Proposed Priorities

A preliminary review

Ken Raeburn
MIT Kerberos Consortium
December 11, 2007
Proposed work

37 proposed tasks in Consortium proposal
10 requested by board members
## Proposed priority tasks

<table>
<thead>
<tr>
<th>Standardized admin interfaces</th>
<th>Improved sysadmin, user docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thorough security audit</td>
<td>Improved browser, email support</td>
</tr>
<tr>
<td>Zero configuration</td>
<td>Common ccache implementation</td>
</tr>
<tr>
<td>Interoper. test suite</td>
<td>Standardized API</td>
</tr>
<tr>
<td>Database support</td>
<td>Propagation mgmt</td>
</tr>
</tbody>
</table>
Standardized admin interfaces

Admin protocol

Propagation

Incremental propagation

Error messages
Standardized admin interfaces
Better administration in mixed env
Data model, LDAP schema work at IETF
MIT and Heimdal currently use different protocols, but similar data and operations
Sun protocol based on MIT's Microsoft?
IETF set/change key/password protocol
Standardized admin interfaces:

Propagation

Requires coordination with Heimdal, commitment on both sides, possibly database format changes, and propagation protocol changes

Heimdal can propagate from MIT dump file
Standardized admin interfaces:

Incremental propagation

Some sites need propagation faster than kprop can achieve it; don't waste cycles when no changes

Sun's incremental propagation patches

May not align with Heimdal iprop model
Standardized admin interfaces:

Error messages

Some work done on improving local error messages

Looking at means of passing detailed, friendly messages from server

  Additional error codes better for i18n
  Customized policies = custom messages?

Standardization across implementations?
Thorough security audit

Both real and perceived code quality issues

Ability to say “did audit” a PR plus

Some reported vulnerabilities are just sloppy or unclear code, difficult to analyze

Some are due to lack of understanding of how to use library
Thorough security audit

Partial audit of some code done a while ago
Deploy tools, practices to reduce future risk
   May eliminate some bugs as side effect
   Survey of some static analysis tools planned
The actual audit: code read-through? 3rd party?
Multiple areas: code; architecture; protocol
Zero configuration

Useful for new installs, browser-based apps; reduce end-user or administrator hassle

Determining client's local realm

Server referrals

Zero-conf more important for clients than servers?
Interoperability test suite

Partly done this summer

Microsoft's “gssmonger” framework in VMs
Mainly GSSAPI testing, not full Kerberos protocol
No tidy reporting mechanism yet (XML table)
Not integrated into regular testing yet
Easy to moderate difficulty
Database support (esp. MySQL)

Comes up for discussion on MySQL lists now and then; haven't seen progress

Needs buy-in from maintainers

Oracle, Postgres have support
Improved sysadmin, user docs

Last significant documentation work was 5+ years ago

Some visible software changes since have been documented, but not all
  Mostly just admin docs and implementors' notes
  And only by programmers, not doc writers

No overall review of docs in years
Improved web browser / mail client support

Many popular clients have support, not 100%

Mobile devices

What barriers to further deployment or use can we remove?

Zero-config would help; KIM for multiple accts

API docs, guidelines, examples; maybe better APIs?
Common ccache implementation

“Secure memory store” wanted

CCAPI port to UNIX under consideration
Linux keyring; MS LSA work
Secure storage is dependent on OS capabilities

Common format and location across MIT, Heimdal, vendors
CCAPI; UNIX file caches
Standardized developer API

GSSAPI
  Could use better docs

Admin protocol

Basic Kerberos protocol
  Apple, Sun, MIT APIs converging
  MIT/Heimdal issues
  First step probably to document current API
Propagation management

Query
Are the slaves up to date?

Control
Simplify slave propagation management
Force propagation