Living and Working with Heimdal

Henry B. Hotz
Jet Propulsion Laboratory
California Institute of Technology

Overview

- Disclaimer:
  - MIT provides a very good system.
  - I've heard good things about Microsoft as well.
- Heimdal has been working fine for two years now, but there are issues we've had to address.
  - Automated monitoring for service failures.
    - Biggest headache is iprop.
  - Operations and maintenance procedures.
    - Recommend a tested procedure for promoting a slave to a master (before you actually need to replace a broken machine).
  - A few customizations (patches) have been necessary.

Automated Monitoring

On every server check:

- Network connectivity
  - Notify Operations of problems
    - If possible! Same network outage may block notifications.
  - Master can see if it can't talk to slaves
    - Alternate notification path
    - Firewall issues may isolate slaves from the master
- OS, NTP, and all daemons up
  - Automatically restart
  - Notify Operations of status changes

Automated Monitoring, Continued

- Also check the functionality actually works
  - KDC provides tickets
  - Password and admin changes processed, and propagated to slaves
    - Force periodic updates to the database on the master
      - cron job
    - Check the age of the database on the slaves.
      - Other cron job
      - Auto-restart ipropd when stop getting updates.

iprop Background

- Has two halves
  - ipropd-master
    - Watches change log file for updates to the database.
      - Sends changes to slaves immediately.
    - Performs heartbeat check that slave connections are still good.
      - Records status in `/var/heimdal/slaves-stats` file.
      - Very useful when slave network connectivity problems prevent other notifications!
    - Downloads entire database if slave is too out-of-date.
  - ipropd-slave
    - Tells master what database version it has.
    - Responds to heartbeat checks.
    - Applies database changes.

iPROPD: The Best and the Worst of Heimdal

- iprop provides instant, low-overhead replication of database changes.
- iprop has lots of "issues".
  - Slave dies if network connection to master hickups.
  - Generally not robust to second-order errors.
  - Sometimes inserts duplicate entries in the change log.
    - Disk-full condition where the database and log files live causes iprop-slave to consume memory before core-ing.
      - Difficult to debug if it's the same partition that your core-file-saving cron job uses.
    - Once saw a disk-full condition on a slave cause log growth and core-ing on the master and the other slaves. (version 0.6.3)

Example Log Sizes

- This data taken just before a planned log reset.
- Slave near master:

```
-rw-------    1 root     other    29384704 Feb  8 17:28 heimdal.db
-rw-------    1 root     other    27872254 Feb  8 17:28 heimdal.log
```

- Inside firewall slave:

```
-rw-------    1 root     other    29384704 Feb  8 17:28 heimdal.db
-rw-------    1 root     other    27281090 Feb  8 17:28 heimdal.log
```

- The inside-firewall slave needs restarting every 3-4 weeks since the 0.7 upgrade, vice 3-4 months in 0.6.

hprop Versus iprop

- `iprop` not advertised as mature in 0.6.
- Wanted easy(er) migration to MIT, if possible.
- Switched to iprop due to bug in hprop.
  - The hdb library had a race condition if an update (from hpropd) conflicts with a read (from the KDC).
  - Pre-emptively restart kdc
    - Based on open connections shown in LSOF.
  - Love provided a patch quickly, but . . .
  - The delayed effect of password changes generated lots of questions from Operations.

Procedures

- Log rotation and backups
  - Consider the slaves to be “hot” backups.
  - Primarily need off-site backups.
  - Backup the master key separately.
    - The database isn’t that sensitive if it’s encrypted.
- Upgrades (next slides)
- Log file truncation (change log, `heimdal.log`)
  - Once per year on each server do

```
.../sbin/iprop-log truncate      Heimdal 0.8
.../sbin/truncate_log            Heimdal 0.7
```

Slave to Master Promotion

- No, we haven't had to do this for real, yet.
  - Recommend you think about it before you need it, though.

1. Update inetd.conf and startup/shutdown scripts
   - enable `kadmind` and `kpasswdd`
   - change `ipropd-slave` to `ipropd-master`
2. Ensure that `kadmin.acl` and `slaves` are up to date.
3. Update DNS entry for the new master.
   - We have a CNAME to make this easy.
   - Also SRV records, if you use them.
4. Update Firewall exceptions to allow access to the new master.
   - `212/1212, 749, 464`

Minor Upgrade Procedure

- Turn off auto-restarts

  `mv /usr/heimdal /usr/heimdal.old`

- Copy in new binary tree (`/usr/heimdal`)
- Restart daemons
- Restart auto-restart monitoring

Major Upgrade Procedure

- Turn off auto-restarts

```
mv /usr/heimdal /usr/heimdal.old
mv /var/heimdal/heimdal.{db,log,etc} /var/heimdal/old
```

- Text dump database
- Copy in new binary tree, config files, and support scripts.
- Restore database dump
  - Had one principal that would not re-import to 0.7.
- Restart daemons
- Restart auto-restart monitoring

Slave Upgrade Procedure

- Turn off auto-restarts

```
mv /usr/heimdal /usr/heimdal.old
mv /var/heimdal/heimdal.{db,log,etc} /var/heimdal/old
```

- Copy in new binary tree, config files, and support scripts.
- Restart daemons
  - Allow iprop to download new database.
- Restart auto-restart monitoring

Customizations

- Timespan logging on all issued tickets.
  - Allows identification of who could access what when.
    - Not yet used for a real intrusion.
- Password quality
  - Stripped down version of the published cracklib example.
  - Password history (in testing)
- Restrict time correction
  - Prohibit forward extension of end time when the client clock is slow.
    - For MIT/Sun compatibility

Customizations, Continued

- Kerberos 99
  - Transarc proprietary extension to Kerberos 4
    - Adds a 64-bit pre-auth field to enable the kaserver's bad password logging and lockout support as in the ka protocol.
    - If used with an MIT server will fall-back to standard K4.
    - If used with Heimdal will never receive a response, and fail.
  - Implemented in the Transarc Windows AFS client
  - We've phased out all such clients and no longer need the patch to support it.

Development environment Support

- Frequently need a development environment with its own KDC (and maybe AFS cell), but want to use production identities to minimize provisioning overhead.
- Setting up cross-realm relationships

  ```
  kadmin dump --decrypt | grep 'OTHER.REALM' >exchange.file
  ```

  Copy file (securely) to Kerberos master of other realm.

  ```
  kadmin merge exchange.file
  ```

  - Gets you both the `krbtgt/THIS.REALM@OTHER.REALM`, and the `krbtgt/OTHER.REALM@THIS.REALM` keys.
  - Allows you to use full-strength random keys.

Future

- Looking forward to PKINIT/KX509 deployment with 0.8.
- Wish we had:
  - Password History
    - We're doing a plug-in for it.
  - Dual kvno support for krbtgt principals
    - Allow transparent update of keys like you can do in a keytab file for application servers.
  - Replay caching
  - More robust error checking/reporting in ipropd
    - Maybe 0.8 is already better, no experience.