auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

标签设置为人类可读的值，在本例中为 **testuser**。**auth-id** 必须是两个十六进制值，在本例中设为 **01**。

4. 在智能卡的新插槽中存储并标记私钥：

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \ --auth-id 01 --id 01 --pin
963214
Using reader with a card: Reader name
```



注意

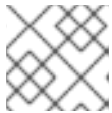
在存储您的私钥并将证书存储在下一步中时，您为 **--id** 指定的值必须相同。建议为 **--id** 指定自己的值，否则它们将更复杂的值由工具计算。

5. 在智能卡上的新插槽中存储并标记该证书：

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \ --auth-id 01 --id 01 --format
pem --pin 963214
Using reader with a card: Reader name
```

6. 可选：在智能卡的新插槽中保存并标记公钥：

```
$ pkcs15-init --store-public-key testuserpublic.key --label testuserpublic_key --auth-id 01 --id
01 --pin 963214
Using reader with a card: Reader name
```



注意

如果公钥与私钥或证书对应，请指定与私钥或证书的 ID 相同的 ID。

7. 可选：某些智能卡要求您通过锁定设置来完成卡：

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

8.9. 使用智能卡登录到 IDM

按照以下流程，使用智能卡登录到 IdM Web UI。

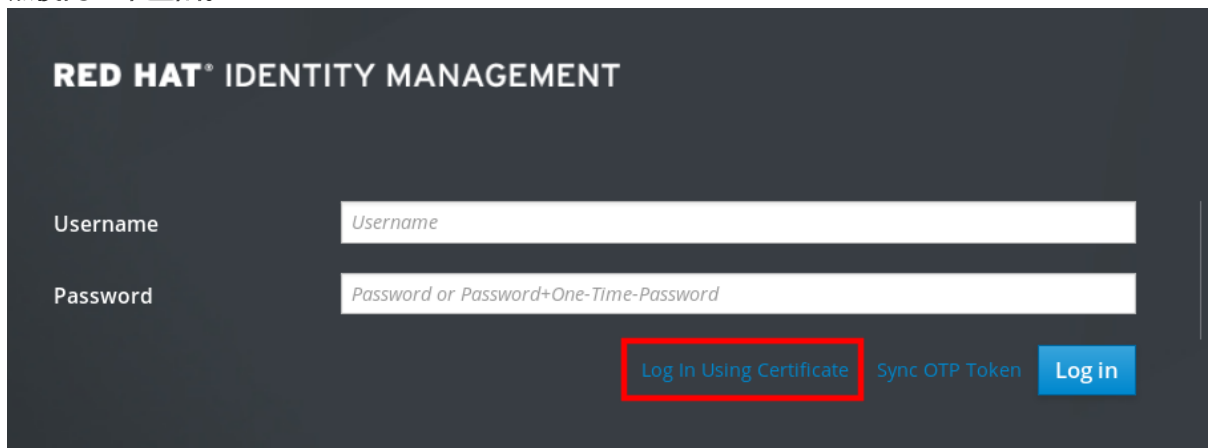
先决条件

- web 浏览器被配置为使用智能卡验证。
- IdM 服务器被配置为智能卡验证。
- 在您的智能卡中安装的证书由 IdM 服务器发出，或者已添加到 IdM 的用户条目中。
- 您知道解锁智能卡所需的 PIN。
- 智能卡已插入到读取器中。

流程

1. 在浏览器中打开 IdM Web UI。

2. 点使用证书登陆。

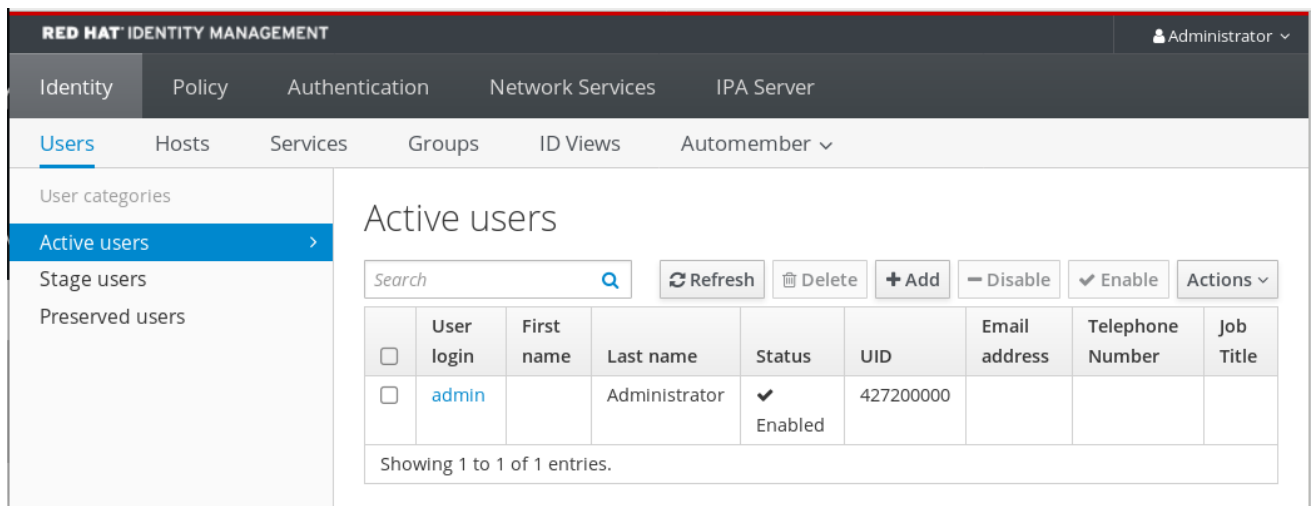


- 如果 **Password Required** 对话框打开，请添加 PIN 来解锁智能卡，然后单击 **OK** 按钮。此时会打开 **User Identification Request** 对话框。

如果智能卡包含多个证书，请在 **选择用于验证的证书** 下方的下拉列表中选择您要用于身份验证的证书。

- 单击 **OK** 按钮。

现在，您已成功登录到 IdM Web UI。



8.10. 在 IDM 客户端中使用智能卡验证登录到 GDM

GNOME 桌面管理器(GDM)需要身份验证。您可以使用您的密码，但是，您也可以使用智能卡进行身份验证。

按照以下流程，使用智能卡验证访问 GDM。

先决条件

- 为智能卡验证配置了系统。详情请参阅[为智能卡验证配置 IdM 客户端](#)。
- 该智能卡包含您的证书和私钥。
- 该用户帐户是 IdM 域的成员。
- 智能卡上的证书通过以下方式映射到用户条目：

- 为特定用户条目分配证书。详情请参阅 [Adding a certificate to a user entry in the IdM Web UI](#) 或 [Adding a certificate to a user entry in the IdM CLI](#) 。
- 应用到该帐户的证书映射数据。详情请查看用于 [在智能卡上配置身份验证的证书映射规则](#)。

流程

1. 在读取器中插入智能卡。
2. 输入智能卡 PIN。
3. 点 **Sign In**。

您成功登录到 RHEL 系统，并且您有一张由 IdM 服务器提供的 TGT。

验证步骤

- 在 **Terminal** 中输入 **klist**，并检查结果：

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM

Valid starting    Expires          Service principal
04/20/2020 13:58:24  04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 04/27/2020 08:58:15
```

8.11. 在 SU 命令中使用智能卡验证

切换到其他用户需要身份验证。您可以使用密码或证书。按照以下流程，通过 **su** 命令使用智能卡。这意味着输入 **su** 命令后，系统会提示您输入智能卡 PIN。

先决条件

- 为智能卡验证配置了您的 IdM 服务器和客户端。
 - 请参阅[为智能卡验证配置 IdM 服务器](#)
 - 请参阅[为智能卡验证配置 IdM 客户端](#)
- 该智能卡包含您的证书和私钥。请参阅[智能卡中的证书](#)
- 该卡插入读卡器并连接到计算机。

步骤

- 在终端窗口中，使用 **su** 命令切换到其他用户：

```
$ su - example.user
PIN for smart_card
```

如果配置正确，会提示您输入智能卡 PIN。

第 9 章 为 IDM 中智能卡验证配置 ADCS 发布的证书

要在 IdM 中为其证书是由活动目录(AD)证书服务发布的用户配置智能卡验证：

- 您的部署是基于身份管理(IdM)和活动目录(AD)之间的跨林信任。
- 您希望允许智能卡验证存储在 AD 中的帐户的用户。
- 证书创建并存储在活动目录证书服务(ADCS)中。

有关智能卡验证的概述，请参阅[了解智能卡验证](#)。

配置通过以下步骤完成：

- [将 CA 和用户证书从活动目录复制到 IdM 服务器和客户端](#)
- [使用 ADCS 证书为智能卡身份验证配置 IdM 服务器和客户端](#)
- [转换 PFX\(PKCS#12\)文件，以便能够将证书和私钥存储到智能卡中](#)
- [在 sssd.conf 文件中配置超时](#)
- [为智能卡身份验证创建证书映射规则](#)

先决条件

- 身份管理(IdM)和活动目录(AD)信任已安装
详情请参阅在[IdM 和 AD 之间安装信任](#)。
- 活动目录证书服务(ADCS)已安装，并且用户证书已生成

9.1. 信任配置和证书使用量所需的 WINDOWS 服务器设置

您必须在 Windows 服务器上配置以下内容：

- 已安装活动目录证书服务(ADCS)
- 创建证书颁发机构
- [可选] 如果您正在使用证书颁发机构 Web 注册，则必须配置互联网信息服务(IIS)

导出证书：

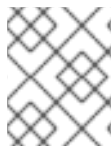
- 密钥必须有 **2048** 位或更多
- 包括一个私钥
- 您将需要以下格式的证书：个人信息交换– **PKCS #12(.PFX)**
 - 启用证书隐私

9.2. 使用 SFTP 从 ACTIVE DIRECTORY 复制证书

要能够使用智能卡身份验证，您需要复制以下证书文件：

- **CER** 格式的根 CA 证书：IdM 服务器上的 **adcs-winservice-ca.cer**。

- 具有 PFX 格式私钥的用户证书：IdM 客户端上的 **aduser1.pfx**。



注意

这个过程预期 SSH 访问是允许的。如果 SSH 不可用，用户必须将文件从 AD 服务器复制到 IdM 服务器和客户端。

步骤

1. 从 **IdM 服务器** 连接，并将 **adcs-winservice-ca.cer** 根证书复制到 IdM 服务器：

```
root@idmserver ~]# sftp Administrator@winservice.ad.example.com
Administrator@winservice.ad.example.com's password:
Connected to Administrator@winservice.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winservice-ca.cer  aduser1.pfx
sftp>
sftp> get adcs-winservice-ca.cer
Fetching <Path to certificates>/adcs-winservice-ca.cer to adcs-winservice-ca.cer
<Path to certificates>/adcs-winservice-ca.cer      100% 1254  15KB/s 00:00
sftp quit
```

2. 从 **IdM 客户端** 连接，并将 **aduser1.pfx** 用户证书复制到客户端：

```
[root@client1 ~]# sftp Administrator@winservice.ad.example.com
Administrator@winservice.ad.example.com's password:
Connected to Administrator@winservice.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx      100% 1254  15KB/s 00:00
sftp quit
```

现在，CA 证书保存在 IdM 服务器上，用户证书存储在客户端机器上。

9.3. 使用 ADCS 证书为智能卡身份验证配置 IDM 服务器和客户端

您必须配置 IdM（身份管理）服务器和客户端，以便能够在 IdM 环境中使用智能卡身份验证。IdM 包含进行了所有必要更改的 **ipa-advise** 脚本：

- 安装所需的软件包
- 配置 IdM 服务器和客户端
- 将 CA 证书复制到期望的位置

您可以在 IdM 服务器中运行 **ipa-advise**。

按照以下流程为智能卡验证配置服务器和客户端：

- 在 IdM 服务器中：准备 **ipa-advise** 脚本来配置您的 IdM 服务器进行智能卡验证。
- 在 IdM 客户端中：准备 **ipa-advise** 脚本来配置您的 IdM 客户端进行智能卡验证。

- 在 IdM 服务器中：使用 AD 证书应用 IdM 服务器上的 **ipa-advise** 服务器脚本。
- 将客户端脚本移动到 IdM 客户端机器中。
- 在 IdM 客户端中：使用 AD 证书应用 IdM 客户端中的 **ipa-advise** 客户端脚本。

先决条件

- 证书已复制到 IdM 服务器。
- 获取 Kerberos 票据。
- 以具有管理权限的用户身份登录。

步骤

1. 在 IdM 服务器中，使用 **ipa-advise** 脚本来配置客户端：

```
[root@idmserver ~]# ipa-advise config-client-for-smart-card-auth > sc_client.sh
```

2. 在 IdM 服务器中，使用 **ipa-advise** 脚本来配置服务器：

```
[root@idmserver ~]# ipa-advise config-server-for-smart-card-auth > sc_server.sh
```

3. 在 IdM 服务器中执行脚本：

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winservice-ca.cer
```

- 它配置 IdM Apache HTTP 服务器。
- 它在 KDC（Key Distribution Center）中启用 PKINIT（Public Key Cryptography for Initial Authentication in Kerberos）。
- 它将 IdM Web UI 配置为接受智能卡授权请求。

4. 将 **sc_client.sh** 脚本复制到客户端系统中：

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
Password:
sc_client.sh          100% 2857  1.6MB/s  00:00
```

5. 将 Windows 证书复制到客户端系统中：

```
[root@idmserver ~]# scp adcs-winservice-ca.cer root@client1.idm.example.com:/root
Password:
adcs-winservice-ca.cer 100% 1254  952.0KB/s  00:00
```

6. 在客户端系统中运行客户端脚本：

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winservice-ca.cer
```

CA 证书以正确格式安装在 IdM 服务器和客户端系统中，下一步是将用户证书复制到智能卡本身。

9.4. 转换 PFX 文件

在将 PFX(PKCS#12)文件存储到智能卡之前，您必须：

- 将文件转换为 PEM 格式
- 将私钥和证书提取到两个不同的文件

先决条件

- PFX 文件被复制到 IdM 客户端机器中。

流程

1. 在 IdM 客户端中，采用 PEM 格式：

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -clcerts -nodes
Enter Import Password:
```

2. 将密钥提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem > aduser1.key
```

3. 将公共证书提取到单独的文件中：

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out aduser1_cert_only.pem > aduser1.crt
```

此时，您可以将 **aduser1.key** 和 **aduser1.crt** 存储到智能卡。

9.5. 安装用来管理和使用智能卡的工具

先决条件

- **gnutls-utils** 软件包已安装。
- **opencsc** 软件包已安装。
- **pcscd** 服务正在运行。

在配置智能卡前，您必须安装相应的工具，该工具可以生成证书，并启动 **pcscd** 服务。

步骤

1. 安装 **opencsc** 和 **gnutls-utils** 软件包：

```
# dnf -y install opencsc gnutls-utils
```

2. 启动 **pcscd** 服务。

```
# systemctl start pcscd
```

验证步骤

- 验证 **pcscd** 服务是否已启动并运行

```
# systemctl status pcscd
```

9.6. 准备智能卡并将证书和密钥上传到智能卡

按照以下流程，使用 **pkcs15-init** 工具配置智能卡，该工具可帮助您进行配置：

- 擦除智能卡
- 设置新的 PIN 和可选的 PIN Unblocking Keys (PUKs)
- 在智能卡上创建新插槽
- 在插槽存储证书、私钥和公钥
- 如果需要，请锁定智能卡设置，因为某些智能卡需要这个类型的最终化



注意

pkcs15-init 工具可能无法使用所有智能卡。您必须使用您使用智能卡的工具。

先决条件

- 已安装 **opensc** 软件包，其中包括 **pkcs15-init** 工具。
如需了解更多详细信息，请参阅 [安装用于管理和使用智能卡的工具](#)。
- 该卡插入读卡器并连接到计算机。
- 您有一个要存储在智能卡上的私钥、公钥和证书。在此流程中，**testuser.key**、**testuserpublic.key** 和 **testuser.crt** 是用于私钥、公钥和证书的名称。
- 您有当前的智能卡用户 PIN 和 Security Officer PIN (SO-PIN)。

流程

1. 擦除智能卡并使用您的 PIN 验证自己：

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

这个卡已经被清除。

2. 初始化智能卡，设置您的用户 PIN 和 PUK，以及您的安全响应 PIN 和 PUK：

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
--pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

pkcs15-init 工具在智能卡上创建一个新插槽。

- 为插槽设置标签和验证 ID :

```
$ pkcs15-init --store-pin --label testuser \ --auth-id 01 --so-pin 65498714 --pin 963214 --puk
321478
Using reader with a card: Reader name
```

标签设置为人类可读的值，在本例中为 **testuser**。**auth-id** 必须是两个十六进制值，在本例中设为 **01**。

- 在智能卡的新插槽中存储并标记私钥 :

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \ --auth-id 01 --id 01 --pin
963214
Using reader with a card: Reader name
```



注意

在存储您的私钥并将证书存储在下一步中时，您为 **--id** 指定的值必须相同。建议为 **--id** 指定自己的值，否则它们将更复杂的值由工具计算。

- 在智能卡上的新插槽中存储并标记该证书 :

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \ --auth-id 01 --id 01 --format
pem --pin 963214
Using reader with a card: Reader name
```

- 可选：在智能卡的新插槽中保存并标记公钥 :

```
$ pkcs15-init --store-public-key testuserpublic.key --label testuserpublic_key --auth-id 01 --id
01 --pin 963214
Using reader with a card: Reader name
```



注意

如果公钥与私钥或证书对应，请指定与私钥或证书的 ID 相同的 ID。

- 可选：某些智能卡要求您通过锁定设置来完成卡 :

```
$ pkcs15-init -F
```

此时您的智能卡在新创建的插槽中包含证书、私钥和公钥。您还创建了您的用户 PIN 和 PUK，以及安全响应 PIN 和 PUK。

9.7. 在 SSSD.CONF 中配置超时

使用智能卡证书进行身份验证的时间可能比 SSSD 使用的默认超时时间更长。超时时间可能是由以下原因造成的：

- 慢的读取器
- 物理设备到虚拟环境的转发

- 保存在智能卡上的证书太多
- 如果使用 OCSP 验证证书，则来自 OCSP（在线证书状态协议）响应器的响应较慢

在这种情况下，您可以在 **sssd.conf** 文件中将以下超时时间延长到 60 秒：

- **p11_child_timeout**
- **krb5_auth_timeout**

先决条件

- 您必须以 root 身份登录。

步骤

1. 打开 **sssd.conf** 文件：

```
[root@idmclient1 ~]# vim /etc/sss/sss.conf
```

2. 更改 **p11_child_timeout** 的值：

```
[pam]
p11_child_timeout = 60
```

3. 更改 **krb5_auth_timeout** 的值：

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. 保存设置。

现在，在验证被认为出现超时故障前，与智能卡的交互可以运行 1 分钟（60 秒）。

9.8. 为智能卡身份验证创建证书映射规则

如果要将一个证书用于 AD(Active Directory)和 IdM 中拥有帐户的用户，您可以在 IdM 服务器上创建证书映射规则。

创建这样的规则后，用户可以在这两个域中使用智能卡进行验证。

有关证书映射规则的详情，请参阅 [用于配置身份验证的证书映射规则](#)。

第 10 章 在身份管理中配置证书映射规则

证书映射规则是允许用户在 Identity Management(IdM)管理员无法访问某些用户证书时使用证书进行身份验证的方法。这通常是因为证书已由外部证书颁发机构发布。

10.1. 用于配置身份验证的证书映射规则

在以下情况下可能需要配置证书映射规则：

- 证书已由活动目录(AD)的证书系统发布，且活动目录与 IdM 域有信任关系。
- 证书已由外部证书颁发机构发布。
- IdM 环境较大，有很多用户使用智能卡的用户。在这种情况下，添加完整证书可能会比较复杂。在大多数情况下，主题和签发者是可预测的，因此与完整证书相比，更容易提前添加。

作为系统管理员，您可以创建证书映射规则，并在向特定用户签发证书前向用户条目添加证书映射数据。签发证书后，用户就可以使用证书登录，即使证书还没有上传到用户条目。

另外，因为证书会定期续订，证书映射规则减少了管理开销。续订用户证书时，管理员不必更新用户条目。例如，如果映射基于 **Subject** 和 **Issuer** 的值，如果新的证书具有与旧证书相同的主题和签发者，则映射仍适用。如果使用完整证书，则管理员必须将新证书上传到用户条目以替换旧证书。

设置证书映射：

1. 管理员必须将证书映射数据或完整证书加载到用户帐户中。
2. 管理员必须创建证书映射规则，以允许为其帐户包含与证书信息匹配的证书映射数据条目的用户成功登录到 IdM。

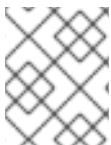
创建证书映射规则后，当最终用户提供保存在 [文件系统](#) 或 [智能卡](#) 中的证书时，身份验证是成功的。



注意

密钥分发中心(KDC)有一个用于证书映射规则的缓存。缓存在第一个 **certauth** 请求进行填充，它有一个硬编码的 300 秒超时。KDC 不会看到对证书映射规则的任何更改，除非它重启了或缓存到期了。

有关构成映射规则的单件，以及如何获取和使用它们的详细信息，请参阅 [IdM 中的身份映射规则组件](#)，以及 [获取证书中的签发者](#)，以便在匹配规则中使用。



注意

您的证书映射规则可取决于您使用证书的用例。例如，如果您使用带有证书的 SSH，则必须有完整的证书来从证书中提取公钥。

10.2. IDM 中身份映射规则的组件

您可以在 IdM 中创建 *身份映射规则* 时配置不同的组件。每个组件都有一个可覆盖的默认值。您可以在 Web UI 或 CLI 中定义这些组件。在 CLI 中，身份映射规则是使用 **ipa certmaprule-add** 命令创建的。

映射规则

映射规则组件将（或 *映射*）证书与一个或多个用户帐户相关联。规则定义了一个 LDAP 搜索过滤器，用于将证书与预期用户帐户相关联。

由不同证书颁发机构(CA)发布的证书可能具有不同的属性，可能在不同的域中使用。因此，IdM 不会以无条件的方式应用映射规则，而是只应用于适当的证书。适当的证书是使用 *匹配规则* 定义的。

请注意，如果您将映射规则选项留空，则会在 **userCertificate** 属性中搜索证书作为编码的二进制文件。

在 CLI 中使用 **--maprule** 选项定义映射规则。

匹配规则

匹配的规则组件选择您要应用映射规则的证书。默认匹配规则与带有 **digitalSignature key** 使用和 **clientAuth extended key** 使用的证书匹配。

使用 **--matchrule** 选项在 CLI 中定义匹配的规则。

域列表

域列表指定您希望 IdM 在处理身份映射规则时搜索用户的身份域。如果您未指定选项，IdM 将仅在 IdM 客户端所属的本地域中搜索用户。

使用 **--domain** 选项在 CLI 中定义域。

优先级

当多个规则适用于证书时，具有最高优先级的规则将具有优先权。所有其他规则将被忽略。

- 数字值越低，身份映射规则的优先级越高。例如，具有优先级 1 的规则的优先级高于优先级 2 的规则。
- 如果规则没有定义优先级值，它具有最低的优先级。

使用 **--priority** 选项在 CLI 中定义映射规则优先级。

证书映射规则示例

要使用 CLI 定义名为 **simple_rule** 证书映射规则，如果该证书上的 **Subject** 与 IdM 中用户帐户中的 **certmapdata** 条目匹配，则允许对 **EXAMPLE.ORG** 机构的 **智能卡 CA** 发布的证书进行身份验证：

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

10.3. 从证书获取数据，以便在匹配规则中使用

这个流程描述了如何从证书获取数据，以便您可以将其复制并粘贴到证书映射规则的匹配规则中。要获得匹配规则所需的数据，请使用 **sssctl cert-show** 或 **sssctl cert-eval-rule** 命令。

先决条件

- 您有 PEM 格式的用户证书。

步骤

1. 创建一个指向证书的变量，该变量还确保其被正确编码，以便您可以检索所需的数据。

```
# CERT=$(openssl x509 -in /path/to/certificate -outform der|base64 -w0)
```


2. 使用 **sssctl cert-eval-rule** 来确定匹配的数据。在以下示例中，使用了证书序列号。

```
# sssctl cert-eval-rule $CERT --match='<ISSUER>CN=adcs19-WIN1-
CA,DC=AD,DC=EXAMPLE,DC=COM' --map='LDAPU1:(altSecurityIdentities=X509:<l>
{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})'
Certificate matches rule.
Mapping filter:

(altSecurityIdentities=X509:<l>DC=com,DC=example,DC=ad,CN=adcs19-WIN1-
CA<SR>0F0000000000DB8852DD7B246C9C0F0000003B)
```

在这种情况下，将 **altSecurityIdentities=** 后的所有内容添加到 AD 中用户的 **altSecurityIdentities** 属性中。如果使用 SKI 映射，请使用 **--map='LDAPU1:(altSecurityIdentities=X509:<SKI>{subject_key_id!hex_u})'**。

3. 另外，要根据指定证书的签发者必须与 **ad.example.com** 域的 **adcs19-WIN1-CA** 匹配的匹配规则在 CLI 中创建一个新的映射规则，证书的序列号必须与用户帐户中的 **altSecurityIdentities** 条目匹配：

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=adcs19-WIN1-
CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule 'LDAPU1:(altSecurityIdentities=X509:<l>
{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})'
```

10.4. 为存储在 IDM 中的用户配置证书映射

如果其配置的证书身份验证存储在 IdM 中的用户在 IdM 中启用了证书映射，则系统管理员必须完成以下任务：

- 设置一个证书映射规则，以便具有与映射规则及其证书映射数据条目中指定的条件匹配的证书的 IdM 用户可以向 IdM 进行身份验证。
- 将证书映射数据输入到 IdM 用户条目中，以便用户可以使用多个证书进行身份验证，只要它们都包含证书映射数据条目中指定的值。

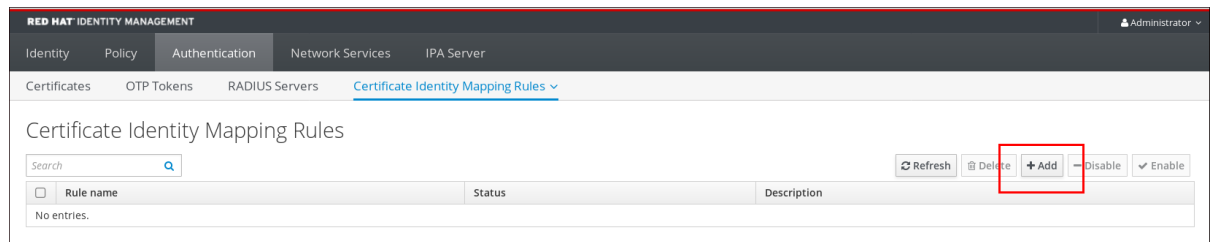
先决条件

- 用户在 IdM 中有一个帐户。
- 管理员具有要添加到用户条目的整个证书或证书映射数据。

10.4.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录到 IdM Web UI。
2. 进入到 **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules**。
3. 点 **Add**。

图 10.1. 在 IdM web UI 中添加新证书映射规则



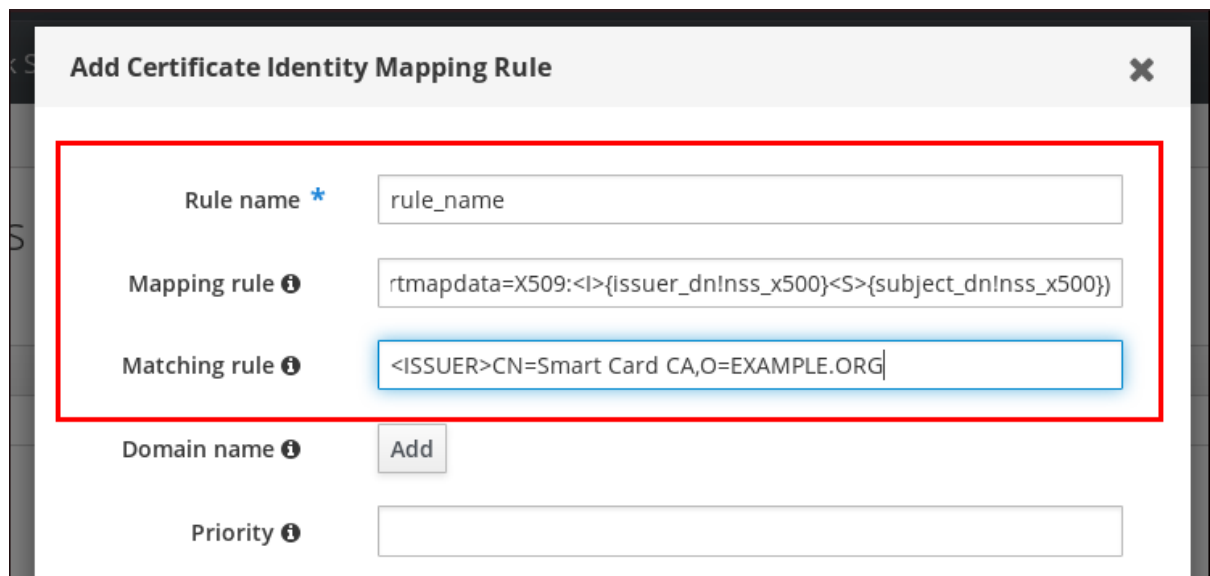
4. 输入规则名称。
5. 输入映射规则。例如，要让 IdM 搜索提供给它们的任何证书中带有 **Issuer** 和 **Subject** 条目，并根据在提供的证书中的这两个条目是否被找到来决定进行验证或不验证。

```
(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
```

6. 输入匹配的规则。例如，只允许由 **EXAMPLE.ORG** 机构的 **智能卡 CA** 签发的证书以向 IdM 验证用户：

```
<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
```

图 10.2. 在 IdM Web UI 中输入证书映射规则的详情



7. 点对话框底部的 **Add**，添加规则并关闭该框。
8. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

