Kerberos V5
Authentication Server

Ibéria Medeiros

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa

Kerberos

- Kerberos is a authentication service for distributed systems based on the protocol by Needham-Schroeder, and therefore uses a *trusted* Key Distribution Center (KDC) and ***resorts only to symmetric cryptography***
- Three main entities: KDC, users, services (offered by servers)
- Based on the following ideas
  - KDC running in a physically secure machine
  - library linked with the applications, which authenticates the users
  - there is an initial authentication based on the user identifier and password, and from then on the machine can authenticate the user in any remote server

- At this moment there two versions in use
  [version 4] simpler and with better performance, but only in TCP/IP
  [version 5] more flexible and greater functionality (eventually substitutes v4)
  (Windows 2000, XP, 2003, Vista, 2008, 7 use a variant of v5)
  (there is a Internet Standard – RFC 4120)

Problems in V4 that were solved in V5

- Limitations related to the environment
  - ***encryption alg dependencies***: support for different encryption algorithms
  - ***IP dependencies***: support distinct types of network addresses
  - ***ordering of bytes in the message***: use of ASN.1 and BER
  - ***ticket lifetime***: arbitrary times
  - ***transmission of authentication***: allows the transmission of user credentials to another machine and user
  - ***authentication among realms***: support an hierarchical structure
- Technical problems
  - ***double encryption***: avoid double ticket encryption of KDC responses
  - ***PCBC encryption***: use standard methods for integrity
  - ***session keys*** : allows different keys between C↔S
  - ***password attacks*** : make it harder to brute force attack the keys

Objectives

- In a distributed environment where users have their PCs and want to request services from remote servers, Kerberos ensures that both the clients and servers authenticate each other
- Requirements (as initially defined in Kerberos)

  security: an eavesdropper can not collect information from the network to impersonate a user/server

  reliability: Kerberos should be highly reliable (through replication) because otherwise users can not make requests

  transparency: besides providing the passwords, authentication should be hidden from the user

  scalability: high numbers of clients and servers should be supported

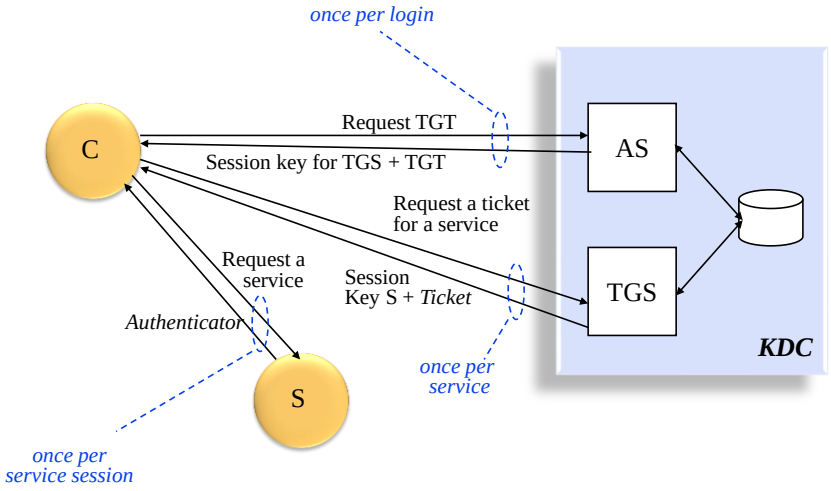
Keys

- Each user and server shares a secret key with the KDC – called the ***master key***
  - the master key of an user is generated from the password
  - the master key of a server can either be generated from a password or from a random number; this key is typically stored in a file, in a well known location in the server machines, which can only be read/written by the server user (often root)
- The KDC keeps a local database with a copy of all master keys (both from users and servers)
- For higher security, the KDC also has a master key that is used to encrypt the keys in the database
- The master key of the KDC is generated from a password (known only to the system administrator)

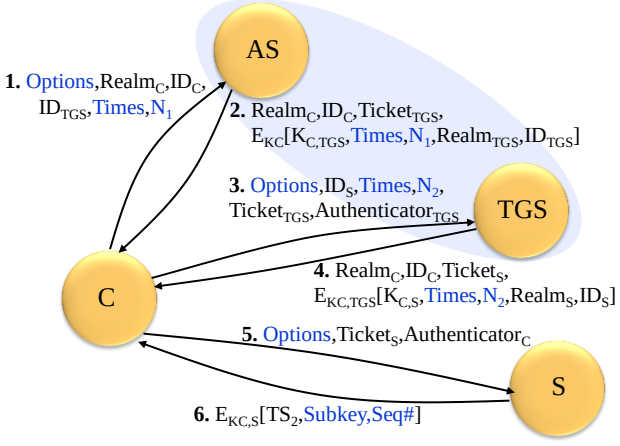
Information Stored at the KDC

- Example information stored per user :
  - **name of the user (*principal*)**
  - **master key**
  - version number of the key
  - maximum lifetime of the tickets created by this user
  - maximum lifetime for renewable tickets
  - version number of the KDC master key that was used to encrypt this data
  - expiration time for this entry in the database
  - time of the last change to this database entry
  - name of the user that made this change
  - what flags can be used
  - time when the password expires and the user is required to change it
  - time of the last change to the password
  - time of the last correct login

