# Kerberos 概覽 - 適用於開放式網路系統的驗證服務

## 目錄

# 簡介

在開放網路運算環境中，對於無法信任其能正確識別網路服務使用者的工作站，Kerberos 提供另一種方式，就是透過使用可信任的第三方驗證服務，來驗證用戶身分。本文件概述如何在 MIT 的 Athena 專案中實作 Kerberos 驗證模型，其中說明了用戶端、伺服器和 Kerberos 用於完成驗證的通訊協定，以及資料庫必備的管理和複寫。本文件亦說明使用者、程式設計人員和管理員可檢視的 Kerberos 檢視畫面。最後，則以宏觀角度說明 Kerberos 在 Athena 中扮演的角色，並提供目前將 Kerberos 用於使用者驗證的應用程式清單。至於在 Sun 網路檔案系統新增 Kerberos 驗證的情形，則視為在現有應用程式中整合 Kerberos 的個案研究加以說明。

# Kerberos作者


- Jennifer G. Steiner， Project Athena， Massachusetts Institute of Technology， Cambridge，MA 02139, steiner@ATHENA.MIT.EDU
- Clifford Neuman，電腦科學系，FR-35，華盛頓大學西雅圖，WA 98195, bcn@CS.WASHINGTON.EDU。Clifford Neuman在Kerberos的設計和初始實施階段是Project Athena團隊的成員。
- Jeffrey I. Schiller， Project Athena， Massachusetts Institute of Technology， Cambridge，MA 02139, jis@ATHENA.MIT.EDU


# Kerberos簡介

本文對Miller和Neuman設計的認證系統Kerberos進行了綜述。，並介紹我們在MIT的Project Athena使用它的經驗。在動機一節中，我們解釋了開放網路需要新的身份驗證模型的原因及其要求。什麼是Kerberos?一節列出了Kerberos軟體的元件，並描述了它們在提供身份驗證服務時如何互動。在Kerberos Names部分中，我們描述了Kerberos命名方案。

Kerberos如何工作提供了Kerberos身份驗證的構建塊 — 票證和身份驗證器。這會引發對兩種身份驗證協定的討論：使用者對Kerberos的初始身份驗證（類似於登入），以及網路服務的潛在消費者和潛在生產者的相互身份驗證協定。

Kerberos需要客戶端相關資訊的資料庫；「Kerberos資料庫」部分介紹了資料庫、資料庫管理以及資料庫修改協定。Kerberos From the Outside Looking In一節向其使用者、應用程式程式設計師和管理員介紹了Kerberos介面。在大圖部分，我們描述了Project Athena Kerberos如何適應於Athena環境的其餘部分。我們還描述了不同Kerberos身份驗證域或領域的互動；在本例中，Project Athena Kerberos和MIT電腦科學實驗室中運行的Kerberos之間的關係。

在「Issues and Open Problems」部分，我們提到尚未解決的問題和問題。最後一節介紹了Project Athena中Kerberos的當前狀態。在附錄中，我們詳細描述了Kerberos如何應用於網路檔案服務，以驗證希望訪問遠端檔案系統的使用者。

# Kerberos概念

在本文中，我們使用了可能不明確、不為讀者所理解的術語，或在其他地方使用不同的術語。下面我們說明我們使用這些術語。

*使用者*、*客戶端*、*服務器* — 按使用者來說，我們指的是使用程式或服務的人員。客戶也使用某種東西，但不一定是人；它可以是一個程式。網路應用通常由兩部分組成：一個程式運行在一台機器上

並請求遠端服務，另一個程式運行在遠端機器上並執行該服務。我們分別稱之為應用程式的客戶端和伺服器端。通常，客戶端會代表使用者聯絡伺服器。

使用Kerberos系統的每個實體（無論是使用者還是網路伺服器）在某種意義上都是客戶端，因為它使用Kerberos服務。因此，為了將Kerberos客戶端與其它服務的客戶端區分開來，我們使用術語主體來表示這樣一個實體。請注意，Kerberos主體可以是使用者或伺服器。（我們將在後續部分介紹Kerberos主體的命名。）

*服務與伺服器* — 我們將服務用作要執行的一些操作的抽象規範。執行這些操作的進程稱為伺服器。在指定的時間，可能有多台伺服器（通常運行在不同電腦上）執行指定的服務。例如，在Athena，我們的每台分時電腦上都運行一個BSD UNIX rlogin伺服器。

*金鑰*、*私鑰*、*密碼*- Kerberos使用私鑰加密。每個Kerberos主體都分配了一個大數字，即只有該主體和Kerberos才知道的私鑰。對於使用者，私鑰是應用於使用者密碼的單向函式的結果。我們使用金鑰作為私鑰的簡寫。

*Credentials* — 很遺憾，該詞對Sun網路檔案系統和Kerberos系統都有特殊含義。我們明確指出我們指的是NFS憑證還是Kerberos憑證，否則該術語在正常英語語言意義上使用。

*Master and Slave* — 可以在多台電腦上運行Kerberos身份驗證軟體。但是，Kerberos資料庫始終只有一個最終副本。存放此資料庫的機器稱為主機器，或者只是主機器。其他電腦可能擁有Kerberos資料庫的只讀副本，這些副本稱為從電腦。

## Kerberos背後的動機

在非聯網的個人計算環境中，可以通過物理保護個人電腦來保護資源和資訊。在分時計算環境中，作業系統保護使用者彼此之間並控制資源。為了確定每個使用者能夠讀取或修改的內容，分時系統需要識別每個使用者。在使用者登入時完成此操作。