现在，您设置了一个证书映射规则，它将把在智能卡证书中找到的映射规则中指定的数据类型与您的 IdM 用户条目中的证书映射数据进行比较。找到匹配项后，它会验证匹配的用户。

10.4.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。例如，要让 IdM 搜索任何提供的证书中的 **Issuer** 和 **Subject** 条目，并根据在提供的证书中找到的这两个条目的信息决定是否进行验证，只允许使用 **EXAMPLE.ORG** 机构的 **智能卡 CA** 发布的证书：

```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
-----
Added Certificate Identity Mapping Rule "rule_name"
-----
Rule name: rule_name
Mapping rule: (ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
Matching rule: <ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
Enabled: TRUE
```

3. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

现在，您设置了一个证书映射规则，它将把在智能卡证书中找到的映射规则中指定的数据类型与您的 IdM 用户条目中的证书映射数据进行比较。找到匹配项后，它会验证匹配的用户。

10.4.3. 在 IdM web UI 中的用户条目中添加证书映射数据

1. 以管理员身份登录 IdM Web UI。
2. 进入 **Users** → **Active users** → **idm_user**。
3. 找到 **Certificate mapping data** 选项并点 **Add**。
4. 选择以下选项之一：
 - 如果您有 **idm_user** 证书：
 - a. 在命令行界面中，使用 **cat** 工具或文本编辑器显示证书：

```
[root@server ~]# cat idm_user_certificate.pem
-----BEGIN CERTIFICATE-----
MIIFFTCCA/2gAwIBAgIBejANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9JRE0
u
RVhBTvBMRS5DT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcm10eTAeFw0x
ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0IETS5F
WEFN
[...output truncated...]
```

- b. 复制证书。
- c. 在 IdM Web UI 中，点 **证书** 旁边的 **Add**，并将证书粘贴到打开的窗口中。

图 10.3. 添加用户的证书映射数据：证书

User: demouser
demouser is a member of:

Settings | User Groups | Netgroups | Roles | HBAC Rules | Sudo Rules

Refresh | Revert | Save | Actions

Identity Settings

Job Title:

First name *:

Last name *:

Full name *:

Display name:

Initials:

GECOS:

Class:

Account Settings

User login: demouser

Password: *****

Password expiration: 2016-07-14 10:14:41Z

UID:

GID:

Principal alias: demouser@IDM.EXAMPLE.COM

Kerberos principal expiration: : UTC

Login shell:

Home directory:

SSH public keys:

Certificates:

- 如果您没有可供使用的 `idm_user` 证书，但知道证书的 **Issuer** 和 **Subject**，请检查 **Issuer** 和 **subject** 单选按钮，并在两个框中输入值。

图 10.4. 添加用户的证书映射数据：签发者和主题

GID: 1997000009

Add Certificate Mapping Data

Certificate mapping data

Certificate mapping data

Certificate

Issuer and subject

Issuer *:

Subject *:

Certificate mapping

5. 点 **Add**。

验证步骤

如果您有访问 `.pem` 格式的整个证书的权限，请验证是否用户和证书已链接：

- 使用 `sss_cache` 实用程序使 SSSD 缓存中的 `idm_user` 记录失效，并强制重新载入 `idm_user` 信息：

```
# sss_cache -u idm_user
```

- 使用包含 IdM 用户证书的文件名称运行 `ipa certmap-match` 命令：

```
# ipa certmap-match idm_user_cert.pem
-----
1 user matched
-----
Domain: IDM.EXAMPLE.COM
User logins: idm_user
-----
Number of entries returned 1
-----
```

输出确认现在已将证书映射数据添加到 **idm_user**，并且存在对应的映射规则。这意味着，您可以使用与定义的证书映射数据匹配的证书，以 **idm_user** 进行身份验证。

10.4.4. 在 IdM CLI 中向用户条目添加证书映射数据

1. 获取管理员凭证：

```
# kinit admin
```

2. 选择以下选项之一：

- 如果您有 **idm_user** 证书，请使用 **ipa user-add-cert** 命令将证书添加到用户帐户中：

```
# CERT=$(openssl x509 -in idm_user_cert.pem -outform der|base64 -w0)
# ipa user-add-certmapdata idm_user --certificate $CERT
```

- 如果您没有 **idm_user** 证书，但知道用户证书的 **Issuer** 和 **Subject**：

```
# ipa user-add-certmapdata idm_user --subject "O=EXAMPLE.ORG,CN=test" --
issuer "CN=Smart Card CA,O=EXAMPLE.ORG"
-----
Added certificate mappings to user "idm_user"
-----
User login: idm_user
Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

验证步骤

如果您有访问 **.pem** 格式的整个证书的权限，请验证是否用户和证书已链接：

1. 使用 **sss_cache** 实用程序使 SSSD 缓存中的 **idm_user** 记录失效，并强制重新载入 **idm_user** 信息：

```
# sss_cache -u idm_user
```

2. 使用包含 IdM 用户证书的文件名称运行 **ipa certmap-match** 命令：

```
# ipa certmap-match idm_user_cert.pem
-----
1 user matched
-----
Domain: IDM.EXAMPLE.COM
User logins: idm_user
```

```
-----
Number of entries returned 1
-----
```

输出确认现在已将证书映射数据添加到 **idm_user**，并且存在对应的映射规则。这意味着，您可以使用与定义的证书映射数据匹配的证书，以 **idm_user** 进行身份验证。

10.5. 使用 ACTIVE DIRECTORY 域信任的证书映射规则

如果 IdM 部署与活动目录(AD)域有信任关系，则可能会有不同的证书映射用例。

根据 AD 配置，可能会出现以下情况：

- 如果证书是由 AD 证书系统发布的，但用户和证书存储在 IdM 中，则身份验证请求的映射和整个处理发生在 IdM 端。有关配置此情境的详情，请参阅[为存储在 IdM 中的用户配置证书映射](#)
- 如果用户存储在 AD 中，则身份验证请求的处理发生在 AD 中。有三个不同的子案例：
 - AD 用户条目包含整个证书。有关在这种情况下配置 IdM 的详情，请参考[AD 用户条目包含整个证书的用户配置证书映射](#)。
 - AD 配置为将用户证书映射到用户帐户。在这种情况下，AD 用户条目不包含整个证书，而是包含名为 **altSecurityIdentities** 的属性。有关如何在这种场景中配置 IdM 的详情，请参阅在[将 AD 配置为将用户证书映射到用户帐户时配置证书映射](#)。
 - AD 用户条目既不包含整个证书也不包括映射数据。在这种情况下，有两个选项：
 - 如果用户证书是由 AD 证书系统发布的，则证书会包含用户主体名称作为 Subject Alternative Name (SAN)，或者如果最新的更新被应用到 AD，则证书在证书的 SID 扩展中包含用户的 SID。它们都可用于将证书映射到用户。
 - 如果用户证书位于智能卡上，要使用智能卡启用 SSH，SSSD 必须从证书派生公共 SSH 密钥，因此需要完整证书。唯一的解决方案是使用 **ipa idoverrideuser-add** 命令将整个证书添加到 IdM 中 AD 用户的 ID 覆盖中。详情请参阅在[AD 用户条目不包含证书或映射数据时配置证书映射](#)。

AD 域管理员可以使用 **altSecurityIdentities** 属性手动将证书映射到 AD 中的用户。此属性支持六个值，但三个映射被视为不安全。作为 [2022 年 5 月 10 日安全更新](#) 的一部分，在安装之后，所有设备都处于兼容模式，如果证书被弱地映射到用户，则身份验证如预期一样。但是，会记录任何标识任何与全强制模式不兼容的证书的警告消息。从 2023 年 11 月 14 日起，所有设备都将更新为全强制模式，如果证书对强映射标准失败，则身份验证将被拒绝。

例如，当 AD 用户使用证书(PKINIT)请求 IdM Kerberos 票据时，AD 需要在内部将证书映射到用户，并对此使用新的映射规则。但是，在 IdM 中，如果 IdM 用于将证书映射到 IdM 客户端上的用户，则以前的规则将继续工作。

IdM 支持新的映射模板，使 AD 管理员更易于使用新规则，而不用维护这两个模板。IdM 现在支持添加到活动目录的新映射模板，以包括：

- 序列号：LDAPU1: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<SR>{serial_number!hex_ur})
- 主题键 Id: LDAPU1: (altSecurityIdentities=X509:<SKI>{subject_key_id!hex_u})
- 用户 SID：LDAPU1: (objectsid={sid})

如果您不想使用新的 SID 扩展重新发布证书，您可以通过将适当的映射字符串添加到 AD 中的 **altSecurityIdentities** 属性来创建手动映射。

10.6. 为 AD 用户条目包含整个证书的用户配置证书映射

这个用例介绍了在 IdM 部署中启用证书映射所需的步骤（如果使用受 Active Directory(AD)信任的 IdM 部署，用户存储在 AD 中，并且 AD 中的用户条目包含整个证书）。

先决条件

- 用户在 IdM 中没有帐户。
- 用户在 AD 中有一个包含证书的帐户。
- IdM 管理员有权访问 IdM 证书映射规则可以基于的数据。



注意

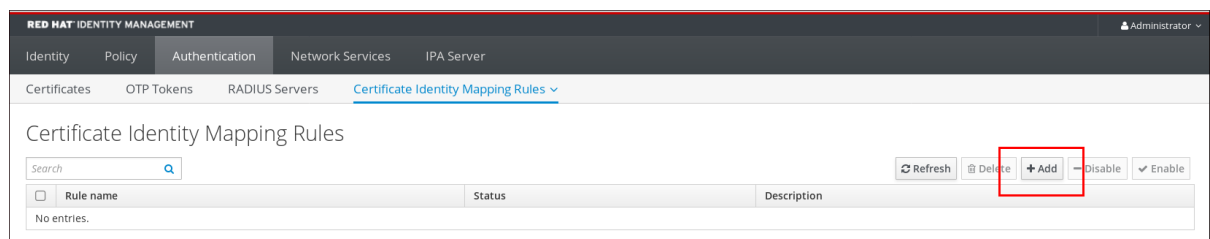
为确保 PKINIT 适用于用户，必须满足以下条件之一：

- 用户条目中的证书包括用户的用户主体名称或 SID 扩展。
- AD 中的用户条目在 **altSecurityIdentities** 属性中有一个合适的条目。

10.6.1. 在 IdM Web UI 中添加证书映射规则

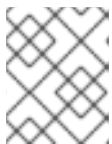
1. 以管理员身份登录 IdM Web UI。
2. 进入到 **Authentication** → **Certificate Identity Mapping Rules** → **Certificate Identity Mapping Rules**。
3. 点 **Add**。

图 10.5. 在 IdM web UI 中添加新证书映射规则



4. 输入规则名称。
5. 输入映射规则。与 AD 中可用的内容相比，会出现 IdM 为身份验证的整个证书：

(userCertificate;binary={cert!bin})



注意

如果使用全证书进行映射，如果续订证书，您必须确保将新证书添加到 AD 用户对象中。

- 输入匹配的规则。例如，只允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书进行验证：

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

图 10.6. 用户使用存储在 AD 中的证书的用户的证书映射规则

- 点 **Add**。
- 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，请在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

10.6.2. 在 IdM CLI 中添加证书映射规则

- 获取管理员凭证：

```
# kinit admin
```

- 输入映射规则以及映射规则所基于的匹配规则。将提供的整个证书与 AD 中可用的证书进行比较，只允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书进行身份验证：

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```



注意

如果使用全证书进行映射，如果续订证书，您必须确保将新证书添加到 AD 用户对象中。

3. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

10.7. 如果将 AD 配置为将用户证书映射到用户帐户，配置证书映射

这个用户故事描述了如果 IdM 部署与活动目录(AD)有信任关系时，在 IdM 中启用证书映射所需的步骤，用户存储在 AD 中，AD 中的用户条目包含证书映射数据。

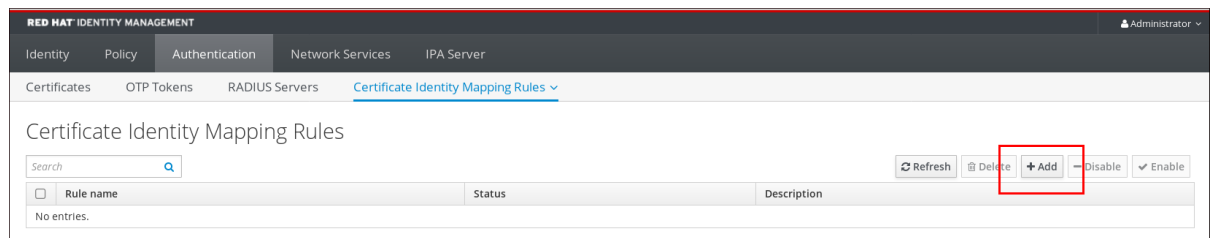
先决条件

- 用户在 IdM 中没有帐户。
- 用户在 AD 中有一个帐户，其中包含 **altSecurityIdentities** 属性（AD 等同于 IdM **certmapdata** 属性）。
- IdM 管理员有权访问 IdM 证书映射规则可以基于的数据。

10.7.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。
2. 进入到 **Authentication** → **Certificate Identity Mapping Rules** → **Certificate Identity Mapping Rules**。
3. 点 **Add**。

图 10.7. 在 IdM web UI 中添加新证书映射规则



4. 输入规则名称。
5. 输入映射规则。例如，要让 AD DC 搜索提供给它们的任何证书中带有 **Issuer** 和 **Subject** 条目，并根据在提供的证书中的这两个条目是否被找到来决定进行验证或不验证。

```
(altSecurityIdentities=X509:<l>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

6. 输入匹配的规则。例如，只允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书在 IdM 中验证用户：

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. 输入域：

```
ad.example.com
```

图 10.8. 如果为映射配置了 AD，则证书映射规则

8. 点 **Add**。
9. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，请在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

10.7.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。例如，若要使 AD 搜索任何呈现的证书中的 **Issuer** 和 **Subject** 条目，并且只允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书：

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule
'<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule
'(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --
domain=ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
-----
```

```
Rule name: ad_configured_for_mapping_rule
Mapping rule: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

10.7.3. 检查 AD 端的证书映射数据

altSecurityIdentities 属性是 IdM 中 **certmapdata** 用户属性等同的 Active Directory(AD)。当将可信 AD 域配置为将用户证书映射到用户帐户时，在 IdM 中配置证书映射时，IdM 系统管理员需要检查是否在 AD 中的用户条目中正确设置了 **altSecurityIdentities** 属性。

先决条件

- 用户帐户必须具有用户管理访问权限。

步骤

- 要检查 AD 是否包含存储在 AD 中的用户的正确信息，请使用 **ldapsearch** 命令。例如，输入以下命令检查 **adserver.ad.example.com** 服务器是否适用以下条件：
 - **altSecurityIdentities** 属性在 **ad_user** 的用户条目中设置。
 - 适用以下条件的匹配规则推断如下：
 - **ad_user** 用于向 AD 进行身份验证的证书由 **ad.example.com** 域的 **AD-ROOT-CA** 发布。
 - 主题为 **<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user**：

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<l>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

10.8. 如果 AD 用户条目不包含证书或映射数据，配置证书映射

这个用例描述了在 IdM 部署中使用 Active Directory(AD)启用证书映射所需的步骤，用户存储在 AD 中，并且 AD 中的用户条目没有包含整个证书也没有证书映射数据。

先决条件

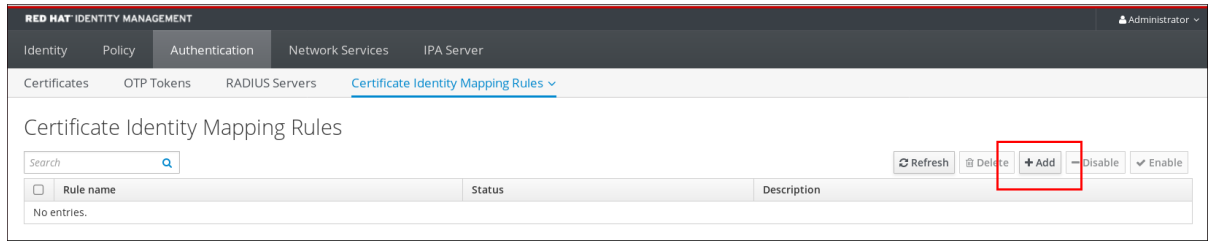
- 用户在 IdM 中没有帐户。
- 用户在 AD 中包含一个帐户，其中没有包含整个证书也没有包括 **altSecurityIdentities** 属性，AD 相当于 IdM **certmapdata** 属性。
- IdM 管理员已完成了以下任务之一：
 - 将整个 AD 用户证书添加到 IdM 中的 **AD 用户 ID 覆盖** 中。
 - 创建一个映射到证书中备用字段的证书映射规则，如 Subject Alternative Name 或用户的 SID。

10.8.1. 在 IdM Web UI 中添加证书映射规则

1. 以管理员身份登录 IdM Web UI。

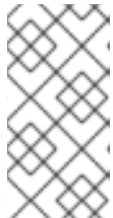
2. 进入到 **Authentication** → **Certificate Identity Mapping Rules** → **Certificate Identity Mapping Rules**。
3. 点 **Add**。

图 10.9. 在 IdM web UI 中添加新证书映射规则



4. 输入规则名称。
5. 输入映射规则。与存储在 IdM 中的 AD 用户条目中的证书相比，会出现 IdM 为进行身份验证的整个证书：

(userCertificate;binary={cert!bin})



注意

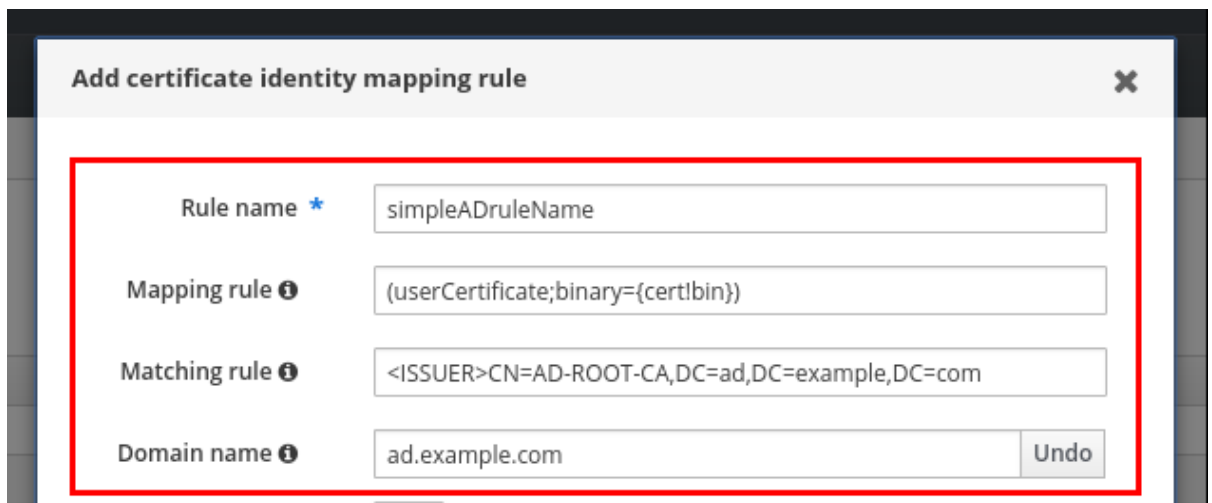
因为证书还包含作为 SAN 的用户主体名称或具有最新更新、证书的 SID 扩展中的用户的 SID,则您也可以使用这些字段将证书映射到用户。例如，如果使用用户的 SID，请将此映射规则替换为 **LDAPU1: (objectsid={sid})**。有关证书映射的更多信息，请参阅 **sss-certmap** 手册页。

6. 输入匹配的规则。例如，只允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书进行验证：

<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com

7. 输入域名。例如，要在 **ad.example.com** 域中搜索用户：

图 10.10. 用户没有证书或映射存储在 AD 中的数据的数据的证书映射规则



8. 点 **Add**。

9. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制新创建的规则立即载入，在 CLI 中重启 SSSD：

```
# systemctl restart sssd
```

10.8.2. 在 IdM CLI 中添加证书映射规则

1. 获取管理员凭证：

```
# kinit admin
```

2. 输入映射规则以及映射规则所基于的匹配规则。将提供的整个证书与 IdM 中的 AD 用户条目的用户 ID 覆盖中所存储的证书进行比较，仅允许 **AD.EXAMPLE.COM** 域的 **AD-ROOT-CA** 签发的证书进行验证：

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```



注意

因为证书还包含作为 SAN 的用户主体名称或具有最新更新、证书的 SID 扩展中的用户的 SID，则您也可以使用这些字段将证书映射到用户。例如，如果使用用户的 SID，请将此映射规则替换为 **LDAPU1: (objectsid={sid})**。有关证书映射的更多信息，请参阅 **sss-certmap** 手册页。

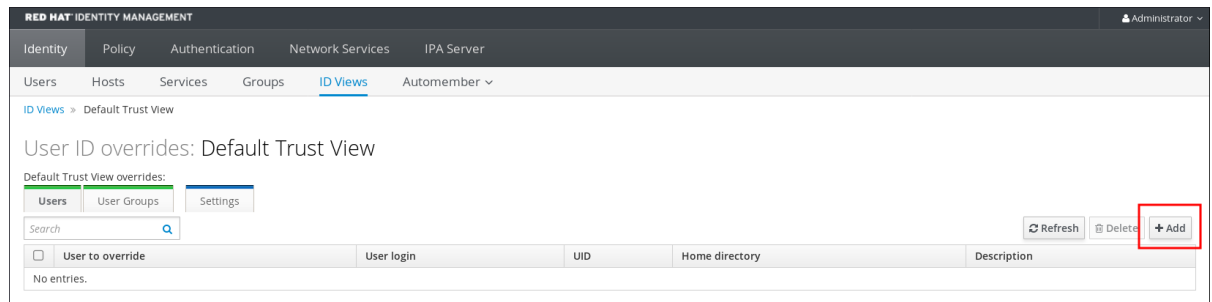
3. 系统安全服务守护进程(SSSD)会定期重新读取证书映射规则。要强制立即载入新创建的规则，重启 SSSD：

```
# systemctl restart sssd
```

10.8.3. 在 IdM Web UI 中添加证书到 AD 用户的 ID 覆盖中

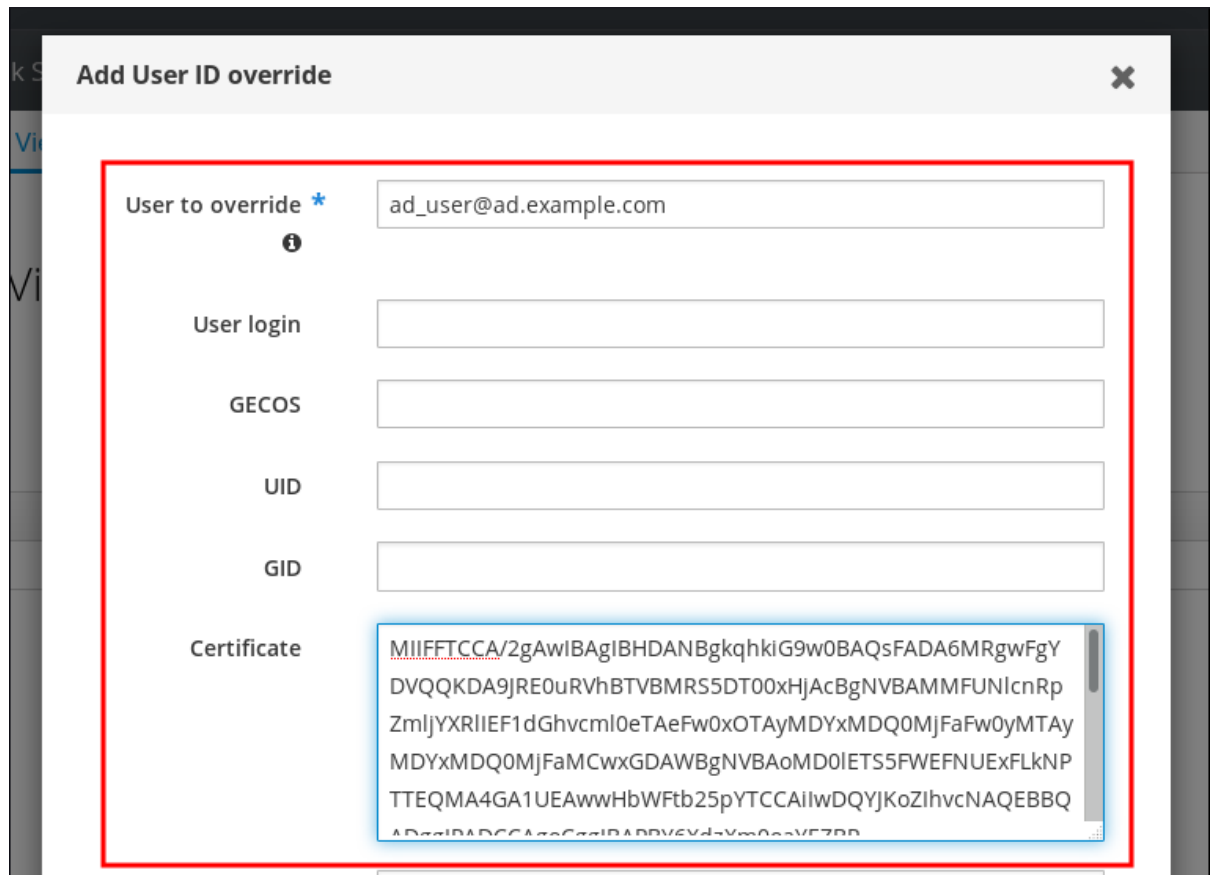
1. 进入 **Identity** → **ID Views** → **Default Trust View**。
2. 点 **Add**。

图 10.11. 在 IdM Web UI 中添加新用户 ID 覆盖



3. 在 **User to override** 字段中，输入 `ad_user@ad.example.com`。
4. 将 `ad_user` 的证书复制并粘贴到 **Certificate** 字段中。

图 10.12. 为 AD 用户配置用户 ID 覆盖



5. 点 **Add**。

验证步骤

验证用户和证书是否已链接：

1. 使用 `sss_cache` 工具使 SSSD 缓存中的 `ad_user@ad.example.com` 记录失效，并强制重新载入 `ad_user@ad.example.com` 信息：

```
# sss_cache -u ad_user@ad.example.com
```

2. 使用包含 AD 用户证书的文件名称运行 `ipa certmap-match` 命令：

```
# ipa certmap-match ad_user_cert.pem
```

```

-----
1 user matched
-----
Domain: AD.EXAMPLE.COM
User logins: ad_user@ad.example.com
-----
Number of entries returned 1
-----

```

输出确认了您已将证书映射数据添加到 `ad_user@ad.example.com`，并且 [如果 AD 用户条目不包含证书或映射数据](#) 中定义的相应的映射规则存在。这意味着，您可以使用与定义的证书映射数据匹配的证书，以 `ad_user@ad.example.com` 进行身份验证。

其他资源

[为活动目录用户使用 ID 视图](#)

10.8.4. 在 IdM CLI 中添加证书到 AD 用户的 ID 覆盖中

1. 获取管理员凭证：

```
# kinit admin
```

2. 将证书 blob 保存在名为 **CERT** 的新变量中：

```
# CERT=$(openssl x509 -in /path/to/certificate -outform der|base64 -w0)
```

3. 使用 `ipa idoverrideuser-add-cert` 命令将 `ad_user@ad.example.com` 的证书添加到用户帐户中：

```
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

验证步骤

验证用户和证书是否已链接：

1. 使用 `sss_cache` 工具使 SSSD 缓存中的 `ad_user@ad.example.com` 记录失效，并强制重新载入 `ad_user@ad.example.com` 信息：

```
# sss_cache -u ad_user@ad.example.com
```

2. 使用包含 AD 用户证书的文件名称运行 `ipa certmap-match` 命令：

```

# ipa certmap-match ad_user_cert.pem
-----
1 user matched
-----
Domain: AD.EXAMPLE.COM
User logins: ad_user@ad.example.com
-----
Number of entries returned 1
-----

```

输出确认了您已将证书映射数据添加到 `ad_user@ad.example.com`，并且 [如果 AD 用户条目不包含证书或映射数据](#) 中定义的相应的映射规则存在。这意味着，您可以使用与定义的证书映射数据匹配的证书，以 `ad_user@ad.example.com` 进行身份验证。

其他资源

[为活动目录用户使用 ID 视图](#)

10.9. 将多个身份映射规则合并到一个中

要将多个身份映射规则合并成一个组合规则，在单个映射规则前面使用 `|` (or) 字符，并使用 `()` 分隔它们，例如：

证书映射过滤示例 1

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \
--maprule='((ipacertmapdata=X509:<l>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<l>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
--matchrule='<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' \
--domain=ad.example.com
```

在上例中，`- maprule` 选项中的过滤器定义包括以下标准：

- `ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}` 是一个过滤器，它将一个智能卡证书中的 `subject` 和 `issuer` 连接到一个 IdM 用户账户中的 `ipacertmapdata` 属性的值，如 [Adding a certificate mapping rule in IdM](#) 部分所述
- `altSecurityIdentities=X509:<l>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}` 是一个过滤器，它将一个智能卡证书中的 `subject` 和 `issuer` 连接到一个 AD 用户账户中的 `altSecurityIdentities` 属性的值，如 [Adding a certificate mapping rule if the trusted AD domain is configured to map user certificates](#) 所述
- 添加 `--domain=ad.example.com` 选项意味着，映射到指定证书的用户不仅在本地 `idm.example.com` 域中进行搜索，也在 `ad.example.com` 域中搜索

在 `--maprule` 选项中的过滤器定义接受逻辑操作符 `|` (or)，以便您可以指定多个条件。在这种情况下，该规则会映射至少满足其中一个条件的所有用户帐户。

