[Oracle Homeage](http://www.oracle.com/)test

[Documentation Home](http://www.oracle.com/technetwork/indexes/documentation/index.html)  > [系统管理指南：安全性服务](https://docs.oracle.com/cd/E19253-01/819-7061/index.html)  > [第 6 部分 Kerberos 服务](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2vak/index.html)  > 第 26 章 Kerberos 服务（参考）

系统管理指南：安全性服务

* [*Previous*: 第 25 章 使用 Kerberos 应用程序（任务）](https://docs.oracle.com/cd/E19253-01/819-7061/user-7/index.html)
* [*Next*: 第 7 部分 Solaris 审计](https://docs.oracle.com/cd/E19253-01/819-7061/audittm-1/index.html)

**第 26 章 Kerberos 服务（参考）**

本章列出了许多属于 Kerberos 产品的文件、命令和守护进程。另外，本章还介绍了有关 Kerberos 验证的工作方式的详细信息。

以下是本章中参考信息的列表。

* [Kerberos 文件](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2vel/index.html)
* [Kerberos 命令](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2vem/index.html)
* [Kerberos 守护进程](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2ven/index.html)
* [Kerberos 术语](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veo/index.html)
* [Kerberos 验证系统的工作方式](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2vep/index.html)
* [使用 Kerberos 获取服务访问权限](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veq/index.html)
* [使用 Kerberos 加密类型](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2ver/index.html)
* [使用 gsscred 表](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2ves/index.html)
* [Solaris Kerberos 和 MIT Kerberos 之间的显著差异](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veu/index.html)

**Kerberos 文件**

表 26–1 Kerberos 文件

| **文件名** | **说明** |
| --- | --- |
| ~/.gkadmin | 用于在 SEAM 管理工具中创建新主体的缺省值 |
| ~/.k5login | 授予 Kerberos 帐户访问权限的主体列表 |
| /etc/krb5/kadm5.acl | Kerberos 访问控制列表文件，其中包含 KDC 管理员的主体名称及其 Kerberos 管理权限 |
| /etc/krb5/kadm5.keytab | 主 KDC 上的 kadmin 服务的密钥表文件 |
| /etc/krb5/kdc.conf | KDC 配置文件 |
| /etc/krb5/kpropd.acl | Kerberos 数据库传播配置文件 |
| /etc/krb5/krb5.conf | Kerberos 领域配置文件 |
| /etc/krb5/krb5.keytab | 网络应用程序服务器的密钥表文件 |
| /etc/krb5/warn.conf | Kerberos 票证到期警告和自动更新配置文件 |
| /etc/pam.conf | PAM 配置文件 |
| /tmp/krb5cc\_*uid* | 缺省凭证高速缓存，其中 *uid* 是用户的十进制 UID |
| /tmp/ovsec\_adm.*xxxxxx* | 口令更改操作生命周期的临时凭证高速缓存，其中 *xxxxxx* 是随机字符串 |
| /var/krb5/.k5.*REALM* | KDC 存储文件，其中包含 KDC 主密钥的加密副本 |
| /var/krb5/kadmin.log | kadmind 的日志文件 |
| /var/krb5/kdc.log | KDC 的日志文件 |
| /var/krb5/principal | Kerberos 主体数据库 |
| /var/krb5/principal.kadm5 | Kerberos 管理数据库，其中包含策略信息 |
| /var/krb5/principal.kadm5.lock | Kerberos 管理数据库锁定文件 |
| /var/krb5/principal.ok | Kerberos 数据库成功初始化时创建的 Kerberos 主体数据库初始化文件 |
| /var/krb5/principal.ulog | Kerberos 更新日志，其中包含增量传播更新 |
| /var/krb5/slave\_datatrans | kprop\_script 脚本用于传播的 KDC 备份文件 |
| /var/krb5/slave\_datatrans\_*slave* | 完全更新指定的 *slave* 时创建的临时转储文件 |

