Kerberos Protocol

Software Security

Steffen Helke
Chair of Software Engineering

14th January 2019

Brandenburgische
Technische Universität
Cottbus - Senftenberg

Kerberos Protocol
– Motivation and Introduction –

Objectives of today's lecture

- ➜ Understanding how the *Kerberos* protocol works
- ➜ Being able to apply important *design criteria* for building security protocols
- ➜ Getting to know different *weaknesses of Kerberos* and countermeasures to increase the security level

Kerberos: The Dog of Hades

- *Kerberos* is a character from the Greek mythology
- Monstrous multi-headed dog that guards the entrance of the underworld to prevent the dead from leaving
- The parable is only partially applicable because the protocol ensures that unauthorized users do not gain access to network resources

Why to use a Kerberos protocol?

Motivation

- Protecting resources from unauthorized access
- *Each* network connection can be potentially insecure
- Network connections should not only be protected outside a subnet, but also within a subnet

Problem

- Protection mechanisms based only on passwords are not practical for all network connections, e.g. if each contact between a user and a server requires entering a password

➜ Instead, a centralized key management should be introduced, supported by a secure protocol

Which features should Kerberos offer?

Requirements

- *Repeated authentication* in decentralized networks should be possible by *entering a password only once* (*single sign-on*)
- *Bidirectional identity verification*, i.e. both client and server must be authenticatable
- Optionally, *confidentiality and integrity* of communication data should be supported (note Kerberos originally only supported authentication)
- The explicit intention is to prevent attacks by fake identities (*man-in-the-middle attack*)
- Other attacks, e.g. based on retransmitting old messages (*replay attack*) should be mitigated by the protocol design

General Remarks and History of Kerberos

Development of the MIT

➜ *Massachusetts Institute of Technology* (MIT) located in USA, Development as part of the Athena project (1983-1991)

- Authentication service based on the symmetric variant of the Needham-Schroeder protocol
- Versions 1 to 3 were only used internally at MIT

Standardization

- Started with version 4 and MIT, IBM and DEC were involved
- In 1993 accepted as an international standard (RFC1510)
- Last detailed protocol update was in 2005 (RFC4120)
- Versions 4 & 5 are public, but v4 should no longer be used

Kerberos in Practice

Implementations

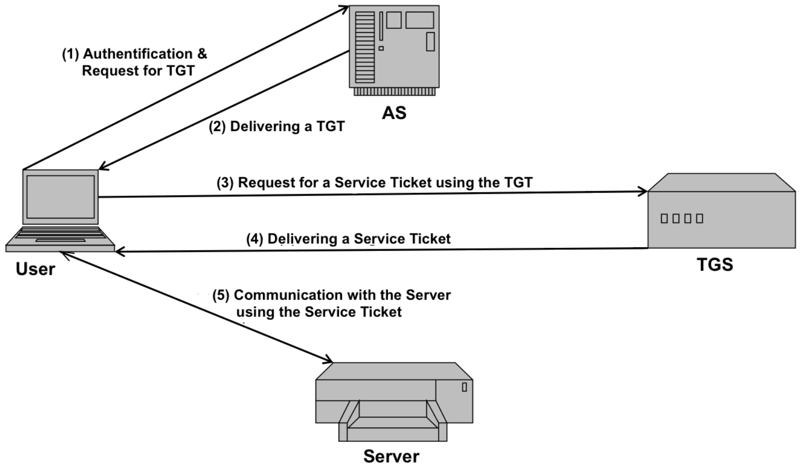
- Reference implementations from MIT for v4 and v5 (KRB5)
- Heimdal of KTS (*Royal Institute of Technology in Sweden*)
- Microsoft's version of Kerberos, used for Microsoft Active Directory from Windows 2000
- ShiShi, GNU GPL (*General Public License*)

Services that support Kerberos

- Secure Shell (ssh)
- Remote Shell (rsh, rlogin), Telnet
- Distributed file system (NFS, AFS)
- Email services

The Kerberos Protocol

Overview of the Kerberos Protocol


(1) Authentification &
Request for TGT
AS
(2) Delivering a TGT
User
(3) Request for a Service Ticket using the TGT
TGS
(4) Delivering a Service Ticket
(5) Communication with the Server
using the Service Ticket
Server

Source for the figure is a student presentation of A. Schlutter, S. Schreck, S. Heidebring & M. Busse, SASWT WS 11/12 TU-Berlin.