在需要來自多台獨立電腦的服務的使用者網路中，可以採用三種方法來進行訪問控制：依靠使用者登入的電腦來防止未經授權的訪問，什麼也做不了；可以要求主機證明自己的身份，但請信任主機的關於使用者身份的資訊；或者可以要求使用者證明其每個所需服務的身份。

在所有的機器都受到嚴格控制的封閉環境中，可以使用第一種方法。當組織控制通過網路通訊的所有主機時，這是一種合理的方法。

在更開放的環境中，您可能只能有選擇地信任受組織控制的主機。在這種情況下，必須要求每台主機證明其身份。rlogin和rsh程式使用此方法。在這些協定中，身份驗證通過檢查建立連線的Internet地址來完成。

在Athena環境中，我們必須能夠處理不受組織控制的主機的請求。使用者完全控制其工作站：他們可以重新啟動它們，將它們獨立啟動，甚至從自己的磁帶上啟動。因此，必須採取第三種方法；使用者必須證明其每個所需服務的身份。伺服器還必須證明其身份。對運行網路伺服器的主機進行物理保護是不夠的；網路中的其他位置可能有人偽裝成給定伺服器。

我們的環境對識別機制提出了幾項要求。首先，它必須是安全的。繞過它必須非常困難，以至於潛在的攻擊者發現身份驗證機制不是薄弱環節。監視網路的人員應該無法獲得模擬其他使用者所需的資訊。其次，它必須是可靠的。訪問許多服務將取決於身份驗證服務。如果不可靠，整個服務系統就不會可靠。第三，它應該是透明的。理想情況下，使用者不應知道正在發生的身份驗證。最後，它應該具有可擴充性。許多系統可以與Athena主機通訊。並非所有這些都能支援我們的機制，但如果它們支援我們的機制，軟體就不應崩潰。

Kerberos是我們滿足上述要求的工作成果。當使用者轉到工作站時，會登入。就使用者所知，此初始標識足以在登入會話期間向所有必需的網路伺服器證明其身份。Kerberos的安全性依賴於幾個身份驗證伺服器的安全性，但不依賴於使用者登入的系統，也不依賴於將要使用的終端伺服器的安全性。認證伺服器向經過正確認證的使用者提供向分散在網路上的伺服器證明其身份的方法。

身份驗證是安全網路環境的基本構建塊。例如，如果伺服器知道某個客戶機的特定身份，它可以決定是否提供服務、是否應該給予使用者特殊許可權、誰應該接收服務的帳單等等。換句話說，授權和計費方案可以建立在Kerberos提供的身份驗證之上，從而產生與單獨的個人電腦或分時系統等效的安全性。

# 什麼是Kerberos?

Kerberos是一種基於Needham和Schroeder提出的模型的可信第三方認證服務。它受信任是因為它的每個客戶端都相信Kerberos對其每個其他客戶端身份的判斷是準確的。已將時間戳（表示當前日期和時間的大數字）新增到原始模型，以幫助檢測重放。當消息從網路中被盜並在以後重新傳送時進行重播。有關重放和其他身份驗證問題的更完整說明，請參見Voydock和Kent。

## Kerberos的作用是什麼？

Kerberos保留其客戶端及其私鑰的資料庫。私鑰是只為Kerberos及其所屬的客戶端所知的大數。如果使用者端是使用者，則密碼為加密密碼。需要身份驗證的網路服務使用Kerberos註冊，希望使用這些服務的客戶端也是如此。在註冊時協商私鑰。

因為Kerberos知道這些私鑰，所以它可以建立消息來使一個客戶端確信另一個客戶端是它所聲稱的真實身份。Kerberos還會生成臨時私鑰（稱為會話金鑰），該私鑰提供給兩個客戶端，而不是其他客戶端。會話金鑰可用於加密雙方之間的消息。

Kerberos提供三個不同的保護級別。應用程式設計人員根據應用程式的需求來決定哪一個是合適的。例如，某些應用程式只要求在網路連線發起時建立真實性，並且可以假設來自給定網路地址的其它消息來自經過身份驗證的一方。經過身份驗證的網路檔案系統使用這種安全級別。

其他應用程式要求驗證每條消息，但不關心是否披露了消息的內容。對於這些情況，Kerberos提供安全消息。然而，私人消息提供了更高級別的安全性，每條消息不僅經過身份驗證，而且經過加密。例如，Kerberos伺服器本身使用私有消息通過網路傳送密碼。

## Kerberos軟體元件

Athena實現包括幾個模組：

- Kerberos應用程式庫
- 加密庫
- 資料庫庫
- 資料庫管理程式
- 管理伺服器
- 驗證伺服器
- db傳播軟體
- 使用者程式
- 應用

Kerberos應用程式庫為應用程式客戶端和應用伺服器提供了一個介面。其中包括用於建立或讀取身份驗證請求的常式以及用於建立安全或私人消息的常式。

Kerberos中的加密基於資料加密標準DES。加密庫實現了這些常式。提供了幾種加密方法，可在速度和安全性之間進行權衡。還提供對DES密碼塊連結(CBC)模式的擴展，稱為傳播CBC模式。在CBC中，錯誤僅通過密碼的當前塊傳播，而在PCBC中，錯誤在消息中傳播。如果發生錯誤，這會使整個消息變得無用，而不僅僅是消息的一部分。加密庫是一個獨立的模組，可以用其他DES實現或不同的加密庫來替換。

