# Module 6: Kerberos

# Module 6:大綱

Module 6-1: 動機(*)

Module 6-2: Kerberos 4(***)

Module 6-3: Kerberos 5(***)

ul

* 初級(basic):基礎性教材內容
**中級(moderate):教師依據學生的吸收情況，選擇性介紹本節的內容
***高級(advanced):適用於深入研究的內容

u1

* 選擇性(optional)介紹的章節:教師依據學生的吸收情況，選擇性介紹本節的內容

** 先進(advanced)的章節:適用於深入研究的內容

user, 2007/1/24

# Module 6-1: 動機(motivation)

# 6-1 動機

- Kerberos身份認證系統，它原本是屬於由美國麻省理工學院所發展的一個雅典娜（Athena）計劃中的一部份，由Miller和Neuman兩位學者負責研發。

- Kerberos是在開放網路或內部網路中所使用的一套身份認證系統。

- 最初 Kerberos 使用於UNIX 作業系統上，然而近年來支援 Kerberos 認證系統的廠商越來越多，如 微軟公司也在其 Windows2000/XP 作業系統以及 Redhat Linux 等皆支援 Kerberos網路身份認證協定。

- 目前Kerberos已發展出兩個版本: Kerberos 4與Kerberos 5。

# Kerberos 的需求

- Kerberos身份鑑別功能的原始需求為:
  - 安全性：
    - » 必需使攻擊者無法經由在網路上竊聽而取得有效的使用者資料，進而偽裝成其他合法使用者。
  - 可靠性：
    - » 採用分散式伺服器架構，讓系統之間能夠相互支援，防止因單一系統失效而導致存取控制認證的空窗期。
  - 透明性：
    - » 除了讓使用者輸入密碼的動作外，使用者毫無感覺認證工作正在運行。
  - 擴充性：
    - » 為了能有效的提供認證服務，最好是將其設計成模組化、分散式的架構。
- Kerberos為以 Needham-Schroeder認證協定為基礎而建置的第三者身份鑑別性協定。

# 6-1 動機

- 設計動機：設計一個適用在分散式網路(可為開放網路或內部網路)，運用集中式的私密鑰匙管理方式，提供主從式架構的第三者身份鑑別性的安全機制
  - 目標是讓使用者可以存取分散式網路上的服務
  - 客戶端須先向伺服端證明自己的身份，身份需要向一個集中式的身份鑑別伺服器(AS)提出申請，伺服端應信任AS提供的客戶端的資料
  - 使用分散式網路上的服務之前，雙方(客戶端與伺服端)需先相互作身份鑑別

# Module 6-2: Kerberos 4

# Kerberos 4 綜觀

- 在 Kerberos 認證系統中總共有四個角色：用戶端，伺服器，身份鑑別伺服器（Authentication Server，AS）及通行證簽發中心（Ticket-Granting Server，TGS）。
- Kerberos 概述
  - 使用者一開始先與身份鑑別伺服端(AS)協商以執行身份鑑別，再透過通行證簽發中心（Ticket-Granting Server，TGS），取得服務伺服器的存取權限
  - 使用者透過由AS發送的TGT (Ticket-Granting Ticket, 通行授與票)，來向TGS 要求取得授權
  - 使用者運用TGS提供的通訊金鑰，與伺服器連線

# Kerberos 4 綜觀

- Kerberos認證系統中是採用DES加密演算法，每一個使用者都會有一把個別的認證金鑰（authentication key），認證金鑰會存在認證伺服器(AS)中。
- 認證伺服器 AS 負責
  - 使用者的身份鑑別
  - 維護使用者和伺服器的資料
- 通行證簽發中心( TGS )專門負責
  - 產生用戶端與伺服器每次通訊時所要使用的通訊金鑰（稱為 session key）

# Kerberos 4 綜觀

- Kerberos認證架構下，認證與授權是獨立的
  - 使用者的身份鑑別，須確認是否真的為該使用者？
  - 確認該使用者在某個時間，是否合法授權使用服務？
- 這種特性與只要經過身分鑑別，就等於獲得授權的Needham-Schroeder認證協定，是完全不同的。
- Kerberos認證，每次的認證與通訊金鑰都會有一個有效期限，這與Needham-Schroeder 的認證是永久有效，也大不相同。
- Kerberos 系統架構如下圖所示：