**Kerberos 命令**

本节列出了 Kerberos 产品中包括的部分命令。

表 26–2 Kerberos 命令

| **命令** | **说明** |
| --- | --- |
| /usr/bin/ftp | 文件传输协议程序 |
| /usr/bin/rcp | 远程文件复制程序 |
| /usr/bin/rdist | 远程文件分发程序 |
| /usr/bin/rlogin | 远程登录程序 |
| /usr/bin/rsh | 远程 shell 程序 |
| /usr/bin/telnet | 基于 Kerberos 的 telnet 程序 |
| /usr/lib/krb5/kprop | Kerberos 数据库传播程序 |
| /usr/sbin/gkadmin | Kerberos 数据库管理 GUI 程序，用于管理主体和策略 |
| /usr/sbin/kadmin | 远程 Kerberos 数据库管理程序（运行时需要进行 Kerberos 验证），用于管理主体、策略和密钥表文件 |
| /usr/sbin/kadmin.local | 本地 Kerberos 数据库管理程序（运行时无需进行 Kerberos 验证，并且必须在主 KDC 上运行），用于管理主体、策略和密钥表文件 |
| /usr/sbin/kclient | Kerberos 客户机安装脚本，可用于安装配置文件，也可不用于安装配置文件 |
| /usr/sbin/kdb5\_util | 创建 Kerberos 数据库和存储文件 |
| /usr/sbin/kproplog | 列出更新日志中更新项的摘要 |

**Kerberos 守护进程**

下表列出了 Kerberos 产品使用的守护进程。

表 26–3 Kerberos 守护进程

| **守护进程** | **说明** |
| --- | --- |
| /usr/sbin/in.ftpd | 文件传输协议守护进程 |
| /usr/lib/krb5/kadmind | Kerberos 数据库管理守护进程 |
| /usr/lib/krb5/kpropd | Kerberos 数据库传播守护进程 |
| /usr/lib/krb5/krb5kdc | Kerberos 票证处理守护进程 |
| /usr/lib/krb5/ktkt\_warnd | Kerberos 票证到期警告和自动更新守护进程 |
| /usr/sbin/in.rlogind | 远程登录守护进程 |
| /usr/sbin/in.rshd | 远程 shell 守护进程 |
| /usr/sbin/in.telnetd | telnet 守护进程 |

**Kerberos 术语**

下一节介绍了 Kerberos 术语及其定义。这些术语可用于整个 Kerberos 文档。要理解 Kerberos 概念，必须先了解这些术语。

**特定于 Kerberos 的术语**

要管理 KDC，您需要了解本节中的术语。

**Key Distribution Center, KDC**（密钥分发中心）是负责颁发凭证的 Kerberos 组件。这些凭证是使用 KDC 数据库中存储的信息创建的。每个领域至少需要两个 KDC，一个主 KDC 以及至少一个从 KDC。所有 KDC 都可生成凭证，但仅有主 KDC 才能处理对 KDC 数据库所做的任何更改。

**stash file（存储文件）**包含 KDC 的主密钥。当重新启动服务器以便在启动 kadmind 和 krb5kdc 命令之前自动验证 KDC 时，将使用此密钥。由于此文件包含主密钥，因此应将此文件及其所有备份安全保存。此文件是使用 root 的只读权限创建的。要确保此文件安全，请勿更改相应的权限。如果此文件已被破坏，则其他用户可能会使用主密钥来访问或修改 KDC 数据库。

**特定于验证的术语**

要了解验证过程，您需要理解本节中的术语。程序员和系统管理员应熟悉这些术语。

**client（客户机）**是在用户工作站上运行的软件。在客户机上运行的 Kerberos 软件会在此过程中发出许多请求。因此，区分此软件的操作和用户非常重要。

术语 *server（服务器）*和 *service（服务）*通常可互换使用。具体而言，术语**服务器**用于定义运行 Kerberos 软件的物理系统。术语**服务**对应于服务器支持的特定功能（例如 ftp 或 nfs）。文档通常会将服务器描述为服务的一部分，但此定义会混淆了这些术语的含义。因此，术语*服务器*是指物理系统，而术语*服务*则是指软件。

Kerberos 产品使用两种类型的密钥。 一种密钥类型是口令派生密钥。 口令派生密钥会被指定给每个用户主体，并仅对该用户和 KDC 公开。 Kerberos 产品使用的另一种密钥类型是与口令无关联的随机密钥，因此不适合用户主体使用。 随机密钥通常用于在密钥表中具有相应项并具有 KDC 生成的会话密钥的服务主体。 服务主体可以使用随机密钥，因为服务可以访问密钥表中允许其以非交互方式运行的密钥。 会话密钥由 KDC 生成，并在客户机和服务之间共享，可用于在两者之间提供安全事务。