证书映射过滤示例 2

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
--maprule='((userCertificate;binary={cert!bin})(ipacertmapdata=X509:<l>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<l>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
--matchrule='<ISSUER>CN=Certificate Authority,O=REALM.EXAMPLE.COM' \
--domain=idm.example.com --domain=ad.example.com
```

在上例中，`- maprule` 选项中的过滤器定义包括以下标准：

- `userCertificate;binary={cert!bin}` 是一个返回包括整个证书的用户条目的过滤器。对于 AD 用户，创建这种类型的过滤器在 [如果 AD 用户条目不包含证书或映射数据，则添加一个证书映射规则](#) 中进行了描述。
- `ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}` 是一个过滤器，它将一个智能卡证书中的 `subject` 和 `issuer` 连接到一个 IdM 用户账户中的 `ipacertmapdata` 属性的值，如 [Adding a certificate mapping rule in IdM](#) 部分所述。

- `altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}` 是一个过滤器，它将一个智能卡证书中的 subject 和 issuer 连接到一个 AD 用户账户中的 `altSecurityIdentities` 属性的值，如 [Adding a certificate mapping rule if the trusted AD domain is configured to map user certificates](#) 所述。

在 `--maprule` 选项中的过滤器定义接受逻辑操作符 `|` (or)，以便您可以指定多个条件。在这种情况下，该规则会映射至少满足其中一个条件的所有用户帐户。

10.10. 其他资源

- 请参阅 `sss-certmap(5)` 手册页。

第 11 章 使用存储在 IDM 客户端中的证书配置身份验证

通过配置身份管理(IdM)，IdM 系统管理员允许使用一个 CA 向用户发布的证书，通过 IdM Web UI 和命令行界面(CLI)进行身份验证。证书存储在 IdM 客户端的桌面上。

Web 浏览器可以在不属于 IdM 域的系统上运行。

在使用证书配置身份验证时请注意以下几点：

- 如果您要使用证书进行身份验证的用户已有证书，则您可以跳过 [请求新的用户证书并将其导出到客户端](#)；
- 如果用户的证书已由 IdM CA 发布了，则您可以跳过 [确保证书和用户链接在一起](#)。



注意

只有身份管理用户可以使用证书登录 Web UI。Active Directory 用户可以使用其用户名和密码登录。

11.1. 在 WEB UI 中为证书验证配置身份管理服务

作为 Identity Management(IdM)管理员，您可以允许用户使用证书来向 IdM 环境进行身份验证。

步骤

作为身份管理管理员：

1. 在身份管理服务器上，获取管理员特权并创建 shell 脚本来配置服务器。
 - a. 运行 **ipa-advise config-server-for-smart-card-auth** 命令，并将其输出保存到一个文件中，例如 **server_certificate_script.sh**：

```
# kinit admin
# ipa-advise config-server-for-smart-card-auth > server_certificate_script.sh
```

- b. 使用 **chmod** 实用程序向该文件添加执行权限：

```
# chmod +x server_certificate_script.sh
```

2. 在 Identity Management 域中的所有服务器上，运行 **server_certificate_script.sh** 脚本
 - a. 如果 IdM CA 是唯一一个您允许向用户签发证书用于验证的 CA，使用 IdM Certificate Authority 证书 **/etc/ipa/ca.crt** 的路径：

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

- b. 如果不同的外部 CA 签署了您要启用证书身份验证的用户的证书，则导致相关 CA 证书的路径作为输入：

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```

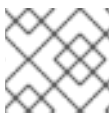


注意

如果您想为整个拓扑中启用的用户认证，请不要忘记在添加到系统的每个新副本上运行脚本。

11.2. 请求新的用户证书并将其导出到客户端

作为 Identity Management (IdM) 管理员，您可以为 IdM 环境中的用户创建证书，并将其导出到您要为用启用证书验证的 IdM 客户端中。



注意

如果要使用证书进行身份验证的用户已有证书，则不需要按照以下流程操作。

步骤

1. (可选) 创建一个新目录，如 `~/certdb/`，并使其成为临时证书数据库。当被要求时，创建一个 NSS 证书 DB 密码来加密在后续步骤中生成的证书的密钥：

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. 创建证书签名请求 (CSR) 并将输出重定向到文件中。例如，要为 **IDM.EXAMPLE.COM** 域中的 **idm_user** 用户创建一个名称为 **certificate_request.csr** 的 4096 位 CSR，请将证书私钥的昵称设为 **idm_user** 以便于查找，并将主题设为 **CN=idm_user,O=IDM.EXAMPLE.COM**：

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s "CN=idm_user,O=IDM.EXAMPLE.COM"
> certificate_request.csr
```

3. 提示时，输入您在使用 **certutil** 创建临时数据库时输入相同的密码。然后继续随机键入直到被告知停止：

Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4. 将证书请求文件提交到服务器。指定要与新签发的证书关联的 Kerberos 主体，输出文件来存储证书，以及证书配置集（可选）。例如，为 **idm_user@IDM.EXAMPLE.COM** 获取 **IECUserRoles** 配置集（添加了用户角色扩展的配置集）的证书，把它保存在 **~/idm_user.pem** 文件中：

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --certificate-out=~/idm_user.pem
```

- 将证书添加到 NSS 数据库。使用 **-n** 选项设置您在之前创建 CSR 时使用的相同的 nickname，以便证书与 NSS 数据库中的私钥匹配。**-t** 选项设置信任级别。详情请查看 `certutil(1)` man page。**-i** 选项指定输入证书文件。例如，要将带有在 `~/idm_user.pem` 文件中定义的 **idm_user** 别名的证书添加到 `~/certdb/` 数据库的 NSS 数据库：

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

- 验证 NSS 数据库中的密钥没有显示 (**orphan**) 作为其 nickname。例如，验证存储在 `~/certdb/` 数据库中的证书是否为孤立：

```
# certutil -K -d ~/certdb/
< 0> rsa 5ad14d41463b87a095b1896cf0068ccc467df395 NSS Certificate
DB: idm_user
```

- 使用 **pk12util** 命令将证书从 NSS 数据库导出到 PKCS12 格式。例如，要将来自 `/root/certdb` NSS 数据库的带有 **idm_user** 别名的证书导出到 `~/idm_user.p12` 文件中：

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

- 将证书传递给您要启用 **idm_user** 的证书身份验证的主机：

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

- 因为安全原因，在证书传输到的主机上，将存储 `.pkcs12` 文件的目录的访问权限设置为 `'other'` 组不能访问它：

```
# chmod o-rwx /home/idm_user/
```

- 为安全起见，请从服务器中删除临时 NSS 数据库和 `.pkcs12` 文件：

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

11.3. 确保证书和用户已链接在一起



注意

如果用户的证书已由 IdM CA 发布，则不需要按照此流程操作。

要使证书身份验证正常工作，您需要确保证书链接到用来进行身份管理(IdM)的用户。

- 如果证书是由不属于您的身份管理环境的证书颁发机构提供的，请根据 [将用户帐户链接到证书](#) 中描述的流程链接用户和证书。

- 如果证书由 Identity Management CA 提供，该证书会自动添加到用户条目中，且您不必将证书链接到用户帐户。有关在 IdM 中创建新证书的详情，请参阅 [请求新的用户证书并将其导出到客户端](#)。

11.4. 配置浏览器以启用证书身份验证

要在使用 WebUI 登录到 Identity Management(IdM)时，可以使用证书进行身份验证，您需要将该用户和相关证书颁发机构(CA)证书导入到 Mozilla Firefox 或 Google Chrome 浏览器。运行浏览器的主机本身不需要是 IdM 域的一部分。

IdM 支持以下浏览器连接到 Web UI :

- Mozilla Firefox 38 及更新的版本
- Google Chrome 46 及更新的版本

以下流程演示了如何配置 Mozilla Firefox 57.0.1 浏览器。

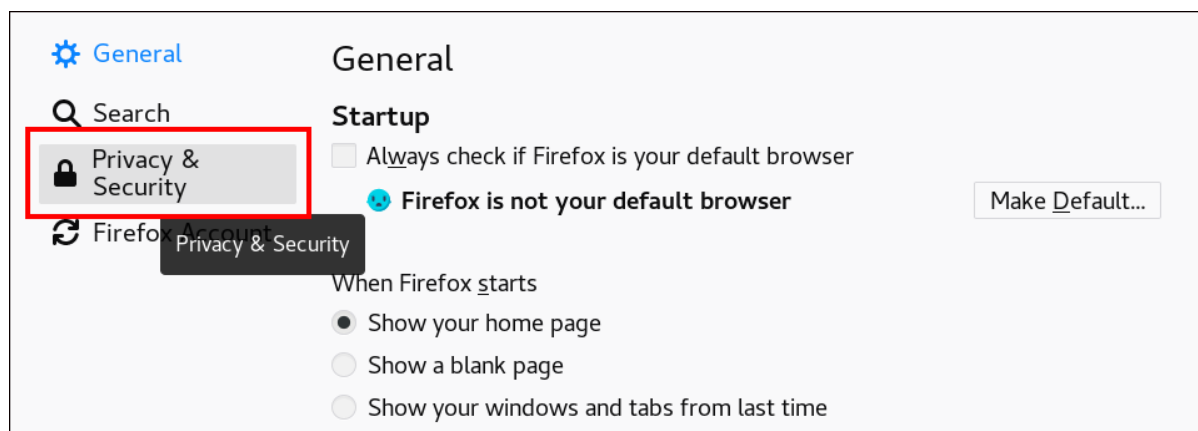
先决条件

- 您有要导入到浏览器的[用户证书](#)（采用 PKCS#12 格式）。

步骤

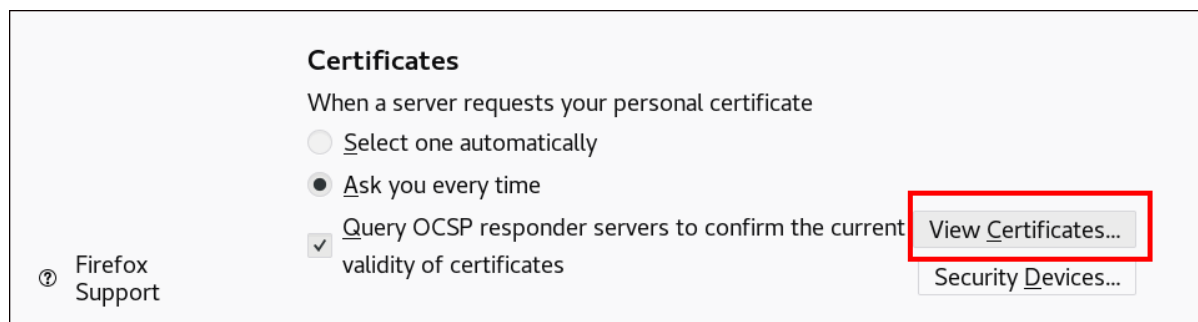
1. 打开 Firefox，进入 **Preferences** → **Privacy & Security**。

图 11.1. 首选项中的隐私和安全部分



2. 点查看证书。

图 11.2. 查看隐私和安全中的证书



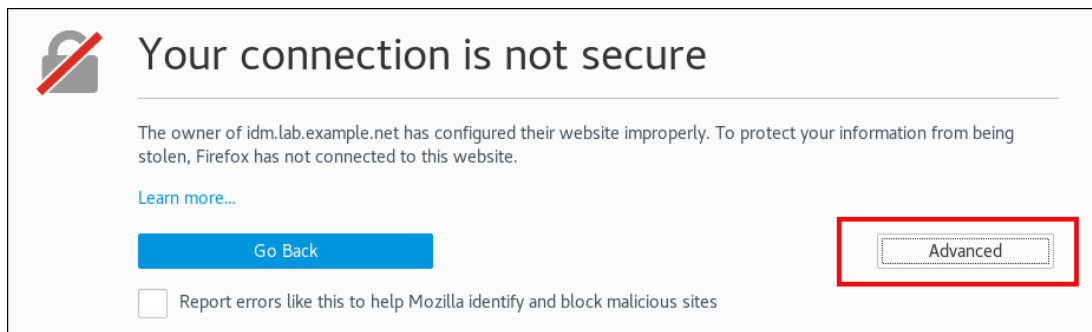
3. 在 **Your Certificates** 选项卡中，点 **Import**。查找并打开用户证书（PKCS12 格式），然后点 **OK** 和 **OK**。

4. 确保 Firefox 将身份管理证书颁发机构识别为可信机构：

a. 本地保存 IdM CA 证书：

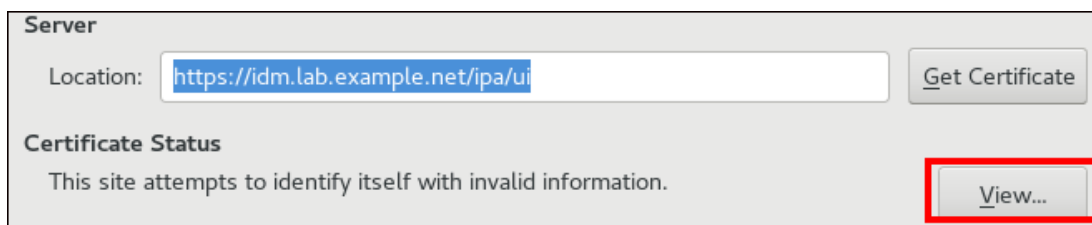
- 通过在 Firefox 地址栏中写入 IdM 服务器的名称，以进入到 IdM web UI。在 Insecure Connection warning 页面上，点 **Advanced**。

图 11.3. 不安全连接



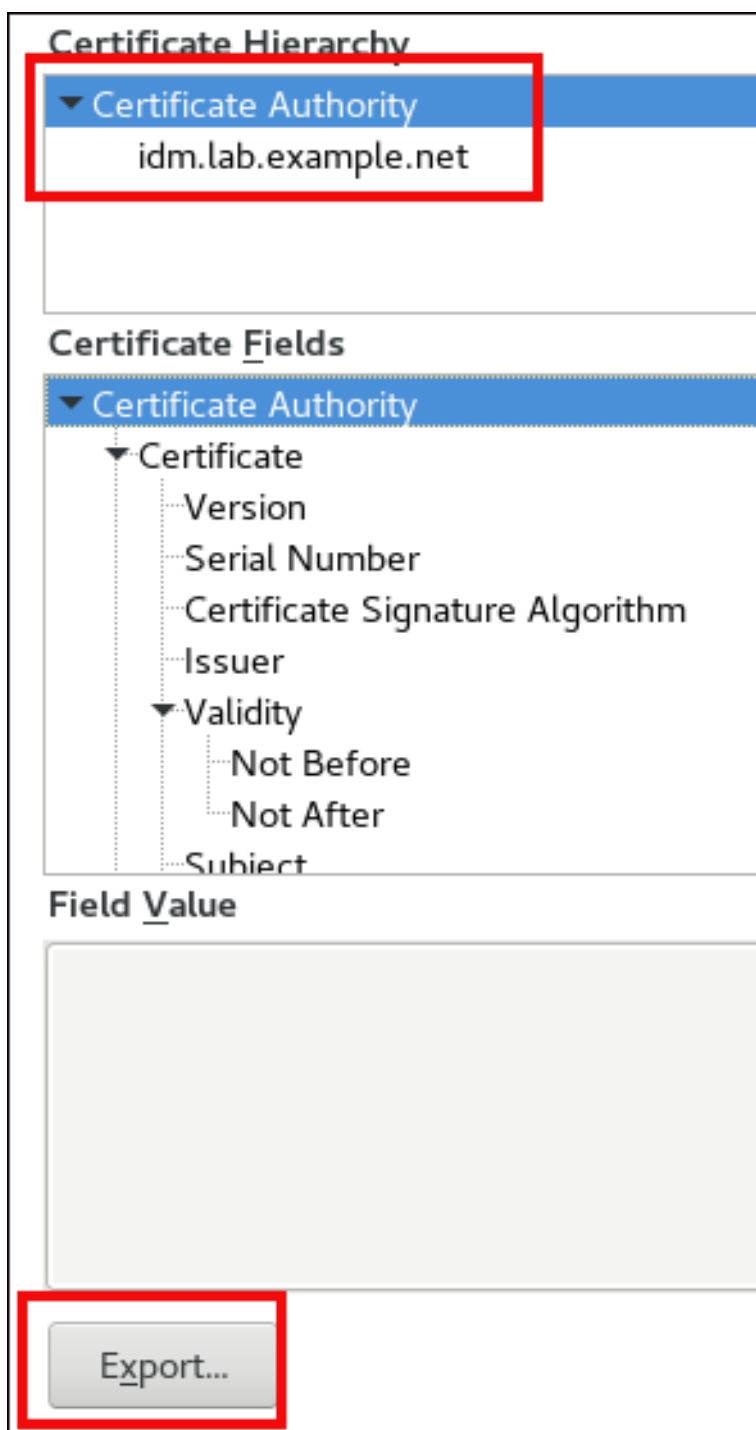
- 添加例外.点查看。

图 11.4. 查看证书详情



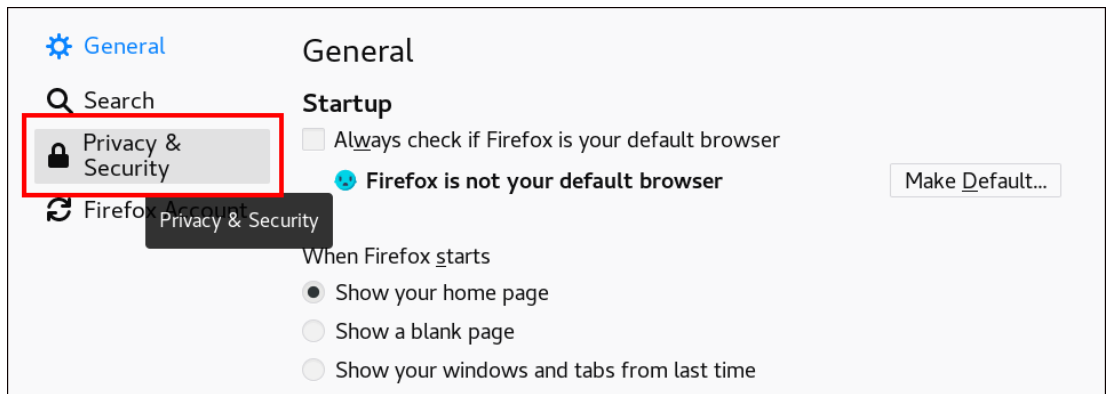
- 在 **Details** 选项卡中，突出显示 **Certificate Authority** 字段。

图 11.5. 导出 CA 证书



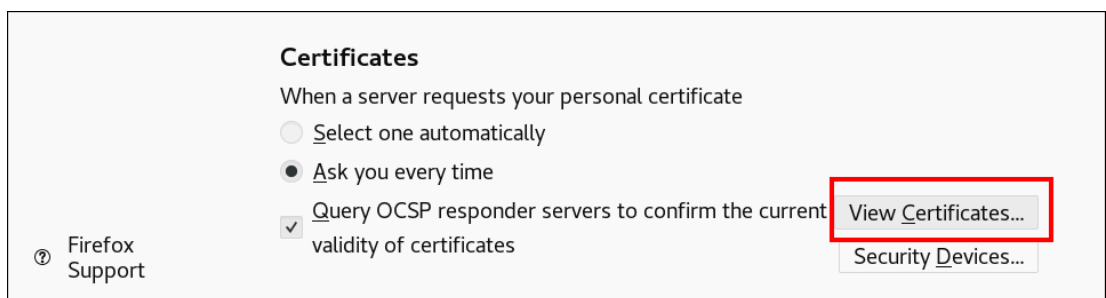
- 单击 **Export**。保存 CA 证书，例如 **CertificateAuthority.crt** 文件，然后点 **Close**，和 **Cancel**。
- b. 将 IdM CA 证书导入到 Firefox 作为可信证书颁发机构证书：
- 打开 Firefox，导航到首选项并单击 **隐私和安全**。

图 11.6. 首选项中的隐私和安全部分



- 点查看证书。

图 11.7. 查看隐私和安全中的证书



- 在 **Authorities** 选项卡中，点 **Import**。在 **CertificateAuthority.crt** 文件中找到并打开在上一步中保存的 CA 证书。信任证书来识别网站，然后点 **OK** 和 **OK**。
5. 继续以身份管理用户的身份使用证书向身份管理 Web UI 进行身份验证。

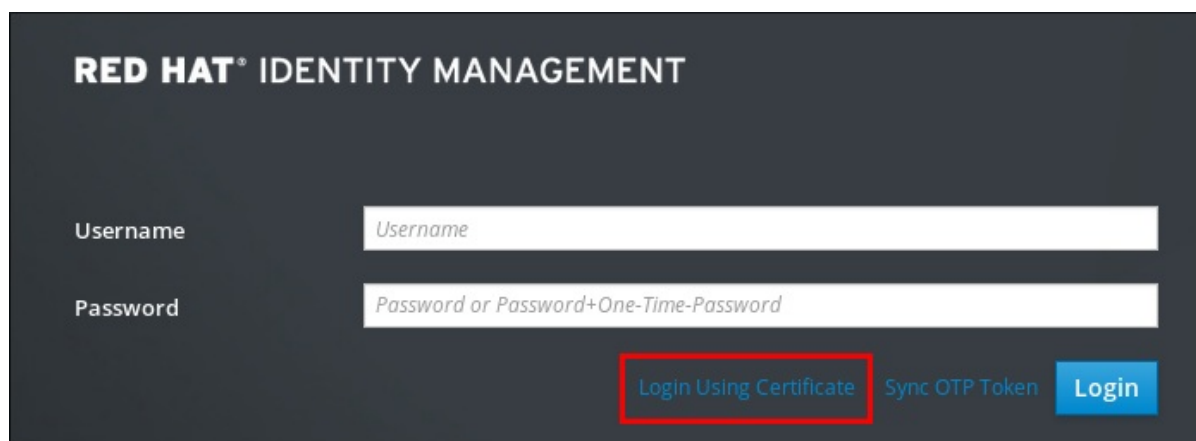
11.5. 以身份管理用户的身份使用证书向身份管理 WEB UI 进行身份验证

按照以下流程，使用存储在身份管理客户端桌面上的证书作为用户向身份管理(IdM) Web UI 进行身份验证。

步骤

1. 在浏览器中，访问 Identity Management web UI（例如 <https://server.idm.example.com/ipa/ui>）。
2. 点使用证书登录。

图 11.8. 在身份管理 Web UI 中使用证书登录



3. 应该已经选择该用户的证书。取消选择 **Remember this decision**，然后点 **OK**。

您现在以与证书对应的用户身份进行身份验证。

其他资源

- 请参阅 [为智能卡验证配置身份管理](#)。

11.6. 配置 IDM 客户端，以启用使用证书对 CLI 进行身份验证

要使证书身份验证可以在 IdM 客户端的命令行界面(CLI)中用于 IdM 用户，请将 IdM 用户的证书和私钥导入到 IdM 客户端。有关创建和传输用户证书的详情，请参阅 [请求新的用户证书并将其导出到客户端](#)。

步骤

- 登录到 IdM 客户端，并具有包含用户证书和密钥的 .p12 文件。要获取并缓存 Kerberos ticket 授予 ticket(TGT)，请使用带用户主体的 **-X** 选项和 **X509_username:/path/to/file.p12** 属性的 **kinit** 命令，指定在哪里查找用户的 X509 身份信息。例如，要获取 **idm_user** 的 TGT，使用保存在 **~/idm_user.p12** 文件中的用户身份信息：

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



注意

命令还支持 .pem 文件格式：**kinit -X X509_username='FILE:/path/to/cert.pem,/path/to/key' user_principal**

第 12 章 使用 IDM CA 续订服务器

12.1. IDM CA 续订服务器说明

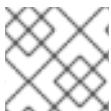
在使用嵌入式证书颁发机构 (CA) 的 Identity Management (IdM) 部署中，CA 续订服务器维护并更新 IdM 系统证书。它确保了强大的 IdM 部署。

IdM 系统证书包括：

- **IdM CA 证书**
- **OCSP 签名证书**
- **IdM CA 子系统证书**
- **IdM CA 审计签名证书**
- **IdM 续订代理 (RA) 证书**
- **KRA 传输和存储证书**

哪个字符化系统证书是其密钥由所有 CA 副本共享。与之相反，IdM 服务证书（如 **LDAP**、**HTTP** 和 **PKINIT** 证书）在不同 IdM CA 服务器上有不同的密钥对和主题名称。

在 IdM 拓扑中，默认情况下，第一个 IdM CA 服务器是 CA 续订服务器。



注意

在上游文档中，IdM CA 名为 **Dogtag**。

CA 续订服务器的角色

IdM CA、**IdM CA 子系统** 和 **IdM RA** 证书对于 IdM 部署至关重要。每个证书都存储在 `/etc/pki/pki-tomcat/` 目录中的 NSS 数据库中，并作为 LDAP 数据库条目。LDAP 中存储的证书必须与存储在 NSS 数据库中的证书匹配。如果不匹配，则 IdM 框架和 IdM CA 之间以及 IdM CA 和 LDAP 之间的身份验证失败。

所有 IdM CA 副本都有每个系统证书的跟踪请求。如果带有集成 CA 的 IdM 部署不包含 CA 续订服务器，每个 IdM CA 服务器会单独请求续订系统证书。这会导致具有不同系统证书的不同 CA 副本，发生身份验证失败。

将一个 CA 副本作为续订服务器，可以在需要时精确续订系统证书，从而防止身份验证失败。

CA 副本上的 certmonger 服务角色

在所有 IdM CA 副本中运行的 **certmonger** 服务都使用 **dogtag-ipa-ca-renew-agent** 续订帮助程序来跟踪 IdM 系统证书。续订帮助程序读取 CA 续订服务器配置。在不是 CA 续订服务器的每个 CA 副本中，续订帮助程序从 **ca_renewal** LDAP 条目中检索最新的系统证书。由于无法决定 **certmonger** 续订尝试发生的准确时间，**dogtag-ipa-ca-renew-agent** 帮助程序有时会在 CA 续订服务器实际续订证书前尝试更新系统证书。如果发生这种情况，则即将过期的证书返回到 CA 副本上的 **certmonger** 服务。**certmonger** 服务因为意识到它是已经存储在其数据库中的相同证书，会重复尝试续订证书（每个尝试间有一个延迟），直到它可以从 CA 续订服务器检索到更新的证书。

IdM CA 续订服务器正常工作

带有内嵌 CA 的 IdM 部署是一个 IdM 部署，该部署使用 IdM CA 安装 - 或者在以后安装 IdM CA 服务器。具有嵌入式 CA 的 IdM 部署必须完全配置一个 CA 副本，作为续订服务器。续订服务器必须在线且完全正常工作，且必须与其他服务器正确复制。

如果使用 `ipa server-del`、`ipa-replica-manage del`、`ipa-csreplica-manage del` 或 `ipa-server-install -uninstall` 命令删除了当前的 CA 续订服务器，其他一个 CA 副本会自动分配为 CA 续订服务器。此策略可确保续订服务器配置保持有效。

此策略不包括以下情况：

- **离线续订服务器**

如果续订服务器在延长期间内离线，则可能会错过续订窗口。在这种情况下，所有非续订的 CA 服务器都会重新安装当前系统证书，直到证书过期为止。当发生这种情况时，IdM 部署会中断，因为即使一个过期的证书可能会导致其他证书的续订失败。

- **复制问题**

如果续订服务器和其它 CA 副本之间存在复制问题，则续订可能会成功，但其他 CA 副本可能无法在过期之前检索更新的证书。

要防止这种情况，请确保您的复制协议可以正常工作。详情请参阅 RHEL 7 *Linux 域身份、身份验证和策略指南* 中的 [常规](#) 或 [特定](#) 复制故障排除指南。

12.2. 更改和重置 IDM CA 续订服务器

当证书颁发机构(CA)续订服务器被删除时，身份管理(IdM)会自动从 IdM CA 服务器列表中选择新的 CA 续订服务器。系统管理员无法影响选择。

要可以选择新的 IdM CA 续订服务器，系统管理员必须手动执行替换。在开始停用当前续订服务器前，选择新的 CA 续订服务器。

如果当前的 CA 续订服务器配置无效，请重置 IdM CA 续订服务器。

完成这个步骤来更改或重置 CA 续订服务器。

先决条件

- 您有 IdM 管理员凭证。

步骤

1. 获取 IdM 管理员凭证：

```
~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. 另外，要找出部署中的 IdM 服务器具有满足新 CA 续订服务器所必需的 CA 角色：

```
~]$ ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server.idm.example.com
Role name: CA server
Role status: enabled
```

```

Server name: replica.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----

```

部署中有两个 CA 服务器。

- 另外，要找出哪个 CA 服务器是当前的 CA 续订服务器，请输入：

```

~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: server.idm.example.com

```

当前的续订服务器是 **server.idm.example.com**。

- 要更改续订服务器配置，请使用 **ipa config-mod** 程序和 **--ca-renewal-master-server** 选项：

```

~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com | grep 'CA renewal'
IPA CA renewal master: replica.idm.example.com

```

重要

您还可以使用以下内容切换到新的 CA 续订服务器：

- **ipa-cacert-manage --renew** 命令。这个命令同时更新 CA 证书，并使您在其上执行新 CA 续订服务器的 CA 服务器。
- **ipa-cert-fix** 命令。当过期的证书会导致失败时，这个命令会恢复部署。它还使您在其上执行新 CA 续订服务器的 CA 服务器。详情请查看[当 IdM 离线时续订过期的系统证书](#)。

第 13 章 管理外部签名的 CA 证书

身份管理(IdM)提供不同类型的证书颁发机构(CA)配置。您可以选择安装带有集成 CA 或带有外部 CA 的 IdM。您必须指定在安装过程中使用的 CA 类型。但是，安装后，您可以从外部签名的 CA 移到自签名 CA，反之亦然。另外，当自动续订自签名 CA 时，您必须确保续订外部签名的 CA 证书。请参考管理外部签名的 CA 证书所需的相关部分。

- 安装带有外部签名 CA 的 IdM :
 - 安装带有集成 DNS 和外部 CA 作为根 CA 的 IdM 服务器。
 - 安装没有集成 DNS 和外部 CA 作为根 CA 的 IdM 服务器。
- 从外部签名的 CA 切换到自签名 CA。
- 从自签名 CA 切换到外部签名的 CA。
- 续订外部签名的 CA 证书。