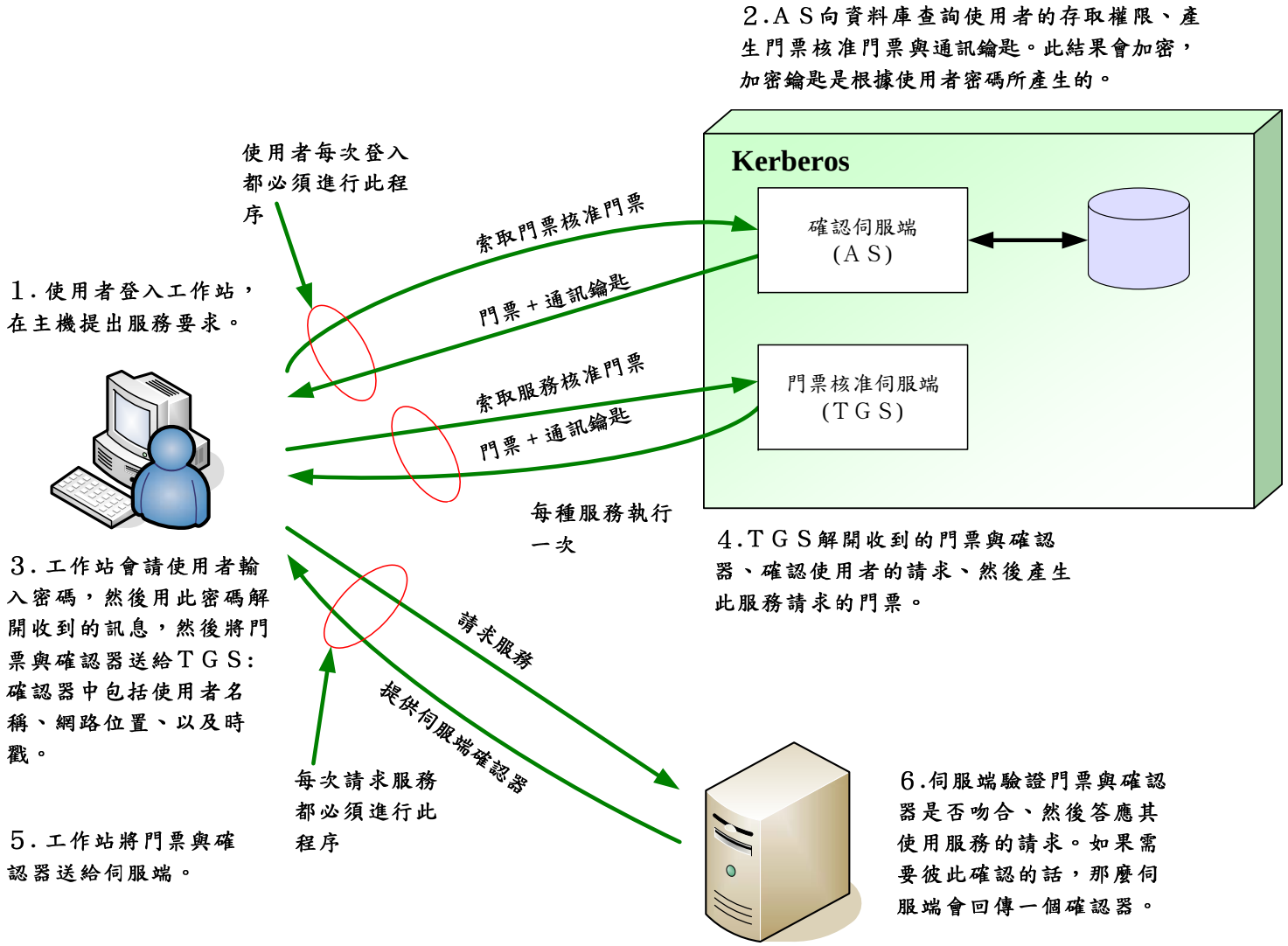


圖6-1 Kerberos 4 系統架構

# Kerberos 領域

- 一個完整的Kerberos 的運作環境稱為「領域(realm)」，具有下列特性：
  - Kerberos4應用於單一的管理領域
  - 由單一個身份鑑別伺服端，及一些客戶端(client)，與應用伺服器所組成
  - 客戶端與伺服端共享私密鑰匙
- 多重Kerberos (Kerberi)可運作於多重領域(Multiple Domain)，此時 Kerberos 伺服端必須彼此信任並共享鑰匙。