**ticket（票证）**是一种信息包，用于将用户身份安全地传递到服务器或服务。一个票证仅对一台客户机以及某台特定服务器上的一项特殊服务有效。票证包含以下内容：

* 服务的主体名称
* 用户的主体名称
* 用户主机的 IP 地址
* 时间标记
* 定义票证生命周期的值
* 会话密钥的副本

所有此类数据都使用服务器的服务密钥进行加密。 请注意，KDC 可颁发嵌入在以下介绍的凭证中。 颁发票证之后，可重用票证直到其到期为止。

**credential（凭证）**是一种信息包，其中包含票证和匹配的会话密钥。凭证使用发出请求的主体的密钥进行加密。通常，KDC 会生成凭证以响应客户机的票证请求。

**authenticator（验证者）**是服务器用于验证客户机用户主体的信息。 验证者包含用户的主体名称、时间标记和其他数据。 与票证不同，验证者只能使用一次，通常在请求访问服务时使用。 验证者使用客户机和服务器共享的会话密钥进行加密。 通常，客户机会创建验证者，并将其与服务器或服务的票证一同发送，以便向服务器或服务进行验证。

**票证类型**

票证具有可管理其使用方式的属性。这些属性是在创建票证时指定给票证的，但稍后可修改票证的属性。例如，可将票证从 forwardable 更改为 forwarded。可以使用 klist 命令查看票证属性。请参见[查看 Kerberos 票证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veg/index.html#user-1)。

以下一个或多个术语对票证进行了描述：

Forwardable（可转发）/forwarded（已转发）

可转发票证可以从一台主机发送到另一台主机，从而使客户机无需对其自身进行重新验证。例如，如果用户 david 在用户 jennifer 的计算机上时获取了一个可转发票证，则前者可登录到自己的计算机，而不必获取新的票证（从而对自身进行重新验证）。有关可转发票证的示例，请参见[示例－创建 Kerberos 票证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veg/index.html#user-21)。

Initial（初始）

初始票证是一种直接颁发的票证，而并非基于票证授予票证的票证。某些服务（如用于更改口令的应用程序）可能会要求将票证标记为初始，以便向这些程序本身确保客户机可知道其私钥。初始票证表明客户机最近已进行了自我验证，而并非依赖于已长期使用的票证授予票证。

Invalid（未生效）

无效票证是一个尚未变为可用的未生效的票证。应用程序服务器会拒绝无效票证，直到其经过验证为止。要进行验证，必须在票证开始时间已过之后由客户机将设置了 **VALIDATE** 标志的票证放置在票证授予服务请求的 KDC 中。

Postdatable（可以后生效）/postdated（以后生效）

以后生效的票证是一种在其创建之后的某个指定时间之前不会生效的票证。例如，此类票证对于计划在深夜运行的批处理作业非常有用，因为如果该票证被盗，则在运行批处理作业之前，将无法使用该票证。颁发以后生效的票证时，该票证未生效，并在其开始时间已过且客户机请求 KDC 进行验证之前一直保持此状态。通常，以后生效的票证在票证授予票证的到期时间之前会一直有效。但是，如果将以后生效的票证标记为可更新，则通常会将其生命周期设置为等于票证授予票证的整个生命周期的持续时间。

Proxiable（可代理）/proxy（代理）

有时，主体需要允许服务代表其执行操作。创建该票证时，必须指定代理的主体名称。Solaris 发行版不支持可代理票证或代理票证。

可代理票证与可转发票证类似，但前者仅对单个服务有效，而可转发票证授予服务对客户机身份的完全使用权限。因此，可以将可转发票证视为一种超级代理。

Renewable（可更新）

由于拥有很长生命周期的票证存在安全风险，因此可将票证指定为可更新票证。可更新票证具有两个到期时间：票证当前实例的到期时间以及任意票证的最长生命周期（一周）。如果客户机要继续使用票证，则可在第一个到期时间之前更新票证。例如，某个票证的有效期可为一个小时，而所有票证的最长生命周期为 10 个小时。如果持有该票证的客户机要将该票证再保留几个小时，则此客户机必须在有效的小时数内更新票证。如果票证到达最长票证生命周期（10 个小时），则该票证将自动到期且无法更新。

有关如何查看票证属性的信息，请参见[查看 Kerberos 票证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veg/index.html#user-1)。