13.1. 在 IDM 中从外部签名的 CA 切换到自签名 CA

完成这个步骤，从外部签名切换到 Identity Management(IdM)证书颁发机构(CA)的自签名证书。使用自签名 CA，自动管理 CA 证书的续订：系统管理员不需要向外部授权提交证书签名请求(CSR)。

从外部签名的 CA 切换到自签名 CA，只替换 CA 证书。上一个 CA 签名的证书仍然有效，仍在使用中。例如，即使您移至自签名 CA，**LDAP** 证书的证书链不会改变：

```
external_CA certificate > IdM CA certificate > LDAP certificate
```

先决条件

- 您有访问 IdM CA 续订服务器和所有 IdM 客户端及服务器的 **root** 权限。

步骤

1. 在 IdM CA 续订服务器中，将 CA 证书更新为自签名：

```
# ipa-cacert-manage renew --self-signed
Renewing CA certificate, please wait
CA certificate successfully renewed
The ipa-cacert-manage command was successful
```

2. 以 **root** 身份 **SSH** 到所有剩余的 IdM 服务器和客户端。例如：

```
# ssh root@idmclient01.idm.example.com
```

3. 在 IdM 客户端上，使用来自服务器的证书更新本地 IdM 证书数据库：

```
[idmclient01 ~]# ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

4. (可选) 检查您的更新是否成功，并且新的 CA 证书已添加到 `/etc/ipa/ca.crt` 文件中：

```
[idmclient01 ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -
print_certs -text -noout
[...]
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 39 (0x27)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
    Validity
      Not Before: Jul  1 16:32:45 2019 GMT
      Not After : Jul  1 16:32:45 2039 GMT
    Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
  [...]

```

输出显示，更新已成功，因为使用较旧的 CA 证书列出新的 CA 证书。

13.2. 在 IDM 中从自签名 CA 切换到外部签名的 CA

在 IdM 中，您可以从自签名 CA 切换到外部签名的 CA。在 IdM 中切换到外部签名的 CA 后，您的 IdM CA 服务器变为外部 CA 的子 CA。另外，不会自动管理 CA 证书的续订，系统管理员必须向外部颁发机构提交证书签名请求(CSR)。

要切换到外部签名的 CA，CSR 必须由外部 CA 签名。按照 [使用外部 CA 续订 IdM CA 续订服务器证书](#) 中的步骤操作，来切换到 IdM 中的自签名 CA。

13.3. 使用外部 CA 续订 IDM CA 续订服务器证书

按照以下流程，使用外部 CA 续订身份管理(IdM)证书颁发机构(CA)证书，来签名证书签名请求(CSR)。在这个配置中，您的 IdM CA 服务器是外部 CA 的子 CA。外部 CA 可以，但不必是 Active Directory 证书服务器(AD CS)。

如果外部证书颁发机构是 AD CS，您可以在 CSR 中指定 IdM CA 证书的模板。证书模板定义了 CA 在收到证书请求时使用的策略和规则。AD 中的证书模板与 IdM 中的证书配置集对应。

您可以根据其对象标识符(OID)定义特定的 AD CS 模板。OID 是各种颁发机构发布的唯一数字值，用于识别数据元素、语法以及分布式应用程序的其他部分。

另外，您还可以通过其名称来定义特定的 AD CS 模板。例如，由 IdM CA 提交的 CSR 中使用的默认配置集的名称是 **subCA**。

要通过在 CSR 中指定 OID 或名称来定义配置集，请使用 **external-ca-profile** 选项。详情请查看 **ipa-cacert-manage** man page。

除了使用可用的证书模板外，您还可以在 AD CS 中创建自定义证书模板，并在 CSR 中使用它。

先决条件

- 有到 IdM CA 续订服务器的 root 访问权限。

步骤

完成这个步骤，使用外部签名续订 IdM CA 的证书，无论当前的 CA 证书是自签名还是外部签名。

1. 创建要提交到外部 CA 的 CSR :

- 如果外部 CA 是一个 AD CS, 请使用 **--external-ca-type=ms-cs** 选项。如果您希望使用默认 **subCA** 模板以外的一个不同的模板, 使用 **--external-ca-profile** 选项指定它 :

```
~]# ipa-cacert-manage renew --external-ca --external-ca-type=ms-cs [--external-ca-profile=PROFILE]
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

- 如果外部 CA 不是 AD CS :

```
~]# ipa-cacert-manage renew --external-ca
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

输出显示已创建 CSR, 并存储在 **/var/lib/ipa/ca.csr** 文件中。

2. 将位于 **/var/lib/ipa/ca.csr** 中的 CSR 提交到外部 CA。这个过程根据要用作外部 CA 的服务的不同而有所不同。

3. 在 base 64 编码的 blob 中检索签发的证书和 CA 证书链, 即 :

- 如果外部 CA 不是一个 AD CS, 则为 PEM 文件。
- 如果外部 CA 是一个 AD CS, 则为 Base_64 证书。
各个证书服务的进程不同。通常, 网页或通知电子邮件中的下载链接允许管理员下载所有需要的证书。

如果外部 CA 是 AD CS, 且您已提通过 Microsoft Windows 认证认证机构管理窗口提交了带有已知模板的 CSR, 则 AD CS 会立即发出证书, Save Certificate 对话框会出现在 AD CS Web 界面中, 要求保存签发的证书。

4. 再次运行 **ipa-cacert-manage renew** 命令, 添加提供完整证书链所需的所有 CA 证书文件。根据需要指定多个文件, 多次使用 **--external-cert-file** 选项 :

```
~]# ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate_1 --external-cert-file=/path/to/external_ca_certificate_2
```

5. 在所有 IdM 服务器和客户端中, 使用来自服务器的证书更新本地 IdM 证书数据库 :

```
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

6. (可选) 检查您的更新是否成功, 并且新的 CA 证书已添加到 `/etc/ipa/ca.crt` 文件中 :

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs -  
text -noout  
[...]  
Certificate:  
Data:  
  Version: 3 (0x2)  
  Serial Number: 39 (0x27)  
  Signature Algorithm: sha256WithRSAEncryption  
  Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority  
  Validity  
    Not Before: Jul  1 16:32:45 2019 GMT  
    Not After : Jul  1 16:32:45 2039 GMT  
  Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority  
[...]
```

输出显示, 更新已成功, 因为使用较旧的 CA 证书列出新的 CA 证书。

第 14 章 当 IDM 离线时续订已过期的系统证书

如果系统证书已过期，Identity Management(IdM)无法启动。IdM 支持使用 **ipa-cert-fix** 工具更新系统证书。

- 通过在主机上输入 **ipactl start --ignore-service-failures** 命令来确保 LDAP 服务正在运行。

14.1. 在 CA 续订服务器中续订已过期的系统证书

按照以下流程对过期的 IdM 证书应用 **ipa-cert-fix** 工具。



重要

如果您在不是 CA 续订服务器的 CA（证书授权机构）主机上运行 **ipa-cert-fix** 工具，且它续订共享的证书，则该主机会自动变为域中的新 CA 续订服务器。域中必须始终只有一个 CA 续订服务器，以避免不一致。

先决条件

- 使用管理权限登录到服务器

步骤

1. （可选）备份系统。这强烈推荐，因为 **ipa-cert-fix** 对 **nssdb** 进行了不可逆的更改。因为 **ipa-cert-fix** 也对 LDAP 进行了更改，因此建议也备份整个集群。
2. 启动 **ipa-cert-fix** 工具以分析系统，并列需要续订的过期证书：

```
# ipa-cert-fix
...
The following certificates will be renewed:

Dogtag sslserver certificate:
Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
Serial: 13
Expires: 2019-05-12 05:55:47
...
Enter "yes" to proceed:
```

3. 输入 **yes** 以开始续订过程：

```
Enter "yes" to proceed: true
Proceeding.
Renewed Dogtag sslserver certificate:
Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
Serial: 268369925
Expires: 2021-08-14 02:19:33
...

Becoming renewal master.
The ipa-cert-fix command was successful
```

ipa-cert-fix 续订所有过期的证书前可能需要一分钟。

4. (可选) 验证所有服务现在是否正在运行：

```
# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

此时，证书已被更新，服务正在运行。下一步是检查 IdM 域中的其他服务器。

**注意**

如果您需要修复跨多个 CA 服务器的证书：

1. 确保 LDAP 复制在拓扑中正常工作后，根据上述流程，首先在一台 CA 服务器上运行 **ipa-cert-fix**。
2. 在另一台 CA 服务器上运行 **ipa-cert-fix** 之前，请通过 **getcert-resubmit**（在另一台 CA 服务器上）触发共享证书的 Certmonger 续订，以避免不必要的共享证书的续订。

14.2. 在续订后验证 IDM 域中的其他 IDM 服务器

使用 **ipa-cert-fix** 工具续订 CA 续订服务器的证书后，您必须：

- 重启域中的所有其它身份管理(IdM)服务器。
- 检查 certmonger 更新证书。
- 如果存在带有过期系统证书的其他证书颁发机构(CA)副本，请使用 **ipa-cert-fix** 工具更新这些证书。

先决条件

- 使用管理权限登录到服务器。

步骤

1. 使用 **--force** 参数重启 IdM：

```
# ipactl restart --force
```

使用 **--force** 参数时，**ipactl** 程序会忽略单个服务启动失败。例如，如果服务器也是过期证书的 CA，**pki-tomcat** 服务无法启动。这是预期的行为，并会被忽略，因为使用了 **--force** 参数。

2. 重启后，验证 **certmonger** 服务是否已更新证书（证书状态为 MONITORING）：

```
# getcert list | egrep '^Request|status:|subject:'
Request ID '20190522120745':
status: MONITORING
```

```

subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
status: MONITORING
subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...

```

在 **certmonger** 更新副本上的共享证书前可能需要一些时间。

3. 如果服务器也是 CA，则上一个命令会为 **pki-tomcat** 服务使用的证书报告 **CA_UNREACHABLE**：

```

Request ID '20190522120835':
status: CA_UNREACHABLE
subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...

```

4. 要更新这个证书，请使用 **ipa-cert-fix** 程序：

```

# ipa-cert-fix
Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM
  Serial: 3
  Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: true
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
  Serial: 15
  Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful

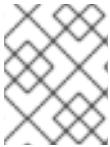
```

现在，所有 IdM 证书都有更新并可以正常工作。

第 15 章 如果 WEB 服务器和 LDAP 服务器证书还没有在 IDM 副本上过期，请替换它们

作为身份管理(IdM)系统管理员，您可以手动替换运行在 IdM 服务器上的 web（或 **httpd**）和 LDAP（或 **Directory**）服务的证书。例如，如果证书接近过期，且 **certmonger** 工具没有配置为自动更新证书，或者证书是由外部证书颁发机构(CA)签名的，则这可能是必需的。

这个示例为运行在 **server.idm.example.com** IdM 服务器上的服务安装证书。您从外部 CA 获取证书。



注意

HTTP 和 LDAP 服务证书在不同的 IdM 服务器上有不同的密钥对和主题名称，因此您必须单独在每台 IdM 服务器上更新证书。

先决条件

- 在 IdM 服务器具有复制协议的拓扑中的至少一个其他的 IdM 副本上，Web 和 LDAP 证书仍有效。这是 **ipa-server-certinstall** 命令的先决条件。命令需要 **TLS** 连接，以与其他 IdM 副本进行通信。但是，如果证书无效，此类连接无法建立，**ipa-server-certinstall** 命令将失败。在这种情况下，请参阅 [如果 web 服务器和 LDAP 服务器证书在整个 IdM 部署中过期，请替换它们](#)。
- 您有访问 IdM 服务器的 **root** 权限。
- 您知道 **目录管理器** 密码。
- 您可以访问存储外部 CA 的 CA 证书链的文件 **ca_certificate_chain_file.crt**。

步骤

1. 将 **ca_certificate_chain_file.crt** 中包含的证书作为额外的 CA 证书安装到 IdM：

```
# ipa-cacert-manage install
```

2. 使用来自 **ca_certificate_chain_file.crt** 的证书更新本地 IdM 证书数据库：

```
# ipa-certupdate
```

3. 使用 **OpenSSL** 工具生成私钥和证书签名请求(CSR)：

```
$ openssl req -new -newkey rsa:4096 -days 365 -nodes -keyout new.key -out new.csr -
addext "subjectAltName = DNS:server.idm.example.com" -subj
'/CN=server.idm.example.com,O=IDM.EXAMPLE.COM'
```

将 CSR 提交给外部 CA。这个过程根据要用作外部 CA 的服务的不同而有所不同。在 CA 为证书签名后，将证书导入到 IdM 服务器。

4. 在 IdM 服务器上，将 Apache Web 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -w --pin=password new.key new.crt
```

在以上命令中：

- **-w** 选项指定您要将证书安装到 Web 服务器中。

- **pin** 选项指定保护私钥的密码。
5. 出现提示时，输入 **目录管理器** 密码。
 6. 将 LDAP 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -d --pin=password new.key new.cert
```

在以上命令中：

- **-d** 选项指定您要将证书安装到 LDAP 服务器中。
 - **pin** 选项指定保护私钥的密码。
7. 出现提示时，输入 **目录管理器** 密码。
 8. 重启 **httpd** 服务：

```
# systemctl restart httpd.service
```

9. 重启 **Directory** 服务：

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

10. 如果服务器上的子 CA 已删除或被替换了，请更新客户端：

```
# ipa-certupdate
```

其他资源

- [转换证书格式以和 IdM 一起工作](#)
- [ipa-server-certinstall\(1\) 手册页](#)

第 16 章 如果 WEB 服务器和 LDAP 服务器证书已在整个 IDM 部署中过期，请替换它们

身份管理(IdM)使用以下服务证书：

- LDAP（或 **Directory**）服务器证书
- web（或 **httpd**）服务器证书
- PKINIT 证书

在没有 CA 的 IdM 部署中，**certmonger** 默认不会跟踪 IdM 服务证书或通知其过期。如果 IdM 系统管理员没有为这些证书手动设置通知，或者配置 **certmonger** 来跟踪它们，则证书将在没有通知的情况下过期。

按照以下流程，为运行在 **server.idm.example.com** IdM 服务器上的 **httpd** 和 LDAP 服务手动替换过期的证书。



注意

HTTP 和 LDAP 服务证书在不同的 IdM 服务器上有不同的密钥对和主题名称。因此，您必须单独更新每个 IdM 服务器上的证书。

先决条件

- HTTP 和 LDAP 证书已在拓扑中的 *所有* IdM 副本上过期。如果没有，请参阅 [如果 web 服务器和 LDAP 服务器证书还没有在 IdM 副本上过期，请替换它们（）](#)。
- 您有访问 IdM 服务器和副本的 **root** 权限。
- 您知道 **目录管理器** 密码。
- 您已创建了以下目录和文件的备份：
 - **/etc/dirsrv/slapd-*IDM-EXAMPLE-COM***
 - **/etc/httpd/alias**
 - **/var/lib/certmonger**
 - **/var/lib/ipa/certs/**

流程

1. （可选）执行 **/var/lib/ipa/private** 和 **/var/lib/ipa/passwds** 的备份。
2. 如果您没有使用相同的 CA 为新证书签名，或者如果已安装的 CA 证书不再有效，请使用包含外部 CA 的有效的 CA 证书链的文件更新本地数据库中有关外部 CA 的信息。该文件接受 PEM 和 DER 证书、PKCS#7 证书链、PKCS#8 和原始私钥，以及 PKCS#12 格式。
 - a. 将 **ca_certificate_chain_file.crt** 中提供的证书作为额外的 CA 证书安装到 IdM 中：

```
# ipa-cacert-manage install ca_certificate_chain_file.crt
```

- b. 使用来自 **ca_certificate_chain_file.crt** 的证书更新本地 IdM 证书数据库：

ipa-certupdate3. 请求 **httpd** 和 LDAP 的证书：

- a. 使用 **OpenSSL** 工具，为在 IdM 实例上运行的 Apache Web 服务器创建到第三方 CA 的证书签名请求(CSR)。

- 创建新私钥是可选的。如果您仍然有原始私钥，您可以将 **-in** 选项与 **openssl req** 命令一起使用，以指定要从中读取请求的输入文件名：

```
$ openssl req -new -nodes -in /var/lib/ipa/private/httpd.key -out /tmp/http.csr -
addext 'subjectAltName = DNS:_server.idm.example.com_,
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/server.idm.example.com@IDM.EX
AMPLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

- 如果要创建一个新密钥：

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout
/var/lib/ipa/private/httpd.key -out /tmp/http.csr -addext 'subjectAltName =
DNS:server.idm.example.com,
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/server.idm.example.com@IDM.EX
AMPLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

- b. 使用 **OpenSSL** 工具，为在 IdM 实例上运行的 LDAP 服务器创建一个到第三方证书的签名请求(CSR)：

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout ~/ldap.key -out /tmp/ldap.csr -
addext 'subjectAltName = DNS:server.idm.example.com,
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:ldap/server.idm.example.com@IDM.EXAMP
LE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

- c. 将 CSR、**/tmp/http.csr** 和 **tmp/ldap.csr** 提交给外部 CA，并获取 **httpd** 的证书和 LDAP 的证书。这个过程根据要用作外部 CA 的服务的不同而有所不同。

4. 安装 **httpd** 的证书：

```
# cp /path/to/httpd.crt /var/lib/ipa/certs/
```

5. 将 LDAP 证书安装到 NSS 数据库中：

- a. [可选] 列出可用的证书：

```
# certutil -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -L
Certificate Nickname                               Trust Attributes
                                                    SSL,S/MIME,JAR/XPI

Server-Cert                                       u,u,u
```

默认证书别名是 **Server-Cert**，但可能应用了不同的名称。

- b. 使用上一步中的证书别名，从 NSS 数据库(NSSDB)中删除旧的无效的证书：

```
# certutil -D -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -n 'Server-Cert' -f
/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

- c. 创建一个 PKCS12 文件，以简化导入到 **NSSDB** 的过程：

```
# openssl pkcs12 -export -in ldap.crt -inkey ldap.key -out ldap.p12 -name Server-Cert
```

- d. 将创建的 PKCS#12 文件安装到 **NSSDB** 中：

```
# pk12util -i ldap.p12 -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -k /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

- e. 检查新证书是否已成功导入：

```
# certutil -L -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM
```

6. 重启 **httpd** 服务：

```
# systemctl restart httpd.service
```

7. 重启 **Directory** 服务：

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

8. 在所有 IdM 副本上执行之前所有的步骤。这是在副本之间建立 **TLS** 连接的先决条件。

9. 将新证书注册到 LDAP 存储：

- a. 将 Apache Web 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -w --pin=password /var/lib/ipa/private/httpd.key /var/lib/ipa/certs/httpd.crt
```

在以上命令中：

- **-w** 选项指定您要将证书安装到 Web 服务器中。
- **pin** 选项指定保护私钥的密码。

- b. 出现提示时，输入 **目录管理器** 密码。

- c. 将 LDAP 服务器的旧私钥和证书替换为新密钥和新签名的证书：

```
# ipa-server-certinstall -d --pin=password /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ldap.key /path/to/ldap.crt
```

在以上命令中：

- **-d** 选项指定您要将证书安装到 LDAP 服务器中。
- **pin** 选项指定保护私钥的密码。

- d. 出现提示时，输入 **目录管理器** 密码。

- e. 重启 **httpd** 服务：


```
# systemctl restart httpd.service
```

f. 重启 **Directory** 服务：

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

10. 在所有其他受影响的副本上执行上一步中的命令。

其他资源

- [转换证书格式以和 IdM 一起工作](#)
- `man ipa-server-certinstall(1)`
- [过期后，如何在 RHEL 8 上手动续订身份管理\(IPA\)证书？\(CA-less IPA\)](#)

第 17 章 在 IDM CA 服务器中生成 CRL

如果您的 IdM 部署使用嵌入的证书颁发机构(CA)，您可能需要将证书撤销列表(CRL)从一个 Identity Management(IdM)服务器移动到另一个。它可能是必要的，例如，当将服务器迁移到其他系统时。

只配置一个服务器来生成 CRL。执行 CRL publisher 角色的 IdM 服务器通常是执行 CA 续订服务器角色的同一服务器，但这不是强制要求。在弃用 CRL publisher 服务器之前，选择并配置另一个服务器来执行 CRL publisher 服务器角色。

17.1. 在 IDM 服务器中停止 CRL 生成

要在 IdM CRL publisher 服务器中停止生成 Certificate Revocation List(CRL)，请使用 **ipa-crlgen-manage** 命令。在禁用生成前，请验证服务器是否生成 CRL。然后您可以禁用它。

先决条件

- 您必须以 root 身份登录。

步骤

1. 检查您的服务器是否生成 CRL：

```
[root@server ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2. 停止在服务器上生成 CRL：

```
[root@server ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on
another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3. 检查服务器是否停止生成 CRL：

```
[root@server ~]# ipa-crlgen-manage status
```

服务器停止生成 CRL。下一步是在 IdM 副本上启用 CRL 生成。

17.2. 在 IDM 副本服务器中启动 CRL 生成

您可以使用 **ipa-crlgen-manage** 命令在 IdM CA 服务器上开始生成证书 Revocation List(CRL)。

先决条件

- RHEL 系统必须是 IdM 证书颁发机构服务器。

- 您必须以 root 身份登录。

步骤

1. 开始生成 CRL :

```
[root@replica1 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

2. 检查是否生成 CRL :

```
[root@replica1 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

17.3. 更改 CRL 更新间隔

默认情况下，证书撤销列表(CRL)文件每四小时由身份管理证书颁发机构(Idm CA)自动生成。您可以按照以下流程更改此间隔。

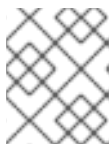
步骤

1. 停止 CRL 生成服务器 :

```
# systemctl stop pki-tomcatd@pki-tomcat.service
```

2. 打开 `/var/lib/pki/pki-tomcat/conf/ca/CS.cfg` 文件，并将 `ca.crl.MasterCRL.autoUpdateInterval` 值改为新的间隔设置。例如，要每 60 分钟生成 CRL :

```
ca.crl.MasterCRL.autoUpdateInterval=60
```



注意

如果您更新了 `ca.crl.MasterCRL.autoUpdateInterval` 参数，则更改将在下一次计划的 CRL 更新后生效。

3. 启动 CRL 生成服务器 :

```
# systemctl start pki-tomcatd@pki-tomcat.service
```

其他资源

- 有关 IdM 副本服务器上 CRL 生成的更多信息，请参阅 [在 IdM 副本服务器上启动 CRL 生成](#)。

第 18 章 停用执行 CA 续订服务器和 CRL 发布者角色的服务器

您可能有一台服务器同时执行证书颁发机构(CA)续订服务器角色和证书吊销列表(CRL)发布者角色。如果您需要将此服务器下线或停用，请选择并配置另一台 CA 服务器来执行这些角色。

在本例中，主机 `server.idm.example.com`，其履行 CA 续订服务器和 CRL 发布者角色，必须停用。此流程将 CA 续订服务器和 CRL 发布者角色转移到主机 `replica.idm.example.com`，并从 IdM 环境中删除 `server.idm.example.com`。



注意

您不需要配置同一服务器来执行 CA 续订服务器和 CRL 发布者角色。

先决条件

- 您有 IdM 管理员凭证。
- 您有要停用的服务器的 root 密码。
- 在您的 IdM 环境中至少有两个 CA 副本。

流程

1. 获取 IdM 管理员凭证：

```
[user@server ~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. (可选) 如果您不确定哪些服务器执行 CA 续订服务器和 CRL 发布者角色：

- a. 显示当前的 CA 续订服务器。您可以从任何 IdM 服务器运行以下命令：

```
[user@server ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: server.idm.example.com
```

- b. 测试主机是否为当前的 CRL 发布者。

```
[user@server ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

不生成 CRL 的 CA 服务器显示 **CRL generation: disabled**。

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: disabled
The ipa-crlgen-manage command was successful
```

继续在 CA 服务器上输入此命令，直到找到 CRL 发布者服务器。

- c. 显示您可以提升的所有其他 CA 服务器，以履行这些角色。此环境有两个 CA 服务器。

```
[user@server ~]$ ipa server-role-find --role 'CA server'
```

```

-----
2 server roles matched
-----
Server name: server.idm.example.com
Role name: CA server
Role status: enabled
Server name: replica.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----

```

3. 将 **replica.idm.example.com** 设为 CA 续订服务器。

```
[user@server ~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com
```

4. 在 **server.idm.example.com** 上 :

- a. 禁用证书更新器任务 :

```
[root@server ~]# pki-server ca-config-set ca.certStatusUpdateInterval 0
```

- b. 重启 IdM 服务 :

```
[root@server ~]# ipactl restart
```

5. 在 **replica.idm.example.com** 上 :

- a. 启用证书更新器任务 :

```
[root@replica ~]# pki-server ca-config-unset ca.certStatusUpdateInterval
```

- b. 重启 IdM 服务 :

```
[root@replica ~]# ipactl restart
```

6. 在 **server.idm.example.com** 上, 停止生成 CRL。

```

[user@server ~]$ ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on
another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful

```

7. 在 **replica.idm.example.com** 上, 开始生成 CRL。

```

[user@replica ~]$ ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg

```

```
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

8. 停止 **server.idm.example.com** 上的 IdM 服务：

```
[root@server ~]# ipactl stop
```

9. 在 **replica.idm.example.com** 上，从 IdM 环境中删除 **server.idm.example.com**。

```
[user@replica ~]$ ipa server-del server.idm.example.com
```

10. 在 **server.idm.example.com** 上，以 root 帐户身份使用 **ipa-server-install --uninstall** 命令：

```
[root@server ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

验证步骤

- 显示当前的 CA 续订服务器。

```
[user@replica ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: replica.idm.example.com
```

- 确认 **replica.idm.example.com** 主机正在生成 CRL。

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

其他资源

- [更改并重置 IdM CA 续订服务器](#)
- [在 IdM CA 服务器上生成 CRL](#)
- [卸载 IdM 副本](#)

第 19 章 使用 CERTMONGER 为服务获取 IDM 证书

19.1. CERTMONGER 概述

当使用集成的 IdM 证书颁发机构(CA)安装 Identity Management(IdM)时，它使用 **certmonger** 服务来跟踪和续订系统和服务证书。当证书达到其过期日期时，**certmonger** 将通过以下方式管理续订过程：

- 使用原始请求中提供的选项重新生成一个证书签名请求(CSR)。
- 使用 IdM API **cert-request** 命令将 CSR 提交到 IdM CA。
- 从 IdM CA 接收证书。
- 如果原始请求指定了，则执行 **pre-save** 命令。
- 在续订请求中指定的位置安装新证书：在 **NSS** 数据库中或在文件中。
- 如果由原始请求指定，则执行 **post-save** 命令。例如，**post-save** 命令可指示 **certmonger** 重启相关服务，以便服务获取新证书。

证书 **certmonger** 跟踪的类型

证书可以分为系统和服务证书。

与服务证书（例如 **HTTP**、**LDAP** 和 **PKINIT**）不同，它们在不同服务器上有不同的密钥对和主题名称，IdM 系统证书及其密钥由所有 CA 副本共享。IdM 系统证书包括：

- **IdM CA** 证书
- **OCSP** 签名证书
- **IdM CA** 子系统证书
- **IdM CA** 审计签名证书
- **IdM** 续订代理 (RA)证书
- **KRA** 传输和存储证书

certmonger 服务跟踪安装带有集成 CA 的 IdM 环境期间请求的 IdM 系统和服务证书。**certmonger** 还跟踪系统管理员为 IdM 主机上运行的其他服务手动请求的证书。**certmonger** 不跟踪外部 CA 证书或用户证书。

certmonger 组件

certmonger 服务由两个主要组件组成：

- **certmonger** 守护进程，即引擎跟踪证书列表和启动续订命令
- 命令行界面 (CLI)的 **getcert** 工具，它允许系统管理员将命令主动发送到 **certmonger** 守护进程。

更具体地说，系统管理员可以将 **getcert** 工具用于：

- [请求新证书](#)
- [查看 **certmonger** track 的证书列表](#)

- [启动或停止跟踪证书](#)
- [续订证书](#)

19.2. 使用 CERTMONGER 为服务获取 IDM 证书

为确保您的 Identity Management(IdM)客户端上运行的浏览器和 Web 服务之间的通信安全并加密，请使用 TLS 证书。从 IdM 证书颁发机构(CA)获取您的 web 服务的 TLS 证书。

按照以下流程，使用 **certmonger** 获取在 IdM 客户端上运行的服务 (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**)的 IdM 证书。

使用 **certmonger** 请求证书时，**certmonger** 会在到期续订时自动管理和更新证书。

有关 **certmonger** 请求服务证书时会发生的情况的视觉表示，请参阅 [第 19.3 节“请求服务证书的 certmonger 的通信流”](#)。

先决条件

- Web 服务器作为 IdM 客户端注册。
- 您有要在其上执行这个过程的 IdM 客户端的 root 访问权限。
- 为其请求证书的服务不需要在 IdM 中预先存在。

步骤

1. 在运行 HTTP 服务的 **my_company.idm.example.com** IdM 客户端中，为与 **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 主体对应的服务请求一个证书，并指定它。
 - 证书将存储在本地 **/etc/pki/tls/certs/httpd.pem** 文件中
 - 私钥将存储在本地 **/etc/pki/tls/private/httpd.key** 文件中
 - **SubjectAltName** 的一个 **extensionRequest** 添加至 **my_company.idm.example.com** 的 DNS 名称的签名请求中：

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

在以上命令中：

- **ipa-getcert request** 命令指定要从 IdM CA 获取的证书。**ipa-getcert request** 命令是 **getcert request -c IPA** 的一个快捷方式。
- **-g** 选项指定如果尚未存在密钥时生成的密钥大小。
- **-D** 选项指定要添加到请求的 **SubjectAltName** DNS 值。
- **-C** 选项指示 **certmonger** 在获取证书后重启 **httpd** 服务。
- 要指定使用特定配置集发布证书，请使用 **-T** 选项。

- 要使用指定 CA 的命名签发者请求证书，请使用 **-X ISSUER** 选项。

2. 另外，要检查请求的状态：

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
  certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
  CA: IPA
[...]
```

