Kerberos 概述 - 开放网络系统的认证服务

目录

简介

在开放式网络计算环境中，无法信任工作站来正确识别其网络服务的用户。Kerberos提供了一种替代方法，其中使用可信第三方身份验证服务来验证用户的身份。本文对麻省理工学院项目雅典娜的Kerberos认证模型进行了概述。它描述了客户端、服务器和Kerberos用于实现身份验证的协议。它还描述了所需数据库的管理和复制。介绍了用户、程序员和管理员看到的Kerberos视图。最后，给出了Kerberos在更大Athena图片中的作用，以及当前使用Kerberos进行用户身份验证的应用列表。我们将Kerberos身份验证添加到Sun网络文件系统作为将Kerberos与现有应用程序集成的案例研究。

Kerberos 作者

- 麻省理工学院雅典娜项目Jennifer G. Steiner，马萨诸塞州剑桥02139,steiner@ATHENA.MIT.EDU
- 华盛顿大学计算机科学系，FR-35，华盛顿州西雅图，WA 98195,bcn@CS.WASHINGTON.EDU。在Kerberos的设计和初始实施阶段，Clifford Neuman是Project Athena员工的成员。
- 麻省理工学院雅典娜项目Jeffrey I. Schiller，马萨诸塞州剑桥02139,jis@ATHENA.MIT.EDU

Kerberos 简介

本文对Miller和Neuman设计的认证系统Kerberos进行了概述。并介绍我们在麻省理工学院雅典娜项目中使用它的经验。在“动机”部分，我们解释了为什么开放网络需要新的身份验证模型，以及其要求。什么是Kerberos?部分列出了Kerberos软件的组件，并描述了它们在提供身份验证服务时如何交互。在“Kerberos名称”部分，我们介绍了Kerberos命名方案。

Kerberos的工作方式提供了Kerberos身份验证的构建块 — 票证和身份验证器。这引出了对两种身份验证协议的讨论：用户对Kerberos的初始身份验证（类似于登录），以及潜在用户和网络服务潜在生成者相互身份验证的协议。

Kerberos需要一个有关其客户端的信息数据库；“Kerberos数据库”部分介绍数据库、其管理和用于修改它的协议。“从外部查找In的Kerberos”部分介绍了与其用户、应用程序程序员和管理员的Kerberos接口。在大图中，我们描述了Athena Kerberos项目如何适合雅典娜环境的其余部分。我们还描述了不同Kerberos身份验证域或领域之间的交互；在本例中，Athena Kerberos项目与麻省理工学院计算机科学实验室运行的Kerberos之间的关系。

在“问题和未解决问题”部分，我们提到尚未解决的未解决问题。最后一部分提供Project Athena中Kerberos的当前状态。在附录中，我们详细介绍了如何将Kerberos应用于网络文件服务，以验证希望访问远程文件系统的用户。

Kerberos 概念

在本文中，我们使用的术语可能不明确、对读者陌生或在其他地方使用不同。下面我们介绍了这些术语的使用。

用户、*客户端*、*服务器* — 按用户，我们指使用程序或服务的人。客户也使用某种东西，但不一定是人；可以是程序。网络应用通常由两部分组成；一个程序在一台计算机上运行并请求远程服务，另一个程序在远程计算机上运行并执行该服务。我们分别将这些客户端和服务器端称为应用。通常，客户端会代表用户与服务器联系。

从某种意义上讲，使用Kerberos系统的每个实体（无论是用户还是网络服务器）都是客户端，因为

它使用Kerberos服务。因此，为了区分Kerberos客户端和其他服务的客户端，我们使用术语“主体”来表示此实体。请注意，Kerberos主体可以是用户或服务器。（我们将在后面的章节中介绍Kerberos主体的命名。）

服务与服务器 — 我们使用服务作为要执行的某些操作的抽象规范。执行这些操作的进程称为服务器。在给定时间，可能有多台服务器（通常在不同的计算机上运行）执行给定服务。例如，在Athena，我们的每台分时机器上都运行一个BSD UNIX登录服务器。

Key、*Private Key*、*Password* - Kerberos使用私钥加密。每个Kerberos主体都分配了一个大数，即其私钥，只知道该主体和Kerberos。对于用户，私钥是应用到用户密码的单向函数的结果。我们使用密钥作为私钥的简写。

凭据 — 遗憾的是，此单词对Sun网络文件系统和Kerberos系统都有特殊含义。我们明确说明我们是指NFS凭证还是Kerberos凭证，否则术语将用于正常的英语意义。

主和从 — 可以在多台计算机上运行Kerberos身份验证软件。但是，Kerberos数据库始终只有一个最终副本。存储此数据库的计算机称为主计算机，或仅称为主计算机。其他计算机可能拥有Kerberos数据库的只读副本，这些副本称为从属。

开发 Kerberos 的动机

在非联网的个人计算环境中，通过物理保护个人计算机，可以保护资源和信息。在分时计算环境中，操作系统保护用户彼此不受影响并控制资源。为了确定每个用户能够读取或修改什么，分时系统需要识别每个用户。这在用户登录时完成。

