Kerberos

Ahmet Burak Can
Hacettepe University
abc@hacettepe.edu.tr

Kerberos

- Advantages:
  - secure authentication
  - single sign-on
  - secure data flow

- Applications benefiting from Kerberos:
  - telnet, ftp
  - BSD tools (rlogin, rsh, rcp)
  - NFS
  - Others (pine, eudora, etc.)

Kerberos

- Kerberos is a network authentication protocol. Requirements:
  - Security
  - Reliability
  - Transparency
  - Scalability
  - Cryptographic authentication for distributed systems
  - Based on symmetric-key authentication with KDC
  - Developed at MIT: two versions: Version 4 and Version 5 (specified as RFC1510)
    - [http://web.mit.edu/kerberos/www](http://web.mit.edu/kerberos/www)

Overview of Kerberos

1. As verified user's access right in database, creates acknowledging ticket and session key. Receipts are encrypted using key derived from user's password.
2. Ticket-granting server (TGS)
3. TGS decrypts ticket and authenticates a session request. Then returns a ticket for requested service.
4. Service verifies that ticket and authentication are correct. Then returns an encrypted token to the client.
5. Client decrypts encrypted token using its private key. Then sends the decrypted token to the server.
6. Server verifies that ticket and authentication are correct. Then returns an acknowledgment to the client.
Protocol Design Motivations

- AS knows passwords for all clients
- AS distributes keys Client-TGS
- TGS distributes keys Client-Server
- Lifetime validity for tickets, include a time validity
- Freshness of messages to prevent replay attacks: use sequence numbers, timestamp or random numbers

Kerberos Keys

- Each principal shares a “master key” with KDC
  - $K_A$: Alice’s master key. Used for initial authentication
  - $K_{TGS}$: The key known by AS and the TGS.
  - $K_{A,TGS}$: The key shared between the TGS and Alice
- Ticket Granting Tickets (TGT):
  - Issued to Alice by AS after login
  - Encrypted with $K_{TGS}$
  - Used to obtain session key $K_{A,TGS}$

Key Relation in Kerberos

Logging into the Network

1. Alice, pwd
2. Alice's terminal

$K_{A,TGS}$ = $E_{K_{A}}(ID_{A} || AD_{A} || ID_{TGS} || TS_{TGS} || Lifetime_{TGS} || Ticket_{TGS})$

$ID_{TGS}$ denotes the identifier of the Ticket Granting Server (TGS)

$K_{A,TGS}$ is the key shared by the TGS and Alice

$K_{A}$ key known by AS and the TGS
Logging into the Network

The workstation,
- converts Alice’s password into a DES key
- when receives the credentials from the server; decrypts them using this DES key
- if decrypts correctly, authentication is successful
- discards Alice’s master key; retains the TGT.
- TGT contains all the information TGS needs about Alice’s session; hence TGS can work without remembering any volatile data.

Obtaining a Ticket from TGS

3- Alice → TGS: $ID_A || Ticket_{tmp} || Authenticator_A$
4- TGS → Alice: $E_{K_{S2}} [ K_{S2} || ID_A || TS_s || Ticket_s ]$

$Authenticator_A = E_{K_{S1}} [ ID_A || AD_A || TS_1 ]$
$Ticket_{tmp} = E_{K_{A10}} [ K_{A10} || ID_A || AD_A || ID_{T2} || TS_2 || Lifetime_2 ]$
$Ticket_s = E_{K_{B}} [ K_{S2} || ID_A || AD_A || ID_{T2} || TS_s || Lifetime_s ]$

$K_B$ is the key shared by the TGS and server B

Client-Server Authentication Exchange

5- Alice → Bob: $Ticket_B || Authenticator_B$
6- Bob → Alice: $E_{K_{B}} [ TS_s * 1 ]$

$Ticket_s = E_{K_{B}} [ K_{S2} || ID_A || AD_A || ID_{T2} || TS_s || Lifetime_s ]$
$Authenticator_B = E_{K_{B}} [ ID_B || AD_B || TS_s ]$