Kerberos for Distributed Systems Security

Cunsheng Ding
HKUST, Hong Kong, CHINA
Agenda

• Distributed system security
• Introduction to Kerberos
• Kerberos Realms
• Authentication with Kerberos in Windows NT 5 and Windows 2000
Distributed Systems Security
Distributed Systems

- **A distributed system**: a collection of computers linked via some network.

- **Characteristic**: The components of the distributed system may be under the authority of different organizations, and may be governed by different security policies.

- **Example**: The Internet
Security Issues in Distributed Systems (1)

• Impersonation of user:
  - A user may gain access to a particular workstation and pretend to be another user operating from that workstation.

• Impersonation of workstation:
  - A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
Security Issues in Distributed Systems (2)

• Replay attacks:
  - A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

• Conclusion:
  - In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access.
Security Services in Distributed Systems

• Authentication

• Guarding the boundaries of internal networks
  - Firewalls

• Access control to distributed objects
  - Access control techniques

• Availability
  - Counter DoS techniques
Security Policies

• **Fact:** In a distributed system, users are not necessarily registered at the node they are accessing an object.

• **Question:** How to authenticate a user?

• **Question:** What is the basic for access control decisions?
Basis for Authentication and Access Control

• The user identity and password;
• the network address the user operates from;
  - e.g., any machine in UST can access Elsevier database;
• the distributed service the user is invoking, i.e., the access operation.
  - Anyone can read but cannot modify documents posted on my personal web page.
Examples: Unix System

- **ftp**: transfer files between Unix systems.
- **telnet, rlogin**: remote access
  - use user identity and password for authentication;
  - use the normal Unix access control.
- **New problem**: How can my password travel through the network securely?
Security Enforcement

- Once you have sorted out security policies, you have to decide where to enforce them!
  - Where in the system do you authenticate a user?
  - Where in the system do you make an access control decision?

**Authentication: Kerberos**
Kerberos Version 4
Kerberos Version 4

- Centralized network authentication service
- Developed in the Project Athena in MIT
Environment Addressed

- An open distributed environment in which
  - Users at workstations wish to access services on servers distributed throughout the network.
  - Servers can:
    - restrict access to authorized users
    - authenticate requests for service.
  - Workstations cannot be trusted to identify its users correctly to network services.
Requirements for Kerberos

- **Secure**: Opponent cannot impersonate a user and the Kerberos service should not be a weak link.
- **Reliable**: Highly reliable Kerberos service to ensure availability of supported services of application servers.
- **Transparent**: Users are only required to enter a password once and don’t know the authentication.
- **Scalable**: System can support large numbers of clients and servers.
Kerberos 4 Overview

• A basic third-party authentication scheme
• Have an Authentication Server (AS)
  - users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
• Have a Ticket Granting server (TGS)
  - users subsequently request access to other services from TGS on the basis of user’s TGT
2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.

5. Workstation sends ticket and authenticator to server.

6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

1. User logs on to workstation and requests service on host.

Once per user logon session

Kerberos

Authentication Server (AS)

Ticket-granting Server (TGS)

Request ticket-granting ticket

Ticket + session key

Request service-granting ticket

Ticket + session key

Once per type of service

Provide server authenticator

Once per service session
Further Information

• Only one symmetric cipher, i.e., DES, is used (in Version 4).
• Each client needs to share a secret key with the AS only.
• ID, timestamp, network address are used for authentication.
• Technical details of the protocol is omitted here (see Appendix).
Kerberos Realms
Kerberos Realm

• Kerberos realm:
  - The environment that one Kerberos server can manage the authentication process.