在需要从许多独立计算机提供服务的用户网络中，访问控制可采用三种方法：无所作为，依靠用户登录的机器来防止未经授权的访问；可以要求主机证明其身份，但信任主机对用户的描述；或者要求用户证明每个所需服务的身份。

在所有机器都受到严格控制的封闭环境中，可以使用第一种方法。当组织控制通过网络通信的所有主机时，这是一种合理的方法。

在更开放的环境中，人们可能只会选择性地信任那些受组织控制的主机。在这种情况下，必须要求每台主机证明其身份。rlogin和rsh程序使用此方法。在这些协议中，身份验证通过检查从中建立连接的Internet地址来完成。

在雅典娜环境中，我们必须能够满足组织控制范围之外的主机的请求。用户完全控制其工作站：他们可以重新启动，独立启动，甚至从自己的磁带上启动。因此，必须采取第三种方法；用户必须证明每个所需服务的身份。服务器还必须证明其身份。物理保护运行网络服务器的主机是不够的；网络中其他位置的某人可能伪装成给定服务器。

我们的环境对识别机制提出了若干要求。首先，它必须是安全的。绕过它一定足够困难，以致潜在攻击者无法发现身份验证机制是弱链路。监视网络的用户应该无法获取模拟其他用户所需的信息。其次，它必须可靠。对许多服务的访问将取决于身份验证服务。如果它不可靠，则服务系统整体将不可靠。第三，它应该是透明的。理想情况下，用户不应知道身份验证正在进行。最后，它应具有可扩展性。许多系统可以与雅典娜主机通信。并非所有这些都支持我们的机制，但如果软件支持，则软件不应中断。

Kerberos是我们为满足上述要求而做的工作的结果。当用户走上到工作站时，他们会登录。据用户所知，此初始标识足以在登录会话期间向所有所需网络服务器证明其身份。Kerberos的安全性依赖于多个身份验证服务器的安全性，但不依赖于用户登录的系统，也不依赖于将要使用的终端服务器的安全性。身份验证服务器为经过正确身份验证的用户提供向分散在网络中的服务器证明其身份的

方法。

身份验证是安全网络环境的基本构建块。例如，如果服务器确定知道客户端的身份，它可以决定是否提供服务、是否应给用户特权、谁应接收服务帐单等。换句话说，授权和记帐方案可以建立在Kerberos提供的身份验证之上，从而产生与单个个人计算机或分时系统等同的安全性。

什么是 Kerberos？

Kerberos是基于Needham和Schroeder提出的模型的可信第三方认证服务。它值得信赖，因为它的每个客户端都相信Kerberos对其他每个客户端身份的判断是准确的。时间戳（表示当前日期和时间的大数）已添加到原始模型中，以帮助检测重放。当消息从网络中被盗并稍后重新发送时，会发生重播。有关重播和其他身份验证问题的更完整说明，请参阅Voydock和Kent。

Kerberos 起什么作用？

Kerberos保留其客户端及其私钥的数据库。私钥是一个只有Kerberos及其所属客户端知道的大数。如果客户端是用户，则是加密密码。需要向Kerberos注册身份验证的网络服务，希望使用这些服务的客户端也是如此。私钥在注册时协商。

由于Kerberos知道这些私钥，因此它可以创建消息来说服一个客户端相信另一个客户端是它声称的真实身份。Kerberos还会生成临时私钥，称为会话密钥，这些私钥给两个客户端，而没有其他客户端。会话密钥可用于加密双方之间的消息。

Kerberos提供三种不同级别的保护。应用程序员根据应用的要求确定哪个是合适的。例如，某些应用程序只需要在网络连接开始时建立真实性，并且可以假设来自给定网络地址的进一步消息来自经过身份验证的一方。我们经过身份验证的网络文件系统使用这种安全级别。

其他应用要求对每条消息进行身份验证，但不关心消息的内容是否被公开。对于这些，Kerberos提供安全消息。但是，私有消息提供了更高级别的安全性，其中每条消息不仅经过身份验证，而且经过加密。私有消息（例如，Kerberos服务器本身使用私有消息通过网络发送密码）。

Kerberos 软件组件

Athena实施包括几个模块：

- Kerberos应用程序库
- 加密库
- 数据库库
- 数据库管理程序
- 管理服务器
- 身份认证服务器：
- DB传播软件
- 用户程序
- 应用

Kerberos应用程序库为应用程序客户端和应用程序服务器提供了一个接口。它包括创建或读取身份验证请求的例程，以及创建安全或专用消息的例程。

Kerberos中的加密基于DES，即数据加密标准。加密库实现这些例程。提供了几种加密方法，在速度和安全之间进行权衡。还提供了DES密码块链(CBC)模式的扩展，称为传播CBC模式。在CBC中，错误仅通过当前密码块传播，而在PCBC中，错误在消息中传播。如果发生错误，而不只是错误