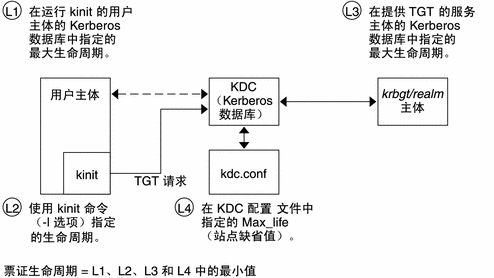
**票证生命周期**

每当主体获取包括票证授予票证 (Ticket–Granting Ticket, TGT) 在内的票证时，都会将票证的生命周期设置为以下生命周期值中的最小值：

* 通过 kinit 的 **-l** 选项指定的生命周期值，前提是使用 kinit 获取票证。缺省情况下，kinit 使用最长生命周期值。
* kdc.conf 文件中指定的最长生命周期值 (max\_life)。
* Kerberos 数据库中为提供票证的服务主体指定的最长生命周期值。如果使用 kinit，则服务主体为 krbtgt/*realm*。
* Kerberos 数据库中为请求票证的用户主体指定的最长生命周期值。

[图 26–1](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veo/index.html#refer-fig-501) 说明了如何确定 TGT 的生命周期以及四个生命周期值的来源。虽然该图说明的是如何确定 TGT 的生命周期，但所有主体获取票证时的情况基本相同。唯一的区别在于，kinit 不提供生命周期值，而提供票证的服务主体（非 krbtgt/*realm* 主体）会提供最长生命周期值。

**图 26–1 如何确定 TGT 的生命周期**



可更新票证生命周期也是由四个值中的最小值确定的，但是使用的却是可更新的生命周期值，如下所示：

* 通过 kinit 的 **-r** 选项指定的可更新生命周期值，前提是使用 kinit 获取或更新票证。
* kdc.conf 文件中指定的最长可更新生命周期值 (max\_renewable\_life)。
* Kerberos 数据库中为提供票证的服务主体指定的最长可更新生命周期值。如果使用 kinit，则服务主体为 krbtgt/*realm*。
* Kerberos 数据库中为请求票证的用户主体指定的最长可更新生命周期值。

**Kerberos 主体名称**

每个票证都使用主体名称进行标识。主体名称可以标识用户或服务。以下是一些主体名称的示例：

表 26–4 Kerberos 主体名称示例

| **主体名称** | **说明** |
| --- | --- |
| changepw/kdc1.example.com@EXAMPLE.COM | 更改口令时，允许访问 KDC 的主 KDC 服务器的主体。 |
| clntconfig/admin@EXAMPLE.COM | kclient 安装实用程序使用的主体。 |
| ftp/boston.example.com@EXAMPLE.COM | ftp 服务使用的主体。此主体可用于替代 host 主体。 |
| host/boston.example.com@EXAMPLE.COM | 基于 Kerberos 的应用程序（例如 klist 和 kprop）和服务（例如 ftp 和 telnet）使用的主体。此主体称为 host 主体或服务主体，用于验证 NFS 挂载。 |
| K/M@EXAMPLE.COM | 主密钥名称主体。一个主密钥名称主体可与每个主 KDC 关联。 |
| kadmin/history@EXAMPLE.COM | 一种主体，其中包含用于保存其他主体的口令历史记录的密钥。每个主 KDC 都具有这些主体之一。 |
| kadmin/kdc1.example.com@EXAMPLE.COM | 允许使用 kadmind 访问 KDC 的主 KDC 服务器的主体。 |
| kadmin/changepw.example.com@EXAMPLE.COM | 一种主体，用于接受来自未运行 Solaris 发行版的客户机的口令更改请求。 |
| krbtgt/EXAMPLE.COM@EXAMPLE.COM | 生成票证授予票证时使用的主体。 |
| krbtgt/EAST.EXAMPLE.COM@WEST.EXAMPLE.COM | 此主体是跨领域的票证授予票证的示例。 |
| nfs/boston.example.com@EXAMPLE.COM | NFS 服务使用的主体。此主体可用于替代 host 主体。 |
| root/boston.example.com@EXAMPLE.COM | 与客户机的 root 帐户关联的主体。此主体称作 root 主体，用于向已挂载 NFS 的文件系统提供 root 访问权限。 |
| *username*@EXAMPLE.COM | 用户的主体。 |
| *username*/admin@EXAMPLE.COM | admin 主体，可用于管理 KDC 数据库。 |

