AFS and Kerberos 5

Ken Hornstein
Naval Research Laboratory

Kerberos Use in AFS, "old school"

- Kerberos implementation based on V4 protocol.
- Client-server transport uses RX.
- Raw Kerberos binary ticket placed in credential cache by klog/aklog (instead of Kerberos AP_REQ).
- AFS kaserver also implements a standard Kerberos V4 server (used by the Windows client)

How does this appear to users/ admins?

- Admins create & manage users via kaserver utilities (kas, uss).
- Users authenticate to AFS via klog, AFS-aware login program, etc etc.
- Generally Kerberos is the only Kerberos service utilized by users.

Kerberos 5 with AFS, the old way

- Different Kerberos implementations allow the import of the kaserver database into the Kerberos 5 DB, so users can keep their passwords during a switch.
- However, AFS only understands a Kerberos V4 ticket, so a vanilla Kerberos 5 ticket won't work.
- The solution is to use the Kerberos 524 service (krb524d) that can translate a Kerberos V5 ticket into a Kerberos V4 ticket.
- Depending on your Kerberos client implementation, this may be done via kinit/login/pam_module or a separate aklog program.

Extra Pieces To Make Life Easier

- To prevent a cell-wide flag day, you want to support the “old” authentication mechanism (klog & friends) during a transition period.

- To make this work, you can use fakeka (MIT) or the Heimdal KDC to support the AFS kaserver protocol.

- This gives you the same user passwords no matter which program (V5 kinit, klog) you use to authenticate.

How does this appear to users/ admins?

- Users use kinit to authenticate, Kerberos-aware login program, appropriate PAM module, etc etc. Depending on you Kerberos, kinit/login may run aklog for you, or you have to run it by hand.
- Since the TGT is always kept around, generally more sites in this configuration use other Kerberos services.
- Admins create & manage users via Kerberos 5 admin suite.
- Since the users no longer appear in a kaserver, uss & kas no longer work, so the old account creation process generally has to change.

Items of Note in this Setup

- Unsupported by IBM/Transarc (hah!)
- The salt algorithm used by AFS, Kerberos V4, and Kerberos V5 are all different.
- The AFS Windows client uses Kerberos V4 to the AFS DB servers.
- The 524 service uses a port used by a recent Windows virus (4444), so admins may be reluctant to unblock it.
- You still need to support single-DES at least for the AFS service (and users if you want klog to still work).

Kerberos 5 with AFS, the new way

- A security bug was discovered in Kerberos V4 cross-realm that was a flaw in the base protocol (in other words, there was no possible patch).
- The solution was to modify the AFS servers to accept a V5 service ticket as well as a V4 service ticket.
- krb524d has been modified in newer Kerberos releases so that it will simply return the unmodified V5 service ticket for the AFS service.
- Other than that, everything is the same as the "old way" of doing Kerberos V5.

A Few Extra Items of Note

- The maximum size of the AFS token is 344 bytes. This normally isn't a problem ... unless you're using MS Kerberos.

- Several possible solutions: use a modified krb524d which can strip out the MS PAC information from the ticket (patches from Doug Engert) or set a flag in the MS Kerberos DB to not include PAC information for that principal (coming soon, according to Doug).

- The krb524d step is not really necessary (since an aklog has access to the V5 credentials), but adding client-side support for this requires an internal API (or hand-coding ASN.1). Next release of aklog may optionally support this (because of firewall issues).

Cross-Realm Authentication and AFS

- We make heavy use of this in production. It works well, but it has a few warts.
- You need to create a PTS entry called system: authuser@foreign.realm (note lower case) and give it a high group quota.
- Users that cross-realm authenticate with a modern aklog will get automatically created PTS entries (user@foreign.realm).
- Foreign realm users don't appear in system:authuser (but do appear in system:authuser@foreign.realm)
- The AFS PTS ID is a high-numbered (>16 bits) id that doesn't match the user's Unix uid. This gives a few utilities some problems, but most things work just fine.
- Uid mis-match confuses some users, but most don't notice.

And that's it!

Any Questions?