• The environment of one realm
  - The Kerberos server of one realm has all users ID and hashed password of all users in the realm.
  - The Kerberos server must share a secret key with each server.
  - All servers are registered with the Kerberos server.
Need for Inter-realm Authentication

• **Problem**: Users in one realm may need access to servers in other realms

• **Question**: How can this be done in a secure way?
Achieving Inter-realm Authentication

- Kerberos provides a mechanism for supporting inter-realm authentication under the following conditions:
  - Kerberos servers in inter-realms share a secret key
  - Assume the Kerberos servers trust each other.
Figure 11.2 Request for Service in Another Realm
Authentication with Kerberos in Windows NT and Windows 2000
Kerberos 4: Protocol Overview

1. Request for TGS ticket
2. Ticket for TGS
3. Request for Server ticket
4. Ticket for Server
5. Request for service
6. Mutual authentication
Authentication in Windows NT 5 and Windows 2000

• The main objective is to present the basic idea without technical details.
• Those who wish to have details should read Kerberos 5 and details of Windows NT 5 and Windows 2000.
The Basic Idea

- Use a KDC to run the AS and TGS in Kerberos.
- The KDC is located in the Domain Controller.
- Use the TGT and service ticket as access tokens.
Initial Kerberos Ticket
Ticket Granting Ticket (TGT)

• First ticket is a Ticket Granting Ticket
  - Used by client to get tickets to other services
  - Contains \textit{authorization data} based on group membership and privileges

• Ticket is encrypted in user’s key known by the KDC
  - Requires knowledge of password to use

• Tickets are stored in a ticket cache managed by LSA (Local Security Authority).
1. User logs on, authenticates to KDC

2. KDC provides TGT to user

3. User wants to use network resource; presents TGT to KDC and requests service ticket

4. KDC provides service ticket

5. User session provides service ticket to resource and gets access

6. For subsequent uses, user simply presents ticket
Comments on Authentication with Kerberos

• Single Sign-On
  - Simple administration
  - Good administrative control
  - Good user productivity
  - Good network security
Appendix: Details of Kerberos V4
# Version 4 Authentication Dialogue (3)

## (a) Authentication Service Exchange: to obtain ticket-granting ticket

1. \( C \rightarrow AS \): \( ID_c \parallel ID_{tgs} \parallel TS_1 \)
2. \( AS \rightarrow C \): \( E_{K_C} [ K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs} ] \)

\[ Ticket_{tgs} = E_{K_{tgs}} [ K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 ] \]

## (b) Ticket-Granting Service Exchange: to obtain service-granting ticket

3. \( C \rightarrow TGS \): \( ID_V \parallel Ticket_{tgs} \parallel Authenticator_c \)
4. \( TGS \rightarrow C \): \( E_{K_{C,tgs}} [ K_{c,v} \parallel ID_V \parallel TS_4 \parallel Ticket_V ] \)

\[ Ticket_{tgs} = E_{K_{tgs}} [ K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 ]. \]
\[ Ticket_V = E_{K_V} [ K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_V \parallel TS_4 \parallel Lifetime_4 ] \]
\[ Authenticator_c = E_{K_{c,tgs}} [ ID_c \parallel AD_c \parallel TS_3 ] \]

## (c) Client/Server Authentication Exchange: to obtain service

5. \( C \rightarrow V \): \( Ticket_V \parallel Authenticator_c \)
6. \( V \rightarrow C \): \( E_{K_{C,V}} [ TS_5 + 1 ] \) (for mutual authentication)

\[ Ticket_V = E_{K_V} [ K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_V \parallel TS_4 \parallel Lifetime_4 ] \]
\[ Authenticator_c = E_{K_{C,V}} [ ID_c \parallel AD_c \parallel TS_5 ] \]
Index

- $k_c$ the secret key shared between C and the AS.
- $k_{c,tgs}$ the session key for C and TGS, generated by the AS.
- $k_{c,v}$ the session key for C and V, generated by the TGS.
- $k_{tgs}$ the secret key shared between the TGS and the AS.
- TS, timestamp
- ID$_c$, C’s ID
- AD$_c$, C’s network address.

C. Ding - Computer Security - L18