输出显示请求处于 **MONITORING** 状态，这表示已获取了证书。密钥对和证书的位置是请求的。

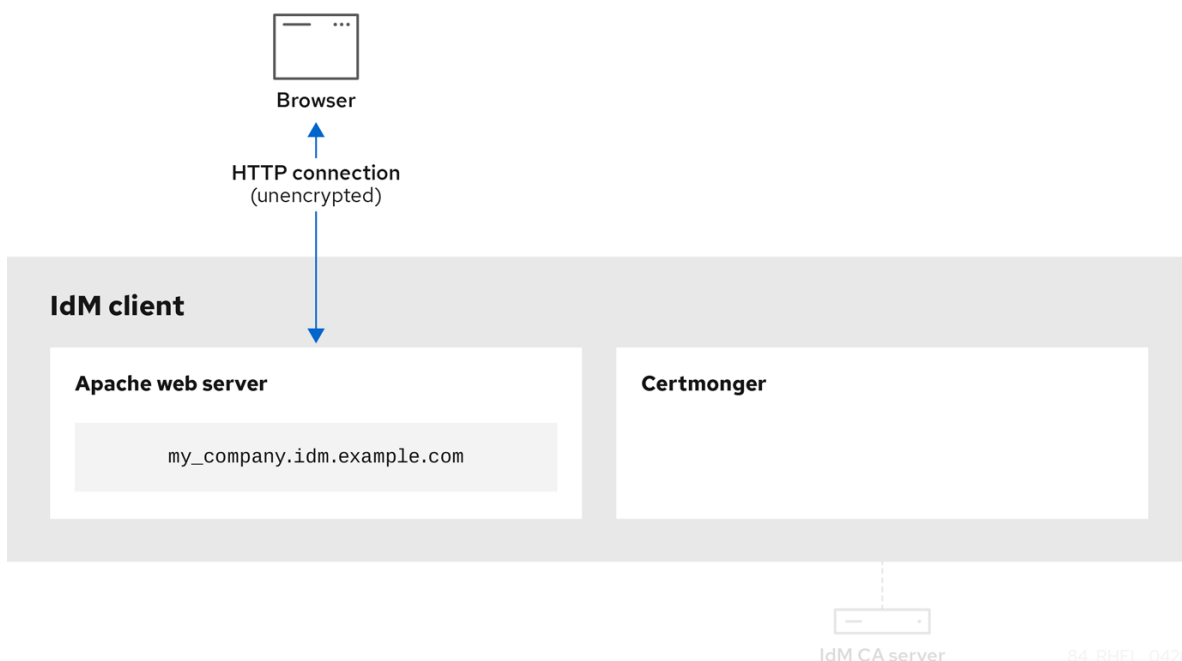
19.3. 请求服务证书的 CERTMONGER 的通信流

这些图显示了当 **certmonger** 从身份管理(IdM)证书认证机构(CA)服务器请求服务证书时发生了什么情况的阶段。序列由以下图表组成：

- 未加密的通信
- 请求服务证书的 **certmonger**
- 发布服务证书的 IdM CA
- 应用服务证书的 **certmonger**
- 当旧的证书接近过期时，请求新证书的 **certmonger**

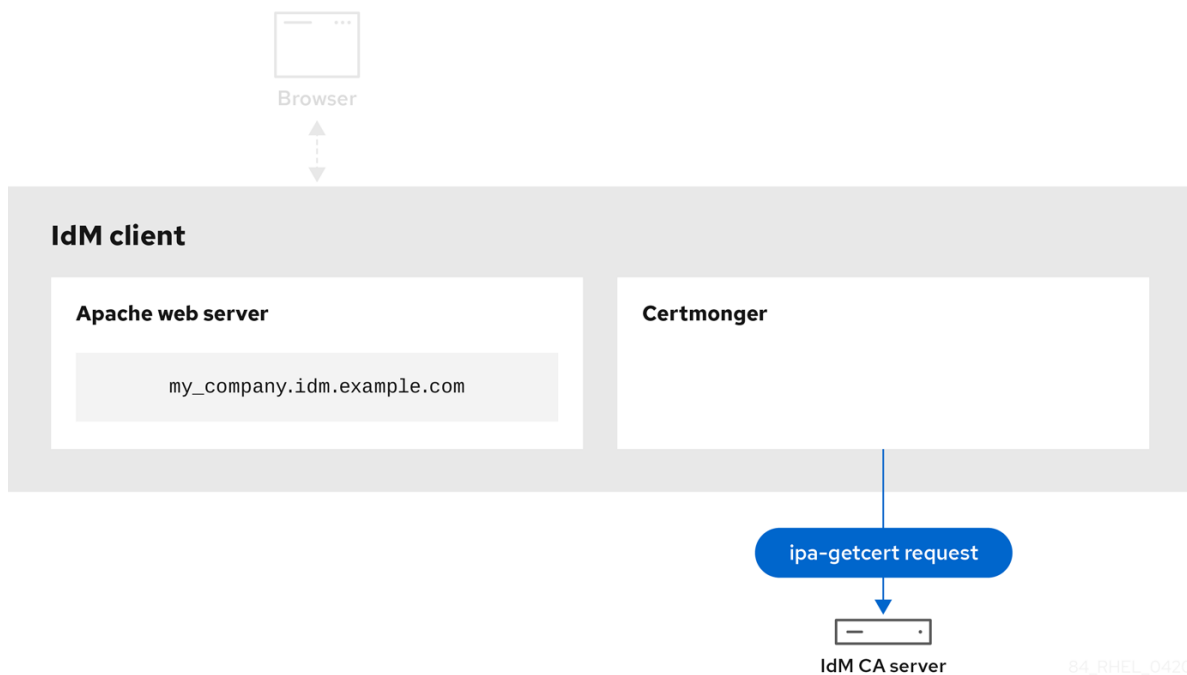
未加密的通信 显示初始情况：没有 HTTPS 证书，Web 服务器和浏览器之间的通信未加密。

图 19.1. 未加密的通信



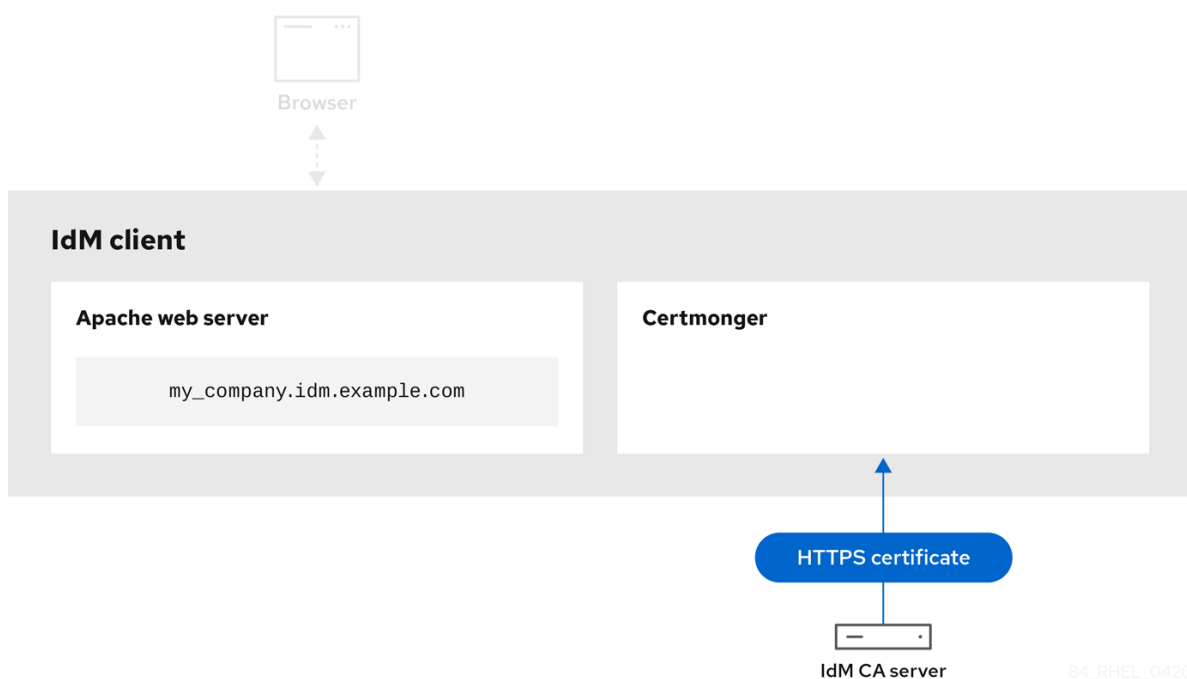
请求服务证书的 `certmonger` 显示系统管理员使用 `certmonger` 来手动为 Apache Web 服务器请求 HTTPS 证书。请注意，当请求 web 服务器证书时，`certmonger` 不会直接与 CA 通信。它通过 IdM 代理。

图 19.2. 请求服务证书的 `certmonger`



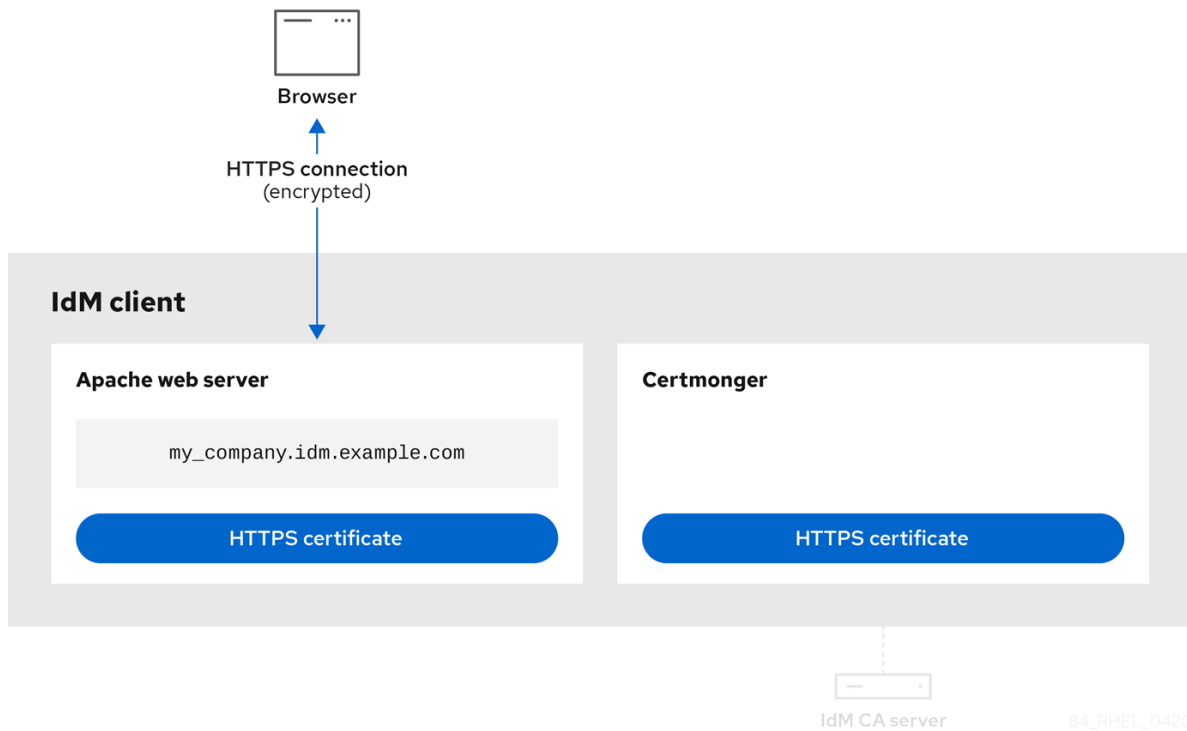
发布服务证书的 IdM CA 显示为 web 服务器发布 HTTPS 证书的 IdM CA。

图 19.3. 发布服务证书的 IdM CA



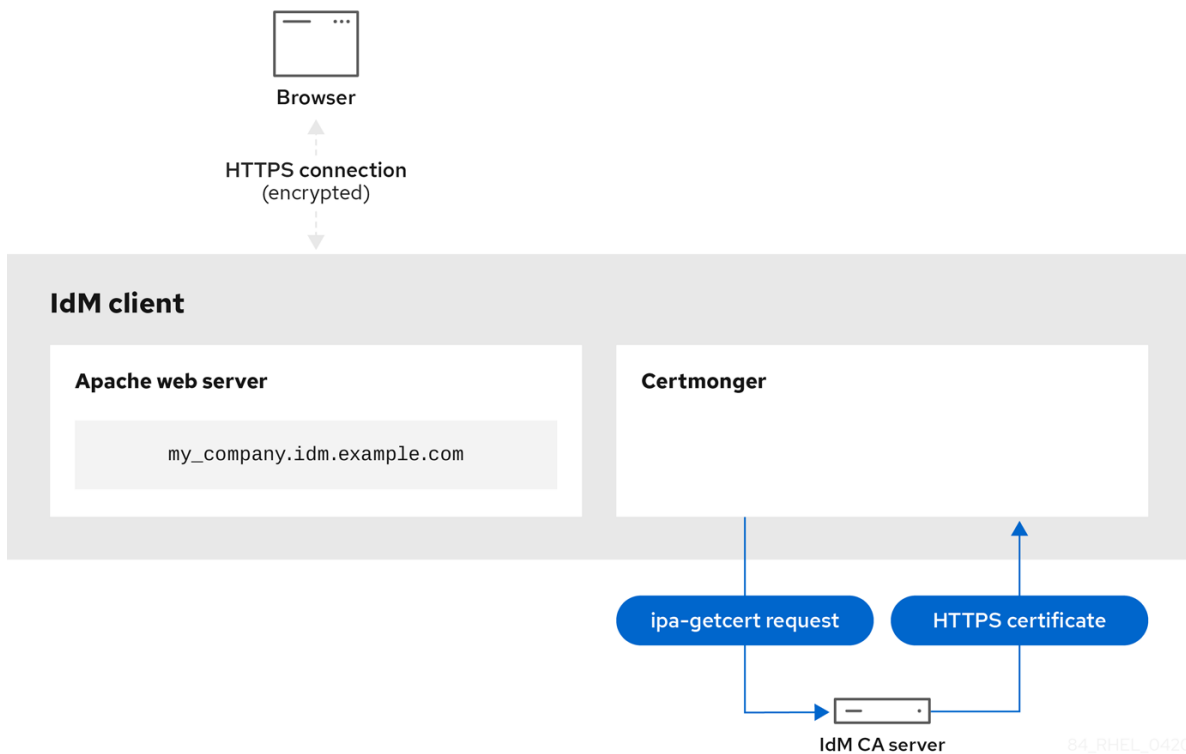
应用服务证书的 `certmonger` 显示将 HTTPS 证书放在 IdM 客户端上合适位置的 `certmonger`，如果指示要这样做，请重新启动 `httpd` 服务。随后，Apache 服务器使用 HTTPS 证书来加密其自身和浏览器之间的流量。

图 19.4. 应用服务证书的 `certmonger`



当旧的证书接近过期时，请求新证书的 `certmonger`，显示 `certmonger` 在证书过期前自动从 IdM CA 请求续订服务证书。IdM CA 发布新证书。

图 19.5. 当旧的证书接近过期时，请求新证书的 certmonger



19.4. 查看由 CERTMONGER 跟踪的证书请求详情

certmonger 服务监控证书请求。当成功签名的证书请求时，它会生成证书。**certmonger** 管理证书请求，包括生成的证书。按照以下流程查看由 **certmonger** 管理的特定证书请求的详情。

步骤

- 如果您知道如何指定证书请求，请列出仅针对该特定证书请求的详细信息。例如，您可以指定：
 - 请求 ID
 - 证书的位置
 - 证书 nickname
 例如，若要查看请求 ID 为 20190408143846 的证书详情，使用 **-v** 选项，以便在证书请求失败时查看所有错误详情：

```
# getcert list -i 20190408143846 -v
Number of certificates and requests being tracked: 16.
Request ID '20190408143846':
  status: MONITORING
  stuck: no
  key pair storage: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfilere.txt'
  certificate: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB'
  CA: IPA
  issuer: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

```

subject: CN=r8server.idm.example.com,O=IDM.EXAMPLE.COM
expires: 2021-04-08 16:38:47 CEST
dns: r8server.idm.example.com
principal name: ldap/server.idm.example.com@IDM.EXAMPLE.COM
key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
eku: id-kp-serverAuth,id-kp-clientAuth
pre-save command:
post-save command: /usr/libexec/ipa/certmonger/restart_dirsrv IDM-EXAMPLE-COM
track: true
auto-renew: true

```

输出显示了几个与证书相关的信息，例如：

- 证书位置；在上面的示例中，它是 **/etc/dirsrv/slapd-IDM-EXAMPLE-COM** 目录中的 NSS 数据库
- 证书 nickname；在上例中是 **Server-Cert**
- 存储 pin 文件；在上面的示例中，它是 **/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt**
- 用于续订证书的证书颁发机构(CA)；在上例中，它是 **IPA CA**
- 到期日期；在上面的示例中是 **2021-04-08 16:38:47 CEST**
- 证书的状态；在上面的示例中，**MONITORING** 状态意味着证书有效且正在跟踪
- post-save 命令；在上例中，它是 **LDAP** 服务的重启
- 如果您不知道如何指定证书请求，请列出 **certmonger** 正在监控或试图获取的所有证书详情：

```
# getcert list
```

其他资源

- 请参阅 **getcert list** 手册页。

19.5. 启动和停止证书跟踪

按照以下流程，使用 **getcert stop-tracking** 和 **getcert start-tracking** 命令来监控证书。这两个命令由 **certmonger** 服务提供。如果您从不同的 IdM 客户端导入了 Identity Management(IdM)证书颁发机构(CA)发布的证书，则启用证书跟踪特别有用。启用证书跟踪也可以是以下置备场景的最后一步：

1. 在 IdM 服务器中，您可以为尚不存在的系统创建一个证书。
2. 您创建新系统。
3. 您可以将新系统注册为 IdM 客户端。
4. 您将从 IdM 服务器上的证书和密钥导入到 IdM 客户端。
5. 开始使用 **certmonger** 跟踪证书，以确保其在到期过期时会被续订。

步骤

- 要禁用对带有 Request ID 20190408143846 的证书的监控：

-

```
# getcert stop-tracking -i 20190408143846
```

有关更多选项，请参阅 `getcert stop-tracking` man page。

- 要启用存储在 `/tmp/some_cert.crt` 文件中的证书监控，其私钥存储在 `/tmp/some_key.key` 文件中：

```
# getcert start-tracking -c IPA -f /tmp/some_cert.crt -k /tmp/some_key.key
```

`certmonger` 无法自动识别签发证书的 CA 类型。因此，如果证书由 IdM CA 签发，在 `getcert start-tracking` 中使用值为 `IPA` 的 `-c` 选项。省略添加 `-c` 选项会导致 `certmonger` 进入 `NEED_CA` 状态。

有关更多选项，请参阅 `getcert start-tracking` man page。



注意

这两个命令不会对证书进行操作。例如，`getcert stop-tracking` 不会删除证书，或者从 NSS 数据库或文件系统中将其删除，但只是从受监控的证书列表中删除证书。同样，`getcert start-tracking` 只会将证书添加到受监控的证书列表中。

19.6. 手动续订证书

当证书接近其过期日期时，`certmonger` 守护进程将使用证书认证机构(CA)帮助程序自动发布续订命令，获取更新的证书，并将之前的证书替换为新证书。

您还可以使用 `getcert resubmit` 命令提前手动续订证书。这样，您可以通过添加主题备用名称(SAN)来更新证书包含的信息。

按照以下流程手动续订证书。

步骤

- 要更新带有 20190408143846 的 Request ID 的证书：

```
# getcert resubmit -i 20190408143846
```

要获得特定证书的 Request ID，请使用 `getcert list` 命令。详情请查看 `getcert list` man page。

19.7. 使 CERTMONGER 恢复跟踪 CA 副本中的 IDM 证书

此流程演示，在证书跟踪被中断后，如何使 `certmonger` 恢复对带有集成证书颁发机构的 IdM 部署很重要的 Identity Management(IdM)系统证书的跟踪。IdM 主机在续订系统证书的过程中无法从 IdM 取消注册，或者复制拓扑无法正常工作。此流程还演示，如何使 `certmonger` 恢复对 IdM 服务证书（即 HTTP、LDAP 和 PKINIT 证书）的跟踪。

先决条件

- 要恢复跟踪系统证书的主机是一个 IdM 服务器，它也是 IdM 证书颁发机构(CA)，而不是 IdM CA 续订服务器。

步骤

1. 获取 subsystem CA 证书的 PIN :

```
# grep 'internal=' /var/lib/pki/pki-tomcat/conf/password.conf
```

2. 在子系统 CA 证书中添加跟踪, 使用上一步中获取的 PIN 替换下面的命令中的 [internal PIN] :

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "caSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"caSigningCert cert-pki-ca"' -T caCACert
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "auditSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"auditSigningCert cert-pki-ca"' -T caSignedLogCert
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "ocspSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"ocspSigningCert cert-pki-ca"' -T caOCSPCert
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "subsystemCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"subsystemCert cert-pki-ca"' -T caSubsystemCert
```

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "Server-Cert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"Server-Cert cert-pki-ca"' -T caServerCert
```

3. 为剩余的 IdM 证书 (HTTP、LDAP、IPA 续订代理和 PKINIT 证书) 添加跟踪 :

```
# getcert start-tracking -f /var/lib/ipa/certs/httpd.crt -k /var/lib/ipa/private/httpd.key -p
/var/lib/ipa/passwds/idm.example.com-443-RSA -c IPA -C
/usr/libexec/ipa/certmonger/restart_httpd -T caIPAserviceCert
```

```
# getcert start-tracking -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -n "Server-Cert" -c IPA
-p /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt -C
'/usr/libexec/ipa/certmonger/restart_dirsrv "IDM-EXAMPLE-COM"' -T caIPAserviceCert
```

```
# getcert start-tracking -f /var/lib/ipa/ra-agent.pem -k /var/lib/ipa/ra-agent.key -c
dogtag-ipa-ca-renew-agent -B /usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_ra_cert -T caSubsystemCert
```

```
# getcert start-tracking -f /var/kerberos/krb5kdc/kdc.crt -k
/var/kerberos/krb5kdc/kdc.key -c dogtag-ipa-ca-renew-agent -B
/usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_kdc_cert -T KDCs_PKINIT_Certs
```

4. 重启 certmonger :

```
# systemctl restart certmonger
```


5. 在 **certmonger** 启动后等待一分钟，然后检查新证书的状态：

```
# getcert list
```

其他资源

- 如果您的 IdM 系统证书已全部过期，请参阅 [这个以知识为中心的支持\(KCS\)解决方案](#)，来手动更新 IdM CA 服务器上的 IdM 系统证书，该服务器也是 CA 续订服务器和 CRL 发布者服务器。然后，请按照 [此 KCS 解决方案](#) 中描述的步骤手动在拓扑中的所有其它 CA 服务器中续订 IdM 系统证书。

19.8. 使用带有 CERTMONGER 的 SCEP

简单证书注册协议(SCEP)是您可以跨不同设备和操作系统使用的证书管理协议。如果您在环境中使用 SCEP 服务器作为外部证书颁发机构(CA)，您可以使用 **certmonger** 获取身份管理(IdM)客户端的证书。

19.8.1. SCEP 概述

简单证书注册协议(SCEP)是您可以跨不同设备和操作系统使用的证书管理协议。您可以使用 SCEP 服务器作为外部证书颁发机构(CA)。

您可以配置一个身份管理(IdM)客户端，以通过 HTTP 直接从 CA SCEP 服务请求并检索证书。此过程由共享 secret 保护，该 secret 通常仅在有限时间内有效。

在客户端上，SCEP 要求您提供以下组件：

- SCEP URL：CA SCEP 接口的 URL。
- SCEP 共享的 secret：一个在 CA 和 SCEP 客户端之间共享的 **challengePassword** PIN，用于获取证书。

然后，客户端通过 SCEP 检索 CA 证书链，并将证书签名请求发送到 CA。

配置带有 **certmonger** 的 SCEP 时，您可以创建一个新的 CA 配置的配置文件，该配置文件指定签发的证书参数。

19.8.2. 通过 SCEP 请求一个 IdM CA 签名的证书

以下示例将 **SCEP_example** SCEP CA 配置添加到 **certmonger**，并在 **client.idm.example.com** IdM 客户端上请求一个新证书。**certmonger** 支持 NSS 证书数据库格式和基于文件的(PEM)格式，如 OpenSSL。

先决条件

- 您知道 SCEP URL。
- 您有 **challengePassword** PIN 共享 secret。

流程

1. 将 CA 配置添加到 **certmonger**：

```
[root@client.idm.example.com ~]# getcert add-scep-ca -c SCEP_example -u SCEP_URL
```

- **-c**: CA 配置的强制别名。稍后可以将相同的值用于其他 **getcert** 命令。
- **-u**: 服务器的 SCEP 接口的 URL。



重要

使用 HTTPS URL 时，还必须使用 **-R** 选项指定 SCEP 服务器 CA 证书的 PEM 格式的副本的位置。

2. 验证 CA 配置是否已成功添加：

```
[root@client.idm.example.com ~]# getcert list-cas -c SCEP_example
CA 'SCEP_example':
  is-default: no
  ca-type: EXTERNAL
  helper-location: /usr/libexec/certmonger/scep-submit -u
  http://SCEP_server_enrollment_interface_URL
  SCEP CA certificate thumbprint (MD5): A67C2D4B 771AC186 FCCA654A 5E55AAF7
  SCEP CA certificate thumbprint (SHA1): FBFF096C 6455E8E9 BD55F4A5 5787C43F
  1F512279
```

如果成功添加了配置，certmonger 会从远程 CA 检索 CA 链。然后，CA 链以指纹的形式显示在命令输出中。当通过未加密的 HTTP 访问服务器时，手动将指纹与 SCEP 服务器中显示的指纹进行比较，以防止中间人攻击。

3. 从 CA 请求一个证书：

- 如果您在使用 NSS：

```
[root@client.idm.example.com ~]# getcert request -l Example_Task -c SCEP_example -
d /etc/pki/nssdb -n ExampleCert -N cn="client.idm.example.com" -L one-time_PIN -D
client.idm.example.com
```

您可以使用选项来指定证书请求的以下参数：

- **-l**: (可选) 任务的名称：请求的跟踪 ID。稍后可以将相同的值用于 **getcert list** 命令。
 - **-c**：将请求提交给的 CA 配置。
 - **-d**：包含 NSS 数据库的目录，以存储证书和密钥。
 - **-n**：证书的别名，在 NSS 数据库中使用。
 - **-n**: CSR 中的主题名称。
 - **-L**：CA 发布的一次性时间限制的 **challengePassword** PIN。
 - **-D**：证书的主题备用名称，通常与主机名相同。
- 如果您在使用 OpenSSL：

```
[root@client.idm.example.com ~]# getcert request -l Example_Task -c SCEP_example -f
/etc/pki/tls/certs/server.crt -k /etc/pki/tls/private/private.key -N
cn="client.idm.example.com" -L one-time_PIN -D client.idm.example.com
```

您可以使用选项来指定证书请求的以下参数：

- **-l**：(可选) 任务的名称：请求的跟踪 ID。稍后可以将相同的值用于 **getcert list** 命令。
- **-c**：将请求提交给的 CA 配置。
- **-f**：到证书的存储路径。
- **-k**：到密钥的存储路径。
- **-n**：CSR 中的主题名称。
- **-L**：CA 发布的一次性时间限制的 **challengePassword** PIN。
- **-D**：证书的主题备用名称，通常与主机名相同。

验证

1. 验证证书是否已颁发，并正确存储在本地数据库中：

- 如果您使用了 NSS，请输入：

```
[root@client.idm.example.com ~]# getcert list -l Example_Task
Request ID 'Example_Task':
  status: MONITORING
  stuck: no
  key pair storage:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
  certificate:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
  signing request thumbprint (MD5): 503A8EDD DE2BE17E 5BAA3A57 D68C9C1B
  signing request thumbprint (SHA1): B411ECE4 D45B883A 75A6F14D 7E3037F1
D53625F4
  CA: IPA
  issuer: CN=Certificate Authority,O=EXAMPLE.COM
  subject: CN=client.idm.example.com,O=EXAMPLE.COM
  expires: 2018-05-06 10:28:06 UTC
  key usage: digitalSignature,keyEncipherment
  eku: iso.org.dod.internet.security.mechanisms.8.2.2
  certificate template/profile: IPSECIntermediateOffline
  pre-save command:
  post-save command:
  track: true
  auto-renew: true
```

- 如果您使用了 OpenSSL，请输入：

```
[root@client.idm.example.com ~]# getcert list -l Example_Task
Request ID 'Example_Task':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/etc/pki/tls/private/private.key'
  certificate: type=FILE,location='/etc/pki/tls/certs/server.crt'
  CA: IPA
```