另一個可替換模組是資料庫管理系統。目前的Athena資料庫庫實現使用ndbm，儘管Ingres最初是使用。也可以使用其他資料庫管理庫。

Kerberos資料庫的需求很簡單；系統會為每個主體儲存一條記錄，其中包含主體的名稱、私鑰和到期日期以及一些管理資訊。(到期日期是條目失效的日期。它通常設定為註冊時距未來幾年時間。)

其他使用者資訊（如真實姓名、電話號碼等）由另一台伺服器Hesiod名稱伺服器保留。這樣，敏感資訊，即密碼，就可以由Kerberos來處理，使用相當高的安全措施；而Hesiod儲存的非敏感資訊處理方式不同；例如，它可以通過網路以未加密的方式傳送。

Kerberos伺服器使用資料庫庫，管理資料庫的工具也是如此。

管理伺服器（或KDBM伺服器）為資料庫提供讀寫網路介面。程式的客戶端可以在網路上的任何機器上運行。但是，伺服器端必須在裝有Kerberos資料庫的電腦上運行，才能對資料庫進行更改。

另一方面，身份驗證伺服器（或Kerberos伺服器）對Kerberos資料庫執行只讀操作，即主體身份驗證和生成會話金鑰。由於此伺服器不修改Kerberos資料庫，因此它可以在裝有主Kerberos資料庫只讀副本的電腦上運行。

資料庫傳播軟體管理Kerberos資料庫的複製。可以將資料庫的副本放在幾台不同的電腦上，並在每台電腦上運行身份驗證伺服器的副本。這些從屬電腦中的每一個以給定的間隔從主電腦接收Kerberos資料庫的更新。

最後，還有一些終端使用者程式用於登入Kerberos、更改Kerberos密碼以及顯示或銷毀Kerberos票證（稍後將介紹票證）。

## Kerberos名稱

驗證實體的一部分是為實體命名。身份驗證過程是驗證客戶端是否為請求中命名的客戶端。一個名字由什麼構成？在Kerberos中，使用者和伺服器都是命名的。就身份驗證伺服器而言，它們是等效的。名稱由主名稱、例項和領域組成，表示為name.instance@realm。

主要名稱是使用者或服務的名稱。該例項用於區分主名稱的各種變體。對於使用者來說，一個例項可能包含一些特殊許可權，如「root」或「admin」例項。對於Athena環境中的服務，例項通常是運行伺服器的電腦的名稱。例如，rlogin服務在不同主機上具有不同的例項：rlogin.priam是名為priam的主機上的rlogin伺服器。Kerberos票證僅適用於單個命名伺服器。因此，需要單獨的票證才能訪問同一服務的不同例項。領域是維護身份驗證資料的管理實體的名稱。例如，不同的機構可能都有各自的Kerberos機器，它們包含不同的資料庫。它們有不同的Kerberos領域。(在與其他Kerberi的互動中將進一步討論領域。)

## Kerberos的運作方式

本節介紹Kerberos身份驗證協定。如上所述，Kerberos認證模型基於Needham和Schroeder金鑰分發協定。當使用者請求服務時，必須建立其身份。為此，向伺服器提供票證，同時提供票證最初發給使用者而非被盜的證據。通過Kerberos進行身份驗證有三個階段。在第一階段，使用者獲得用於

請求訪問其他服務的憑據。在第二階段，使用者請求對特定服務進行身份驗證。在最後階段，使用者將這些憑證提供給終端伺服器。

## Kerberos憑證

Kerberos身份驗證模型中使用了兩種型別的憑據：票證和驗證器。兩者都基於私鑰加密，但它們使用不同的金鑰進行加密。票證用於在認證伺服器和終端伺服器之間安全地傳遞票證被簽發人的身份。票證還會傳遞一些資訊，這些資訊可用於確保票證的使用者與票證發行對象相同。鑑別器包含附加資訊，當與票證中的附加資訊比較時，該附加資訊證明呈現票證的客戶端與發行票證的客戶端相同。

票證適用於單個伺服器和單個客戶端。它包含伺服器名稱、客戶端名稱、客戶端的Internet地址、時間戳、生存期和隨機會話金鑰。此資訊將使用票證所使用的伺服器金鑰加密。票證發佈後，指定客戶端可以多次使用票證來訪問指定伺服器，直到票證過期。請注意，由於票證在伺服器的金鑰中加密，因此允許使用者將票證傳遞到伺服器，而不必擔心使用者修改票證，這是安全的。

與票證不同，驗證器只能使用一次。每次客戶端要使用服務時，必須生成一個新服務。這不會造成問題，因為使用者端可以建立驗證器本身。身份驗證器包含客戶端名稱、工作站的IP地址和當前工作站時間。驗證器在票證中的會話金鑰中加密。

## 獲取初始Kerberos票證

當使用者走向工作站時，只有一條資訊可以證明他/她的身份：使用者的密碼。與認證伺服器的初始交換被設計為最小化密碼被洩露的可能性，同時不允許使用者在不知道該密碼的情況下正確認證自己。對使用者來說，登入過程與登入到分時系統過程相同。但在幕後，情況則大不相同。

