Kerberos 5 for DESY

Wolfgang Friebel

Useful URL's

- K5 protocol:http://www.isi.edu/people/bcn/krb-revisions
- FAQ:http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html
- MIT K5:http://web.mit.edu/kerberos/www
- KTH K4:http://www.pdc.kth.se/kth-krb
- Heimdal K5:http://www.pdc.kth.se/heimdal
- Kerberos Projects:http://www.nectar.cc/krb
- Heimdal PAM:http://www.rit.bme.hu/~balsa/pam_krb5
- GSS-API:http://docs.sun.com (search for GSSAPI)
- Gssklog:ftp://achilles.ctd.anl.gov/pub/DEE

Motivation

- Kerberos4 used internal to AFS
  - no easy replacement available
  - is superseded by Kerberos5
  - has security weaknesses
- Kerberos5 is a supported standard
  - lots of tools are K5 enabled (or prepared)
    - token extension in LSF, SGEEE, UW-imap server, lprNG, Cisco Routers,...
  - AFS can be configured to work with K5
  - K5 offers desired features missing in K4
  - K5 comes with an implementation of the GSS-API (another one is the GSI from Globus)

Kerberos Terminology

- Kerberos is a protocol to authenticate users and services (= principals)
- The Key Distribution Center (KDC) issues proofs of identity (= tickets, containing short living session keys)
- Identity is checked by exchanging messages (challenge - response) using a short lived session key, no passwords are sent over the network
- A key is a bit string used to en- or decrypt messages
- Services use randomly generated keys instead of passwords

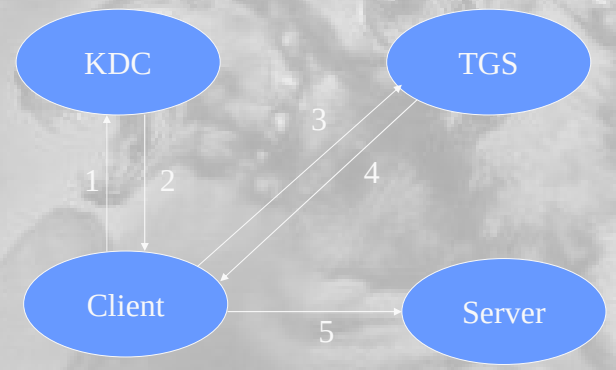
Kerberos Keys and Tickets

- The KDC('s) have keys for all principals
- Services need access to (permanent) keys to authenticate against the KDC. They are stored in a file (usually in krb5.keytab)
- User passwords are transformed into keys by selectable (predefined) algorithms
- Authenticated users have a Ticket Granting Ticket (TGT), that is a service ticket for the Ticket Granting Service (TGS) stored in a file
- An AFS token is a service ticket for the service "afs" and is cached in kernel memory

Features of a ticket

- Valid for a limited time period (e.g. 25h)
- Can be renewed if valid and within the allowed time period for renewal (e.g. 14d)
- Can be made invalid initially
- Can be made forwardable to other hosts
- (TGT) can be used to obtain further (service) tickets
- Can be destroyed if not longer needed
- Is usually stored in a file (/tmp/...)

Getting a Service Ticket



Kerberos auth (simplified)

- Step 1: Requesting a ticket
  a) Client requests a ticket for the ticket granting service (TGS)
  b) Server returns ticket (contains newly generated session key encrypted with user key)
  c) Further communication is encrypted (session key)
- Step 2: User Authentication
  a) User types password
  b) Client converts password into key and sends a request for a Ticket granting Ticket
  c) Server issues TGT
- Step 3: Request service tickets with the TGT

Preauthentication (K5 only)

- In Keberos4 Users can request a TGT for arbitrary principals
  - KDC sends a response which is encrypted with the principals key
  - can be used for offline password attack
- Kerberos5 adds Preauthentication
  - already at step 1a the user has to type a password, which is converted to a key
  - timestamp encrypted with that key is sent with the initial request (or smart card data)

Kerberos Realms and AFS

- All principals stored in the KDC database belong to a common realm
- AFS assumes that a user in afs cell my.cell maps to a K4 principal user.@MY.CELL
- There is documentation stating that a K5 realm different from K4 is possible, but it is harder to get and I was unable to make it work

Available Software

- Kerberos Servers:
  - MIT distribution
  - Heimdal distribution
  - DCE
  - A KDC comes with Windows 2000
- Kerberos Clients
  - Collection of clients contained in all server packs
  - At least 5 different PAM implementations
  - Standalone Software that is Kerberos enabled through the use of GSS-API (useful for Grid) or through native Kerberos calls (MIT and Heimdal differ!)