# Kerberos 4 運作程序

- 在 Kerberos 系統中，當用戶端要和伺服器通訊時，用戶端會先向身份鑑別伺服器 AS請求一張與通行證簽發中心 TGS 溝通所需要的通行證
  - 該通行證中除了記錄有一把通訊金鑰（session key [Kc,tgs]）
  - 尚包含了用戶端ID、用戶端網路位址、通行證簽發中心 ID、時間戳記及有效期限
- 當用戶端拿到了與 TGS 通訊的通行證後，用戶端使用此通行證再向TGS 要一張與伺服器連線所需要的通行證
  - 此通行證中即包含用戶端和伺服器交談所需要用的通訊金鑰（session key [Kc,v]）

# Kerberos 4 運作程序

- 最後，用戶端利用該通訊金鑰，將它和伺服器之間的通訊內容編碼，以利網路上傳送
  - 值得一提的是，在與TGS 通訊所使用的通行證過期之前，用戶端若要再和別的伺服器連線的話，就只要透過TGS拿伺服器的通行證即可，不必再透過 AS作身份鑑別。

# Kerberos v4認證的詳細流程(**)

- 以下我們介紹Kerberos v4認證協定的詳細流程，總共分為三大步驟：
  - 使用者鑑別交換（Authentication Service Exchange）：使用者先向身份鑑別伺服器證明自己的身份，以便取得與 TGS 通訊所要使用的通行證，稱為 Ticket-Granting Ticket (TGT)。
  - 取得與伺服器通訊的通行證（Ticket-Granting Service Exchange）：使用者向 TGS 取得一張於使用服務時與伺服器連線用的伺服器通行證（Service-Granting Ticket）。
  - 用戶端與伺服器通訊（Authentication Exchange）：使用經由伺服器對伺服器通行證的驗證，取得服務授權許可，使用服務。

# Kerberos v4認證的詳細流程(一) (**)

每個步驟所交換的訊息詳述如下：

使用者身份鑑別交換

步驟1. C→AS：$ID$c|| $ID_{tgs}$||$TS_1$

步驟2. AS→C：$E_{Kc}[Kc,_{tgs}||ID_{tgs}||TS_2||Lifetime_2||\ Ticket_{tgs}]$

其中

- $Ticket_{tgs} = EK_{tgs}[K_{c,tgs}||ID\text{c}||AD\text{c}||ID_{tgs}||TS_2||Lifetime_2]$
- 在上述的步驟1中，$ID$c、$ID_{tgs}$及$TS_1$是用來讓AS驗證用戶端的身份，並且要求與TGS溝通，此訊息是在時間戳記$TS_1$時產生。
- 在步驟2中，所有訊息都使用Kc（用戶端與認證伺服器所共享的密碼）加密傳送給用戶端，以避免資料被別人截取。

# Kerberos v4認證的詳細流程(一) (**)

- $K_{c,tgs}$是由AS所產生的通訊金鑰，用來讓用戶端和TGS溝通用，如此一來用戶端與TGS便不需要共享一個金鑰。
- $ID_{tgs}$是用來表示此通行證是用來與TGS溝通用的。
- $Lifetime_2$用來告訴用戶端$Ticket_{tgs}$的有效期限。
- $TS_2$為另一個時間戳記，用來通知用戶端關於$Ticket_{tgs}$ 的產生的時間。
- $Ticket_{tgs}$ 是給用戶端用來與TGS通訊用的通行證，該通行證是用AS和TGS所共享的金鑰加密，以避免被用戶端竄改資料。