Protocol Summary



Kerberos V5



Terminology

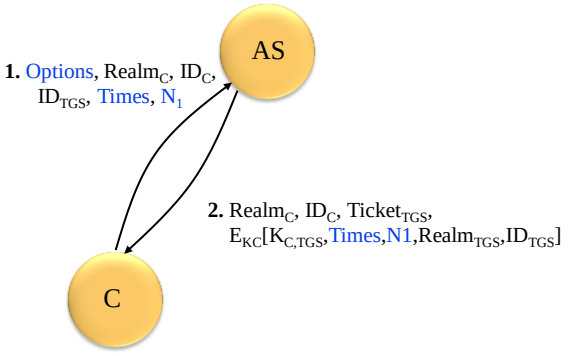
AS - *Authentication Server*
TGS - *Ticket-Granting Server*
ID - identification
AD – network level address
Realm - realm
TS - timestamp
Times - *(from,till,renew_time)*
N - nonce
K – secret key
E$_K$[X] – encrypt X with key K

Ticket$_{TGS}$ = E$_{KTGS}$[Flags,K$_{C,TGS}$,Realm$_C$,ID$_C$,AD$_C$,Times]
Ticket$_S$ = E$_{KS}$[Flags,K$_{C,S}$,Realm$_C$,ID$_C$,AD$_C$,Times]

Authenticator$_{TGS}$=E$_{KC,TGS}$[Realm$_C$,ID$_C$,TS$_1$]
Authenticator$_C$=E$_{KC,S}$[Realm$_C$, ID$_C$,TS$_2$,
Subkey,Seq#]

Kerberos V5

Authentication service exchange



Terminology

AS - *Authentication Server*
TGS - *Ticket-Granting Server*
ID - identification
AD – network level address
Realm - realm
TS - timestamp
Times - *(from,till,renew_time)*
N - nonce
K – secret key
E_K[X] – encrypt X with key K

Ticket$_{TGS}$ = E$_{KTGS}$[Flags,K$_{C,TGS}$,Realm$_C$,ID$_C$,AD$_C$,Times]

NOTES

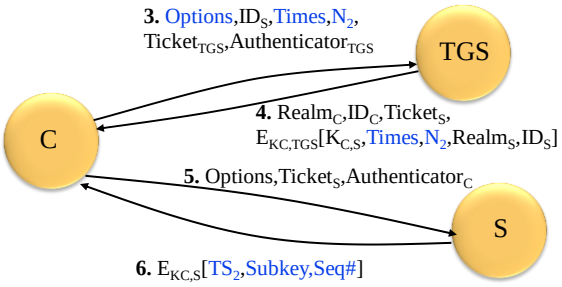
- Kerberos can *optionally* require some fields of message **1.** to be encrypted with the user master key with the objective of increasing the difficulty of brute force attacks on the passwords

- Message 1
  - Options : requests certain flags to be included in the ticket
  - Times : requests the ticket to be valid during (*from, till, rtime*)
  - N_1 : random value that is repeated in **2.** to detect replays

- Message 2
  - Ticket$_{TGS}$: information to be sent to the TGS
  - Flags : specific to this ticket (*see later*)
  - K$_{C,TGS}$: session key with the TGS

Kerberos V5

Ticket-granting service exchange



Terminology

AS - *Authentication Server*
TGS - *Ticket-Granting Server*
ID - identification
AD – network level address
Realm - realm
TS - timestamp
Times - *(from,till,renew_time)*
N - nonce
K – secret key
E_K[X] – encrypt X with key K

Client/server authentication exchange

$Ticket_{TGS} = E_{KTGS}[Flags,K_{C,TGS},Realm_C,ID_C,AD_C,Times]$ $\quad Authenticator_{TGS}=E_{KC,TGS}[Realm_C,ID_C,TS_1]$

$Ticket_S = E_{KS}[Flags,K_{C,S},Realm_C,ID_C,AD_C,Times]$ $\quad Authenticator_C=E_{KC,S}[Realm_C,ID_C,TS_2, Subkey,Seq\#]$

NOTES

- *$Ticket_S$* can be used several times by the user for authentication at the server (while the ticket is within its lifetime)

- Message 5
  - Subkey : (optional) the key chosen by the client to protect the interactions with the server during this session; if this field is not filled then they use $K_{C,S}$
  - Seq# : (optional) indicates the starting value of the sequence number used in the messages by the server (to detect replays)

- Message 6
  - Subkey : (optional) substitutes the subkey of message 5
  - Seq# : (optional) indicates the starting value of the sequence number used in the messages by the client (to detect replays)

Delegation of Access Rights (Authentication)