**Kerberos 验证系统的工作方式**

如果您可以提供证明您身份的票证和匹配的会话密钥，则应用程序允许您登录到远程系统。会话密钥包含特定于用户以及要访问的服务的信息。所有用户首次登录时，KDC 都会为其创建票证和会话密钥。票证和匹配的会话密钥可组成凭证。使用多个网络服务时，用户可以收集许多凭证。对于在特定服务器上运行的每个服务，用户都需要拥有一个凭证。例如，访问名为 boston 的服务器中的 ftp 服务需要凭证。访问其他服务器中的 ftp 服务需要对应于该服务的凭证。

创建和存储凭证的过程是透明的。凭证是由将凭证发送到请求程序的 KDC 创建的。收到凭证后，会将其存储在凭证高速缓存中。

**使用 Kerberos 获取服务访问权限**

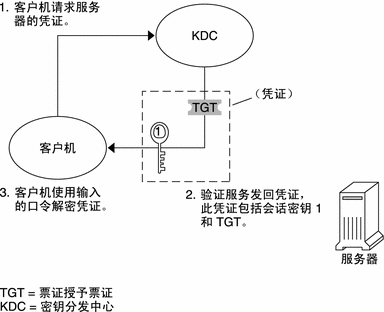
要访问特定服务器上的特定服务，用户必须获取两个凭证。第一个凭证用于票证授予票证（称为 TGT）。票证授予服务对此凭证进行解密之后，该服务即可为用户请求访问的服务器创建第二个凭证。然后，可使用第二个凭证来请求访问该服务器中的相应服务。该服务器成功解密第二个凭证后，便会授予用户访问权限。以下各节详细介绍了此过程。

**获取用于票证授予服务的凭证**

1. 要启动验证过程，客户机需要向验证服务器发送针对特定用户主体的验证请求。该请求在发送时未加密。由于请求中未包含安全信息，因此无需加密。
2. 验证服务收到该请求后，将在 KDC 数据库中查找该用户的主体名称。如果主体与数据库中的项匹配，则验证服务可获取该主体的私钥。然后，验证服务将生成一个供客户机和票证授予服务使用的会话密钥（称为会话密钥 1），以及一个用于票证授予服务的票证（票证 1）。此票证也称作**票证授予票证** (Ticket-Granting Ticket, TGT)。会话密钥和票证均使用该用户的私钥进行加密，并且会将信息发回客户机。
3. 客户机通过该用户主体的私钥，使用此信息对会话密钥 1 和票证 1 进行解密。由于该私钥仅对此用户和 KDC 数据库公开，因此包中的信息应是安全的。客户机将该信息存储在凭证高速缓存中。

在此过程中，通常会提示用户输入口令。如果用户指定的口令与用于生成存储在 KDC 数据库中的私钥的口令相同，则客户机可以成功解密验证服务发送的信息。现在，客户机便拥有了用于票证授予服务的凭证。客户机现在可以请求用于服务器的凭证。

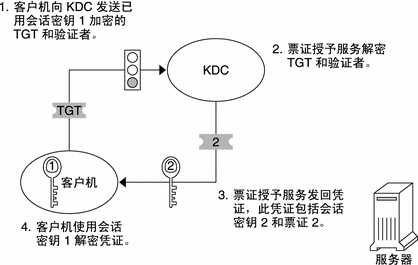
**图 26–2 获取用于票证授予服务的凭证**



**获取用于服务器的凭证**

1. 要请求访问特定服务器，客户机必须首先从验证服务获取用于该服务器的凭证。请参见[获取用于票证授予服务的凭证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veq/index.html#refer-16)。然后，客户机会向票证授予服务发送请求，其中包含服务主体名称、票证 1 以及使用会话密钥 1 加密的验证者。票证 1 最初是由验证服务使用票证授予服务的服务密钥加密的。
2. 由于票证授予服务的服务密钥对票证授予服务公开，因此可以解密票证 1。票证 1 中的信息包括会话密钥 1，因此票证授予服务可以解密验证者。此时，可使用票证授予服务验证用户主体。
3. 成功验证后，票证授予服务将为用户主体和服务器生成一个会话密钥（会话密钥 2），以及一个用于服务器的票证（票证 2）。然后，使用会话密钥 1 加密会话密钥 2 和票证 2。由于会话密钥 1 仅对该客户机和票证授予服务公开，因此此信息是安全的并可在网络上安全发送。
4. 客户机收到此信息包后，将使用存储在凭证高速缓存中的会话密钥 1 解密此信息。客户机即获取用于服务器的凭证。现在，客户机可以请求访问该服务器中的特定服务。