的一部分，这会使整个消息变得无用。该加密库是独立的模块，可以被其它DES实现或不同的加密库替换。

另一个可更换模块是数据库管理系统。数据库库的当前Athena实现使用ndbm，尽管Ingres最初使用。也可以使用其他数据库管理库。

Kerberos数据库需求非常简单；每名被委托人都有记录，包括被委托人的姓名、私钥、到期日期以及一些管理信息。(到期日期是条目不再有效的日期。注册时，通常设定为未来几年。)

其他用户信息（如实名、电话号码等）由另一台服务器（即Hesiod名称服务器）保存。这样，敏感信息即密码就可以由Kerberos处理，使用相当高的安全措施；而赫西奥德保存的非敏感信息处理方式不同；例如，它可以通过网络未加密地发送。

Kerberos服务器使用数据库库，管理数据库的工具也使用数据库库。

管理服务器（或KDBM服务器）为数据库提供读写网络接口。程序的客户端可以在网络上的任何计算机上运行。但是，服务器端必须在包含Kerberos数据库的计算机上运行，才能对数据库进行更改。

另一方面，身份验证服务器（或Kerberos服务器）对Kerberos数据库执行只读操作，即主体身份验证和会话密钥的生成。由于此服务器不修改Kerberos数据库，因此它可能在包含主Kerberos数据库的只读副本的计算机上运行。

数据库传播软件管理Kerberos数据库的复制。可以在多台不同的计算机上拥有数据库副本，并且每台计算机上都运行着身份验证服务器副本。这些从机中的每台都以给定的时间间隔从主机接收Kerberos数据库的更新。

最后，还有一些最终用户程序，用于登录Kerberos、更改Kerberos密码以及显示或销毁Kerberos票证（稍后将介绍票证）。

Kerberos 名称

对实体进行身份验证的部分内容是对其命名。身份验证过程是验证客户端是请求中命名的客户端。名称由什么组成？在Kerberos中，用户和服务器均被命名。就身份验证服务器而言，它们是等效的。名称由主名称、实例和领域组成，表示为name.instance@realm。

主名称是用户或服务的名称。该实例用于区分主名称上的各种变体。对于用户，实例可能具有特殊权限，例如“根”或“管理员”实例。对于Athena环境中的服务，实例通常是服务器运行所在计算机的名称。例如，rlogin服务在不同主机上具有不同的实例：rlogin.priam是名为priam的主机上的rlogin服务器。Kerberos票证仅适用于单个命名服务器。因此，需要单独的票证才能访问同一服务的不同实例。领域是维护身份验证数据的管理实体的名称。例如，不同的机构可能各自拥有自己的Kerberos机器，其中包含不同的数据库。它们有不同的Kerberos领域。(在与其他Kerberi的交互中进一步讨论领域。)

Kerberos 工作原理

本节介绍Kerberos身份验证协议。如上所述，Kerberos认证模型基于Needham和Schroeder密钥分发协议。当用户请求服务时，必须建立其身份。为此，系统会向服务器显示票证，并提供票证最初是发给用户而非被盗的证明。通过Kerberos进行身份验证有三个阶段。在第一阶段，用户获取用于请求访问其他服务的凭证。在第二阶段，用户请求对特定服务进行身份验证。在最后阶段，用户向终端服务器提供这些凭证。

Kerberos 证书

Kerberos身份验证模型中使用两种类型的凭据：票证和身份验证器。两者都基于私钥加密，但使用不同的密钥加密。票证用于在身份验证服务器和终端服务器之间安全地将票证颁发者的身份传递给其。票证还传递信息，这些信息可用于确保使用票证的人员与签发票证的人员相同。验证器包含附加信息，当与票证中的信息进行比较时，证明提供票证的客户端与发出票证的客户端相同。

单个服务器和单个客户端都适合使用票证。它包含服务器名称、客户端名称、客户端的Internet地址、时间戳、生存期和随机会话密钥。此信息使用将使用票证的服务器的密钥进行加密。票证发出后，指定客户端可多次使用该票证以访问指定服务器，直到票证过期。请注意，由于票证在服务器的密钥中加密，因此允许用户将票证传递到服务器是安全的，而不必担心用户修改票证。

与票证不同，身份验证器只能使用一次。每次客户端想使用服务时，都必须生成新服务。这不会出现问题，因为客户端可以自行构建身份验证器。身份验证器包含客户端的名称、工作站的IP地址和当前工作站时间。身份验证器在作为票证一部分的会话密钥中加密。

获取初始Kerberos票证

当用户走上工作站时，只有一条信息可以证明其身份：用户的密码。与身份验证服务器的初始交换设计为最小化密码被泄露的可能性，同时不允许用户在不知道该密码的情况下正确地对她/自己进行身份验证。登录过程对用户而言与登录分时系统相同。然而，在幕后，情况却截然不同。

系统会提示用户输入其用户名。一旦输入，请求就会发送到包含用户名和特殊服务名称（称为票证授予服务）的身份验证服务器。

身份验证服务器检查它是否知道客户端。如果是，它会生成随机会话密钥，稍后将在客户端和票证授予服务器之间使用。然后，它为票证授予服务器创建票证，该票证授予服务器包含客户端名称、票证授予服务器的名称、当前时间、票证的生存期、客户端的IP地址和刚创建的随机会话密钥。所有这些都在只有票证授予服务器和身份验证服务器知道的密钥中加密。