```

issuer: CN=Certificate Authority,O=EXAMPLE.COM
subject: CN=client.idm.example.com,O=EXAMPLE.COM
expires: 2018-05-06 10:28:06 UTC
eku: id-kp-serverAuth,id-kp-clientAuth
pre-save command:
post-save command:
track: true
auto-renew: true

```

状态 **MONITORING** 表示成功检索了签发的证书。**getcert-list(1)** 手册页列出了其他可能的状态及其含义。

其他资源

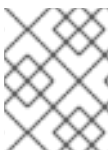
- 有关请求证书时的更多选项，请参阅 **getcert-request(1)** 手册页。

19.8.3. 使用 **certmonger** 自动续订 AD SCEP 证书

当 **certmonger** 发送 SCEP 证书续订请求时，此请求使用现有的证书私钥进行签名。但是，**certmonger** 发送的续订请求默认还包括用于最初获取证书的 **challengePassword** PIN。

作为 SCEP 服务器工作的活动目录(AD)网络设备注册服务(NDES)服务器会自动拒绝包含最初 **challengePassword** PIN 的任何续订请求。因此，续订会失败。

要使带有 AD 的续订正常工作，您需要配置 **certmonger**，以发送没有 **challengePassword** PIN 的签名续订请求。您还需要配置 AD 服务器，使其不会在续订时比较主题名称。



注意

除了 AD，SCEP 服务器也会拒绝包含 **challengePassword** 的请求。在这些情况下，您可能还需要以这种方式更改 **certmonger** 配置。

先决条件

- RHEL 服务器必须正在运行 RHEL 8.6 或更新版本。

流程

1. 在 AD 服务器上打开 **regedit**。
2. 在 **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MSCEP** 子键中，添加一个新的 32 位 **REG_DWORD** 条目 **DisableRenewalSubjectNameMatch**，并将其值设为 **1**。
3. 在运行 **certmonger** 的服务器上，打开 **/etc/certmonger/certmonger.conf** 文件，并添加以下部分：

```

[scep]
challenge_password_otp = yes

```

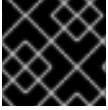
4. 重启 **certmonger**：

```

# systemctl restart certmonger

```

第 20 章 在 IDM 中部署和管理 ACME 服务



重要

这个功能是一个技术预览。

自动证书管理环境(ACME)是用于自动标识符验证和证书颁发的协议。它的目标是通过缩短证书生命周期并避免证书生命周期管理中的手动过程来提高安全性。

使用 RHEL 身份管理(IdM)，管理员可从单个系统轻松地部署和管理拓扑范围的 ACME 服务。

20.1. IDM 中的 ACME 服务



重要

这个功能是一个技术预览。



注意

IdM 目前仅在启用了 Random Certificate Serial Numbers(RSNv3)的 RHEL 9.2 或更高版本中支持 ACME。

ACME 使用质询和响应身份验证机制来证明客户端能够控制标识符。在 ACME 中，标识符是通过解决挑战来获得证书的所有权证明。在身份管理(IdM)中，ACAC 目前支持以下挑战：

- **dns-01**，客户端创建 DNS 记录来证明它对标识符有控制
- **http-01**，客户端提供 HTTP 资源以证明其对标识符有控制

在 IdM 中，ACAC 服务使用 PKI ACME 响应器。ACME 子系统会自动部署在 IdM 部署中的每个 CA 服务器上，但它不会服务请求，直到管理员启用它。服务器使用名称 **ipa-ca.DOMAIN** 来发现。所有 IdM CA 服务器都使用此 DNS 名称注册，因此请求通过轮询来平衡负载。

当管理员使用 **ipa-server-upgrade** 命令升级服务器时，还会部署 ACME，但禁用它。

ACME 作为 Apache Tomcat 中的单独服务运行。ACME 配置文件存储在 **/etc/pki/pki-tomcat/acme** 中，PKI 将 ACME 信息记录到 **/var/log/pki/pki-tomcat/acme/** 中。

在发布 ACME 证书时，IdM 使用 **acmeIPAServerCert** 配置文件。签发的证书的有效期为 90 天。因此，强烈建议将 ACME 设置为自动删除过期的证书，以便它们不会在 CA 中累积，因为这可能会对性能造成负面影响。

提供不同的 ACME 客户端。要与 RHEL 一起使用，所选客户端必须支持 **dns-01** 或 **http-01** 质询。目前，以下客户端已经过测试，并已知可在 RHEL 中与 ACME 一起工作：

- **certbot** 带有 **http-01** 和 **dns-01** 质询
- **mod_md**，其只支持 **http-01** 质询

20.2. 在 IDM 中启用 ACME 服务

**重要**

这个功能是一个技术预览。

默认情况下，部署了 ACME 服务，但已禁用。启用 ACME 服务可在整个 IdM 部署中的所有 IdM CA 服务器上启用它。这通过复制进行处理。

在本例中，您启用了 ACME，并将其设置为在每个月的第一天的午夜自动删除过期的证书。

先决条件

- IdM 部署中的服务器运行启用了随机证书序列号(RSNv3)的 RHEL 9.2 或更新版本。
- 在运行该流程的 IdM 服务器上您需要有 root 权限。

流程

1. 在整个 IdM 部署中启用 ACME：

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

2. 将 ACME 设置为从 CA 中自动删除过期的证书：

```
# ipa-acme-manage pruning --enable --cron "0 0 1 * *"
```

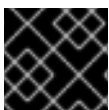
**注意**

过期的证书会在保留周期后被删除。默认情况下，这是过期后 30 天。

验证步骤

- 要检查 ACME 服务是否已安装并启用，请使用 **ipa-acme-manage status** 命令：

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

20.3. 在 IDM 中禁用 ACME 服务**重要**

这个功能是一个技术预览。

禁止 ACME 服务在整个 IdM 部署中禁用它。这通过复制进行处理。

先决条件

- IdM 部署中的服务器运行启用了随机证书序列号(RSNv3)的 RHEL 9.2 或更新版本。
- 在运行该流程的 IdM 服务器上您需要有 root 权限。

流程

1. 在整个 IdM 部署中禁用 ACME :

```
# ipa-acme-manage disable  
The ipa-acme-manage command was successful
```

2. (可选) 禁用自动删除过期的证书 :

```
ipa-acme-manage pruning --disable
```

验证步骤

- 要检查是否安装了 ACME 服务，但禁用了，请使用 **ipa-acme-manage status** 命令 :

```
# ipa-acme-manage status  
ACME is disabled  
The ipa-acme-manage command was successful
```

第 21 章 使用 RHEL 系统角色请求证书

您可以使用 **certificate** 系统角色发布和管理证书。

21.1. CERTIFICATE RHEL 系统角色

使用 **certificate** 系统角色，您可以使用 Ansible Core 管理发布和更新 TLS 和 SSL 证书。

该角色使用 **certmonger** 作为证书提供者，目前支持发布和续订自签名证书及使用 IdM 集成认证机构 (CA)。

您可以使用 **certificate** 系统角色，在 Ansible playbook 中使用以下变量：

certificate_wait

来指定任务是否应该等待要发布的证书。

certificate_requests

来表示要发布的每个证书及其参数。

其他资源

- `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 文件
- `/usr/share/doc/rhel-system-roles/certificate/` 目录

21.2. 使用 CERTIFICATE RHEL 系统角色请求一个新的自签名证书

使用 **certificate** 系统角色，您可以使用 Ansible Core 发布自签名证书。

此过程使用 **certmonger** 提供者，并通过 **getcert** 命令请求证书。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 `~/playbook.yml`：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
        dns: "*.example.com"
        ca: self-sign
```

- 将 **name** 参数设为所需证书的名称，如 **mycert**。

- 将 `dns` 参数设为要在证书中包含的域，如 `*.example.com`。
- 将 `ca` 参数设为 `self-sign`。

默认情况下，`certmonger` 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 `auto_renew` 参数设为 `no` 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook:

```
$ ansible-playbook ~/playbook.yml
```

其他资源

- `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 文件
- `/usr/share/doc/rhel-system-roles/certificate/` 目录

21.3. 使用 CERTIFICATE RHEL 系统角色从 IDM CA 请求一个新证书

使用 `certificate` 系统角色，您可以在使用带有集成证书颁发机构(CA)的 IdM 服务器时，使用 `ansible-core` 来发布证书。因此，当使用 IdM 作为 CA 时，您可以高效且一致地为多个系统管理证书信任链。

此过程使用 `certmonger` 提供者，并通过 `getcert` 命令请求证书。

先决条件

- [您已准备好控制节点和受管节点](#)
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 `sudo` 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 `~/playbook.yml`：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        principal: HTTP/www.example.com@EXAMPLE.COM
        ca: ipa
```

- 将 `name` 参数设为所需证书的名称，如 `mycert`。

- 将 **dns** 参数设要在证书中包含的域，如 **www.example.com**。
- 将 **principal** 参数设为指定 Kerberos 主体，如 **HTTP/www.example.com@EXAMPLE.COM**。
- 将 **ca** 参数设为 **ipa**。

默认情况下，**certmonger** 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 **auto_renew** 参数设为 **no** 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook:

```
$ ansible-playbook ~/playbook.yml
```

其他资源

- [/usr/share/ansible/roles/rhel-system-roles.certificate/README.md](#) 文件
- [/usr/share/doc/rhel-system-roles/certificate/](#) 目录

21.4. 使用 CERTIFICATE RHEL 系统角色指定在证书颁发之前或之后要运行的命令

使用 **certificate** 系统角色，您可以使用 Ansible Core 在签发或更新证书前后执行命令。

在以下示例中，管理员确保在为 **www.example.com** 发布或更新自签名证书前停止 **httpd** 服务，然后再重启该服务。

先决条件

- 您已准备好控制节点和受管节点
- 以可在受管主机上运行 playbook 的用户登录到控制节点。
- 用于连接到受管节点的帐户具有 **sudo** 权限。

流程

1. 创建一个包含以下内容的 playbook 文件，如 **~/playbook.yml**：

```
---
- hosts: managed-node-01.example.com
  roles:
    - rhel-system-roles.certificate
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
```

```
ca: self-sign
run_before: systemctl stop httpd.service
run_after: systemctl start httpd.service
```

- 将 **name** 参数设为所需证书的名称，如 **mycert**。
- 将 **dns** 参数设要在证书中包含的域，如 **www.example.com**。
- 将 **ca** 参数设为您要用来发布证书的 CA，如 **self-sign**。
- 将 **run_before** 参数设为在签发或续订证书之前要执行的命令，如 **systemctl stop httpd.service**。
- 将 **run_after** 参数设为在签发或续订此证书后要执行的命令，如 **systemctl start httpd.service**。

默认情况下，**certmonger** 会在证书过期前自动尝试续订证书。您可以通过将 Ansible playbook 中的 **auto_renew** 参数设为 **no** 来禁用此功能。

2. 验证 playbook 语法：

```
$ ansible-playbook --syntax-check ~/playbook.yml
```

请注意，这个命令只验证语法，不会防止错误但有效的配置。

3. 运行 playbook:

```
$ ansible-playbook ~/playbook.yml
```

其他资源

- `/usr/share/ansible/roles/rhel-system-roles.certificate/README.md` 文件
- `/usr/share/doc/rhel-system-roles/certificate/` 目录

第 22 章 将应用程序限制为只信任证书子集

如果您的身份管理(IdM)安装被配置为带有集成的证书系统(CS)证书颁发机构(CA)，您可以创建轻量级的子 CA。您创建的所有子 CA 都从属到证书系统的主 CA，**ipa CA**。

在这个上下文中的 *lightweight sub-CA* 代表一个子 CA 为特定目的发布证书。例如，可以使用轻量级子 CA 配置一个服务（如 VPN 网关，Web 浏览器），仅接受由子 CA A 签发的证书。通过将其他服务配置为仅接受子 CA B 签发的证书，您可以防止他们接受子 CA A，主 CA (**ipa CA**) 以及两者之间的任何中间子 CA 签发的证书。

如果您撤销了一个子 CA 的中间证书，正确配置的客户端会将由这个子 CA 发布的所有证书都视为无效。root CA、**ipa** 或另一个子 CA 直接发布的所有其他证书均保持有效。

本节使用 Apache Web 服务器示例来说明如何将应用程序限制为仅信任某个证书子集。完成这个部分，将在 IdM 客户端中运行的 web 服务器限制为使用由 **webserver-ca** IdM 子 CA 发布的证书，并要求用户使用 **webclient-ca** IdM 子 CA 发布的用户证书向 Web 服务器进行身份验证。

您需要执行的步骤有：

1. [创建 IdM 子 CA](#)
2. [从 IdM WebUI 下载子 CA 证书](#)
3. [创建 CA ACL，指定用户、服务和 CA 的正确组合，以及使用的证书配置集](#)
4. [为 IdM 子 CA 在 IdM 客户端中运行的 web 服务请求证书](#)
5. [设置单实例 Apache HTTP 服务器](#)
6. [在 Apache HTTP 服务器中添加 TLS 加密](#)
7. [在 Apache HTTP 服务器中设置支持的 TLS 协议版本](#)
8. [在 Apache HTTP 服务器中设置支持的密码](#)
9. [在 web 服务器上配置 TLS 客户端证书身份验证](#)
10. [从 IdM 子 CA 请求用户的证书并将其导出到客户端](#)
11. [将用户证书导入到浏览器中，并将浏览器配置为信任 sub-CA 证书](#)

22.1. 管理轻量级子 CA

本节描述了如何管理轻量级从属证书颁发机构(sub-CA)。您创建的所有子 CA 都从属到证书系统的主 CA，**ipa CA**。您还可以禁用和删除子 CA。



注意

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 **notAfter** 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子 CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

有关管理子 CA 的详情，请参阅：

- [从 IdM WebUI 创建子 CA](#)
- [从 IdM WebUI 删除子 CA](#)
- [从 IdM CLI 创建子 CA](#)
- [从 IdM CLI 禁用子 CA](#)
- [从 IdM CLI 删除子 CA](#)

22.1.1. 从 IdM WebUI 创建子 CA

按照以下流程，使用 IdM WebUI 创建名为 `webserver-ca` 和 `webclient-ca` 的新子 CA。

先决条件

- 确保您已获取了管理员的凭据。

步骤

1. 在 **Authentication** 菜单中，点 **Certificates**。
2. 选择证书授权并点添加。
3. 输入 `webserver-ca` 子 CA 的名称。在 Subject DN 字段中输入 Subject DN，如 `CN=WEBSERVER,O=IDM.EXAMPLE.COM`。请注意，Subject DN 在 IdM CA 基础架构中必须是唯一的。
4. 输入 `webclient-ca` 子 CA 的名称。在 Subject DN 字段中输入 Subject DN `CN=WEBCLIENT,O=IDM.EXAMPLE.COM`。
5. 在命令行界面中，运行 `ipa-certupdate` 命令，来为 `webserver-ca` 和 `webclient-ca` 子 CA 证书创建 `certmonger` 追踪请求：

```
[root@ipaserver ~]# ipa-certupdate
```



重要

在创建子 CA 后如果忘记运行 `ipa-certupdate` 命令，则意味着在子 CA 证书过期时，即使最终的证书没有过期，子 CA 签发的最终证书也会被视为无效，。

验证

- 验证新子 CA 的签名证书是否已添加到 IdM 数据库中：

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

Certificate Nickname	Trust Attributes
	SSL,S/MIME,JAR/XPI
caSigningCert cert-pki-ca	CTu,Cu,Cu
Server-Cert cert-pki-ca	u,u,u
auditSigningCert cert-pki-ca	u,u,Pu

```
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca u,u,u
subsystemCert cert-pki-ca u,u,u
```



注意

新的子 CA 证书将自动传输到安装了证书系统实例的所有副本。

22.1.2. 从 IdM WebUI 删除子 CA

按照以下流程删除 IdM Web UI 中的轻量级子 CA。



注意

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 **notAfter** 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子 CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

先决条件

- 确保您已获取了管理员的凭据。
- 您已在 IdM CLI 中禁用了子 CA。请参阅 [从 IdM CLI 禁用子 CA](#)

步骤

1. 在 IdM Web UI 中，打开 **身份验证** 选项卡，然后选择 **证书** 子选项卡。
2. 选择 **证书颁发机构**。
3. 选择要删除的子 CA，然后单击“**删除**”。

图 22.1. 在 IdM Web UI 中删除子 CA

The screenshot shows the IdM Web UI interface. The top navigation bar includes 'Identity', 'Policy', 'Authentication', 'Network Services', and 'IPA Server'. The 'Authentication' section is active, showing 'Certificates', 'OTP Tokens', 'RADIUS Servers', and 'Certificate Identity Mapping Rules'. The 'Certificate Authorities' page is displayed, featuring a search bar, 'Refresh', 'Delete', and 'Add' buttons. A table lists three entries: 'ipa', 'webclient-ca', and 'webserver-ca'. The 'webserver-ca' entry is selected with a checkmark in the first column.

<input type="checkbox"/>	Name	Subject DN	Description
<input type="checkbox"/>	ipa	CN=Certificate Authority,O=IPA.TEST	IPA CA
<input type="checkbox"/>	webclient-ca	CN=WEBCIENT,O=IDM.EXAMPLE.COM	
<input checked="" type="checkbox"/>	webserver-ca	CN=WEBSERVER,O=IDM.EXAMPLE.COM	

Showing 1 to 3 of 3 entries.

4. 单击 **删除** 以确认。

子 CA 从 **证书颁发机构** 列表中删除。

22.1.3. 从 IdM CLI 创建子 CA

按照以下流程，使用 IdM CLI 创建名为 **webserver-ca** 和 **webclient-ca** 的新子 CA。

先决条件

- 确保您已获取了管理员的凭据。
- 确保您已登录到一个 CA 服务器的 IdM 服务器。

步骤

1. 输入 **ipa ca-add** 命令，指定 **webserver-ca** 子 CA 的名称及其 Subject Distinguished Name(DN)：

```
[root@ipaserver ~]# ipa ca-add webserver-ca --
subject="CN=WEBSERVER,O=IDM.EXAMPLE.COM"
-----
Created CA "webserver-ca"
-----
Name: webserver-ca
Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

名称

CA 的名称。

颁发机构 ID

自动创建，为 CA 创建单独的 ID。

主题 DN

主题可辨识名称(DN)。Subject DN 在 IdM CA 基础架构中必须是唯一的。

签发者 DN

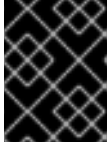
发布子 CA 的父 CA。所有子 CA 都作为 IdM root CA 的子 CA 创建。

2. 创建 **webclient-ca** 子 CA 以将证书发送到 Web 客户端：

```
[root@ipaserver ~]# ipa ca-add webclient-ca --
subject="CN=WEBCLIENT,O=IDM.EXAMPLE.COM"
-----
Created CA "webclient-ca"
-----
Name: webclient-ca
Authority ID: 8a479f3a-0454-4a4d-8ade-fd3b5a54ab2e
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

3. 运行 **ipa-certupdate** 命令，来为 **webserver-ca** 和 **webclient-ca** 子 CAs 证书创建 **certmonger** 追踪请求：

```
[root@ipaserver ~]# ipa-certupdate
```



重要

如果您在创建子 CA 后忘记了运行 `ipa-certupdate` 命令，且子 CA 证书已过期，则该子 CA 发布的最终身份证书被视为无效，即使最终身份证书没有过期。

验证步骤

- 验证新子 CA 的签名证书是否已添加到 IdM 数据库中：

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

Certificate Nickname	Trust Attributes
	SSL,S/MIME,JAR/XPI
caSigningCert cert-pki-ca	CTu,Cu,Cu
Server-Cert cert-pki-ca	u,u,u
auditSigningCert cert-pki-ca	u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc	u,u,u
ocspSigningCert cert-pki-ca	u,u,u
subsystemCert cert-pki-ca	u,u,u



注意

新的子 CA 证书将自动传输到安装了证书系统实例的所有副本。

22.1.4. 从 IdM CLI 禁用子 CA

按照以下流程，从 IdM CLI 禁用子 CA。如果子 CA 发布的证书还有未过期的，则您不应该删除它，但可以禁用它。如果您删除了子 CA，则对该子 CA 的吊销检查将不再工作。

先决条件

- 确保您已获取了管理员的凭据。

步骤

1. 运行 `ipa ca-find` 命令来确定您要删除的子 CA 的名称：

```
[root@ipaserver ~]# ipa ca-find
-----
3 CAs matched
-----
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webserver-ca
```



```

Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
-----

```

```

-----
Number of entries returned 3
-----

```

2. 运行 **ipa ca-disable** 命令来禁用您的子 CA，在本例中为 **webserver-ca**：

```

ipa ca-disable webserver-ca
-----
Disabled CA "webserver-ca"
-----

```

22.1.5. 从 IdM CLI 删除子 CA

按照以下流程从 IdM CLI 删除轻量级子 CA。



注意

- 如果您删除了子 CA，则该子 CA 的吊销检查将不再工作。只有当子 CA 不再发布证书，且其 **notAfter** 过期时间在未来时，才可以删除该子 CA。
- 只有当子 CA 发布的证书仍然未过期时，才应禁用该子 CA。如果子 CA 发布的所有证书都已过期，您可以删除该子 CA。
- 您不能禁用或删除 IdM CA。

先决条件

- 确保您已获取了管理员的凭据。

步骤

1. 要显示子 CA 和 CA 的列表，请运行 **ipa ca-find** 命令：

```

# ipa ca-find
-----
3 CAs matched
-----
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webserver-ca
Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM

```

```
Issuer DN: CN=Certificate Authority,O=IPA.TEST
-----
```

```
Number of entries returned 3
-----
```

- 运行 **ipa ca-disable** 命令来禁用您的子 CA，在本例中为 **webserver-ca**：

```
# ipa ca-disable webserver-ca
-----
```

```
Disabled CA "webserver-ca"
-----
```

- 删除子 CA，在本例中为 **webserver-ca**：

```
# ipa ca-del webserver-ca
-----
```

```
Deleted CA "webserver-ca"
-----
```

验证

- 运行 **ipa ca-find** 来显示 CA 和子 CA 的列表。 **webserver-ca** 不再位于列表中。

```
# ipa ca-find
-----
```

```
2 CAs matched
-----
```

```
Name: ipa
```

```
Description: IPA CA
```

```
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
```

```
Subject DN: CN=Certificate Authority,O=IPA.TEST
```

```
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webclient-ca
```

```
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
```

```
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
```

```
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
-----
Number of entries returned 2
-----
```

22.2. 从 IDM WEBUI 下载子 CA 证书

先决条件

- 确保您已获取 IdM 管理员的凭证。

步骤

- 在 **Authentication** 菜单中，点击 **Certificates > Certificates**。

图 22.2. 证书列表中的子 CA 证书

<input type="checkbox"/>	268173326	CN=WEBSERVER,O=IDM.EXAMPLE.COM	ipa	VALID
<input type="checkbox"/>	268238849	CN=idm_user,O=IDM.EXAMPLE.COM	ipa	VALID

2. 点 sub-CA 证书的序列号打开证书信息页面。
3. 在证书信息页面中，点 **Actions > Download**。
4. 在 CLI 中，将 sub-CA 证书移动到 `/etc/pki/tls/private/` 目录中：

```
# mv path/to/the/downloaded/certificate /etc/pki/tls/private/sub-ca.crt
```

22.3. 为 WEB 服务器和客户端身份验证创建 CA ACL

证书颁发机构访问控制列表(CA ACL)规则定义哪些配置集可用于发布哪些用户、服务或主机的证书。通过关联配置集、主体和组，CA ACL 允许主体或组使用特定配置集请求证书。

例如，使用 CA ACL，管理员可以将只用于伦敦办事处工作的员工的配置文件限制为与伦敦办事处相关的组的成员。

22.3.1. 在 IdM CLI 中查看 CA ACL

按照以下流程查看 IdM 部署中提供的证书颁发机构访问控制列表(CA ACL)以及特定 CA ACL 的详情。

步骤

1. 要在 IdM 环境中查看所有 CA ACL，请输入 `ipa caacl-find` 命令：

```
$ ipa caacl-find
-----
1 CA ACL matched
-----
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
```

2. 要查看 CA ACL 的详情，请输入 `ipa caacl-show` 命令并指定 CA ACL 名称。例如，要查看 `hosts_services_calPAserviceCert` CA ACL 的详情，请输入：

```
$ ipa caacl-show hosts_services_calPAserviceCert
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
Host category: all
Service category: all
CAs: ipa
Profiles: calPAserviceCert
Users: admin
```

22.3.2. 使用由 `webserver-ca` 发布的证书为 Web 客户端创建 CA ACL

按照以下流程，在为 `HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM` 服务请求证书时，创建一个需要系统管理员使用 `webserver-ca` 子 CA 和 `calPAserviceCert` 配置文件的 CA ACL。如果用户从其他子 CA 或不同配置集请求证书，则请求会失败。唯一的例外是，当存在另一个启用了匹配的 CA ACL 时。要查看可用的 CA ACL，请参阅[在 IdM CLI 中查看 CA ACL](#)。

先决条件

- 确保 `HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM` 服务是 IdM 的一部分。
- 请确定您已获取了 IdM 管理员的凭证。

步骤

1. 使用 `ipa caacl` 命令创建 CA ACL，并指定其名称：

```
$ ipa caacl-add TLS_web_server_authentication
-----
Added CA ACL "TLS_web_server_authentication"
-----
ACL name: TLS_web_server_authentication
Enabled: TRUE
```

2. 使用 `ipa caacl-mod` 命令修改 CA ACL，以指定 CA ACL 的描述：

```
$ ipa caacl-mod TLS_web_server_authentication --desc="CAACL for web servers
authenticating to web clients using certificates issued by webserver-ca"
-----
Modified CA ACL "TLS_web_server_authentication"
-----
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
Enabled: TRUE
```

3. 将 `webserver-ca` 子 CA 添加到 CA ACL 中：

```
$ ipa caacl-add-ca TLS_web_server_authentication --ca=webserver-ca
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
Enabled: TRUE
CAs: webserver-ca
-----
Number of members added 1
-----
```

4. 使用 `ipa caacl-add-service` 指定其主体可以请求证书的服务：

```
$ ipa caacl-add-service TLS_web_server_authentication --
service=HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-----
Number of members added 1
-----
```

- 使用 `ipa caacl-add-profile` 命令为请求的证书配置集指定：

```
$ ipa caacl-add-profile TLS_web_server_authentication --
certprofiles=calPAserviceCert
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Profiles: calPAserviceCert
  Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-----
Number of members added 1
-----
```

您可以直接使用新创建的 CA ACL。它默认在创建后启用。



注意

CA ACL 的作用是，可为来自特定主体或组的请求指定允许哪些 CA 和配置集组合。CA ACL 不会影响证书验证或信任。它们不会影响签发的证书的使用方式。

22.3.3. 使用由 `webclient-ca` 发布的证书为 Web 服务器验证用户 Web 浏览器创建 CA ACL

按照以下流程，在请求证书时，创建一个需要系统管理员使用 `webclient-ca` 子 CA 和 `IECUserRoles` 配置文件的 CA ACL。如果用户从其他子 CA 或不同配置集请求证书，则请求会失败。唯一的例外是，当存在另一个启用了匹配的 CA ACL 时。要查看可用的 CA ACL，请参阅[在 IdM CLI 中查看 CA ACL](#)。

先决条件

- 确保您已获取 IdM 管理员的凭据。

步骤

- 使用 `ipa caacl` 命令创建 CA ACL 并指定其名称：

```
$ ipa caacl-add TLS_web_client_authentication
-----
Added CA ACL "TLS_web_client_authentication"
-----
  ACL name: TLS_web_client_authentication
  Enabled: TRUE
```

- 使用 `ipa caacl-mod` 命令修改 CA ACL，以指定 CA ACL 的描述：

```
$ ipa caacl-mod TLS_web_client_authentication --desc="CAACL for user web
browsers authenticating to web servers using certificates issued by webclient-ca"
-----
Modified CA ACL "TLS_web_client_authentication"
-----
  ACL name: TLS_web_client_authentication
```

```
Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
Enabled: TRUE
```

3. 将 **webclient-ca** 子 CA 添加到 CA ACL 中：

```
$ ipa caacl-add-ca TLS_web_client_authentication --ca=webclient-ca
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca
-----
Number of members added 1
-----
```

4. 使用 **ipa caacl-add-profile** 命令为请求的证书配置集指定：

```
$ ipa caacl-add-profile TLS_web_client_authentication --certprofiles=IECUserRoles
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca
Profiles: IECUserRoles
-----
Number of members added 1
-----
```

5. 使用 **ipa caacl-mod** 命令修改 CA ACL，以指定 CA ACL 适用于所有 IdM 用户：

```
$ ipa caacl-mod TLS_web_client_authentication --usercat=all
-----
Modified CA ACL "TLS_web_client_authentication"
-----
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
Enabled: TRUE
User category: all
CAs: webclient-ca
Profiles: IECUserRoles
```

您可以直接使用新创建的 CA ACL。它默认在创建后启用。



注意

CA ACL 的作用是，可为来自特定主体或组的请求指定允许哪些 CA 和配置集组合。CA ACL 不会影响证书验证或信任。它们不会影响签发的证书的使用方式。

22.4. 使用 CERTMONGER 为服务获取 IDM 证书

要确保您的 IdM 客户端上运行的浏览器和 Web 服务之间的通信安全并加密，请使用 TLS 证书。如果要将 Web 浏览器限制为信任由 **webserver-ca** 子 CA 而不是其他 IdM 子 CA 签发的证书，从 **webserver-ca** 子 CA 获取 web 服务的 TLS 证书。

按照以下流程，使用 **certmonger** 获取在 IdM 客户端上运行的服务 (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**) 的 IdM 证书。

使用 **certmonger** 请求证书时，**certmonger** 会在到期续订时自动管理和更新证书。

有关 **certmonger** 请求服务证书时会发生的情况的视觉表示，请参阅 [第 22.5 节“请求服务证书的 certmonger 的通信流”](#)。

先决条件

- Web 服务器作为 IdM 客户端注册。
- 您有要在其上执行这个过程的 IdM 客户端的 root 访问权限。
- 为其请求证书的服务不需要在 IdM 中预先存在。

步骤

1. 在运行 HTTP 服务的 **my_company.idm.example.com** IdM 客户端中，为与 **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** 主体对应的服务请求一个证书，并指定它。

- 证书将存储在本地 **/etc/pki/tls/certs/httpd.pem** 文件中
- 私钥将存储在本地 **/etc/pki/tls/private/httpd.key** 文件中
- **webserver-ca** 子 CA 是签发的证书颁发机构
- **SubjectAltName** 的一个 **extensionRequest** 添加至 **my_company.idm.example.com** 的 DNS 名称的签名请求中：

```
# ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -X webserver-ca -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