系統會提示使用者輸入其使用者名稱。輸入後，向認證伺服器傳送請求，該請求包含使用者的名稱和被稱為票證授予服務的特殊服務的名稱。

驗證伺服器會檢查它是否知道使用者端。如果是，則生成隨機會話金鑰，該金鑰稍後將在客戶端和票證授予伺服器之間使用。然後，它為票證授予伺服器建立一個票證，其中包含客戶端的名稱、票證授予伺服器的名稱、當前時間、票證的生存時間、客戶端的IP地址以及剛建立的隨機會話金鑰。所有這一切都以只有票證授予伺服器和身份驗證伺服器已知的金鑰加密。

然後，身份驗證伺服器將票證連同隨機會話金鑰的副本和一些附加資訊傳送回客戶端。此響應在客戶端的私鑰中加密，只有Kerberos和客戶端知道，該私鑰是從使用者密碼派生的。

客戶端收到響應後，會要求使用者輸入其密碼。密碼被轉換為DES金鑰並用於解密來自身份驗證伺服器的響應。票證和會話金鑰連同一些其它資訊被儲存以供將來使用，並且使用者的密碼和DES金鑰被從儲存器中擦除。

一旦交換完成，工作站就擁有資訊，它可用於在票證授予票證的生存期內證明其使用者的身份。只要工作站上的軟體之前未被篡改，就不會存在允許他人假冒使用者超出票證生命週期的資訊。

## 請求Kerberos服務

現在，我們假設使用者已經擁有所需伺服器的票證。為了訪問伺服器，應用程式會構建一個包含客戶端名稱、IP地址以及當前時間的身份驗證器。然後，在隨伺服器票證接收的會話金鑰中加密身份驗證器。然後，客戶端將驗證器和票證一起按單個應用程式定義的方式傳送到伺服器。

一旦伺服器已經接收到驗證器和票證，伺服器解密票證，使用票證中包含的會話金鑰解密驗證器

，比較票證中的資訊與驗證器中的資訊、接收請求的IP地址以及當前時間。如果所有內容都匹配，則允許請求繼續。

假定時鐘在幾分鐘內同步到。如果請求中的時間過長，則在將來或過去，伺服器會將該請求視為重放先前請求的嘗試。還允許伺服器跟蹤所有過去請求，其時間戳仍然有效。為了進一步阻止重放攻擊，可以丟棄與已接收相同的票證和時間戳接收的請求。

最後，如果客戶端指定它希望伺服器也證明其身份，則伺服器會向客戶端在身份驗證器中傳送的時間戳新增一個，加密會話金鑰中的結果，並將結果傳送回客戶端。

在此交換結束時，伺服器確定，根據Kerberos，客戶端是它所說的。如果發生相互驗證，則客戶端也確信伺服器是可信的。此外，客戶端和伺服器共用一個他人不知道的金鑰，並且可以安全地假設在該金鑰中加密的合理的最近的消息源自另一方。

### 獲取Kerberos伺服器票證

回想一下，票證只適用於單個伺服器。因此，必須為客戶想要使用的每項服務獲取單獨的票證。可以從票證授予服務獲取單個伺服器的票證。由於票證授予服務本身就是一項服務，因此它使用上一節中介紹的服務訪問協定。

當程式需要尚未請求的票證時，它會將請求傳送到票證授予伺服器。該請求包含請求票證的伺服器的名稱，以及票證授予票證和構建的驗證器（如上一節所述）。

然後，票證授予伺服器按照上述步驟檢查驗證器和票證授予票證。如果有效，則票證授予伺服器生成要在客戶端和新伺服器之間使用的新隨機會話金鑰。然後，它將為新伺服器構建票證，其中包含客戶端名稱、伺服器名稱、當前時間、客戶端的IP地址以及剛生成的新會話金鑰。新票證的生存期是票證授予票證的剩餘生存期的最小值和服務的預設值。

然後，票證授予伺服器將票證連同會話金鑰和其他資訊傳送回客戶端。但是，這一次，回覆將在作為票證授予票證一部分的會話金鑰中加密。如此一來，使用者便無需再次輸入其密碼。

## Kerberos資料庫

到目前為止，我們已經討論了需要對Kerberos資料庫進行只讀訪問的操作。這些操作由身份驗證服務執行，該服務可在主電腦和從電腦上運行。

在本節中，我們將討論需要對資料庫進行寫訪問的操作。這些操作由名為Kerberos資料庫管理服務(KDBM)的管理服務執行。 目前的實現規定只能對主Kerberos資料庫進行更改；從屬副本是只讀的。因此，KDBM伺服器只能在主Kerberos電腦上運行。

請注意，儘管仍可能進行身份驗證（在從屬電腦上），但如果主電腦已關閉，則無法處理管理請求。根據我們的經驗，這並沒有帶來問題，因為行政要求很少。

KDBM處理使用者更改密碼的請求。此程式的客戶端（通過網路向KDBM傳送請求）是kpasswd程式。KDBM還接受Kerberos管理員的請求，管理員可以向資料庫中新增承擔者，並更改現有承擔者的口令。管理程式的客戶端是kadmin程式，該程式也通過網路向KDBM傳送請求。

### KDBM伺服器

KDBM伺服器接受向資料庫新增承擔者或更改現有承擔者的口令的請求。此服務的唯一性在於，票證授予服務不會為其頒發票證。相反，必須使用身份驗證服務本身（用於獲取票證授予票證的相同

服務）。 此程式的目的是要求使用者輸入密碼。如果不是，那麼如果使用者離開其工作站而無人看管，則過路人可以走上前更改其密碼，這是應該防止的。同樣，如果管理員未保護其工作站，則密碼可以更改系統中的任何密碼。

