Kerberos

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

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

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

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

Overview of Kerberos

1. User logs on to workstation and requests service on host
2. AS verifies user’s credentials in database, creates ticket-granting ticket and assigns key. Results are encrypted using key derived from user’s password.
3. Workstation prompts user for password and uses password to decrypt ticket-granting ticket, then sends ticket and authentication data that contains user’s name, network address, and time to TGS.
4. TGS decrypts ticket and authenticates, verifies request, then creates ticket for requested service.
5. Server verifies that ticket and authenticator match, allows access to service. If mutual authentication is required, server returns an authenticator.
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 $K_{A,TGS}$

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.

Ticket$_{tgs}$ = $E_{K_{tgs}} [K_{A,tgs} || ID_A || AD_A || ID_{tgs} || TS || LifeTime_{tgs}]$

$ID_{tgs}$ denotes the identifier of the Ticket Granting Server (TGS)
$K_{A,tgs}$ is the key shared by the TGS and Alice
$K_{tgs}$ key known by AS and the TGS
Obtaining a Ticket from TGS

3- Alice → TGS: \( ID_B \ || \ Ticket_{tg} || \text{Authenticator}_A \)
4- TGS → Alice: \( E_{K_{AB}} [ K_{AB} || ID_B || TS_4 || Ticket_B ] \)

\( \text{Authenticator}_A = E_{K_{A,TG}} [ ID_A || AD_A || TS_3 ] \)
\( Ticket_{tg} = E_{K_{A,TG}} [ K_{A,TG} || ID_A || AD_A || ID_{tg} || TS_2 || \text{Lifetime}_2 ] \)
\( Ticket_B = E_{K_B} [ K_{AB} || ID_A || AD_A || ID_B || TS_4 || \text{Lifetime}_4 ] \)

\( K_B \) is the key shared by the TGS and server B

Client-Server Authentication Exchange

5- Alice → Bob: \( Ticket_B || \text{Authenticator}_A \)
6- Bob → Alice: \( E_{K_{AB}} [ TS_5 + 1 ] \)

\( Ticket_B = E_{K_B} [ K_{AB} || ID_A || AD_A || ID_B || TS_4 || \text{Lifetime}_4 ] \)
\( \text{Authenticator}_A = E_{K_{AB}} [ ID_A || AD_A || TS_3 ] \)

Key Relation in Kerberos

[Diagram showing key relations between Alice, AS, Bob, TGS, and the keys K_A, K_B, K_{AB}, K_{A,TG}, K_{TGS}]