MIT Kerberos Software Development Roadmap

Tom Yu
MIT Kerberos Consortium
November 3, 2008
Overview

Timeline
Completed krb5-1.7 goals
Areas for improvement
Process changes
Interface change strategy
Timeline

Target 18-month cycle

krb5-1.7
  Branch Jan. 2009
  Release Apr. 2009

krb5-1.8
  Branch Jul. 2010
  Release Oct. 2010

krb5-1.9
  Branch Jan. 2012
  Release Apr. 2012
Completed krb5-1.7 Goals

Enhanced GSS-API error messages
Cross-platform CCAPI (Mac and Windows)
Kerberos Identity Management (KIM) API
Areas for Improvement

Modularity
Credential management
End-user experience
Administrator experience
Performance
Protocol evolution
Code quality
Modularity

Support readily building subsets (1.8)

“Lite” client
“Lite” server
GSS-API: context estab. vs msg. protection
e.g. Solaris user/kernel space split

Crypto (1.8)

Native (accelerated) crypto API support
Performance optimizations (caching, etc.)
New API design 1.7+
Modularity (cont’d)

GSS-API mechanism glue
  At least rough form for NTLM support (1.7)
  Possible refinements later (1.8)

KDC Database (long-term)
  Track IETF data model work
  New API for 1.8
  New implementation for 1.9

Secure co-processor (“would be nice”)
End-user Experience

Enhanced error messages for GSS-API (done)

Credential management
  KIM API (done)
  Cross-platform CCAPI
    Done for Mac & Windows
    UNIX implementation (1.7+)

Referrals (1.7)
  DNS independence via referrals

Localization of static error strings (1.7+)
Administrator Experience

Incremental propagation (1.7)
  Integrated; needs cleanup

Improve key rollover
  Master key (1.7)
  Application service keys (1.8)

Audit support (log all ticket requests) (1.7+)

Disable DES by default (1.8)
Performance

Decrease DNS traffic (1.7)
  Stop trying to crawl up to the root

Replay cache ("rcache")
  Disable on KDC (1.7)
    Avoid known false-positive issues
  Collision avoidance (1.7+)
  Improve implementation (1.7+)
  Disable by service type name (1.7+)

New crypto API (1.8) facilitates optimizations
Protocol Evolution

Encryption algorithm negotiation (1.7)
Microsoft Kerberos extensions (1.7)
Improved PKINIT support (1.7)
Anonymous PKINIT (1.8)
FAST (1.8; IETF)
International strings in protocol (1.8+; IETF)
Timestamp-independence (1.8, 1.9)
Replay-proofing protocols (1.8, 1.9)
Code Quality

Remove krb4 (1.7)

Use safer library functions (ongoing)
  - Avoid false positives
  - Avoid need to validate “unsafe” calls
  - Stop using strcpy, strcat, sprintf, etc.
    - Mostly done
    - New internal APIs for complex operations

Reduce commitment to “difficult” platforms
  - More effectively focus resources
Supported Platforms

Mac OS X
  “Darwin” command-line build

GNU/Linux (OS family)
  Currently Debian, Ubuntu, or Red Hat on x86_64 and x86

Solaris (SPARC or x86_64/x86)

BSD (OS family)
  Currently NetBSD on x86_64 and x86
Process Changes

Streamline project proposal process

Community resources
- Wiki for developers – k5wiki.kerberos.org
- Source browsers – OpenGrok, FishEye
- White papers, tutorials, best practices

Incrementally adopt style, review guidelines

Improve testing infrastructure

Analysis tools
- Coverity, compiler warnings (static)
- Valgrind, Purify (runtime)
Interface Change Strategy

Crypto, KDB, etc.

Incremental, staged approach

Design new interface

Upper layer on new interface

Implement new interface on top of old

New lower layer

Compatibility interface on top of new interface

If needed