當KDBM伺服器接收到請求時，它通過將更改的請求者的已驗證主體名稱與請求的目標主體名稱進行比較來授權該請求。如果相同，則允許請求。如果它們不同，KDBM伺服器會查詢訪問控制清單（儲存在主Kerberos系統上的檔案中）。 如果在此檔案中找到請求者的主體名稱，則允許該請求，否則將拒絕該請求。

按照慣例，具有NULL例項（預設例項）的名稱不會出現在訪問控制清單檔案中；而是使用管理員例項。因此，要使使用者成為Kerberos管理員，必須建立該使用者名稱的管理員例項，並將其新增到訪問控制清單中。此約定允許管理員對Kerberos管理使用不同的密碼，然後管理員將使用該密碼進行正常登入。

記錄對KDBM程式的所有請求（無論是允許還是拒絕）。

# kadmin和kpasswd程式

Kerberos的管理員使用kadmin程式將主體新增到資料庫中，或更改現有主體的密碼。管理員在呼叫kadmin程式時需要輸入其管理員例項名稱的密碼。此密碼用於獲取KDBM伺服器的票證。

使用者可以使用kpasswd程式更改其Kerberos密碼。在呼叫程式時，需要輸入舊密碼。此密碼用於獲取KDBM伺服器的票證。

## Kerberos資料庫複製

每個Kerberos領域都有一個主Kerberos電腦，該電腦包含身份驗證資料庫的主副本。在系統中其他位置的從屬機器上擁有資料庫的附加只讀副本是可能的（儘管不是必需的）。有多個資料庫副本的優點通常是指用於複製的優勢：更高的可用性和更好的效能。如果主電腦關閉，仍可以在其中一個從電腦上實現身份驗證。在幾台機器中的任意一台機器上執行身份驗證的能力降低了主機器出現瓶頸的可能性。

保留資料庫的多個複製會帶來資料一致性問題。我們發現非常簡單的方法足以處理不一致問題。每小時都會轉儲主資料庫。該資料庫將全部傳送到從屬電腦，從屬電腦隨後更新其自己的資料庫。主主機上的程式（稱為kprop）將更新傳送給每台從電腦上運行的名為kprop的對等程式。第一個kprop傳送它即將傳送的新資料庫的校驗和。校驗和在Kerberos主資料庫和從Kerberos電腦都擁有的Kerberos主資料庫金鑰中加密。然後，資料通過網路傳輸到從屬機器上的kpropd。從傳播伺服器計算其已接收資料的校驗和，如果它與主伺服器傳送的校驗和匹配，則使用新資訊更新從資料庫。

Kerberos資料庫中的所有密碼均以主資料庫金鑰加密。因此，通過網路從主資料庫傳遞到從資料庫的資訊對竊聽者沒有用。但是，從裝置必須只接受來自主主機的資訊，並且必須檢測資料篡改，從而檢測校驗和。

# Kerberos從外部查詢

本部分從實際應用的角度對Kerberos進行了描述，首先從使用者角度出發，然後從應用程式設計人員的角度出發，最後通過Kerberos管理員的任務進行描述。

## Kerberos使用者的觀點

如果一切順利，使用者幾乎不會注意到Kerberos的存在。在我們的UNIX實現中，票證授予票證是作為登入過程的一部分從Kerberos獲取的。更改使用者的Kerberos密碼是passwd程式的一部分。當使用者註銷時，將自動銷毀Kerberos票證。

如果使用者的登入會話持續時間超過票證授予票證的生存期（當前為8小時），使用者將注意到Kerberos的存在，因為下次執行Kerberos驗證的應用程式時，該應用程式將失敗。它的Kerberos票證將過期。此時，使用者可以運行kinit程式來獲得票證授予伺服器的新票證。與登入時一樣，必須提供密碼才能獲取該密碼。出於好奇而執行klist命令的使用者可能會對以她/他的名義為需要Kerberos身份驗證的服務默示獲取的所有票證感到驚訝。

## 從程式設計師的角度看克里伯斯

編寫Kerberos應用程式的程式設計師通常會將身份驗證新增到由客戶端和伺服器端組成的現有網路應用程式。我們把這個過程稱為「Kerberizing」程式。Kerberos化通常涉及呼叫Kerberos庫，以便在初始服務請求時執行身份驗證。它還可能涉及呼叫DES庫以加密隨後在應用客戶端和應用伺服器之間傳送的消息和資料。

最常用的庫函式是客戶端上的krb_mk_req和伺服器端的krb_rd_req。krb_mk_req常式將請求的目標伺服器的名稱、例項和領域以及要傳送的資料的校驗和作為引數。然後，客戶端通過網路將krb_mk_req呼叫返回的消息傳送到應用程式的伺服器端。伺服器收到此消息時，會呼叫庫常式krb_rd_req。該常式會返回對發件人所聲稱身份的真實性的判斷。

如果應用程式要求客戶端和伺服器之間傳送的消息是保密的，則可以對krb_mk_priv(krb_rd_priv)進行庫呼叫，以加密（解密）會話金鑰中雙方現在共用的消息。

## Kerberos管理員的作業

Kerberos管理員的作業開始於運行程式以初始化資料庫。必須運行另一個程式以在資料庫中註冊基本主體，例如Kerberos管理員的名稱與管理員例項。必須啟動Kerberos身份驗證伺服器和管理伺服器。如果存在從屬資料庫，管理員必須安排定期啟動將資料庫更新從主資料庫傳播到從屬資料庫的程式。

