Kerberos and GSI Collaboration

Daniel Kouřil

CERN, November 18–19, 2002

Kerberos

- *provides means for entity authentication and establishment of shared keys*
- *conventional (symmetric) cryptography*
- *trusted third-party model (Key Distribution Centre – KDC)*
- *KDC maintains a database of all principals and their secret keys*
- *version 5 standardized by IETF RFC 1510 (message format, protocols)*
- *krb5 implementations available:*
  - ** MIT, Heimdal*
  - ** Microsoft Windows 2000*

Kerberos – Basic Principles

- *based on the use of tickets issued by KDC*
- *tickets can be viewed as certificates binding an entity identification with a session key; tickets are signed by KDC*
- *a ticket can be read only by a service principal that it was issued for*

ticket = {client_id, server_id, session_key, lifetime}$_{server\_key}$

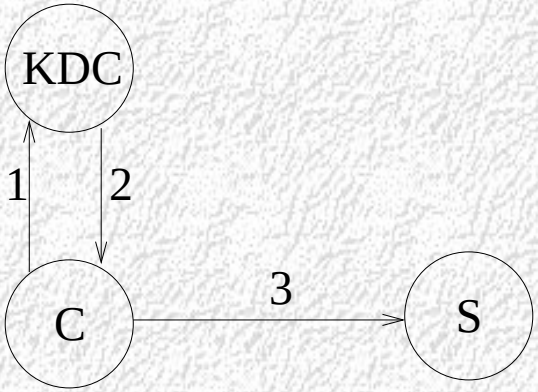
Krb5 credential = ticket + session key (possibly encrypted with client's key)

Kerberos – Basic Principles

- *based on the use of tickets issued by KDC*
- *tickets can be viewed as certificates binding an entity identification with a session key; tickets are signed by KDC*
- *a ticket can be read only by a service principal that it was issued for*

ticket = {client_id, server_id, session_key, lifetime}$_{server\_key}$

Krb5 credential = ticket + session key (possibly encrypted with client's key)



Kerberos vs. GSI

- *symmetric vs. asymmetric cryptography*
  (CPU overhead for PKI)
- *both support a simple mapping/authorization mechanism*
  ($HOME/.k5login, GSI gridmap-files)
- *Kerberos requires on-line available KDC*
- *security of users' credentials*
  short krb5 passwords can be remembered, GSI private keys are stored on disks
- *better scalability of PKI*
  (heterogenous solutions combining both the mechanisms)

Kerberos and GSI Interoperability

Applications development

- *for our purposes portability = a means of writing and running application so that they can use multiple security mechanisms without (or with minimal) changes of source code (`ssh daemon`).*
- *how to easy develop and maintain portable applications?*

Kerberos and GSI Interoperability

Applications development

- *for our purposes portability = a means of writing and running application so that they can use multiple security mechanisms without (or with minimal) changes of source code (`ssh daemon`).*
- *how to easy develop and maintain portable applications?*
- *GSS-API*

Kerberos and GSI Interoperability

Applications development

- *for our purposes portability = a means of writing and running application so that they can use multiple security mechanisms without (or with minimal) changes of source code (`ssh daemon`).*
- *how to easy develop and maintain portable applications?*
- *GSS-API*

Usage of multiple types of credentials

- *a user is using one type of credential (GSI proxy) and wants to obtain another type (Krb5 ticket) by means of the the first credential.*

Kerberos and GSI Interoperability

Applications development

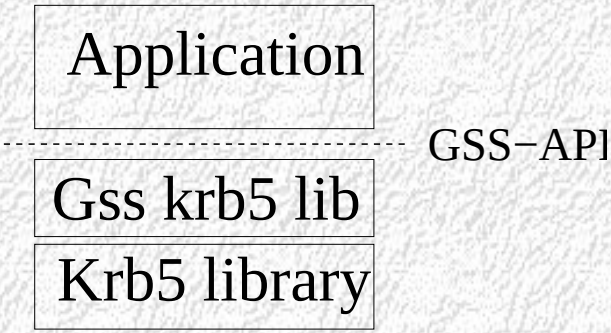
- *for our purposes portability = a means of writing and running application so that they can use multiple security mechanisms without (or with minimal) changes of source code (`ssh daemon`).*
- *how to easy develop and maintain portable applications?*
- *GSS-API*

Usage of multiple types of credentials

- *a user is using one type of credential (GSI proxy) and wants to obtain another type (Krb5 ticket) by means of the the first credential.*
- *various "transformation mechanisms"*