然后，身份验证服务器将票证连同随机会话密钥的副本和一些附加信息发送回客户端。此响应在客户端的私钥中加密，该私钥仅对Kerberos和客户端知道，从用户密码派生。

客户端收到响应后，会要求用户输入其密码。密码被转换为DES密钥，用于解密来自身份验证服务器的响应。票证和会话密钥连同一些其它信息一起被存储以供将来使用，并且用户的密码和DES密钥从存储器中被擦除。

一旦交换完成，工作站就具有它可用于证明其用户在票证授予票证的有效期内的身份的信息。只要工作站上的软件之前未被篡改过，就不存在允许其他人在票证有效期之外模拟用户的信息。

请求Kerberos服务

目前，让我们假装用户已经拥有所需服务器的票证。为了获得对服务器的访问权，应用程序会构建包含客户端名称和IP地址以及当前时间的身份验证器。然后，身份验证器在随服务器票证收到的会话密钥中加密。然后，客户端以由单个应用定义的方式将身份验证器连同票证一起发送到服务器。

一旦服务器接收到验证器和票证，服务器将解密该票证，使用票证中包含的会话密钥解密验证器，将票证中的信息与验证器中的信息、从中接收请求的IP地址以及当前时间进行比较。如果所有内容都匹配，则允许请求继续。

假设时钟在几分钟内同步到。如果请求中的时间过长，或过去，服务器会将请求视为重播先前请求的尝试。还允许服务器使用仍然有效的时间戳跟踪所有过去的请求。为了进一步阻止重播攻击，可

以丢弃与已接收的请求具有相同票证和时间戳的已接收请求。

最后，如果客户端指定它希望服务器也证明其身份，则服务器会将一个添加到客户端在身份验证器中发送的时间戳，加密会话密钥中的结果，并将结果发送回客户端。

在此交换结束时，服务器确定，根据Kerberos，客户端是它所说的自己。如果进行相互身份验证，客户端也确信服务器是可信的。此外，客户端和服务器共享一个其他人都不知道的密钥，并且可以安全地假设在该密钥中加密的合理最近的消息源自另一方。

获取Kerberos服务器票证

回想一下，票证仅适用于单台服务器。因此，必须为客户端要使用的每项服务获取单独的票证。可从票证授予服务获取单个服务器的票证。由于票证授予服务本身是一项服务，因此它利用了上一节中描述的服务访问协议。

当程序需要尚未请求的票证时，它会向票证授予服务器发送请求。该请求包含请求票证的服务器的名称，以及票证授予票证和构建的验证器，如上一节所述。

然后，票证授予服务器检查身份验证器和票证授予票证，如上所述。如果有效，票证授予服务器将生成一个新的随机会话密钥，以在客户端和新服务器之间使用。然后，它为新服务器创建票证，其中包含客户端名称、服务器名称、当前时间、客户端的IP地址以及刚生成的新会话密钥。新票证的有效期是授予票证的剩余寿命的最小值和服务的默认寿命。

然后，票证授予服务器将票证连同会话密钥和其他信息发送回客户端。但是，这次回复在作为票证授予票证一部分的会话密钥中加密。这样，用户就无需再次输入其密码。

Kerberos 数据库

到目前为止，我们已讨论了需要对Kerberos数据库进行只读访问的操作。这些操作由身份验证服务执行，该服务可在主机和从机上运行。

在本节中，我们将讨论需要对数据库进行写入访问的操作。这些操作由管理服务(称为Kerberos数据库管理服务(KDBM))执行。 当前实施规定，只能对主Kerberos数据库进行更改；从属副本为只读。因此，KDBM服务器只能在主Kerberos计算机上运行。

请注意，虽然身份验证仍然可以（在从设备上）进行，但如果主计算机关闭，则无法处理管理请求。根据我们的经验，这并没有带来问题，因为管理请求很少。

KDBM处理用户更改其密码的请求。此程序的客户端通过网络向KDBM发送请求，即kpasswd程序。KDBM还接受Kerberos管理员的请求，这些管理员可能会向数据库添加主体，并更改现有主体的密码。管理程序的客户端也通过网络向KDBM发送请求，即kadmin程序。

KDBM 服务器

KDBM服务器接受向数据库添加主体或更改现有主体密码的请求。此服务的独特之处在于，票证授予服务不会为其颁发票证。相反，必须使用身份验证服务本身（用于获取票证授予票证的相同服务）。 这样做的目的是要求用户输入密码。否则，如果用户将其工作站留在无人看管的位置，路人可以走上前去更改其密码，这应该防止。同样，如果管理员将其工作站置于未保护状态，则密码可能会更改系统中的任何密码。

当KDBM服务器接收到请求时，它通过将更改的请求者的经过验证的主体名称与请求的目标的主体

名称进行比较来授权它。如果它们相同，则允许请求。如果它们不相同，KDBM服务器将查询访问控制列表（存储在主Kerberos系统上的文件中）。 如果在此文件中找到请求者的主体名称，则允许请求，否则拒绝请求。