- **IDEA**: allows someone to transmit its rights of access to some resource to another user or server
- The delegation of access rights usually has the limitations
  - time: limits the period during which the other user can utilize the rights
  - utilization: only a subset of the rights can be utilized
- How can we delegate rights in Kerberos?
  - *Basic method* : *Alice* has to
    - » ask the KDC for a new TGT ***or*** for a ticket to a specific service
    - » with a different address from hers (assuming that Bob is in another machine), with several addresses, or no address
    - » then, she sends the TGT/ticket and session key to Bob
  - *Optional data* : Alice can request the addition to the TGT/ticket of a field *AUTHORIZATION-DATA* with information to be interpreted by the server application (for example, restricting what Bob can do)

Delegation of Access Rights (cont)

- A TGT with the following associated flags allows
  - **proxiable** : indicates that this TGT can be used to obtain *tickets* for services that will be used in another addresses (e.g., from A goes to B); the new tickets will have the proxy flag
  - **forwardable** : indicates that this TGT can be used to obtain a *TGT* with another address (e.g., from A goes to B); the new TGT will have the forwarded flag (and the tickets created with the new TGT also have the forwarded flag)

    If the user indicates that the new TGT is also forwardable, then the new user can also generate TGT for other addresses (e.g., from A goes to B, and then to C)
- Tradeoffs on Kerberos delegation scheme
  - Advantages : requires an explicit request for the delegation of rights, which allows the KDC to store some auditing information
  - Disadvantages : performance is penalized due to the extra messages that have to be exchanged

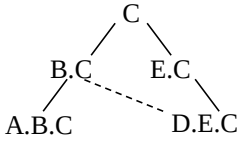
Realms

- Allow the definition of several security domains
- For example, in a organization with several departments, each department can manage its own users and services, and at the same time access the services of other departments
- Each realm has its own KDC (AS + TGS)

- For a user of a realm A to use a service of a realm B
  - the identifier of the service indicates that it belongs to realm B
  - the TGS of B must be registered as a service in the KDC of realm A

Hierarchical Realms

- V4 required each realm to share a key with every other realm with whom its users would need to authenticate => may require a large number of keys
- V5 allows transitivity on the creation of tickets, but requires the inclusion of the field ***TRANSITED*** in the tickets, which contains the names of the realms that used from the beginning of the path
- It is the responsibility of the service (i.e., Bob) to decide if it should trust or not the realms that were used in the authentication



A good policy to decide if a ticket should be accepted is to verify if the ticket went through a minimal path until it reached the destination

Ticket Lifetime

- Tickets in V5 can have very long lifetimes (on the contrary to V4 where the maximum lifetime was 21 hours)
- On the other hand, very long lifetimes can create security problems because typically it is very difficult to cancel tickets

- Kerberos solution
  - **renewable tickets** : tickets can be valid for a long time (e.g., 100 years) but they have to be periodically renewed; should have **renewable** flag
    - » before expiring, the ticket should be sent to the KDC to be renewed
    - » one can indicate to the KDC that a certain ticket can no longer be renewed
    - » the KDC does not renew tickets outside their validity period

  NOTE: with this solution, the KDC only has to memorize revoked tickets until the instant when they have to be renewed, instead of until the end of their lifetime period

Ticket Lifetime (cont)

- Kerberos solution
  - **post-dated tickets** : the ticket starts to be valid at some future instant
    - » the client asks for a TGT with the **may-postdate** flag to be able to request postdated tickets
    - » to a allow a ticket to be invalidated in the interval between its creation and the instant when it starts to be valid
      - the ticket is created with an **invalid** + **postdated** flag
      - when one wants to utilize the ticket, it is necessary to contact the KDC to remove the **invalid** flag
      - in the meantime, if the ticket had been canceled, then the KDC would refuse to remove the flag, preventing its use

Attack: Lateral Movement

- Techniques that enable an adversary to **access and control remote systems** on a network and could, but does not necessarily, include execution of tools on remote systems
- **Pass the ticket (PtT)** uses Kerberos tickets for authentication **without** having access to an account's password by compromising one system in the network
  - **valid user TGT and tickets** are captured by ***Credential Dumping***, depending on the level of access
  - a service ticket allows for access to a particular resource, whereas a TGT can be used to request service tickets from the TGS to access any resource the user has privileges to access
- **Silver Tickets** can be obtained for services that use Kerberos as an authentication mechanism and are used to *generate tickets* to access that particular resource and the system that hosts the resource (e.g., SharePoint)
- **Golden Tickets** can be obtained for the domain using *the KRBTGT account NTLM hash*, which enables *generation of TGTs* for any account in Active Directory

Search for tool: mimikatz

Bibliography

- C. Kaufman, R. Perlman, M. Speciner, *Network Security : Private Communication in a Public World (Second Edition)*, Prentice Hall, 2002 (capítulo 14)
- W. Stallings, Cryptography and Network Security: Principles and Practice, Six Edition, 2014 (pages 489-495)