Portable Applications Development – GSS-API

- *IETF RFC 2743 (API specifications), IETF RFC 2744 (C-bindings)*
- *powerful tool for the development of mechanism-independent applications*
- *the goal should be a source code without any functions of any particular security mechanism*



Development of Portable Applications – GSS-API Extensions

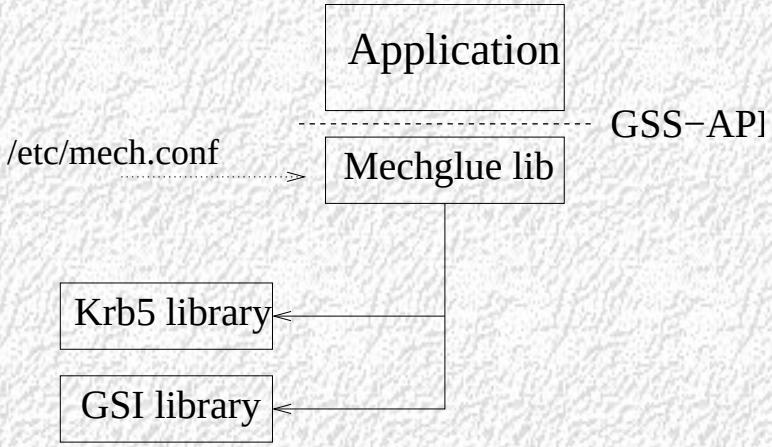
- *some areas are not covered by GSS-API and must be solved by applications via krb5 or GSI API.*
  (functions for storing and reading credentials to/from a disk, credential delegation at any time, mapping of GSS-API identities to local ones)
- *GGF GSI Security WG produced the GSS-API Extension draft addressing these issues*
- *the only thing not solved by the GSS-API Extension draft is mapping of GSS names to local uids*
- *implementations of GSS-API Extensions are available for Globus (GSI) and for Heimdal*

Running of Portable Applications

- *GSS-API and GSS-API Extensions enable the development of reasonably portable applications*
- *almost no changes in the source code are needed to port an application to another security mechanism (ID mapping)*
- *it's sufficient to either re-link (static) or just re-start (dynamic) the binary with appropriate GSS-API library*
- *multiple binaries (each supporting a different mechanism) for one application are still needed*
  (separate starting scripts, cannot share some resources – e.g. must run on different TCP ports, application must able to run parallely)

Running of Portable Applications – Mechglue

- *Sun Microsystem, open-source, part of the MIT Kerberos5 distribution*
- *an abstract GSS-API implementation, instead of implementing a real mechanism, it creates a layer between an application and real GSS-API libraries available on a system.*



- *when mechglue is used it's possible for an application to support multiple mechanisms at once (in one binary)*
- *it's no more needed to maintain more binaries, application can run on one TCP port, etc.*

Running of Portable Applications – Mechglue

- *the original implemenation suffer from several drawbacks (support of GSS-API version 1 (obsoleted), obviously not really tested with multiple mechanisms, doesn't support GSS-API Extensions)*
- *a patch is available fixing the above problems (Jim Basney, Doug E. Engert, Daniel Kouril)*
- *mechglue is supported by current GsiSSH (maintaned by NSCA); a mechglue-enabled CVS server has been running for 2 months for our WP1, internal Cesnet repository and another Grid project*

Credentials Transformations

- *a user has one type of credential and wants to access a service requiring another type.*

Kerberos v5 ticket → GSI proxy

- *kCA (server) + KX.509 (client)*

GSI Proxy → Kerberos v5 ticket

- *SSLk5*
  expansion of MIT Kerberos5 with SSL authentication in the initial stage (contacting of KDC) by Doug Engert; mainly for Globus, current status ?
- *PKINIT*
  endeavour of IETF Kerberos WG of adding PKI to the Kerberos v.5 protocol, still draft (but hopefully close to RFC)

Another types

OTP → Kerberos, AFS aklog using GSI proxy by Doug Engert

PKINIT

- *change of the first two authentication exchanges of the Kerberos5 protocol (between client and KDC), rest of the protocol is not changed*
- *doesn't use SSL, only X.509 certificates*
- *implementation available for Microsoft Windows 2000 (unfortunately according to draft 9, which is incompatible with current version)*
- *implementation for Heimdal:*
  - *Client support of Win2k KDC (Heimdal client can authenticate to a Win2k KDC via PKINIT)*
  - *smartcard support*
  - *support of GSI (clients can use GSI proxy certificates, KDC is able to read gridmap-files)*
- *http://meta.cesnet.cz/software/heimdal/*