Server Infrastructure

Authentication Server (AS)

- Authenticates clients (*Principals*) based on passwords and is able to generate tickets for the TGS

Ticket-Granting Server (TGS)

- Authenticates clients based on submitted tickets (issued by AS) and creates service tickets for requested servers

Key-Distribution Center (KDC)

- Provides all necessary services for Kerberos
- Services of AS/TGS are often on *the same* server deployed

➜ Assumption: Services of the KDC are trustworthy
(*Trusted Third Party*)

Credentials

Ticket-Granting Ticket (TGT)

- Ticket is issued by AS
- Using this ticket client can request service tickets at TGS

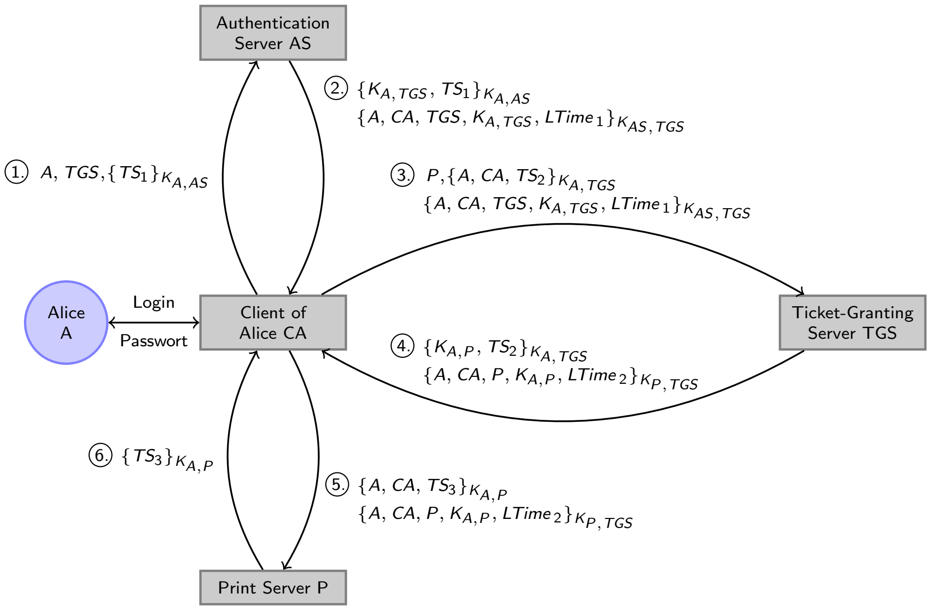
Service Ticket

- Ticket is issued by TGS
- Using this ticket client is able to request services at ordinary servers and to use their resources (e.g. print services)

Authenticator

- For verification of authenticity of principals
- Must be shown to both ordinary servers *and* and the TGS

Protocol Steps for Kerberos



Authentication Server AS

② $\{K_{A,TGS}, TS_1\}_{K_{A,AS}}$
$\{A, CA, TGS, K_{A,TGS}, LTime_1\}_{K_{AS,TGS}}$

① $A, TGS, \{TS_1\}_{K_{A,AS}}$

③ $P, \{A, CA, TS_2\}_{K_{A,TGS}}$
$\{A, CA, TGS, K_{A,TGS}, LTime_1\}_{K_{AS,TGS}}$

Alice A

Login

Passwort

Client of Alice CA

Ticket-Granting Server TGS

④ $\{K_{A,P}, TS_2\}_{K_{A,TGS}}$
$\{A, CA, P, K_{A,P}, LTime_2\}_{K_{P,TGS}}$

⑥ $\{TS_3\}_{K_{A,P}}$

⑤ $\{A, CA, TS_3\}_{K_{A,P}}$
$\{A, CA, P, K_{A,P}, LTime_2\}_{K_{P,TGS}}$

Print Server P

Weaknesses of the Kerberos Protocol

Weaknesses of the Kerberos Protocol

Weak Points for Attacks

1. Key management of KDC
2. Time synchronisation and lifetime of tickets
3. Weak passwords (*dictionary attack*)
4. Session keys on the client
5. One-time authentication only (*single sign on*)

Weak Point 1: Key management of KDC

- KDC is a *single point of failure*
- KDC keys are only protected with a single master key
- DoS attacks highly effective (*denial of service attack*)

Ticket Lifetime

Weak Point 2: Replay Attacks

1. Log the messages of Step ⑤
2. Replay these messages at a later time
3. Use a service (e.g. print service) as a different person

Assumptions

- Lifetime of messages from Step ⑤ has not expired at the time of replay, or
- Clock of the print server was manipulated

➜ Kerberos is not secure against replay attacks, because the attack can only mitigated by using time stamps!