完成這些初始步驟後，管理員使用kadmin程式通過網路操作資料庫。通過該程式，可以新增新的承擔者，也可以更改密碼。

具體來說，在將新的Kerberos應用程式新增到系統中時，Kerberos管理員必須執行幾個步驟才能使其正常工作。伺服器必須在資料庫中註冊，並分配一個私鑰（通常是自動生成的隨機金鑰）。 然後，必須從資料庫中提取某些資料（包括伺服器的金鑰），並將其安裝在伺服器電腦上的檔案中。預設檔案為/etc/srvtab。伺服器呼叫的krb_rd_req庫常式（請參見上一節）使用該檔案中的資訊解密在伺服器的私鑰中加密傳送的消息。/etc/srvtab檔案驗證伺服器，作為在終端上鍵入的密碼驗證使用者。

Kerberos管理員還必須確保Kerberos電腦在物理上是安全的，並且最好是維護Master資料庫的備份。

# Kerberos的大局

在本節中，我們介紹Kerberos如何適應Athena環境，包括它被其他網路服務和應用程式使用，以及它如何與遠端Kerberos領域互動。有關雅典娜環境的更完整描述，請參見G.W. Treese。

## 其他網路服務使用Kerberos

已修改多個網路應用程式以使用Kerberos。rlogin和rsh命令首先嘗試使用Kerberos進行身份驗證。具有有效Kerberos票證的使用者可以登入到另一台Athena電腦，而無需設定.rhosts檔案。如果Kerberos驗證失敗，程式將回退到其常用的授權方法，在本例中為.rhosts檔案。

我們修改了郵局協定，使用Kerberos對希望從「郵局」檢索其電子郵件的使用者進行身份驗證。Athena最近開發了一個名為Zephyr的消息傳遞程式，它使用Kerberos進行身份驗證。

註冊新使用者的程式稱為註冊程式，它使用服務管理系統(SMS)和Kerberos。它通過SMS確定潛在的新雅典娜使用者輸入的資訊（例如姓名和MIT標識號）是否有效。然後它會使用Kerberos檢查所請求的使用者名稱是否唯一。如果一切順利，則會向Kerberos資料庫建立一個新條目，其中包含使用者名稱和密碼。

有關使用Kerberos保護Sun網路檔案系統的詳細討論，請參閱附錄。

### 與其他Kerberi的互動

預期不同的管理組織將希望使用Kerberos進行使用者身份驗證。此外，在許多情況下，一個組織的使用者希望使用另一個組織的服務。Kerberos支援多個管理域。Kerberos中的名稱規範包含一個名為領域的欄位。此欄位包含要在其中對使用者進行身份驗證的管理域的名稱。

服務通常在單個領域註冊，並且只接受該領域的身份驗證伺服器頒發的憑據。使用者通常在單一領域（本地領域）中註冊，但他/她可以藉助本地領域提供的身份驗證獲取另一個領域（遠端領域）頒發的憑據。在遠端領域有效的憑據指示使用者最初經過身份驗證的領域。遠端領域中的服務可以選擇是否採用這些憑據，具體取決於所需的安全級別以及最初驗證使用者時領域中的信任級別。

為了執行跨領域身份驗證，每對領域的管理員必須選擇在各自領域之間共用的金鑰。然後，本地領域中的使用者可以從本地認證伺服器為遠端領域中的票證授予伺服器請求票證授予票證。當使用票證時，遠端票證授予伺服器識別該請求不是來自其自己的領域，並且它使用先前交換的金鑰來解密票證授予票證。然後，它將按正常方式發出票證，不同之處在於，客戶端的realm（領域）欄位包含最初驗證客戶端的領域的名稱。

這種方法可以擴展，允許一個人通過一系列領域驗證自己，直到到達具有所需服務的領域。不過，為了做到這一點，有必要記錄整個採用的路徑，而不僅僅是使用者進行身份驗證的初始領域的名稱。在這種情況下，伺服器所知道的全部是A說B說C說使用者是如此等等。僅當沿途的所有人都同樣受信任時，才能信任此語句。

## Kerberos問題和開放問題

與Kerberos驗證機制相關的許多問題和開放問題。問題包括如何確定票證的正確生命週期、如何允許代理以及如何保證工作站的完整性。

機票壽命問題就是如何在安全性和便利性之間做出恰當的權衡。如果票證的使用時間較長，那麼如果票證及其相關會話金鑰被盜或放錯位置，則可以使用更長時間。如果使用者忘記從公共工作站註銷，此類資訊可能會被竊取。或者，如果一個使用者在允許多個使用者的系統上通過身份驗證，則另一個具有超級使用者許可權的使用者可能會找到使用盜版票證所需的資訊。不過，為票證提供較短生命週期的問題在於，票證到期後，使用者必須獲得新的票證，而新票證需要使用者再次輸入密碼。

一個開放的問題是代理問題。經過身份驗證的使用者如何允許伺服器代表其獲取其他網路服務？這一點很重要的一個示例是使用直接從檔案伺服器訪問受保護檔案的服務。此問題的另一個示例是我們稱為身份驗證轉發。如果使用者登入到工作站並登入到遠端主機，則當使用者能夠在遠端主機上

運行程式時訪問本地提供的相同服務時，情況會更好。造成此困難的原因在於使用者可能不信任遠端主機，因此身份驗證轉發在所有情況下都是不可取的。目前我們還沒有解決這個問題的辦法。