Kerberos Servers

- Initial tests done (at CERN) with 3 servers
- Heimdal
  - Integrated Kerberos4 and AFS support
  - Easy conversion of the K4 user database
  - Incremental propagation of database changes
  - Fewer applications available which successfully build against Heimdal K5 libs than against MIT libraries
  - CLI differs from MIT and from W2000 ones

Kerberos Servers (2)

- MIT
  - A separate daemon needed for AFS, otherwise AFS-K5 integration provided
  - Complicated procedure to convert AFS users to K5
  - Comes precompiled with Linux (Redhat)
  - Some applications ready to use MIT libraries only
  - No integrated server replication, it is done dumping the full database regularly (CERN > 10MB each time)
  - Used by many US sites (Fermilab)

Kerberos Servers (3)

- W2000
  - No smooth migration of AFS accounts to W2000, users without W2000 accounts need to get new account and password
  - Authentication for AFS with Windows password
  - Separate daemon required to obtain AFS tokens (tested and working: gssklog)
  - Almost no Windows applications known that use the Kerberos protocol
  - TGT contains data internal to W2000, this rules out to use non W2000 KDC's

Kerberos Servers (4)

- W2000 (cont.)
  - No source code available, no chance to fix bugs quickly and no guarantee that K5 will survive
  - There is a simple recipe to crash (reboot) W2000 KDC's, bug reported to Microsoft
  - Other errors seem to be known for a long time but not fixed (some telnet core dumps)
  - Only subset of K5 standard implemented
  - Administration of K5 interwoven with W2000 internals and very different from both MIT and Heimdal administration

Design decisions

- Slow migration from K4 to K5
  - Start with a limited number of users
  - Free choice between K4 and K5 for users
  - All changes in the user base must be in K4
- Use Heimdal for the following reasons:
  - Incremental and synchronous K5 DB propagation
  - One daemon only for K5, K4, AFS
  - Sync of K4 and K5 without service interrupt possible, could be triggered by K4 DB change
- Use PAM to get Tickets and AFS token
  - Transparent for the users
  - No change in PAM aware applications

Implementation

- Built Kerberos 5 realm CERN.CH
  - Used Heimdal distribution
  - Master KDC is on Solaris 8
  - One slave server planned (also Solaris 8)
- Software compiled for Linux + Solaris
  - Server and clients Heimdal
  - Heimdal compliant PAM
- Work repeated for realm IFH.DE (DESY)
  - Newer releases of K4 and K5 used

Implementation (2)

- Script to configure clients and servers
  - krb5setuphd (ksh script)
- Script to keep K4 and K5 DB in sync
  - updatekrb (perl script)
- Init scripts to start KDC and slaves
  - Generated from krb5setuphd

Implementation (3)

- Ticket lifetimes
  - Need to be short for security reasons
  - Need to be renewable for a long time for batch jobs (now unlimited, proposal: 30d)
  - Users could make use of it to refresh AFS tokens almost forever without typing a password (security concerns?)
  - Default ticket lifetime now in sync with AFS (25h)

Interoperability issues

- K4 vs K5 database
  - K4 gets updated by several methods
  - K5 is updated asynchronously (by krbupdate, could be done often >5min)
  - password out of sync for short period
- K5 integrated K4 vs original K4
  - Both K4 KDC's are fully functional
  - Old K4 selected from krb.conf and CellServDB

Does one really need to keep K4 (because of access from external sites) ?

Interoperability issues (2)

- Heimdal vs MIT clients
  - Bug in MIT lib? (Clients did not get proper AFS cell name from Heimdal KDC, has been fixed in MIT 1.2.6)
  - MIT clients do not get K4 TGT and AFS token (need to call afslog to get token)

Interoperability issues (3)

- Heimdal vs. W2000
  - On W2000 client could be installed to contact Heimdal KDC and obtain AFS token
  - Software available (gssklog(d) and others)
  - Need to investigate: obtaining AFS token by presenting the W2000 TGT without entering passwords

Trust between different realms

- Needed to do mutual authentication between different sites (e.g CERN-DESY) or on site Unix and Windows realms
- Realms share a common secret stored in keys krbtgt/realmA@realmB and ..B@..A
- User from realm A is trusted in realm B after obtaining ticket from realm A
- User has also in realmB only a ticket user@realmA
- AFS ACL's in realm B will not be honored and the user is not in system:authuser
- Needs to be handled by extra PTS entries

Next steps

- Deploy K5 clients for a limited number of users
- Do the PAM configuration
- Test more applications (ssh!)
- Test more platforms
  - Platform dependent PAM configuration
  - Possible compilation problems
- Port K4 applications
- Replace the K4 KDC by K5

Conclusions

- K5 deployment is fairly easy
  - Users can already now authenticate against K5 realms CERN.CH, IFH.DE
- K5 integration with services more tedious
  - Need to prepare and test lots of services and applications (AFS, ssh, SGEEE, CVS, PAM)
- W2000 interoperability is possible
  - harder and less useful than thought

Conclusions (2)

- Users will not notice a difference in behaviour (except e.g. different output of klist) if PAM properly configured
- Sessions with long living AFS tokens will become possible (without storing passwords)
- Usage of Kerberos5 will lead to more secure systems and easier configuration of software