Weak Passwords

Weak Point 3: Dictionary Attack

1. Intercept messages of Step ① and ②
2. Convert potential passwords into DES/AES keys using the selected hash function (e.g. MD5 or SHA1)
3. Decrypt messages of Step ② with AES/DES key and perform a plausibility check, e.g. check for validity of time stamps

Kerberos v4 has an additional weakness

- no *Pre-Authentication*, i.e. Step ① gives no information on whether the applicant knows a password

➜ AS sends TGT for Step ② without any check

➜ The attacker is able to send any requests in the name of the person to be attacked in order to evaluate the answers

Weaknesses of the Kerberos Protocol

Weak Point 4: Session keys on the client

- Assumption: Client is a single-user machine is not realistic
- Attacker accesses session keys on the client and get access to the complete network

➜ Authentication via client IP address is useless

➜ Kerberos v5 supports additional challenge-response authentication

Weak Point 5: One-time authentication

- Single sign-on is an advantage for easy handling
- Disadvantage: Attacks have a large impact, protection of sensitive data requires additional security mechanisms

Kerberos Realms

– How works interrealm authentication? –

Kerberos Realms

Realm

- Consists of a KDC and the users assigned to it
- Realm name is often based on DNS names
  name[/*instance*]@*REALM* (e.g. *helke*/*admin*@*TU-COTTBUS.DE*)

Cross-Realm Authentification (only Kerberos v5)

- Using services of other realms without entering password
- Assumption: There exists a trust relationship between the user realm and the other realm

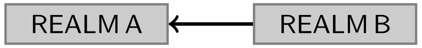
Relations between Realms

1. Direct trust relationship
2. Transitive trust relationship
3. Hierarchical trust relationship

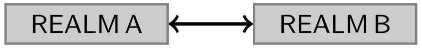
Relations between Realms

1. Direct Trust

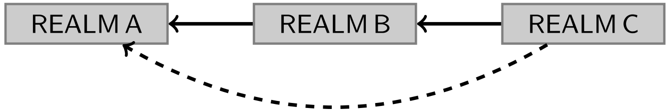
➜ **REALM B** trusts **REALM A**



➜ **REALM A** trusts additionally **REALM B**



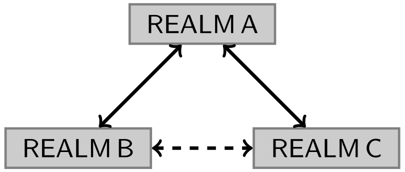
2. Transitive Trust



➜ **REALM C** trusts additionally **REALM A**

Relations between Realms (2)

3. Hierarchical Trust



➜ **REALM C** trusts additionally **REALM B** and vice versa

➜ Trust relationship is derived via transitivity

Cross-Realm Authentication

Procedure

1. Request a **TGT** from your own **AS** that will be accepted by your own **TGS**
2. Request a **TGT** from your own **TGS** that will be accepted by an external **TGS**
3. Request a **Service Ticket** from the external **TGS** that will be accepted by an external Service

Summary

Comparison of Kerberos v4 and v5

Lifetime of Tickets

- v4: maximum 21 hours
- v5: maximum until 31.12.1999
- In addition, v5 allows to renew tickets and to define the validity of a ticket into the future

Encryption & Hash Functions

- v4: DES & MD5 are fixed
- v5: selectable (e.g. AES & SHA1)

Pre-Authentication

- v4: not supported ➜ useable for active dictionary attacks
- v5: request ① is encrypted and is used for authentication

Summary and Conclusions

- Kerberos protocol is used to authenticate communication partners in a network and to exchange a session key
- Kerberos v4 has many security vulnerabilities and should therefore not be used anymore
- Kerberos v5 is also vulnerable, but with the right configuration it increases the security level considerably
- Kerberos is widely used and can be deployed on various platforms