另一個問題（在Athena環境中很重要的一個問題）是如何保證在工作站上運行的軟體的完整性。在專用工作站上，這並不是什麼大問題，因為要使用它的使用者可以控制它。但在公共工作站上，可能有人走了進來，修改了登入程式以儲存使用者的密碼。目前在我們環境中可用的唯一解決方案是讓人們難以修改在公共工作站上運行的軟體。更好的解決方案要求使用者的金鑰從不離開使用者知道可以信任的系統。一種方法是，如果使用者擁有能夠執行身份驗證協定中要求的加密的智慧卡，就可以實現這一點。

## Kerberos狀態

Kerberos的原型版本於1986年9月投入生產。自1987年1月以來，Kerberos一直是Project Athena唯一驗證其5,000個使用者、650個工作站和65台伺服器的手段。此外，現在使用Kerberos來代替.rhosts檔案來控制幾個Athena分時系統中的訪問。

## Kerberos確認

Kerberos最初由Steve Miller和Clifford Neuman設計，由Jeff Schiller和Jerry Saltzer提出。從那時起，許多其他人參與了這個專案。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、Brian Murphy、John Ostlund Ken Raeburn、Chris Reed、Jon Rochlis、Mike Shanzer、Bill Sommerfeld、Ted T'so、Win Treese和Stan Zanarotti。

我們感謝Dan Geer、Kathy Lieben、Josh Lubarr、Ken Raeburn、Jerry Saltzer、Ed Steiner、Robbert van Renesse和Win Treese的建議，他們的建議大大改進了本文的初稿。

Jedlinsky、J.T. Kohl和W.E. Sommerfeld在Usenix會議記錄（1988年，冬季）中寫道：「The Zephyr Notification System」。

M.A. Rosenstein、D.E. Geer和P.J. Levine，在Usenix會議記錄(Winter， 1988)中。

R. Sandberg， D. Goldberg， S. Kleiman， D. Walsh和B. Lyon，「Sun Network Filesystem的設計與實現」，在Usenix Conference Proceedings（1985年夏）中。

## 附錄：Kerberos在SUN網路檔案系統(NFS)中的應用

Project Athena工作站系統的一個關鍵元件是使用者工作站與其專用檔案儲存（主目錄）之間的網路介面。 所有專用儲存都位於專用於此目的的一組電腦（當前為VAX 11/750）上。這允許我們在公共可用的UNIX工作站上提供服務。當使用者登入到這些公共工作站之一時，而不是根據本地駐留的密碼檔案驗證其名稱和密碼，我們使用Kerberos來確定其真實性。登入程式會提示輸入使用者名稱（如同在任何UNIX系統上）。 此使用者名稱用於獲取Kerberos票證授予票證。登入程式使用密碼生成用於解密票證的DES金鑰。如果解密成功，通過諮詢Hesiod命名服務找到使用者的主目錄，並通過NFS進行裝載。然後，登入程式將控制權移交給使用者的shell，然後shell就可以運行傳統的每使用者自定義檔案，因為主目錄現在已「連線」到工作站。Hesiod服務還用於在本地密碼檔案中構建條目。（這有利於在/etc/passwd中查詢資訊的程式。）

從多個提供遠端檔案服務的選項中，我們選擇了Sun的網路檔案系統。然而，這個系統未能以關鍵的方式滿足我們的需求。NFS假定所有工作站都歸為兩類（從檔案伺服器的角度進行檢視）：可信和不可信。不受信任的系統根本無法訪問任何檔案，受信任可以。可信系統完全可信。假設可信系

統是由友好管理來管理的。具體來說，可以從受信任的工作站偽裝成檔案服務系統的任何有效使用者，從而獲得對系統上幾乎每個檔案的訪問許可權。（只有「root」擁有的檔案才會免除。）

在我們的環境中，工作站的管理（在傳統的UNIX系統管理意義上）由當前正在使用它的使用者負責。我們不對工作站上的根密碼保密，因為我們意識到，一個真正不友好的使用者可能因為他/她與電腦位於同一物理位置而破解，而且能夠訪問所有控制檯功能。因此我們無法真正信任我們的工作站在NFS解釋中的信任。為了允許環境中適當的訪問控制，我們必須對基本NFS軟體進行一些修改，並將Kerberos整合到方案中。

## Kerberos未修改NFS

在我們開始實施的NFS中（來自威斯康辛大學），身份驗證以每個NFS請求中包含的資料片段的形式提供（在NFS術語中稱為「憑證」）。 該憑證包含關於請求者的唯一使用者識別符號(UID)的資訊以及請求者的成員資格的組識別符號(GID)的清單。然後NFS伺服器將使用此資訊來進行訪問檢查。受信任工作站與非受信任工作站之間的區別在於NFS伺服器是否接受其憑據。

# Kerberos修改的NFS

在我們的環境中，若且唯若憑證指示工作站使用者的UID時，NFS伺服器必須接受來自工作站的憑證，而不能接受其他憑證。

一個顯而易見的解決方案是改變憑證的性質，從僅僅的UID和GID指示符變成完全吹的Kerberos驗證資料。但是，如果採用此解決方案，將會支付很高的效能損失。在每個NFS操作（包括所有磁碟讀取和寫入活動）上交換憑證。在每筆磁碟事務上新增Kerberos身份驗證會為每個事務新增相當數量的完全加密（在軟體中完成），而且根據我們的信封計算，會提供不可接受的效能。（它還需要將Kerberos庫常式放在核心地址空間中。）