按照惯例，具有NULL实例（默认实例）的名称不会出现在访问控制列表文件中；而是使用管理实例。因此，用户要成为Kerberos的管理员，必须创建该用户名的管理实例，并将其添加到访问控制列表。此约定允许管理员使用不同的密码进行Kerberos管理，然后她/他将使用该密码进行正常登录。

记录对KDBM程序的所有请求，无论允许还是拒绝。

kadmin 与 kpasswd 程序

Kerberos的管理员使用kadmin程序将主体添加到数据库，或更改现有主体的密码。管理员在调用kadmin程序时需要输入其管理实例名称的密码。此密码用于获取KDBM服务器的票证。

用户可以使用kpasswd程序更改其Kerberos密码。当他们调用程序时，必须输入其旧密码。此密码用于获取KDBM服务器的票证。

Kerberos 数据库复制

每个Kerberos领域都有一个主Kerberos计算机，其中包含身份验证数据库的主副本。在系统中其他位置的从属计算机上可以拥有数据库的其他只读副本（尽管不必要）。拥有多个数据库副本的优势是通常用于复制的优势：更高的可用性和更好的性能。如果主机关闭，仍然可以在其中一台从机上实现身份验证。在多台计算机中的任何一台上执行身份验证的能力降低了主计算机出现瓶颈的可能性。

保留数据库的多个副本会带来数据一致性问题。我们发现非常简单的方法足以处理不一致。主数据库每小时转储一次。该数据库被完整地发送到从机，然后从机更新自己的数据库。主主机上的程序kprop将更新发送到在每台从机上运行的对等程序kpropd。First kprop发送它将要发送的新数据库的校验和。校验和在主Kerberos计算机和从Kerberos计算机都拥有的Kerberos主数据库密钥中加密。然后，数据通过网络传输到从设备上的kpropd。从传播服务器计算其所接收数据的校验和，如果它与主传播服务器发送的校验和匹配，则新信息用于更新从传播服务器的数据库。

Kerberos数据库中的所有密码都在主数据库密钥中加密。因此，通过网络从主数据库传送到从数据库的信息对窃听者并不有用。但是，只有来自主主机的信息被从主机接受，并且检测到数据的篡改，从而检测校验和，这是非常重要的。

Kerberos 简介

本节从实际的角度介绍Kerberos，首先由用户看到，然后从应用程序程序员的角度，最后通过Kerberos管理员的任务。

用户眼中的 Kerberos

如果一切顺利，用户将不会注意到Kerberos存在。在我们的UNIX实施中，票证授予票证是作为登录过程的一部分从Kerberos获取的。更改用户的Kerberos密码是passwd程序的一部分。当用户注销时，Kerberos票证会自动销毁。

如果用户的登录会话持续的时间超过票证授予票证的有效期（当前为8小时），则用户将注意到Kerberos的存在，因为下次执行经Kerberos身份验证的应用程序时，它将失败。其Kerberos票证将

过期。此时，用户可以运行kinit程序来获取票证授予服务器的新票证。与登录时一样，必须提供密码才能获取密码。出于好奇而执行klist命令的用户可能会惊讶于为需要Kerberos身份验证的服务默默地获得的所有票证。

从程序员的观点看 Kerberos

编写Kerberos应用程序的程序员通常会向包含客户端和服务器端的现有网络应用程序添加身份验证。我们称此过程为“Kerberizing”程序。Kerberizing通常包括对Kerberos库进行调用，以便在初始服务请求时执行身份验证。它还可能涉及调用DES库来加密随后在应用客户端和应用服务器之间发送的消息和数据。

最常用的库函数是客户端的krb_mk_req和服务器端的krb_rd_req。krb_mk_req例程将目标服务器的名称、实例和领域以及可能要发送的数据的校验和作为参数。然后，客户端通过网络将krb_mk_req调用返回的消息发送到应用的服务器端。当服务器收到此消息时，它会调用库例程krb_rd_req。例行程序会就发件人所声称身份的真实性作出判决。

如果应用程序要求在客户端和服务器之间发送的消息是加密的，则可以对krbmkpriv(krbrdpriv)进行库调用，以加密（解密）会话密钥中的消息，而会话密钥现在由双方共享。

Kerberos 管理员的工作

Kerberos管理员的作业首先运行程序来初始化数据库。必须运行另一个程序才能在数据库中注册基本主体，例如Kerberos管理员的名称和管理员实例。必须启动Kerberos身份验证服务器和管理服务器。如果有从属数据库，管理员必须安排定期启动从主数据库向从属数据库传播数据库更新的程序。

执行这些初始步骤后，管理员使用kadmin程序通过网络操纵数据库。通过该程序，可以添加新的主体，并更改密码。

特别是，当新的Kerberos应用程序添加到系统时，Kerberos管理员必须执行几个步骤才能使其正常工作。服务器必须在数据库中注册，并分配私钥（通常是自动生成的随机密钥）。 然后，必须从数据库中提取某些数据（包括服务器的密钥），并将其安装到服务器计算机上的文件中。默认文件为/etc/srvtab。服务器调用的krb_rd_req库例程（请参阅上一节）使用该文件中的信息解密以服务器私钥中加密的发送的消息。/etc/srvtab文件将服务器验证为终端上键入的密码对用户进行身份验证。