# Kerberos v4認證的詳細流程(二) (**)

取得與伺服器通訊的通行證

步驟3. C→TGS：$IDs||Ticket_{tgs}||Authenticator_c$

步驟4. TGS→C：$EK_{c,tgs}[K_{c,s}||\ IDs\ ||TS_4||Ticket_s]$

- 其中
  - $Ticket_{tgs}=E_{Ktgs}[K_{c,tgs}||ID_c||AD_c||ID_{tgs}\ ||TS_2||Lifetime_2]$
  - $Ticket_s=E_{Ks}[Kc,s||ID_c||AD_c||IDs||TS_4||Lifetime_4]$
  - $Authenticator_c=E_{Kc,tgs}[ID_c||AD_c||TS_3]$
  - *IDs*:在訊息3中用來表示用戶端C要求與伺服器IDs進行通訊
  - $Ticket_{tgs}$:由步驟2中所取得的，用來證明C已通過AS的認證。
  - $Authenticator_c$用來證明自己是$Ticket_{tgs}$的擁有者。

# Kerberos v4認證的詳細流程(二) (**)

- $Ticket_{tgs}$中的$ID_c$可用來表示此通行證是給用戶端使用，其中的$AD_c$為用戶端電腦的網路位址，用來限制只有同樣網路位址的機器才能使用此通行證，這可避免別人用同樣的*ID*並冒用此通行證企圖取得存取權限。
- 通行證中加入$ID_{tgs}$可用來驗證此$Ticket_{tgs}$解密是否成功。
- $TS_2$表示此$Ticket_{tgs}$產生的時間，*Lifetime*2用來表示此$Ticket_{tgs}$的有效期限。
- $Authenticator_c$是由用戶端所產生，其使用期限非常短，用以減少遭受他人重送攻擊（replay attack）的機會。

# Kerberos v4認證的詳細流程(二) (**)

- 其中包括一個$TS_3$時間戳記表示此$Authenticator_c$產生的時間。$ID_c$和$AD_c$都是用來與$Ticket_{tgs}$中的$ID_c$和$AD_c$做比對之用，以證明用戶端的身份。
- 在訊息4中，所有訊息都使用$K_{c,tgs}$加密傳送給用戶端，以避免資料在網路上被別人截取。此訊息包含了$K_{c,s}$以讓用戶端能夠與伺服器端做加解密資料。
- $ID_S$用來表示$Ticket_s$是用來與伺服器S溝通之用。
- $TS_4$時間戳記用來表示此$Ticket_s$的產生時間，而$Ticket_s$係由通行證簽發中心所簽發，用來給用戶端與伺服器通訊之用的通行證。

# Kerberos v4認證的詳細流程(三) (**)

用戶端與伺服器通訊

步驟5. C→S：$Ticket_s||Authenticator_c$

步驟6. S→C：$E_{Kc,s}[TS_5+1]$

其中

- $Ticket_s=E_{Ks}[K_{c,s}||ID_c||AD_c||ID_s||TS_4||\ Lifetime_4]$
- $Authenticator_c=E_{Kc,s}[ID_c|AD_c||TS_5]$

# Kerberos v4認證的詳細流程(三) (**)

- 步驟5與步驟4作用類似，在此不再贅述
- 步驟6是作為用戶端認證伺服器之用，以證明該伺服器有能力解密步驟5的$Authenticator_c$，並回傳另一個時間戳記 $TS_5+1$。
- 在網路遭受攻擊時，有可能會有入侵者企圖假造伺服器以取得用戶端的一些機密資料，因此在用戶端與伺服器端連線時，用戶端也是有必要確認伺服器端的身份，而不是單向的由伺服器端認證用戶端而已。

# Kerberos v4跨網域間的認證(***)

- 在跨網域架構下，各網域的Kerberos認證系統必須存在某種信賴關係才能建立起跨網域的認證
  - 兩個不同領域間的Kerberos伺服器須互相註冊，並共同擁有一把秘密鑰匙
  - 不同領域下的Kerberos伺服器除須彼此信任外，也須相信經由另一領域Kerberos伺服器認證過的使用者