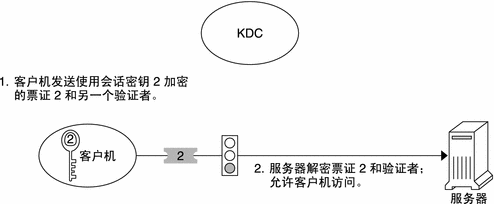
**图 26–3 获取用于服务器的凭证**



**获取对特定服务的访问权限**

1. 要请求访问特定服务，客户机必须首先从验证服务器获取用于票证授予服务的凭证，然后从票证授予服务获取服务器凭证。请参见[获取用于票证授予服务的凭证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veq/index.html#refer-16)和[获取用于服务器的凭证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2veq/index.html#refer-17)。然后，客户机可将包含票证 2 和另一个验证者的请求发送到该服务器。该验证者使用会话密钥 2 进行加密。
2. 票证 2 是由票证授予服务使用该服务的服务密钥进行加密的。由于服务密钥对服务主体公开，因此该服务可以解密票证 2 并获取会话密钥 2。然后，可使用会话密钥 2 解密验证者。如果成功解密验证者，则可授予客户机对该服务的访问权限。

**图 26–4 获取对特定服务的访问权限**



**使用 Kerberos 加密类型**

加密类型可标识执行加密操作时要使用的加密算法和模式。使用 aes、des3-cbc-sha1 和 rc4–hmac 加密类型可以创建用于较高强度加密操作的密钥。这些较高强度的操作可增强 Kerberos 服务的整体安全性。

**注 –**

如果安装了非随附的强加密软件包，则可以将 aes256-cts-hmac-sha1-96 加密类型用于 Kerberos 服务。

客户机从 KDC 请求票证时，KDC 必须使用其加密类型与客户机和服务器兼容的密钥。尽管 Kerberos 协议允许客户机请求 KDC 对客户机的票证回复部分使用特定的加密类型，但该协议不允许服务器为 KDC 指定加密类型。

**注 –**

如果安装了未运行 Solaris 10 发行版的主 KDC，则在升级主 KDC 之前，必须将从 KDC 升级到 Solaris 10 发行版。Solaris 10 主 KDC 将使用新的加密类型，而较早版本的从 KDC 将无法处理这些加密类型。