Kerberos管理员还必须确保Kerberos计算机在物理上是安全的，而且最好维护主数据库的备份。

Kerberos 的广阔前景

在本节中，我们将介绍Kerberos如何融入Athena环境，包括其他网络服务和应用程序的使用，以及它如何与远程Kerberos领域交互。有关雅典娜环境的更完整说明，请参阅G.W. Treese。

在其他网络服务中使用 Kerberos

一些网络应用程序已修改为使用Kerberos。rlogin和rsh命令首先尝试使用Kerberos进行身份验证。具有有效Kerberos票证的用户可以重新登录到另一台Athena计算机，而无需设置.rhosts文件。如果Kerberos身份验证失败，程序将回退到其常用的授权方法，在本例中为.rhosts文件。

我们修改了邮局协议，使用Kerberos对希望从“邮局”检索其电子邮件的用户进行身份验证。Athena最近开发了一个名为Zephyr的消息传递程序，它也使用Kerberos进行身份验证。

用于注册新用户的程序称为注册，它同时使用服务管理系统(SMS)和Kerberos。通过SMS，它确定新雅典娜用户输入的信息（如姓名和MIT标识号）是否有效。然后，它会与Kerberos一起检查请求的用户名是否唯一。如果一切顺利，则会向Kerberos数据库添加新条目，其中包含用户名和密码。

有关使用Kerberos保护Sun网络文件系统的详细讨论，请参阅附录。

与其它 Kerberi 的交互

预计不同的管理组织会希望使用Kerberos进行用户身份验证。在许多情况下，预计一个组织中的用户会希望在另一个组织中使用服务。Kerberos支持多个管理域。Kerberos中名称的规范包括一个称为领域的字段。此字段包含要对其用户进行身份验证的管理域的名称。

服务通常在单个领域中注册，并且只接受由身份验证服务器为该领域颁发的凭证。用户通常在单个领域（本地领域）中注册，但是，她/他可以根据本地领域提供的验证，获得由另一领域（远程领域）颁发的凭证。在远程领域有效的凭证指示用户最初经过身份验证的领域。远程领域的服务可以根据所需的安全程度和最初对用户进行身份验证的领域的信任程度来选择是否执行这些凭证。

要执行跨领域身份验证，每对领域的管理员必须选择要在其领域之间共享的密钥。然后，本地领域中的用户可以从本地认证服务器为远程领域中的票证授予服务器请求票证授予票证。当使用该票证时，远程票证授予服务器识别该请求不来自其自己的领域，并且它使用之前交换的密钥解密票证授予票证。然后，它会像通常一样发出票证，但客户端的领域字段包含客户端最初经过身份验证的领域的名称除外。

此方法可以扩展为允许通过一系列领域对自己进行身份验证，直到使用所需服务到达领域。但是，为了执行此操作，需要记录所采用的整个路径，而不仅仅是用户通过身份验证的初始领域的名称。在这种情况下，服务器只知道A说B说C说用户是个个。只有沿途的每个人都受信任时，才能信任此语句。

Kerberos 问题与未解决的问题

与Kerberos身份验证机制相关的问题和开放问题很多。问题包括如何确定故障单的正确生命期、如何允许代理以及如何保证工作站完整性。

票证寿命问题是在安全性和便利性之间选择适当的折衷。如果票证的寿命较长，则如果票证及其关联会话密钥被盗或放错位置，则它们可以被使用更长的时间。如果用户忘记从公共工作站注销，此类信息可能会被窃取。或者，如果用户在允许多个用户的系统上经过身份验证，则具有根访问权限的其他用户可能能够找到使用被盗票证所需的信息。但是，给票证提供短生命期的问题在于，当票证过期时，用户将必须获得一个新票证，该票证要求用户再次输入密码。

开放问题是代理问题。通过身份验证的用户如何允许服务器代表其获取其他网络服务？这很重要的一个示例是使用一个服务，该服务将直接从文件服务器访问受保护的文件。此问题的另一个示例是我们所谓的身份验证转发。如果用户登录工作站并登录到远程主机，则当用户在远程主机上运行程序时，如果用户能够访问本地可用的相同服务，则情况会很好。使此难以实现的是，用户可能不信任远程主机，因此在所有情况下，身份验证转发都不理想。我们目前没有解决这个问题的办法。

另一个问题是如何保证在工作站上运行的软件的完整性，这也是Athena环境中的一个重要问题。这在专用工作站上并不是什么问题，因为将使用它的用户可以控制它。但是，在公共工作站上，可能有人来修改登录程序以保存用户密码。目前我们环境中唯一可用的解决方案是让人们很难修改在公共工作站上运行的软件。更好的解决方案要求用户的密钥永远不要离开用户知道可以信任的系统。一种方法是，如果用户拥有能够执行身份验证协议中所需加密的智能卡，就可以实现此目的。