# Kerberos v4跨領域認證流程(***)

以下是Kerberos v4跨網域間認證的詳細流程：

1. 由 AS 取得本地端通行證簽發中心（local TGS）取得通行證

C→AS：$ID_c||ID_{tgs}||TS_1$

AS→C：$E_{Kc}[K_{c,tgs}||ID_{tgs}||TS_2||Lifetime_2||Ticket_{tgs}]$

2. 用戶端請求取得遠端通行證簽發中心（remote TGS）的通行證

C→TGS：

$ID_{tgsrem}||Ticket_{tgs}||Authenticator_c$

TGS→C：

$E_{Kc,tgs}[K_{c,tgsrem}||ID_{tgsrem}||TS_4||Ticket_{tgsrem}]$

# Kerberos v4跨領域認證流程(***)

3. 用戶端請求取得遠端伺服器之通行證

C→TGSrem：$ID_{srem}||Ticket_{tgsrem}||Authenticator_c$

TGSrem→C：$E_{Kc,tgsrem}[K_{c,srem}||ID_{srem}||TS_6||Ticket_{srem}]$

4. 用戶端向遠端伺服器提出服務請求

C→Srem：$Ticket_{srem}||Authenticator_c$

- 下圖為Kerberos在跨網域認證的架構圖。

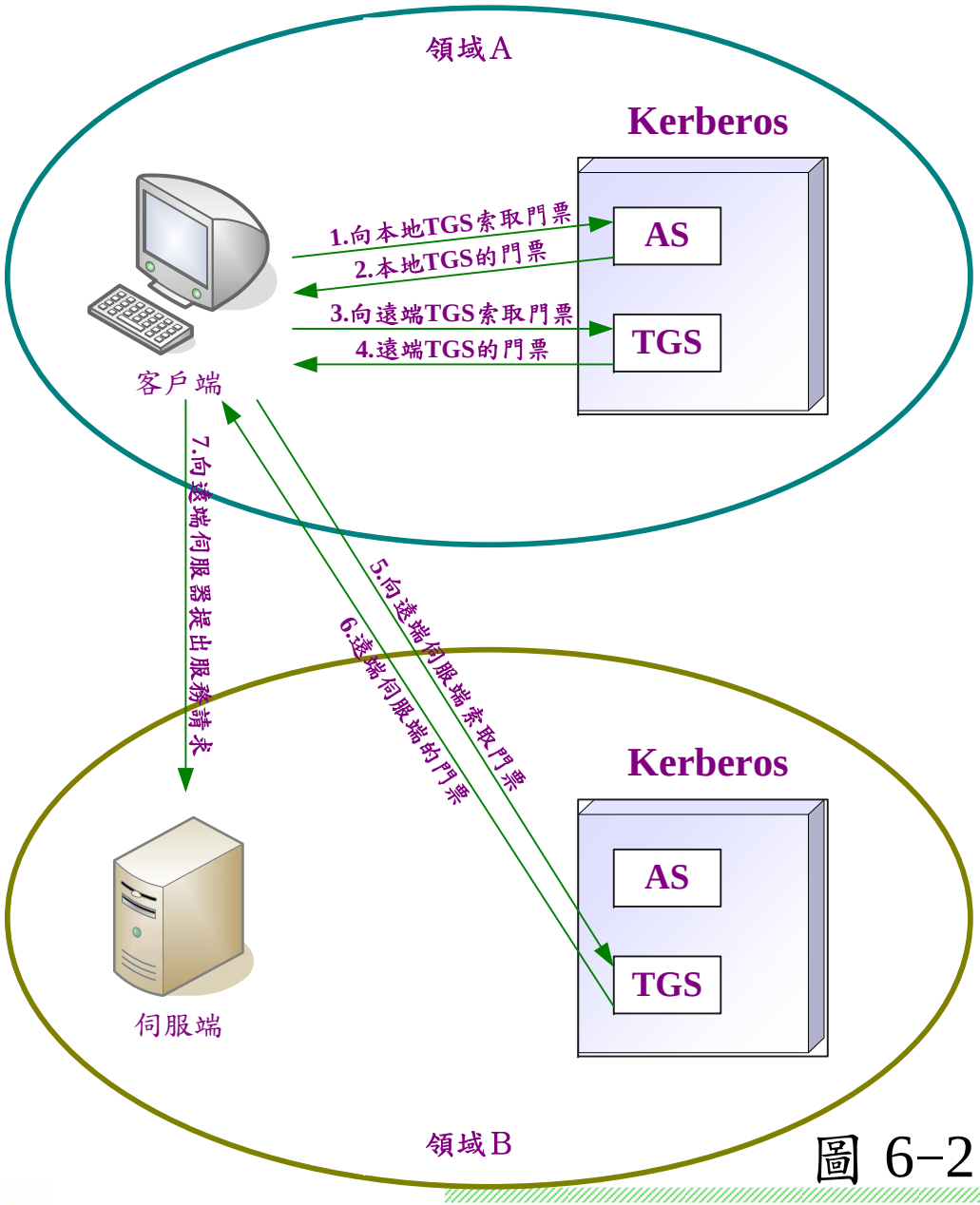


圖 6-2 Kerberos 跨網域認證架構圖

# Module 6-3: Kerberos 5

# 6.3 Kerberos 版本 5 (**)

- 於1990年代中葉發展，改良版本v4的缺點如下
  - 著重在運作環境的缺失
    - 加密方法、網路協定、資料位元組順序、通行證有效期限、認證的轉移、以及跨領域的認證性
  - 技術缺失
    - 雙重加密、非標準模式的使用、通訊鑰匙、以及密碼的破解
- Kerberos v5 加入了一些新的機制以作為一種跨網域的認證協定，並已被指定為網路標準 RFC 1510。

# Kerberos v4與v5比較

- 以下針對Kerberos v4、v5的運作環境及技術二方面予以分析比較。
- 運作環境方面：
  - 加密方法：加密方法Kerberos v4只能支援DES加密法，而在v5中可支援各類的加密方法。
  - 網路協定：各種協定Kerberos v4中只能用在 IP 定址網路上，而v5則能夠使用在其它定址方式(OSI網路)。
  - 資料位元組順序：Kerberos v4中資料位元順序是由其中的一個標號（tags）來決定；而在v5中則採用抽象文法表示(Abstract Syntax Notation One, ASN.1）及基本編碼規則(Basic Encoding Rule, BER）來定義訊息的結構，及決定資料位元順序。