我們需要一種混合方法，如下所述。其基本思想是從客戶端工作站接收的NFS伺服器將憑證對映到伺服器系統上的有效（可能不同）憑證。此對映在每次NFS事務上的伺服器核心中執行，並在「裝載」時由使用者級進程設定，該進程在建立有效的核心憑據對映之前進行Kerberos仲裁身份驗證。

為了實現這一目的，我們向核心新增了一個新的系統呼叫（僅在伺服器系統上需要，而在客戶端系統上不需要），該呼叫提供了對對映功能的控制，該對映功能將來自客戶端工作站的傳入憑據對映到可在伺服器上使用的憑據（如果有）。 基本對映函式對映元組：

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

伺服器系統上的有效NFS憑據。從客戶端系統提供的NFS請求資料包中提取CLIENT-IP-ADDRESS。附註：客戶端生成的憑據中的所有資訊（UID-ON-CLIENT除外）將被丟棄。

如果不存在對映，則伺服器會以兩種方式中的一種進行反應，具體取決於所配置的方式。在我們的友好配置中，我們將不可對映的請求預設為使用者「nobody」的憑證中，該使用者「nobody」沒有特權訪問許可權並具有唯一的UID。當找不到傳入的NFS憑據的有效對映時，不友好的伺服器會返回NFS訪問錯誤。

我們的新系統呼叫用於新增和刪除核心駐留對映中的條目。它還提供了刷新對映到伺服器系統中特定UID的所有條目或刷新指定CLIENT-IP-ADDRESS的所有條目的能力。

我們修改了裝載守護程式（處理伺服器系統上的NFS裝載請求）以接受新的事務型別，即Kerberos身份驗證對映請求。基本上，作為安裝過程的一部分，客戶端系統提供Kerberos身份驗證器以及工作站上他/他的UID-ON-CLIENT（在Kerberos身份驗證器中加密）的指示。伺服器的裝載

守護程式將Kerberos主體名稱轉換為本地使用者名稱。然後，將在特殊檔案中查詢此使用者名稱，以生成使用者的UID和GID清單。為了提高效率，此檔案是一個使用使用者名稱作為金鑰的ndbm資料庫檔案。根據此資訊，會構建NFS憑證並將其作為此請求的<CLIENT-IP-ADDRESS，CLIENT-UID>元組的有效對映傳遞給核心。

在解除安裝時，將向裝載守護進程傳送請求，以從核心中刪除以前新增的對映。也可以在註銷時傳送請求，使有關伺服器上的當前使用者的所有對映無效，從而在工作站可供下一個使用者使用之前清除所有剩餘對映（儘管它們不應存在）。

## 修改後的NFS的Kerberos安全意義

此實施並非完全安全。首先，使用者資料仍以未加密形式通過網路傳送，因此可以擷取。低級別、每事務身份驗證基於請求資料包中未加密的<CLIENT-IP-ADDRESS， CLIENT-UID>對。這些資訊可能被偽造，從而危及安全。但是，應注意，只有當使用者主動使用其檔案（即登入時）時，才有有效的對映存在，因此這種形式的攻擊僅限於有問題的使用者登入時。當使用者未登入時，任何IP地址偽造都不會允許對他/她的檔案進行未經授權的訪問。

## Kerberos引用


1. S.P. Miller、B.C. Neuman、J.I. Schiller和J.H. Saltzer，E.2.1節：Kerberos身份驗證和授權系統，麻省劍橋Athena麻省理工專案（1987年12月21日）。
2. E. Balkovich、S.R. Lerman和R.P. Parmelee，「高等教育中的計算：The Athena Experience，」 Communications of the ACM， Vol. 28(11),pp. 1214-1224, ACM（1985年11月）。
3. R.M. Needham和M.D. Schroeder，「Using Encryption for Authentication in Large Networks of Computers」，《ACM通訊》，第21(12)卷，第993-999頁（1978年12月）。
4. V.L. Voydock和S.T. Kent，「Security Mechanism in High-Level Network Protocols」，《Computing Survey》，第15(2)卷，ACM（1983年6月）。
5. 國家標準局，「資料加密標準」，聯邦資訊處理標準發佈46，政府印刷辦公室，華盛頓(1977)。
6. SP Dyer，「Hesiod」，在Usenix會議記錄（1988年冬季）中。
7. W.J. Bryant，Kerberos程式設計師教程，麻省理工學院雅典娜專案（正在準備）。
8. W.J. Bryant， Kerberos管理員手冊， MIT Project Athena（正在準備）。
9. G.W. Treese，「Berkeley Unix on 1000 Workstations:Athena Changes to 4.3BSD，」在Usenix會議記錄(Winter， 1988)中。
10. C.A. DellaFera、M.W. Eichin、R.S. French、D.C. Jedlinsky、J.T. Kohl和W.E. Sommerfeld，「The Zephyr Notification System」，在Usenix會議記錄(Winter， 1988)中。
11. M.A. Rosenstein、D.E. Geer和P.J. Levine，在Usenix會議記錄(Winter， 1988)中。
12. R. Sandberg， D. Goldberg， S. Kleiman， D. Walsh和B. Lyon，「Sun Network Filesystem的設計與實現」，在Usenix Conference Proceedings（1985年夏）中。


## 相關資訊

- Kerberos支援頁面
- 技術支援與文件 - Cisco Systems