在以上命令中：

- **ipa-getcert request** 命令指定要从 IdM CA 获取的证书。**ipa-getcert request** 命令是 **getcert request -c IPA** 的一个快捷方式。
 - **-g** 选项指定如果尚未存在密钥时生成的密钥大小。
 - **-D** 选项指定要添加到请求的 **SubjectAltName** DNS 值。
 - **-X** 选项指定证书的签发者必须是 **webserver-ca**，而不是 **ipa**。
 - **-C** 选项指示 **certmonger** 在获取证书后重启 **httpd** 服务。
 - 要指定使用特定配置集发布证书，请使用 **-T** 选项。
2. 另外，要检查请求的状态：

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
  certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
  CA: IPA
  issuer: CN=WEBSERVER,O=IDM.EXAMPLE.COM

[...]
```

输出显示请求处于 **MONITORING** 状态，这表示已获取了证书。密钥对和证书的位置是请求的。

22.5. 请求服务证书的 CERTMONGER 的通信流

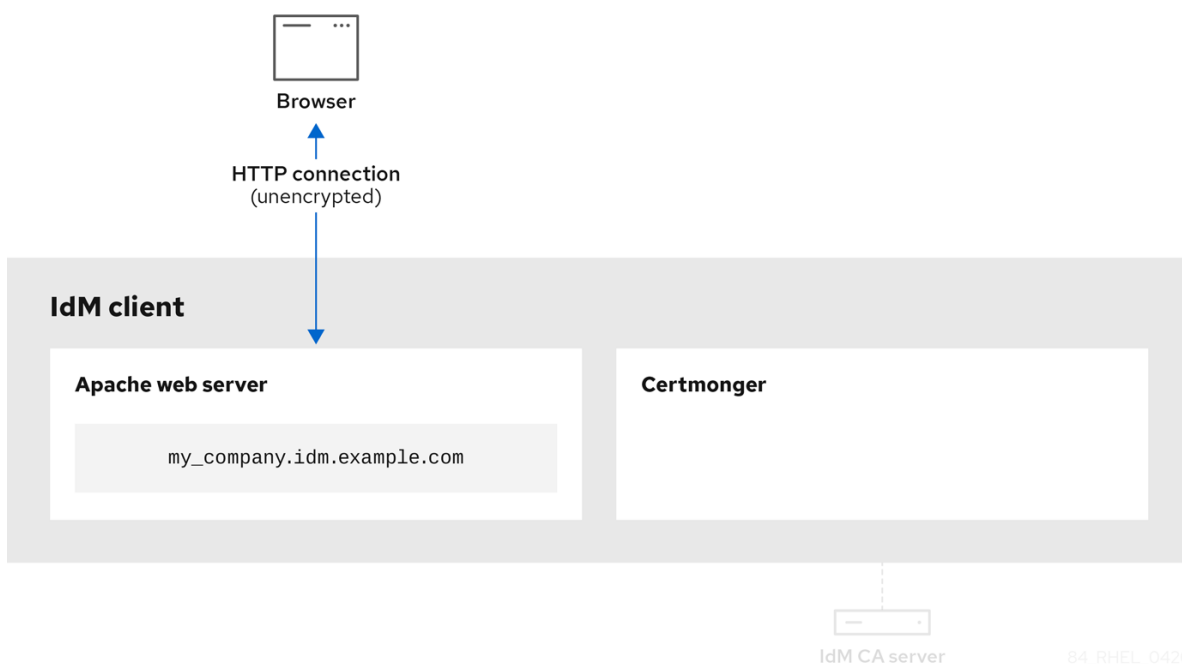
这些图显示了当 **certmonger** 从身份管理(IdM)证书认证机构(CA)服务器请求服务证书时发生了什么情况的阶段。序列由以下图表组成：

- 未加密的通信
- 请求服务证书的 **certmonger**
- 发布服务证书的 IdM CA
- 应用服务证书的 **certmonger**
- 当旧的证书接近过期时，请求新证书的 **certmonger**

在图中，**webserver-ca** 子 CA 由通用 **IdM CA 服务器** 代表。

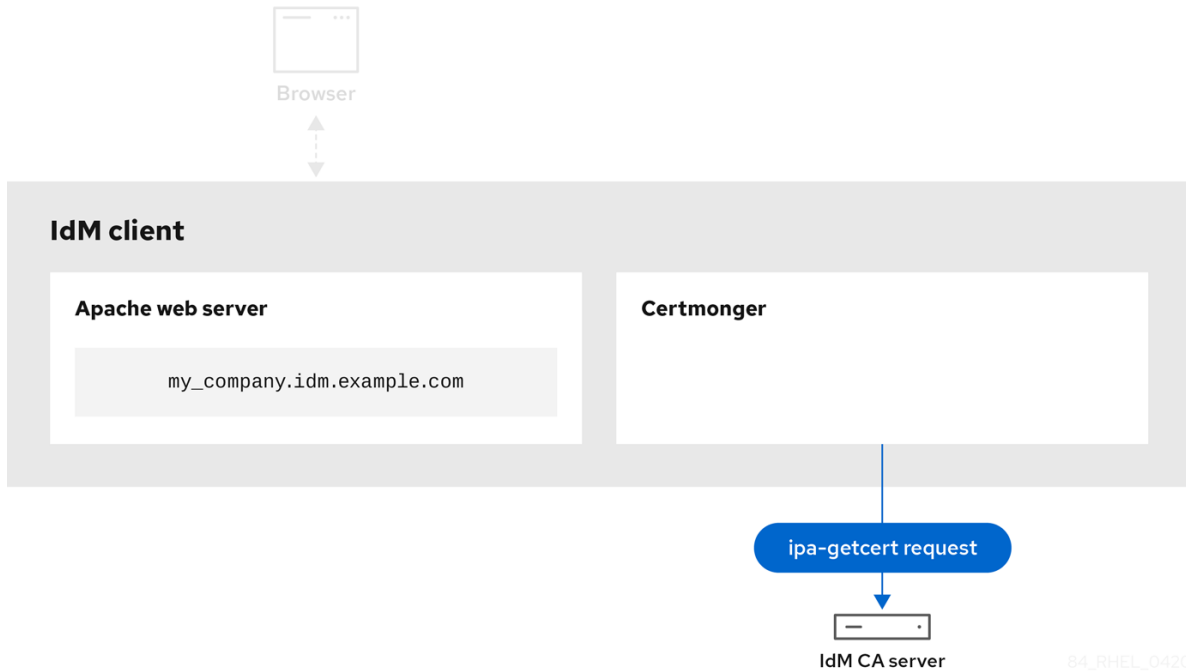
未加密的通信 显示初始情况：没有 HTTPS 证书，Web 服务器和浏览器之间的通信是未加密的。

图 22.3. 未加密的通信



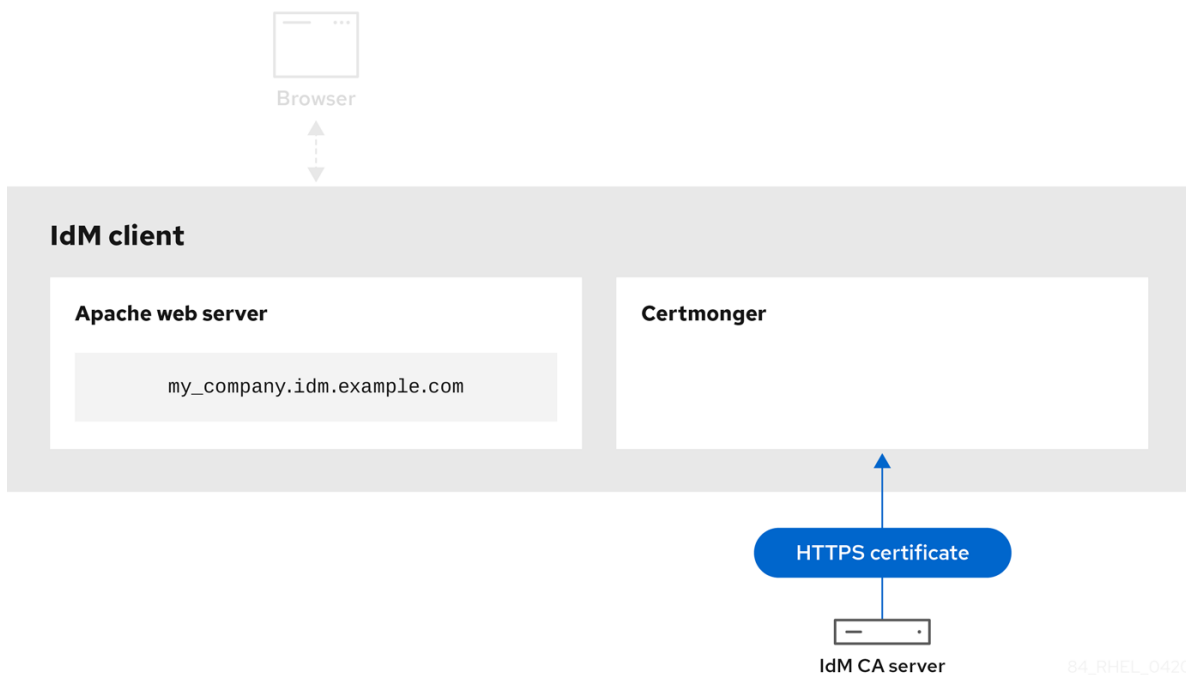
请求服务证书的 `certmonger` 显示系统管理员使用 `certmonger` 来手动为 Apache Web 服务器请求 HTTPS 证书。请注意，当请求 web 服务器证书时，`certmonger` 不会直接与 CA 通信。它通过 IdM 代理。

图 22.4. 请求服务证书的 `certmonger`



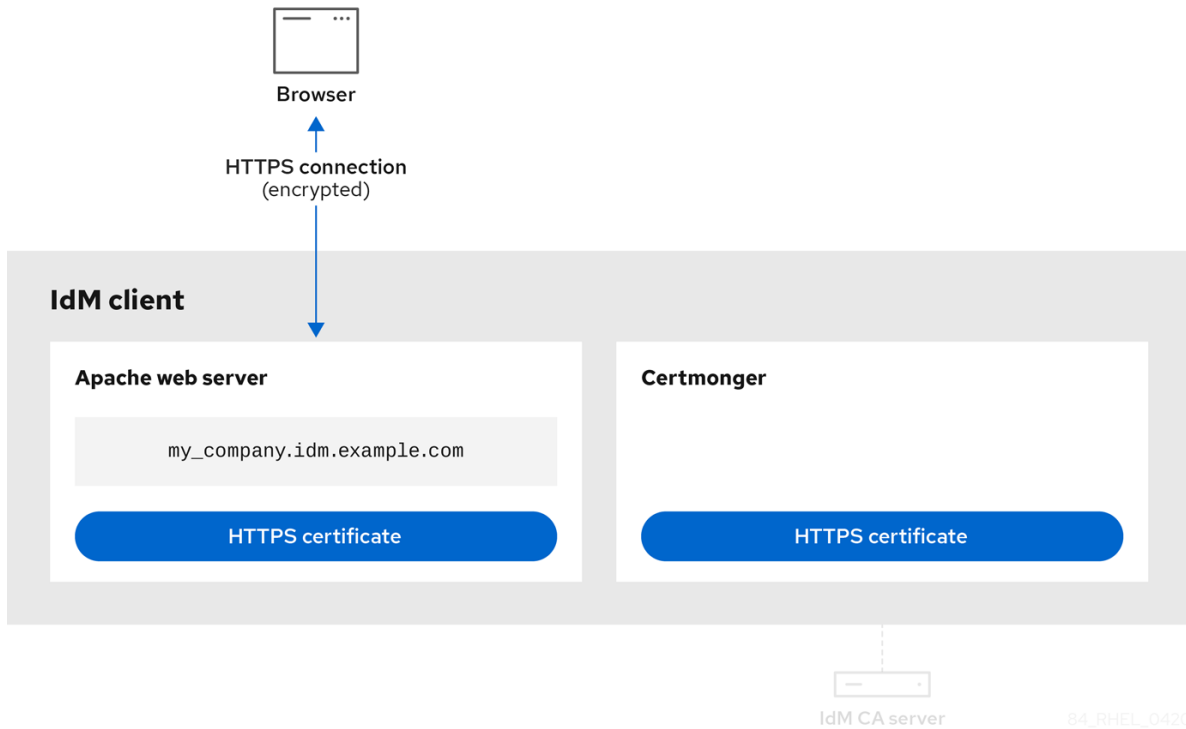
发布服务证书的 IdM CA 显示为 web 服务器发出 HTTPS 证书的 IdM CA。

图 22.5. 发布服务证书的 IdM CA



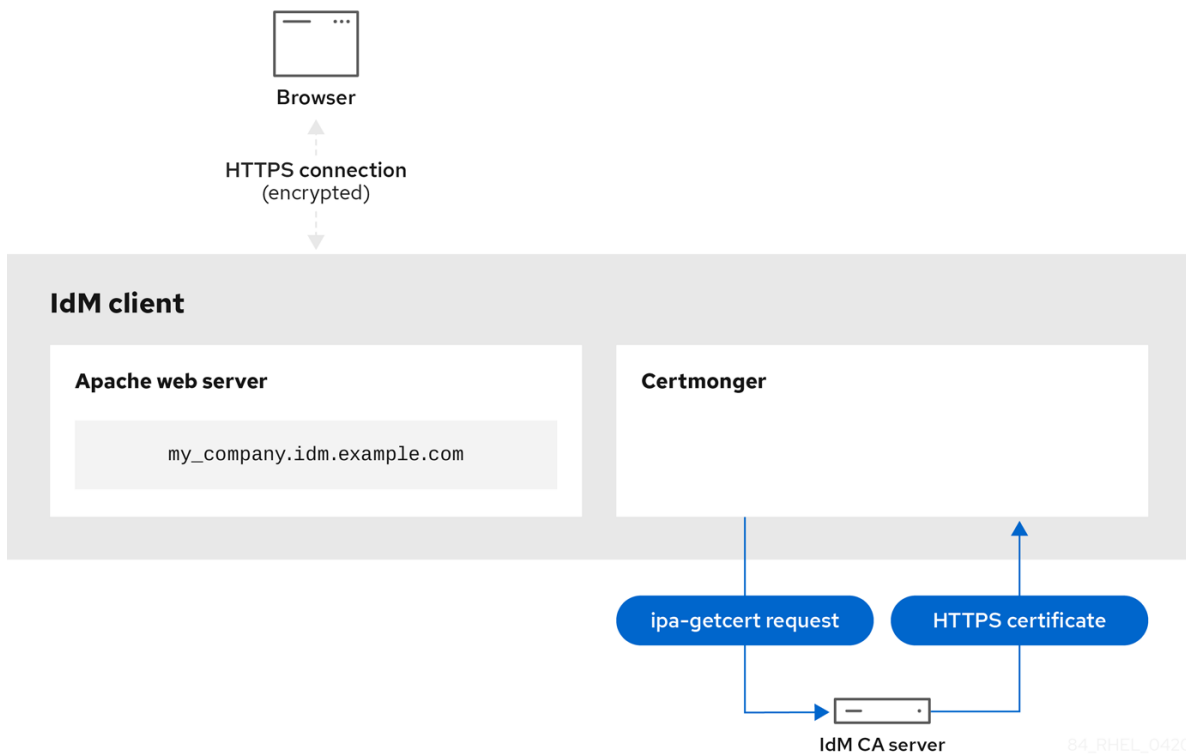
应用服务证书的 `certmonger` 显示将 HTTPS 证书放在 IdM 客户端上合适位置的 `certmonger`，如果指示要这样做，请重新启动 `httpd` 服务。随后，Apache 服务器使用 HTTPS 证书来加密其自身和浏览器之间的流量。

图 22.6. 应用服务证书的 `certmonger`



当旧证书接近过期时，请求新证书的 `certmonger`，显示 `certmonger` 在证书过期前自动从 IdM CA 请求续订服务证书。IdM CA 发布新证书。

图 22.7. 当旧的证书接近过期时，请求新证书的 certmonger



22.6. 设置单实例 APACHE HTTP 服务器

您可以设置一个单实例 Apache HTTP 服务器，来提供静态 HTML 内容。

如果 Web 服务器应该为与服务器关联的所有域提供相同的内容，请按照以下流程操作。如果要为不同的域提供不同的内容，请设置基于名称的虚拟主机。详情请参阅[配置 Apache 基于名称的虚拟主机](#)。

步骤

1. 安装 **httpd** 软件包：

```
# dnf install httpd
```

2. 如果使用 **firewalld**，请在本地防火墙中打开 TCP 端口 **80**：

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

3. 启用并启动 **httpd** 服务：

```
# systemctl enable --now httpd
```

4. 可选：将 HTML 文件添加到 **/var/www/html/** 目录中。



注意

在向 `/var/www/html/` 添加内容时，在 `httpd` 默认运行的情况下，文件和目录必须可被用户读取。内容所有者可以是 `root` 用户和 `root` 用户组，也可以是管理员所选择的其他用户或组。如果内容所有者是 `root` 用户和 `root` 用户组，则文件必须可被其他用户读取。所有文件和目录的 SELinux 上下文必须为 `httpd_sys_content_t`，其默认应用于 `/var/www` 目录中的所有内容。

验证步骤

- 使用 Web 浏览器连接到 `http://my_company.idm.example.com/` 或 `http://server_IP/`。如果 `/var/www/html/` 目录为空，或者不包含 `index.html` 或 `index.htm` 文件，则 Apache 会显示 **Red Hat Enterprise Linux 测试页面**。如果 `/var/www/html/` 包含具有不同名称的 HTML 文件，您可以通过输入 URL 到该文件来加载这些文件，如 `http://server_IP/example.html` 或 `http://my_company.idm.example.com/example.html`。

其他资源

- Apache 手册：[安装 Apache HTTP 服务器手册](#)。
- 请参见 `httpd.service(8)` 手册页。

22.7. 在 APACHE HTTP 服务器中添加 TLS 加密

您可以对 `idm.example.com` 域的 `my_company.idm.example.com` Apache HTTP 服务器启用 TLS 加密。

先决条件

- `my_company.idm.example.com` Apache HTTP 服务器已安装并运行。
- 您已从 `webserver-ca` 子 CA 获取了 TLS 证书，并将其存储在 `/etc/pki/tls/certs/httpd.pem` 文件中，如[使用 certmonger 获取服务的 IdM 证书](#) 中所述。如果您使用其他路径，请调整该流程的对应步骤。
- 对应的私钥存储在 `/etc/pki/tls/private/httpd.key` 文件中。如果您使用其他路径，请调整该流程的对应步骤。
- `webserver-ca` CA 证书存储在 `/etc/pki/tls/private/sub-ca.crt` 文件中。如果您使用其他路径，请调整该流程的对应步骤。
- 客户端和 `my_company.idm.example.com` Web 服务器会将服务器的主机名解析为 web 服务器的 IP 地址。
- 如果服务器运行 RHEL 9.2 或更高版本，并且启用了 FIPS 模式，则客户端必须支持 Extended Master Secret(EMS)扩展或使用 TLS 1.3。没有 EMS 的 TLS 1.2 连接会失败。如需更多信息，请参阅[强制 TLS 扩展"Extended Master Secret"](#) 知识库文章。

步骤

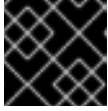
1. 安装 `mod_ssl` 软件包：

```
# dnf install mod_ssl
```

- 编辑`/etc/httpd/conf.d/ssl.conf`文件，并将以下设置添加到 `<VirtualHost _default_:443>` 指令中：

- 设置服务器名称：

```
ServerName my_company.idm.example.com
```



重要

服务器名称必须与证书的 **Common Name** 字段中设置的条目匹配。

- 可选：如果证书在 **Subject Alt Names (SAN)** 字段中包含额外的主机名，您可以配置 `mod_ssl` 来为这些主机名提供 TLS 加密。要配置此功能，请添加具有对应名称的 `ServerAliases` 参数：

```
ServerAlias www.my_company.idm.example.com
server.my_company.idm.example.com
```

- 设置到私钥、服务器证书和 CA 证书的路径：

```
SSLCertificateKeyFile "/etc/pki/tls/private/httpd.key"
SSLCertificateFile "/etc/pki/tls/certs/httpd.pem"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

- 出于安全考虑，配置成只有 **root** 用户才可以访问私钥文件：

```
# chown root:root /etc/pki/tls/private/httpd.key
# chmod 600 //etc/pki/tls/private/httpd.key
```



警告

如果私钥被设置为可以被未授权的用户访问，则需要撤销证书，然后再创建一个新私钥并请求一个新证书。否则，TLS 连接就不再安全。

- 如果您使用 **firewalld**，请在本地防火墙中打开端口 **443**：

```
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

- 重启 **httpd** 服务：

```
# systemctl restart httpd
```



注意

如果您使用密码来保护私钥文件，则必须在每次 **httpd** 服务启动时都输入此密码。

- 使用浏览器并连接到 https://my_company.idm.example.com。

其他资源

- [SSL/TLS 加密](#)。
- [RHEL 8 中 TLS 的安全注意事项](#)

22.8. 在 APACHE HTTP 服务器中设置支持的 TLS 协议版本

默认情况下，RHEL 上的 Apache HTTP 服务器使用定义了安全默认值的系统范围的加密策略，这些值也与最新的浏览器兼容。例如，**DEFAULT**策略定义了只在 apache 中只启用 **TLSv1.2**和**TLSv1.3**协议版本。

您可以手动配置 my_company.idm.example.com Apache HTTP 服务器支持哪个 TLS 协议版本。如果您的环境只需要启用特定的 TLS 协议版本，请按照以下步骤操作，例如：

- 如果您的环境要求客户端也可以使用弱 **TLS1** (TLSv1.0)或**TLS1.1**协议。
- 如果你想将 Apache 配置为只支持**TLSv1.2**或**TLSv1.3**协议。

先决条件

- 在 my_company.idm.example.com 服务器上启用了 TLS 加密，如 [向 Apache HTTP 服务器添加 TLS 加密](#) 中所述。
- 如果服务器运行 RHEL 9.2 或更高版本，并且启用了 FIPS 模式，则客户端必须支持 Extended Master Secret(EMS)扩展或使用 TLS 1.3。没有 EMS 的 TLS 1.2 连接会失败。如需更多信息，请参阅 [强制 TLS 扩展"Extended Master Secret"](#) 知识库文章。

步骤

1. 编辑 `/etc/httpd/conf/httpd.conf` 文件，并将以下设置添加到您要为其设置 TLS 协议版本的`<VirtualHost>`指令中。例如，只启用**TLSv1.3**协议：

```
SSLProtocol -All TLSv1.3
```

2. 重启httpd服务：

```
# systemctl restart httpd
```

验证步骤

1. 使用以下命令来验证服务器是否支持**TLSv1.3**:

```
# openssl s_client -connect example.com:443 -tls1_3
```

2. 使用以下命令来验证服务器是否不支持**TLSv1.2**：

```
# openssl s_client -connect example.com:443 -tls1_2
```

如果服务器不支持该协议，命令会返回一个错误：

```
140111600609088:error:1409442E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol
version:ssl/record/rec_layer_s3.c:1543:SSL alert number 70
```

3. 可选：重复用于其他 TLS 协议版本的命令。

其他资源

- [update-crypto-policies\(8\)](#) 手册页
- [使用系统范围的加密策略。](#)
- 有关 `SSLProtocol` 参数的详情，请参考 Apache 手册中的 `mod_ssl` 文档：[安装 Apache HTTP 服务器手册](#)。

22.9. 在 APACHE HTTP 服务器中设置支持的密码

默认情况下，Apache HTTP 服务器使用定义了安全默认值的系统范围的加密策略，这些值也与最新的浏览器兼容。有关系统范围加密允许的密码列表，请查看 `/etc/crypto-policies/back-ends/openssl.config` 文件。

您可以手动配置 `my_company.idm.example.com` Apache HTTP 服务器支持哪些密码。如果您的环境需要特定的加密系统，请按照以下步骤操作。

先决条件

- 在 `my_company.idm.example.com` 服务器上启用了 TLS 加密，如 [向 Apache HTTP 服务器添加 TLS 加密](#) 中所述。

步骤

1. 编辑 `/etc/httpd/conf/httpd.conf` 文件，并将 `SSLCipherSuite` 参数添加到您要为其设置 TLS 密码的 `<VirtualHost>` 指令中：

```
SSLCipherSuite
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!SHA1:!SHA256"
```

这个示例只启用 `EECDH+AESGCM`、`EDH+AESGCM`、`AES256+EECDH` 和 `AES256+EDH` 密码，并禁用所有使用 `SHA1` 和 `SHA256` 消息身份验证码 (MAC) 的密码。

2. 重启 `httpd` 服务：

```
# systemctl restart httpd
```

验证步骤

1. 显示 Apache HTTP 服务器支持的密码列表：
 - a. 安装 `nmap` 软件包：

```
# dnf install nmap
```

- b. 使用 `nmap` 工具来显示支持的加密：

```
# nmap --script ssl-enum-ciphers -p 443 example.com
...
PORT      STATE SERVICE
443/tcp   open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
...
```

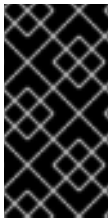
其他资源

- [update-crypto-policies\(8\) 手册页](#)
- [使用系统范围的加密策略。](#)
- [安装 Apache HTTP 服务器手册 - SSLCipherSuite](#)

22.10. 配置 TLS 客户端证书身份验证

客户端证书身份验证可让管理员只允许使用证书进行身份验证的用户访问

`my_company.idm.example.com` Web 服务器上的资源。您可以为 `/var/www/html/Example/` 目录配置客户端证书身份验证。



重要

如果 `my_company.idm.example.com` Apache 服务器使用 TLS 1.3 协议，则某些客户端需要额外的配置。例如，在 Firefox 中，将 `about:config` 菜单中的 `security.tls.enable_post_handshake_auth` 参数设置为 `true`。详情请查看 [Red Hat Enterprise Linux 8 中的传输层安全版本 1.3](#)。

先决条件

- 在 `my_company.idm.example.com` 服务器上启用了 TLS 加密，如 [向 Apache HTTP 服务器添加 TLS 加密](#) 中所述。

步骤

1. 编辑 `/etc/httpd/conf/httpd.conf` 文件，并将以下设置添加到你要为其配置客户端验证的 `<VirtualHost>` 指令中：

```
<Directory "/var/www/html/Example/">
    SSLVerifyClient require
</Directory>
```

`SSLVerifyClient require` 设置定义了服务器必须成功验证客户端证书，然后客户端才能访问 `/var/www/html/Example/` 目录中的内容。

2. 重启 `httpd` 服务：

```
# systemctl restart httpd
```


验证步骤

1. 使用 **curl** 在没有客户端身份验证的情况下访问 **https://my_company.idm.example.com/Example/** URL :

```
$ curl https://my_company.idm.example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 alert
certificate required, errno 0
```

这个错误表示 **my_company.idm.example.com** web 服务器需要客户端证书身份验证。

2. 将客户端私钥和证书以及 CA 证书传递给**curl**以便使用客户端身份验证来访问相同的URL :

```
$ curl --cacert ca.crt --key client.key --cert client.crt
https://my_company.idm.example.com/Example/
```

如果请求成功, **curl**会显示存储在**/var/www/html/Example/**目录中的**index.html**文件。

其他资源

- [安装 Apache HTTP 服务器手册 - mod_ssl 配置](#)

22.11. 请求新的用户证书并将其导出到客户端

作为 Identity Management(IdM)管理员, 您可以配置在 IdM 客户端上运行的 web 服务器, 要求在使用 Web 浏览器访问服务器时使用特定 IdM 子 CA 发布的证书进行身份验证。按照以下流程, 从特定的 IdM 子 CA 请求用户证书, 并将主机上的证书和对应的私钥导出到用户希望使用 Web 浏览器访问 Web 服务器的主机上。之后, [将证书和私钥导入到浏览器](#)。

步骤

1. (可选) 创建一个新目录, 如 **~/certdb/**, 并使其成为临时证书数据库。当被要求时, 创建一个 NSS 证书 DB 密码来加密在后续步骤中生成的证书的密钥 :

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. 创建证书签名请求(CSR)并将输出重定向到文件中。例如, 要为 **IDM.EXAMPLE.COM** 域中的 **idm_user** 用户创建一个名称为 **certificate_request.csr** 的 4096 位 CSR, 请将证书私钥的昵称设为 **idm_user** 以便于查找, 并将主题设为 **CN=idm_user,O=IDM.EXAMPLE.COM** :

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s "CN=idm_user,O=IDM.EXAMPLE.COM"
> certificate_request.csr
```

3. 提示时, 输入您在使用 **certutil** 创建临时数据库时输入相同的密码。然后继续随机键入直到被告知停止 :

```
Enter Password or Pin for "NSS Certificate DB":
```

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

- 将证书请求文件提交到服务器。指定要与新签发的证书关联的 Kerberos 主体，输出文件来存储证书，以及证书配置集（可选）。指定要发布证书的 IdM 子 CA。例如，要为 **idm_user@IDM.EXAMPLE.COM** 主体从 **webclient-ca** 获取 **IECUserRoles** 配置集的证书（这个配置集带有添加了用户角色扩展），并将证书保存到 **~/idm_user.pem** 文件中：

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --ca=webclient-ca --certificate-out=~/idm_user.pem
```

- 将证书添加到 NSS 数据库。使用 **-n** 选项设置您在之前创建 CSR 时使用的相同的 nickname，以便证书与 NSS 数据库中的私钥匹配。**-t** 选项设置信任级别。详情请查看 `certutil(1)man` page。**-i** 选项指定输入证书文件。例如，要将带有在 **~/idm_user.pem** 文件中定义的 **idm_user** 别名的证书添加到 **~/certdb/** 数据库的 NSS 数据库：

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

- 验证 NSS 数据库中的密钥没有显示 (**orphan**) 作为其 nickname。例如，验证存储在 **~/certdb/** 数据库中的证书是否为孤立：