# Kerberos v4與v5比較

- 通行證有效期限：Kerberos v4中，一張通行證的有效期限最多只有21小時限制，而在v5中則可直接指定起用時間及終止時間，因此沒有時間上的限制。
- 認證的轉移：在Kerberos v4中，並沒有代理認證的功能存在，而在v5中則支援此功能。
- 跨網域的認證：Kerberos v5 藉由通行票旗標值的設定，提供更有效的跨網域認證方式。

# Kerberos v4與v5比較

- 技術方面：
  - 雙重加密：Kerberos v4在步驟2與4，傳送給用戶端的通行授予票($Ticket_{tgs}$)及服務授與通行票($Ticket_s$)經過二次加密，此加密不是必要的，造成運算上的浪費，v5則選擇不加密。
  - 非標準模式的使用：Kerberos v4使用的DES不是標準的，而是傳遞式連鎖密文區塊加密模式(Propagating cipher block chaining,PCBC)，此模式證實易遭交換密文區塊攻擊，v5採用標準的密文區段串接加密模式(Cipher block chaining,CBC)。

# Kerberos v4與v5比較

- 通訊鑰匙： Kerberos v4中同一張通行票可能重覆地被拿來對某一個伺服器申請需求，所以可能遭受重送攻擊，v5中用戶端與伺服器可以自行產生一把時效只有一次，只能於那次連線期間才能使用的子連線金鑰(sub key)。
- 密碼的破解： Kerberos v4與v5同樣可能受到破解密碼的攻擊，v5提供一種「事先認證」機制，使破解密碼的困難度增加。

- 以下介紹Kerberos v5認證的基本流程。

# Kerberos v5認證的詳細流程(一) (***)

- 使用者身份鑑別交換

  步驟1. C→AS：$Options||ID_c||ID_{tgs}||Times||Nonce_1$

  步驟2. AS→C：$Realm_c||ID_c||Ticket_{tgs}||EK_c[Kc,tgs||Times||Nonce_1||Realm_{tgs}||ID_{tgs}]$

其中

- $Ticket_{tgs} = E_{Ktgs}[Flags||K_{c,tgs}||Realm_c||ID_c||AD_c||Times]$。
- *Realm*(領域)：代表使用者、TGS或是伺服器所在的領域。
- *Options*(選擇項目)：要求TGS為將要回傳的通行票設定某些旗標值。
- *Times*(時間)：用戶端要求TGS將有關時間的設定以from、till、rtime做調整。
- *Nonce*(隨意值)：確保雙方的回應是最新的，而不是重送攻擊。