Kerberos 状态

1986年9月，Kerberos的原型版开始生产。自1987年1月以来，Kerberos一直是Project Athena对其5,000个用户、650个工作站和65台服务器进行身份验证的唯一方法。此外，Kerberos现在正用于取代.rhosts文件，以控制Athena的多个分时系统中的访问。

Kerberos 致谢

Kerberos最初由Steve Miller和Clifford Neuman设计，Jeff Schiller和Jerry Saltzer提供建议。自那时起，许多其他人参与了该项目。其中包括Jim Aspnes， Bob Baldwin， John Barba， Richard Basch， Jim Bloom， Bill Bryant， Mark Colan， Rob French， Dan Geer， John Kohl， John Kubiatowicz， Bob Mckie， Murphy， John On， On， Ostln， Rerh， Ch， Ch， Ch， Ch， Ch Ch， Ch Ch Ch Ch Ch Ch， Ch Ch Ch， Ch， Ch Ch， Chris Ch， Chris Ch Chris Ch， Ch Ch Chris Ch Chis Ro Ch Ch， Ch， Ch ChChlis、Mike Shanzer、Bill Sommerfeld、Ted T'so、Win Treese和Stan Zanarotti。

我们感谢Dan Geer、Kathy Lieben、Josh Lubarr、Ken Raeburn、Jerry Saltzer、Ed Steiner、Robbert van Renesse和Win Treese，他们的建议大大改善了本白皮书的早期草稿。

J.T. Kohl和W.E. Sommerfeld，Zephyr通知系统，在Usenix会议记录（1988年冬季）中。

M.A. Rosenstein、D.E. Geer和P.J. Levine，在Usenix会议记录（冬季，1988年）。

R. Sandberg， D. Goldberg， S. Kleiman， D. Walsh和B. Lyon，《太阳网络文件系统的设计和实施》，在Usenix会议（1985年夏季）中。

附录：Kerberos在SUN网络文件系统(NFS)中的应用

Project Athena工作站系统的一个关键组件是用户工作站与其专用文件存储（主目录）之间的网络交互。 所有专用存储都驻留在一组专用于此目的的计算机（当前为VAX 11/750）上。这允许我们在公开可用的UNIX工作站上提供服务。当用户登录到这些可公开使用的工作站之一，然后根据本地驻留的密码文件验证其姓名和密码时，我们使用Kerberos确定其真实性。登录程序会提示输入用户名（与任何UNIX系统一样）。 此用户名用于获取Kerberos票证授予票证。登录程序使用密码生成用于解密票证的DES密钥。如果解密成功，则通过咨询Hesiod命名服务并通过NFS装载用户的主目录。然后，登录程序将控制权交给用户的外壳，后者可以运行传统的按用户定制文件，因为现在主目录已“附加”到工作站。Hesiod服务还用于在本地密码文件中构造条目。（这是为在/etc/passwd中查找信息的程序提供的。）

从提供远程文件服务的几个选项中，我们选择了Sun的网络文件系统。但是，此系统无法以关键方式满足我们的需求。NFS假设所有工作站分为两类（从文件服务器的角度来看）：可信和不可信。不受信任的系统根本无法访问任何文件，受信任的系统可以。可信系统是完全可信的。假设可信系统由友好管理管理。具体而言，可以从受信任工作站伪装为文件服务系统的任何有效用户，从而仅获得对系统上几乎所有文件的访问。（仅“根”拥有的文件免除。）

在我们的环境中，工作站的管理（在传统意义上是UNIX系统管理）由当前使用它的用户负责。我们不会对工作站的根密码加密，因为我们意识到，真正不友好的用户可能会因为与计算机位于同一物理位置并且能够访问所有控制台功能而侵入。因此，我们无法真正信任我们的工作站对信任的NFS解释。为了在我们的环境中允许适当的访问控制，我们必须对基本NFS软件进行一些修改，并将Kerberos集成到方案中。

未经 Kerberos 修改的 NFS

在我们开始实施的NFS中（来自威斯康星大学），身份验证是以每个NFS请求中包含的数据片段（在NFS术语中称为“凭证”）的形式提供的。 此凭证包含关于请求者的唯一用户标识符(UID)的信息和请求者成员身份的组标识符(GID)的列表。然后，NFS服务器使用此信息进行访问检查。受信任工作站与不受信任工作站之间的区别在于NFS服务器是否接受其凭证。

被 Kerberos 修改过的 NFS

在我们的环境中，如果且仅当凭证指示工作站用户的UID时，NFS服务器才必须接受工作站的凭证，而不是其他凭证。

一个显而易见的解决方案是将凭证的性质从单纯的UID和GID指标更改为完整的Kerberos身份验证数据。然而，如果采用此解决方案，将会支付重大绩效惩罚。在每个NFS操作（包括所有磁盘读写活动）上交换凭据。在每个磁盘事务上包含Kerberos身份验证会为每个事务添加相当数量的完整加密（在软件中完成），并且根据我们的信封计算，会提供不可接受的性能。（它还需要将Kerberos库例程放入内核地址空间。）