```
# certutil -K -d ~/certdb/
< 0> rsa 5ad14d41463b87a095b1896cf0068ccc467df395 NSS Certificate
DB:idm_user
```

- 使用 **pk12util** 命令将证书从 NSS 数据库导出到 PKCS12 格式。例如，要将来自 **/root/certdb** NSS 数据库的带有 **idm_user** 别名的证书导出到 **~/idm_user.p12** 文件中：

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

- 将证书传递给您要启用 **idm_user** 的证书身份验证的主机：

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

- 因为安全原因，在证书传输到的主机上，将存储 **.pkcs12** 文件的目录的访问权限设置为 **'other'** 组不能访问它：

```
# chmod o-rwx /home/idm_user/
```

- 为安全起见，请从服务器中删除临时 NSS 数据库和 **.pkcs12** 文件：

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

22.12. 配置浏览器以启用证书身份验证

要在使用 WebUI 登录到 Identity Management(IdM)时，可以使用证书进行身份验证，您需要将该用户和相关证书颁发机构(CA)证书导入到 Mozilla Firefox 或 Google Chrome 浏览器。运行浏览器的主机本身不需要是 IdM 域的一部分。

IdM 支持以下浏览器连接到 Web UI：

- Mozilla Firefox 38 及更新的版本
- Google Chrome 46 及更新的版本

以下流程演示了如何配置 Mozilla Firefox 57.0.1 浏览器。

先决条件

- 您有要导入到浏览器的用户证书（采用 PKCS#12 格式）。
- 您已下载了子 CA 证书，并使其处于 PEM 格式。

步骤

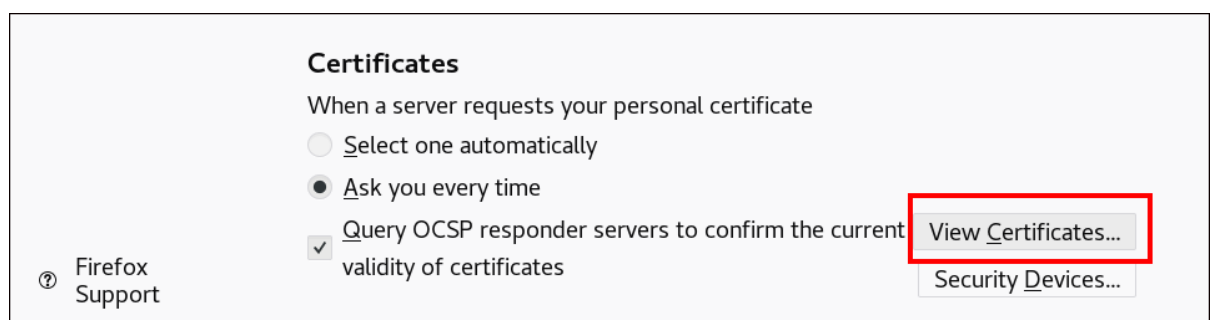
1. 打开 Firefox，进入 **Preferences** → **Privacy & Security**。

图 22.8. 首选项中的隐私和安全部分



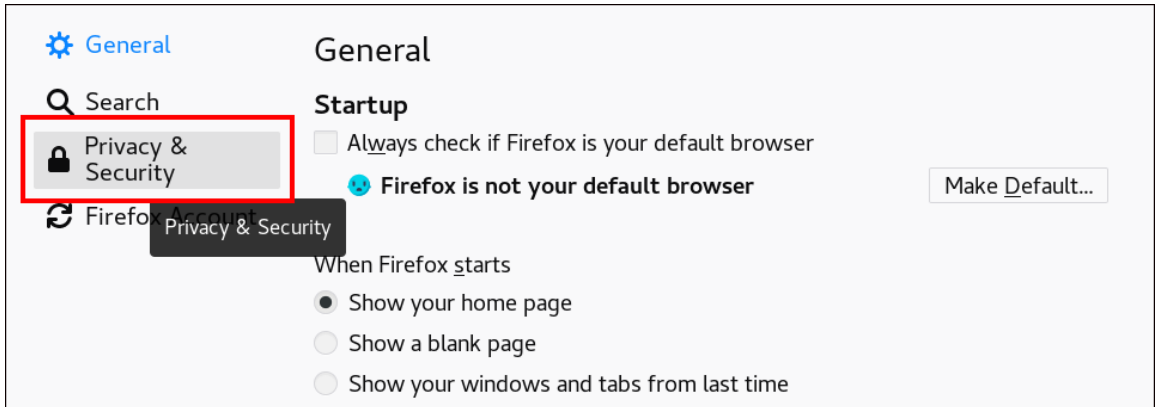
2. 点查看证书。

图 22.9. 查看隐私和安全中的证书



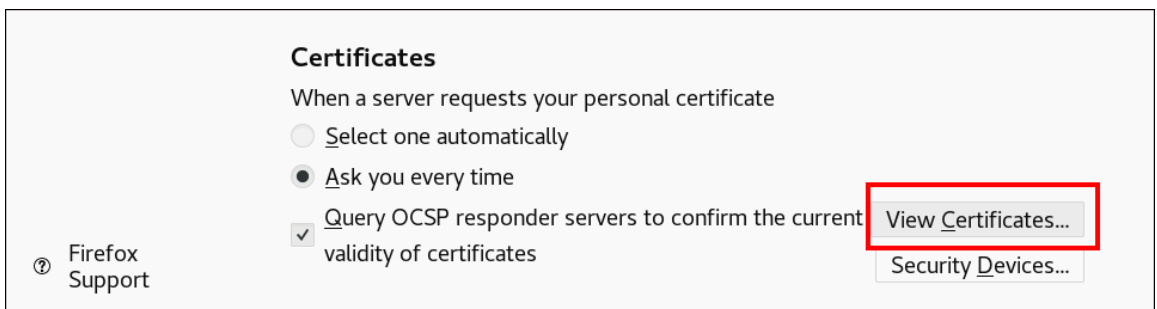
3. 在 **Your Certificates** 选项卡中，点 **Import**。查找并打开用户证书（PKCS12 格式），然后点 **OK** 和 **OK**。
4. 要确保您的 IdM 子 CA 被 Firefox 识别为可信颁发机构，请导入您在 [从 IdM Web UI 下载子 CA 证书](#) 中作为可信证书颁发机构证书保存的证书：
 - a. 打开 Firefox，导航到首选项并单击 **隐私和安全**。

图 22.10. 首选项中的隐私和安全部分



- b. 点查看证书。

图 22.11. 查看隐私和安全中的证书

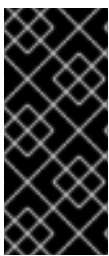


- c. 在 **Authorities** 选项卡中，点 **Import**。找到并打开 sub-CA 证书。信任证书来识别网站，然后点 **OK** 和 **OK**。

第 23 章 快速地使相关证书的特定组无效

作为系统管理员，如果您想要快速地使相关证书的特定组无效：

- 设计应用程序，以便它们仅信任由特定轻量级身份管理(IdM)子 CA 发布的证书。之后，您可以通过撤销发布这些证书的 Identity Management(IdM)子 CA 的证书，使所有这些证书无效。有关如何在 IdM 中创建和使用轻量级子 CA 的详情，请参考[快速地使特定的相关证书组无效](#)。
- 为确保由撤销的 IdM 子 CA 发布的所有证书都立即无效，请配置依赖此类证书的应用程序使用 IdM OCSP 响应器。例如，要将 Firefox 浏览器配置为使用 OCSP 响应器，请确保 **Query OCSP 响应器服务器确认证书复选框的当前有效期**（在 Firefox 首选项中检查）。在 IdM 中，证书吊销列表(CRL)每四个小时更新一次。要使 IdM 子 CA 发布的所有证书无效，请参阅[吊销 IdM 子 CA 证书](#)。此外，[禁用相关的 CA ACL](#)，并考虑[禁用 IdM 子 CA](#)。禁用子 CA 可防止子 CA 发布新证书，但会允许证书状态协议 (OCSP) 对之前发布的证书进行响应，这是因为子 CA 的签名密钥被保留。



重要

如果您在环境中使用 OCSP，请不要删除子 CA。删除子 CA 会删除子 CA 的签名密钥，防止对子 CA 发布的证书的 OCSP 响应进行生产环境。

删除子 CA 好于禁用它的唯一场景是，您希望创建一个新的 sub-CA，它具有相同对象可区分名称(DN)但需要使用一个新的签名密钥。

23.1. 在 IDM CLI 中禁用 CA ACL

当您要停用 IdM 服务或 IdM 服务组时，请考虑禁用任何现有的对应的 CA ACL。

按照以下流程禁用 [TLS_web_server_authentication](#) CA ACL，其限制运行在 IdM 客户端上的 Web 服务器请求由 **webserver-ca** IdM 子 CA 发布的证书，并禁用 [TLS_web_client_authentication](#) CA ACL，其限制 IdM 用户请求由 **webclient-ca** IdM 子 CA 发布的用户证书。

步骤

1. 另外，要在 IdM 环境中查看所有 CA ACL，请输入 **ipa caacl-find** 命令：

```
$ ipa caacl-find
-----
3 CA ACLs matched
-----
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE

ACL name: TLS_web_server_authentication
Enabled: TRUE

ACL name: TLS_web_client_authentication
Enabled: TRUE
```

2. 另外，要查看 CA ACL 的详情，请输入 **ipa caacl-show** 命令并指定 CA ACL 名称。

```
$ ipa caacl-show TLS_web_server_authentication
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates issued
```

```
by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Profiles: calPAserviceCert
Services: HTTP/rhel8server.idm.example.com@IDM.EXAMPLE.COM
```

3. 要禁用 CA ACL，输入 **ipa caacl-disable** 命令，并指定 CA ACL 名称。

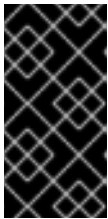
- 要禁用 **TLS_web_server_authentication** CA ACL，请输入：

```
$ ipa caacl-disable TLS_web_server_authentication
-----
Disabled CA ACL "TLS_web_server_authentication"
-----
```

- 要禁用 **TLS_web_client_authentication** CA ACL，请输入：

```
$ ipa caacl-disable TLS_web_client_authentication
-----
Disabled CA ACL "TLS_web_client_authentication"
-----
```

唯一启用的 CA ACL 现在是 **hosts_services_calPAserviceCert** CA ACL。



重要

请注意，禁用 **hosts_services_calPAserviceCert** CA ACL。如果禁用了 **hosts_services_calPAserviceCert**，且没有其他使用带有 **calPAserviceCert** 配置集的 **ipa** CA 的 CA ACL 授权 IdM 服务器时，在续订 IdM **HTTP** 和 **LDAP** 证书时会失败。已过期的 IdM **HTTP** 和 **LDAP** 证书最终将导致 IdM 系统失败。

23.2. 禁用 IDM 子 CA

在撤销 IdM 子 CA 的 CA 证书以使该子 CA 发布的所有证书无效后，如果您不再需要 IdM 子 CA，请考虑禁用它。您可以稍后重新启用 sub-CA。

禁用子 CA 可防止子 CA 发布新证书，但会允许线证书状态协议 (OCSP) 对之前发布的证书进行响应，这是因为子 CA 的签名密钥被保留。

先决条件

- 以 IdM 管理员身份登录。

步骤

- 输入 **ipa ca-disable** 命令并指定 sub-CA 的名称：

```
$ ipa ca-disable webserver-CA
-----
Disabled CA "webserver-CA"
-----
```

第 24 章 使用 IDM HEALTHCHECK 验证证书

了解更多有关理解和使用身份管理(IdM)中的 Healthcheck 工具，以识别由 **certmonger** 维护的 IPA 证书的问题。

详情请参阅 [IdM 中的 Healthcheck](#)。

24.1. IDM 证书 HEALTHCHECK 测试

Healthcheck 工具包括几个测试，用于验证 Identity Management(IdM)中由 certmonger 维护的证书状态。有关 certmonger 的详情，请参阅[使用 certmonger 为服务获取 IdM 证书](#)。

这个测试套件会检查过期、验证、信任和其他问题。可能会为相同的底层问题抛出多个错误。

要查看所有证书测试，请使用 **--list-sources** 选项运行 **ipa-healthcheck**：

```
# ipa-healthcheck --list-sources
```

您可以在 **ipahealthcheck.ipa.certs** 源中找到所有测试：

IPACertmongerExpirationCheck

此测试会检查 **certmonger** 中的过期时间。
如果报告错误，代表证书已过期。

如果显示警告，代表证书将很快过期。默认情况下，这个测试会在证书过期前的 28 天或更短的天数内应用。

您可以在 **/etc/ipahealthcheck/ipahealthcheck.conf** 文件中配置天数。打开文件后，更改 default 部分中的 **cert_expiration_days** 选项。

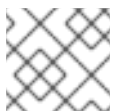


注意

certmonger 加载并维护其证书过期视图。此检查不会验证磁盘上的证书。

IPACertfileExpirationCheck

此测试会检查是否无法打开证书文件或 NSS 数据库。此测试还会检查过期时间。因此，仔细阅读错误或警告输出中的 **msg** 属性。消息指定了问题。



注意

此测试会检查磁盘上的证书。如果缺少证书且不可读取，也会引发单独的错误。

IPACertNSSTrust

此测试会比较 NSS 数据库中存储的证书的信任。对于 NSS 数据库中的预期跟踪证书，信任与预期值进行比较，导致在非匹配时引发错误。

IPANSSChainValidation

此测试会验证 NSS 证书的证书链。测试执行：**certutil -V -u V -e -d [dbdir] -n [nickname]**

IPAOpenSSLChainValidation

此测试会验证 OpenSSL 证书的证书链。为了可以与这里的 **NSSChain** 验证比较，执行 OpenSSL 命令：

-

```
openssl verify -verbose -show_chain -CAfile /etc/ipa/ca.crt [cert file]
```

IPARAAgent

此测试将磁盘上的证书与 `uid=ipara,ou=People,o=ipaca` 中的 LDAP 中的等效记录进行比较。

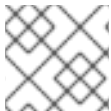
IPACertRevocation

此测试使用 `certmonger` 验证证书没有被撤销。因此，测试只能查找与由 `certmonger` 维护的证书连接的问题。

IPACertmongerCA

此测试会验证 `certmonger` 证书颁发机构(CA)配置。IdM 无法在没有 CA 的情况下发布证书。`certmonger` 维护一组 CA 帮助程序。在 IdM 中，有一个名为 IPA 的 CA，它会在主机或服务证书中以主机或用户主体身份通过 IdM 发出证书。

另外，还有 `dogtag-ipa-ca-renew-agent` 和 `dogtag-ipa-ca-renew-agent-reuse`，它们续订 CA 子系统证书。



注意

当尝试检查问题时，在所有 IdM 服务器中运行这些测试。

24.2. 使用 HEALTHCHECK 工具检查证书

按照以下流程，使用 `Healthcheck` 工具运行身份管理(IdM)证书健康检查的独立的手动测试。

`Healthcheck` 工具包括许多测试，您可以对结果进行简化：

- 排除所有成功的测试: `--failures-only`
- 仅包含证书测试: `--source=ipahealthcheck.ipa.certs`

先决条件

- 您必须以 `root` 用户身份执行 `Healthcheck` 测试。

步骤

- 要运行带有警告的 `Healthcheck`，有关证书的错误和严重问题，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.ipa.certs --failures-only
```

成功测试显示空的括号：

```
[]
```

失败测试显示以下输出：

```
{
  "source": "ipahealthcheck.ipa.certs",
  "check": "IPACertfileExpirationCheck",
  "result": "ERROR",
  "kw": {
```



```
"key": 1234,  
"dbdir": "/path/to/nssdb",  
"error": [error],  
"msg": "Unable to open NSS database '/path/to/nssdb': [error]"  
}  
}
```

这个 `IPACertfileExpirationCheck` 测试在打开 NSS 数据库时失败。

其他资源

- 请参阅 `man ipa-healthcheck`。

第 25 章 使用 IDM HEALTHCHECK 验证系统证书

了解如何使用 Healthcheck 工具识别身份管理(IdM)中系统证书的问题。

详情请查看

[IdM 中的健康检查。](#)

25.1. 系统证书健康检查测试

Healthcheck 工具包括多个用于验证系统(DogTag)证书的测试。

要查看所有测试，请使用 `--list-sources` 选项运行 `ipa-healthcheck`:

```
# ipa-healthcheck --list-sources
```

您可以在 `ipahealthcheck.dogtag.ca` 源中找到所有测试：

DogtagCertsConfigCheck

此测试会将其 NSS 数据库中的 CA(Certificate Authority)证书与 `CS.cfg` 中存储的相同值进行比较。如果不匹配，CA 无法启动。

特别是，它会检查：

- `auditSigningCert cert-pki-ca` against `ca.audit_signing.cert`
- `ocspSigningCert cert-pki-ca` against `ca.ocsp_signing.cert`
- `caSigningCert cert-pki-ca` against `ca.signing.cert`
- `subsystemCert cert-pki-ca` against `ca.subsystem.cert`
- `Server-Cert cert-pki-ca` 与 `ca.sslserver.cert`

如果安装了密钥恢复授权(KRA)：

- `transportCert cert-pki-kra` against `ca.connector.KRA.transportCert`

DogtagCertsConnectivityCheck

此测试会验证连接。此测试等同于 `ipa cert-show 1` 命令，它检查：

- Apache 中的 PKI 代理配置
- IdM 可以找到 CA
- RA 代理客户端证书
- CA 回复请求的更正

请注意，测试会检查带有串行 #1 的证书，因为您想要验证证书是否可以被执行，并从 CA 返回预期的结果（证书或未找到证书）。



注意

当尝试查找问题时，在所有 IdM 服务器中运行这些测试。

25.2. 使用 HEALTHCHECK 输出系统证书

按照以下流程，使用 Healthcheck 工具运行身份管理(IdM)证书的独立的手动测试。

由于 Healthcheck 工具包括许多测试，您可以通过只包括 DogTag 测试：`--source=ipahealthcheck.dogtag.ca` 来缩小结果范围

步骤

- 要运行 Healthcheck 限制为 DogTag 证书，请输入：

```
# ipa-healthcheck --source=ipahealthcheck.dogtag.ca
```

成功测试示例：

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: SUCCESS",
  "uuid: 9b366200-9ec8-4bd9-bb5e-9a280c803a9c",
  "when: 20191008135826Z",
  "duration: 0.252280",
  "kw:" {
    "key": "Server-Cert cert-pki-ca",
    "configfile": "/var/lib/pki/pki-tomcat/conf/ca/CS.cfg"
  }
}
```

一个失败的测试示例：

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: CRITICAL",
  "uuid: 59d66200-1447-4b3b-be01-89810c803a98",
  "when: 20191008135912Z",
  "duration: 0.002022",
  "kw:" {
    "exception": "NSDB /etc/pki/pki-tomcat/alias not initialized",
  }
}
```

其他资源

- 请参阅 `man ipa-healthcheck`。

第 26 章 了解 IDM 内部使用的证书

您可以安装带有集成证书颁发机构(CA)或没有 CA 的 Red Hat Identity Management (IdM)服务器。访问和管理 IdM 所需的证书会根据您的 CA 是否是集成的而不同：

- 集成的 CA：证书由 **certmonger** 自动创建和跟踪。**certmonger** 会自动续订证书，确保 IdM 服务持续有效。
- 没有 CA：证书是从第三方授权请求的。在这种情况下，您需要监控其过期，并确保它们被续订，以确保 IdM 服务的持续有效。

26.1. 关于 IDM 中的内部证书

Red Hat Identity Management (IdM)使用许多使用网络访问的服务，包括 LDAP 服务器和 HTTP 服务器。您可以使用 SSL/TLS 端口访问这些服务，该端口需要服务器证书。在安装 IdM 服务器的过程中需要 HTTP 和 LDAP 服务器证书。

您可以根据您安装和配置 IdM 的方式以多种方式获取证书：

- 集成的 CA 可以是自签名的或由外部 CA 签名的：IdM 会为由 IdM 管理的用户、主机和服务发出所有证书，您不需要提供证书文件。**certmonger** 会自动监控证书的到期日期，它们在需要时会自动续订。
- 使用外部签名的 CA：安装是一个多个步骤的过程。
 - 您需要使用 **--external-ca** 选项运行安装来生成 CSR。
 - 将 CSR 提交给外部 CA，并以 PEM 文件或 Base64 编码证书的形式检索发布的证书和 CA 证书链。
 - 再次运行 IdM 服务器安装，指定新发布的 CA 证书和 CA 链文件的位置和名称。您的 IdM 证书颁发机构被配置为外部 CA 的子 CA，这个子 CA 发布所需的 HTTP 和 LDAP 服务器证书。**certmonger** 会自动监控证书的到期日期，它们在需要时会自动续订。
- 没有 CA：要求您从第三方认证机构请求以下证书：
 - LDAP 服务器证书
 - Apache 服务器证书
 - PKINIT 证书
 - 发布 LDAP 和 Apache 服务器证书的 CA 完整 CA 证书链
这些证书不会被 **certmonger** 跟踪，管理员负责在它们过期日期之前续订证书。

其他资源

- [规划您的 CA 服务。](#)

26.2. IDM 内部的证书

您的内部证书可以取决于您是如何安装 IdM 的，以及该安装中包含了哪些组件。根据该安装，您可能在您的系统上存储了以下证书。

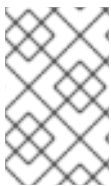
IdM CA 证书

IdM 使用 IdM CA 证书为所有其他证书签名。请注意，它没有出现在 CA-less 安装中。

caSigningCert	描述
文件系统位置	<ul style="list-style-type: none"> • <code>/etc/pki/pki-tomcat/alias</code> NSS 数据库中的 <code>nickname=caSigningCert cert-pki-ca</code> • <code>/etc/ipa/nssdb/</code> 和 <code>/etc/ipa/ca.crt</code> 中的 <code>nickname=REALM.NAME IPA CA</code> (从 LDAP 填充)
LDAP 位置	cn=REALM.NAME IPA CA,cn=certificates,cn=ipa,cn=etc,dc=realm,dc=name 和 ou=authorities,ou=ca,o=ipaca
发布者	由外部 CA 自签名或签名
主题	O = REALM.NAME, CN = Certificate Authority <p>请注意，这是默认值，但可以在 IdM 服务器安装过程中自定义。</p>
附加信息	必须具有 CA:true 关键约束，且必须在 NSS 数据库中有 CT,C,C 信任标记。

外部 CA 证书

如果您使用外部 CA，则 IdM 中必须有外部 CA 链来验证 IdM 证书。对于无 CA 的安装，外部 CA 证书必须存在于不同的位置，包括 LDAP 和 `/etc/ipa/ca.crt` 目录中，以验证 HTTPD 和 LDAP 证书。



注意

您不必手动将外部 CA 证书添加到所有所需的位置，因为这在安装过程中自动完成。但是，如果以后更新了外部 CA 证书，您应该按照 [使用外部 CA 重新更新 IdM CA 续订服务器证书](#) 中的步骤，以确保将新证书添加到每个需要的位置。

外部证书	描述
文件系统位置	<code>/etc/pki/pki-tomcat/alias nssdb</code> ，并作为 <code>/etc/ipa/ca.crt</code> 中的链的一部分 (从 LDAP 填充)
LDAP 位置	cn=SUBJECT,cn=certificates,cn=ipa,cn=etc,dc=realm,dc=name 和 ou=authorities,ou=ca,o=ipaca
发布者	外部 CA 签名的
主题	外部 CA 主题

外部证书	描述
附加信息	您必须在链中有 DER 格式的所有证书，您必须将它们导入到 LDAP 中。在 NSS 数据库中必须有 CT,C,C 信任标记。

子系统 CA 证书

此证书用于在写入 LDAP 数据库时向 LDAP 服务器进行身份验证。在无 CA 的安装中没有此证书。

subsystemCert	描述
文件系统位置	nickname=subsystemCert cert-pki-ca in /etc/pki/pki-tomcat/alias nssdb
LDAP 位置	uid=pkidbuser,ou=people,o=ipaca
发布者	IPA CA
主题	CN=CA Subsystem,O=REALM.NAME
附加信息	注意 LDAP 中的序列号和 Blob 不匹配。例如， 2;SERIAL;CN=Certificate Authority,O=REALM.NAME;CN=CA Subsystem,O=REALM.NAME 和 userCertificate 必须与文件系统上的证书匹配。

审计签名证书

此证书用于签名审计日志。请注意，它没有出现在 CA-less 安装中。

auditSigningCert	描述
文件系统位置	nickname=auditSigningCert cert-pki-ca in /etc/pki/pki-tomcat/alias nssdb
LDAP 位置	没有专用的 LDAP 位置，通过 ou=certificateRepository,ou=ca,o=ipaca 共享
发布者	IPA CA
主题	CN=CA Audit,O=REALM.NAME
附加信息	在 NSS 数据库中必须有 „P 信任标记。

OCSP 签名证书

此证书用于提供在线证书状态协议(OCSP)服务。请注意，它没有出现在 CA-less 安装中。

ocspSigningCert	描述
文件系统位置	nickname=ocspSigningCert cert-pki-ca in /etc/pki/pki-tomcat/alias nssdb
LDAP 位置	没有专用的 LDAP 位置，通过 ou=certificateRepository,ou=ca,o=ipaca 共享
发布者	IPA CA
主题	CN=OCSP Subsystem,O=REALM.NAME
附加信息	

Tomcat servlet 证书

当客户端联系 PKI 时，使用此证书。请注意，这个服务器证书特定于主机，它不会出现在无 CA 的安装中。

Server-Cert	描述
文件系统位置	<ul style="list-style-type: none"> nickname = /etc/pki/pki-tomcat/alias nssdb+ 中的 Server-Cert cert-pki-ca
LDAP 位置	
发布者	IPA CA
主题	CN=\$HOSTNAME,O=REALM.NAME
附加信息	

注册颁发机构证书

certmonger 以及 IdM 框架用来对 PKI 进行身份验证的证书。例如，如果您运行 **ipa cert-show 1**，则 HTTPD 使用此证书与 PKI 进行通信和身份验证。没有出现在无 CA 的安装中。

RA 代理	描述
文件系统位置	/var/lib/ipa/ra-agent.pem (在 RHEL 7.4 之前位于 /etc/httpd/alias 中)
LDAP 位置	uid=ipara,ou=people,o=ipaca
发布者	IPA CA

RA 代理	描述
主题	CN=IPA RA,O=REALM.NAME
附加信息	注意 LDAP 中的序列号和 Blob 不匹配。例如, 2;SERIAL;CN=Certificate Authority,O=REALM.NAME;CN=IPA RA,O=REALM.NAME 和 userCertificate 必须与文件系统上的证书匹配。

HTTPD 前端证书

用于 HTTPD 前端的证书, 来保护与 Web UI 和 API 的连接。必须存在。

HTTPD	描述
文件系统位置	/var/lib/ipa/certs/httpd.crt (RHEL 8 之前在 /etc/httpd/alias 中)
LDAP 位置	
发布者	无 CA 安装中的 IPA CA 或外部 CA
主题	CN=\$HOSTNAME,O=REALM.NAME
附加信息	必须包含一个主体名称为 otherName = 1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/\$HOSTNAME@REALM , DNS name = \$HOSTNAME 的 Certificate Subject Alt Name 扩展

LDAP TLS 和 STARTTLS 证书

用于 LDAP TLS 和 STARTTLS 连接的证书。必须存在。

LDAP	描述
文件系统位置	/etc/dirsrv/slapd-DOMAIN NSS 数据库中的 nickname=Server-Cert (可以是其他昵称, 与 dse.ldif 中的 nsSSLPersonalitySSL 匹配)
LDAP 位置	
发布者	无 CA 安装中的 IPA CA 或外部 CA
主题	CN=\$HOSTNAME,O=REALM.NAME

LDAP	描述
附加信息	必须包含一个主体名称为 otherName = 1.3.6.1.4.1.311.20.2.3;UTF8:ldap/\$HOSTNAME@REALM, DNS name = \$HOSTNAME 的 Certificate Subject Alt Name 扩展。

KDC 证书

用于 IdM KDC 的 PKINIT 的证书。

KDC	描述
文件系统位置	/var/kerberos/krb5kdc/kdc.crt
LDAP 位置	
发布者	无 CA 安装中的 IPA CA 或外部 CA
主题	CN=\$HOSTNAME,O=REALM.NAME
附加信息	必须有扩展的密钥使用 id-pkinit-KPkdc (1.3.6.1.5.2.3.5) , 主体名称为 otherName = 1.3.6.1.4.1.311.20.2.3;UTF8:krbtgt/REALM@REALM, DNS name = \$HOSTNAME .

26.3. IDM 内部证书续订过程

默认情况下, **certmonger** 跟踪内部证书, 触发续订并请求 IdM CA 发布新证书。

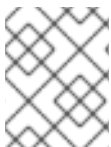
如果您使用外部 CA, 且您的内部证书由这个 CA 发布, 则它们不会自动续订。在这种情况下, 您应该监控证书的过期日期, 以确保在过期前续订它们。续订过程会非常耗时, 如果您没有仔细跟踪过期日期, 您的证书将过期, 某些服务将不再可用。



警告

如果您的内部 Red Hat Identity Management (IdM) 证书过期了, 则 IdM 无法启动。

IdM CA 续订服务器在过期日期前 28 天续订共享的内部证书。**certmonger** 会触发此续订, 并将新证书上传到 **cn=<nickname>,cn=ca_renewal,cn=ipa,cn=etc,\$BASEDN**。**certmonger** 还触发其他 IdM 服务器上的续订过程, 但它是在非 CA 续订服务器上执行的, 它不会请求新证书, 但会从 LDAP 下载证书。请注意, **Server-Cert cert-pki-ca**、HTTP、LDAP 和 PKINIT 证书特定于每个副本, 其中在主题中包含主机名。



注意

如果您在证书过期前使用 **getcert** 手动续订共享证书，则不会在其他副本上触发续订过程，且您必须在其他副本上运行 **getcert**，以执行从 LDAP 下载更新的证书。

26.4. 其他资源

- [使用 IdM CA 续订服务器](#)
- [IdM 离线时续订过期的系统证书](#)
- [如果 web 服务器和 LDAP 服务器证书还没有在 IdM 副本上过期，请替换它们](#)
- [如果 web 服务器和 LDAP 服务器证书已在整个 IdM 部署中过期，请替换它们](#)