# Kerberos v5認證的詳細流程(二) (***)

- 取得與伺服器通訊的通行證

  步驟3. C→TGS：$Options||ID_v||Times||Nonce_2||Ticket_{tgs}$
  $||Authenticator_c$

  步驟4. TGS→C：$Realm_c||ID_c||Ticket_{v|}||E_{Kc,tgs}[K_{c,v}||$
  $Times||Nonce_2||Realm_v||ID_v]$

其中

- $Ticket_{tgs} = E_{Ktgs}[Flags||K_{c,tgs}||Realm_c||ID_c||AD_c||Times]$。
- $Ticket_v = E_{Kv}[Flags||K_{c,v}||Realm_c||ID_c||AD_c||Times]$。
- $Authenticator_c = E_{Kc,tgs}[ID_c||Realm_c||TS_1]$。

# Kerberos v5認證的詳細流程(三) (***)

- 用戶端與伺服器通訊

  步驟5. C→V：$Options||Ticket_v||Authenticator_c$

  步驟6. V→C：$E_{Kc,v}[TS_2||Subkey||Seq\#]$

其中

- $Ticket_v=E_{Kv}[Flags||Kc,v||Realm_c||ID_c||AD_c||Times]$。
- $Authenticator_c=E_{Kc,v}[IDc||Realm_c||TS_2||Subkey||Seq\#]$。
- *Subkey*(子金鑰)：用戶端可選擇是否用另外一把金鑰來加密這次連線間的傳輸資料，若省略，則意謂使用原有的Kc,v來加密。
- *Seq#*(Sequence number,序列號碼)：此為選擇性欄位，伺服器可以利用此設定所有要傳送給用戶端訊息的起始序列號碼，以防止重送攻擊。

# Kerberos v5 旗標值(***)

- Kerberos v5 的通行票旗標值提供擴充性的功能，請詳見於下表。

| 旗標名 | 作用 |
|---|---|
| **INITIAL** | 這張票是由AS所發出，並不是基於通行授與票來發行 |
| **PRE-AUTHENT** | 最初的認證過程中，用戶端需先經過KDC認證後，才能發與通行票 |
| **HW-AUTHENT** | 啟始初始認證協定中所需使用的硬體，只有指名的客戶端才會擁有 |
| **RENEWABLE** | 告知TGS這張票能夠在過期後，用來取得替代的通行票 |
| **MAY-POSTDATED** | 告知TGS這張通行授與通行票可以用來發行"日後的"通行票 |
| **POSTDATED** | 表示這張票已經將有效日期延長，伺服器可以檢查authtime欄位確定最早的認證時間 |
| **INVALID** | 這張票目前是無效的，必須經過KDC使其生效後才能使用 |
| **PROXIABLE** | 告知TGS這張票可以用來允許發行新的並且具有不同網路位置的服務授與通行票 |
| **PROXY** | 表示這張票具有代理人的效用 |
| **FORWARDABLE** | 告知TGS這張票可以用來允許發行新的並且具有不同網路位置的通行授與通行票 |
| **FORWARDED** | 表示這張票可能是經由轉送或者是因為認證中包含轉送的通行授與通行票才被發行的 |

表 6-1 Kerberos v5 旗標值

# ※相關應用趨勢與研發議題

- 作業系統的應用：

  微軟已將Kerberos標準應用在 Windows 2000 Server 及 Windows Server 2003 等伺服器平台，作為用戶端系統與伺服器、伺服器與伺服器之間的認證協定。

  資料來源：
  http://www.microsoft.com/taiwan/technet/columns/profwin/kerberoswin.mspx

- 應用系統的應用
  - PAM Kerberos， PAM全名為可插入認證模組 (Pluggable Authentication Module)
  - 此模組是一個在 HP-UX 上提供多項認證技術支援架構
  - PAM Kerberos v1.24 是 PAM 模組，提供 Open group RFC 86.0 通訊協定的身份認證支援

  資料來源： http://docs.hp.com/zh_tw/5991-6563/ch08s03.html