我们需要一种混合方法，如下所述。其基本思想是将从客户端工作站接收的NFS服务器映射凭据发送到服务器系统上的有效（可能不同）凭据。此映射在每个NFS事务上在服务器的内核中执行，并由用户级进程在“装载”时设置，该进程在建立有效的内核凭据映射之前参与Kerberos审核身份验证。

为了实现此目的，我们向内核添加了新的系统调用（仅在服务器系统上需要，而不是在客户端系统上需要），该调用提供了映射功能的控制，该映射功能将来自客户端工作站的传入凭证映射到服务器上有效使用的凭证（如果有）。 基本映射函数映射元组：

```
<CLIENT-IP-ADDRESS, UID-ON-CLIENT>
```

到服务器系统上的有效NFS凭证。CLIENT-IP-ADDRESS从客户端系统提供的NFS请求数据包中提取。注意：除UID-ON-CLIENT外，客户端生成的凭证中的所有信息都将被丢弃。

如果不存在映射，则服务器会以两种方式之一进行响应，具体取决于其配置。在友好配置中，我们将不可映射请求默认为没有特权访问且具有唯一UID的用户“nobody”的凭证。当找不到传入NFS凭证的有效映射时，不友好服务器返回NFS访问错误。

我们的新系统调用用于从内核驻留映射添加和删除条目。它还能够刷新映射到服务器系统上特定UID的所有条目，或刷新给定CLIENT-IP-ADDRESS中的所有条目。

我们修改了装载守护程序（它在服务器系统上处理NFS装载请求）以接受新的事务类型，即Kerberos身份验证映射请求。基本上，作为安装过程的一部分，客户端系统在工作站上提供Kerberos身份验证器以及其UID-ON-CLIENT（在Kerberos身份验证器中加密）的指示。服务器的装载守护程序将Kerberos主体名称转换为本地用户名。然后，在特殊文件中查找此用户名，以生成用户的UID和GID列表。为提高效率，此文件是以用户名为密钥的ndbm数据库文件。根据此信息，将构建NFS凭证并将其作为此请求的<CLIENT-IP-ADDRESS， CLIENT-UID>元组的有效映射交给内核。

在卸载时，会向装载守护程序发送请求，以从内核中删除之前添加的映射。也可以在注销时发送请求，使有关服务器上当前用户的所有映射失效，从而在工作站可供下一用户使用之前清除所有剩余的映射（尽管它们不应存在）。

修改后的 NFS 的 Kerberos 隐含安全问题

此实施并不完全安全。首先，用户数据仍以未加密的形式通过网络发送，因此可以拦截。每事务低级身份验证基于请求数据包中未加密的<CLIENT-IP-ADDRESS， CLIENT-UID>对。此信息可能会被伪造，从而影响安全。但是，应注意，只有当用户主动使用其文件（即，登录时）时，才存在有效映射，因此这种攻击形式仅限于有关用户登录时。当用户未登录时，任何数量的IP地址伪造都不允许未经授权访问其文件。

Kerberos 参考文献

1. S.P. Miller， B.C. Neuman， J.I. Schiller和J.H. Saltzer， E.2.1节：Kerberos身份验证和授权系统，麻省理工学院项目雅典娜，马萨诸塞州剑桥（1987年12月21日）。
2. E. Balkovich、S.R. Lerman和R.P. Parmelee，“高等教育中的计算：雅典娜体验》，ACM通讯，第28(11)卷，第1214-1224页，ACM（1985年11月）。
3. R.M. Needham和M.D. Schroeder，《在大型计算机网络中使用加密进行身份验证》，ACM通讯，第21(12)卷，第993-999页（1978年12月）。
4. V.L. Voydock和S.T. Kent，《高级网络协议中的安全机制》计算调查，第15(2)卷，ACM（1983年6月）。
5. 美国国家标准局，“数据加密标准”，联邦信息处理标准出版物46，华盛顿政府印刷办公室(1977)。
6. SP Dyer，《赫西奥德》，于1988年冬季在乌塞尼斯会议上发表。
7. W.J. Bryant，Kerberos程序员教程，MIT Project Athena（正在准备）。
8. W.J. Bryant，Kerberos管理员手册，麻省理工学院项目雅典娜（正在筹备中）。
9. G.W. Treese，“1000工作站上的Berkeley Unix:Athena Changes to 4.3BSD”，在Usenix会议记录（1988年冬季）中。
10. C.A. DellaFera， M.W. Eichin， R.S. French， D.C. Jedlinsky， J.T. Kohl和W.E. Sommerfeld， The Zephyr通知系统，在Usenix会议记录（1988年冬季）中。
11. M.A. Rosenstein、D.E. Geer和P.J. Levine，在Usenix会议记录（冬季，1988年）。
12. R. Sandberg， D. Goldberg， S. Kleiman， D. Walsh和B. Lyon，《太阳网络文件系统的设计和实施》，在Usenix会议（1985年夏季）中。

相关信息

- Kerberos 支持页
- 技术支持和文档 - Cisco Systems