以下列出了更改加密类型之前必须考虑的一些问题：

* KDC 假定服务器支持与主体数据库中的服务器主体项关联的第一个密钥/加密类型。
* 在 KDC 上，应确保生成的主体密钥与将验证主体的系统兼容。 缺省情况下，kadmin 命令将为所有支持的加密类型创建密钥。 如果验证主体的系统不支持该缺省加密类型集，则在创建主体时应限制加密类型。通过在 kadmin **addprinc** 中使用 **-e** 标志，或在 kdc.conf 文件中将 supported\_enctypes 参数设置为此子集，可以限制加密类型。如果 Kerberos 领域中的大多数系统都支持缺省加密类型集的子集，则应使用 supported\_enctypes 参数。如果设置 supported\_enctypes，则将指定 kadmin **addprinc** 在为特定领域创建主体时使用的加密类型的缺省集。 一般情况下，最好使用这两种方法之一来控制 Kerberos 使用的加密类型。
* 确定系统支持的加密类型时，请考虑系统中运行的 Kerberos 版本，以及为其创建服务器主体的服务器应用程序所支持的加密算法。 例如，创建 nfs/hostname 服务主体时，应将加密类型限制为相应主机中的 NFS 服务器所支持的类型。 请注意，在 Solaris 10 发行版中，NFS 服务器也支持所有受支持的 Kerberos 加密类型。
* kdc.conf 文件中的 master\_key\_enctype 参数可用于控制加密主体数据库中各项的主密钥的加密类型。 如果已创建 KDC 主体数据库，请勿使用此参数。可在创建数据库时使用 master\_key\_enctype 参数，以便将缺省主密钥加密类型从 des-cbc-crc 更改为更强的加密类型。配置从 KDC 时，请确保所有从 KDC 都支持选择的加密类型，并且其 kdc.conf 中具有相同的 master\_key\_enctype 项。 另外，如果在 kdc.conf 中设置了 supported\_enctypes，则还应确保将 master\_key\_enctype 设置为 supported\_enctypes 中的加密类型之一。如果未正确处理其中任一问题，则主 KDC 可能无法使用从 KDC。
* 在客户机上，可以控制客户机在通过 krb5.conf 中的多个参数从 KDC 获取票证时请求的加密类型。 default\_tkt\_enctypes 参数用于指定客户机在通过 KDC 请求票证授予票证 (Ticket-Granting Ticket, TGT) 时要使用的加密类型。 客户机可使用 TGT 以一种更有效的方式获取其他服务器票证。 如果客户机使用 TGT 请求服务器票证（称为 TGS 请求），则设置 default\_tkt\_enctypes 将提供客户机对加密类型（用于保护客户机和 KDC 之间的通信）的部分控制。 请注意，default\_tkt\_enctypes 中指定的加密类型必须至少与 KDC 中存储的主体数据库中的一种主体密钥加密类型匹配。否则，TGT 请求将会失败。 在大多数情况下，最好不要设置 default\_tkt\_enctypes，因为此参数会导致互操作性问题。 缺省情况下，客户机代码会请求所有受支持的加密类型，而 KDC 会基于在主体数据库中找到的密钥来选择加密类型。
* default\_tgs\_enctypes 参数可限制客户机在其 TGS 请求（用于获取服务器票证）中请求的加密类型。 另外，此参数还可限制 KDC 在创建客户机和服务器共享的会话密钥时使用的加密类型。例如，如果客户机要在执行安全 NFS 时仅使用 3DES 加密，则应将 default\_tgs\_enctypes 设置为 des3-cbc-sha1。请确保客户机主体和服务器主体在主体数据库中具有 des-3-cbc-sha1 密钥。 与 default\_tkt\_enctype 一样，在大多数情况下，最好不要设置此参数，因为如果在 KDC 和服务器上未正确设置凭证，则此参数会导致互操作性问题。
* 在服务器上，可使用 kdc.conf 中的 permitted\_enctypes 来控制服务器可接受的加密类型。 此外，还可指定创建 keytab 项时使用的加密类型。 同样，一般情况下最好不要使用其中任一方法来控制加密类型，而应由 KDC 来确定要使用的加密类型，因为 KDC 不会与服务器应用程序进行通信以确定要使用的密钥或加密类型。

**使用 gsscred 表**

如果缺省映射不满足要求，则 NFS 服务器尝试标识 Kerberos 用户时，将使用 gsscred 表。NFS 服务使用 UNIX ID 来标识用户。这些 ID 不属于用户主体或凭证。gsscred 表提供从 GSS 凭证到 UNIX UID 的其他映射（通过口令文件）。填充 KDC 数据库后，必须创建和管理该表。有关更多信息，请参见[将 GSS 凭证映射到 UNIX 凭证](https://docs.oracle.com/cd/E19253-01/819-7061/6n91j2vb4/index.html)。

接收到客户机请求时，NFS 服务便会尝试将凭证名称映射到 UNIX ID。如果映射失败，则会检查 gsscred 表。

**Solaris Kerberos 和 MIT Kerberos 之间的显著差异**

Kerberos 服务的 Solaris 10 版本基于 MIT Kerberos 版本 1.2.1。以下列出了 Solaris 10 发行版中提供的增强功能，在 MIT 1.2.1 版本中不会提供这些功能：

* Solaris 远程应用程序的 Kerberos 支持
* KDC 数据库的增量传播
* 客户机配置脚本
* 已本地化的错误消息
* BSM 审计记录支持
* Kerberos 通过 GSS-API 安全使用线程
* 使用密码学的加密框架

此版本还包括某些已发布的 MIT 1.2.1 错误修复。特别是，添加了 1.2.5 二叉树错误修复和 1.3 TCP 支持。

* [*Previous*: 第 25 章 使用 Kerberos 应用程序（任务）](https://docs.oracle.com/cd/E19253-01/819-7061/user-7/index.html)
* [*Next*: 第 7 部分 Solaris 审计](https://docs.oracle.com/cd/E19253-01/819-7061/audittm-1/index.html)
* © 2010, Oracle Corporation and/or its affiliates