

If it moves, Kerberize It!

Simon Wilkinson, University of Edinburgh

[<simon@sxw.org.uk>](mailto:simon@sxw.org.uk)

Overview

- Informatics Background & Motivation
- Case studies - political & technical
 - OpenSSH,
 - Thunderbird,
 - Jabber
- Summary of issues encountered
- Questions ...

Informatics@Edinburgh

- ~2000 hosts, ~4500 users
- 5 years of production Kerberos use
- Distributed, highly integrated environment
- Decreed that key applications must have Kerberos support before, or early on in roll-out

Why bother?

- Usability - “The fax effect”
- Security - reduces passwords in cleartext and cached locally

Open Source Kerberos

- Historically varied quality and degree of Kerberos support
- We've put effort put into ...
 - improving support where it exists
 - implementing support where it doesn't

OpenSSH introduction

- Two main vendors - OpenSSH vs ssh.com
- Two protocol versions v1 vs v2
- Two OpenSSH trees - portable & BSD

OpenSSH beginnings

- Initially no Kerberos support
- Daniel Kouril added protocol v1 support
- ... but Heimdal only (used raw API)
- Informatics added MIT version
- Code easily accepted into the portable tree

OpenSSH - take 2

- Still no Kerberos support for protocol v2
- Jeff Hutzleman et al wrote an I-D
- 2 different mechanisms!
- Informatics implemented both of these for Kerberos + Von Welch contributed GSI
- Release early, release often ?? ...

The saga begins

- Specification had some bugs, which were found during implementation
- Implementation had some bugs, which were found through inter-op testing
- Specification had a nasty security issue, requiring protocol changes
- Upshot - lots (3) of incompatible versions!

The saga continues

- But why still a patch?
- Specification is now stable, and an RFC
- Many vendors ship with support
- “Why can’t I do GSSAPI key exchange with a stock OpenSSH?”

... and continues

- Its not easy to get code into OpenSSH
- kerberos-2@ssh.com
- code complexity
- feature complexity
- lack of understanding of enterprise requirements & politics!

OpenSSH Today

- User auth support is in the distribution
- ... but disabled by default
- Still no key exchange, and little chance of it
- What next?

Thunderbird

- Approach from lead developer to add Kerberos support
- Lots of support from Mozilla Foundation throughout

Thunderbird coding

- Existing library to intergrate with GSSAPI
- Existing SASL support (DIGEST-MD5, etc.)
- Implementation had to be internal
- Extended GSSAPI interface to do SASL
GSSAPI - available to all Mozilla based code
- Protocol specific code added for POP3, IMAP and SMTP

Thunderbird downsides

- No security layers
- Hey, they're hard - have to wrap every IO call
- TLS makes this a little easier...
- Assumptions in existing code that you'll only ever need 1 round trip

Thunderbird binaries

- Don't want run-time dependencies
- So, dynamically load GSSAPI library
- Vital for Windows (SSPI vs KfW)
- Removes client library dependence
- Also simplifies build issues
- But, you don't always get what you want!

GSSAPI imposters

- NFSv4 libgssapi doesn't play nicely with the other children!
- Incomplete API implementation
- Calls exit when misconfigured
- Makes users sad
- ... and developers request code removal

If your name's not on the list ...

- Thunderbird lists 'good' gssapi libraries
- ... but NFSv4 library shares a revision with Heimdal's library
- ... so, check for symbols in the library
- The things we do to get in the default build!

Jabber

- Consider a whole set of applications now
- Protocol uses SASL for authentication
- Not one Open Source application offers Kerberos/GSSAPI authentication

Requirement dangers

- XMPP RFCs require DIGEST-MD5 and PLAIN
- Cyrus SASL considered over complex
- Everyone rolled their own!
- Can't add new mechanisms by throwing a switch

Jabberd2

- Ripped out existing SASL library - scod
- Replaced it with Cyrus SASL
- Implementation tricky, but now in CVS for 2.1
- Easy to get code accepted, but project may be stagnating

Gaim

- If a things worth doing its worth doing twice ...

Gaim - round 1

- First implementation Cyrus SASL based
- Replaces existing code with Cyrus calls
- In CVS for 2.0
- No use on Windows or Mac OS X
- Can't prompt for passwords
- Fallback hard to do neatly

Gaim - round 2

- Add internal GSSAPI SASL mechanism
- Works on Windows and Mac OS X
- Can prompt for passwords when required
- Still under development - see gaim-devel

Psi

- Development version uses Cyrus SASL
- But assumes ...
 - that every account will have a password!
 - that if one mech fails, they all will

Lessons - political

- Never underestimate the difficulty of getting your patch accepted
- Get lead developers onside early on
- But, its often difficult to convince them of the need for enterprise features
- Making Kerberos invisible to normal users is vital

Lessons - technical

- It's not over once the code's in
- Most developers build without your code
- Hardly anyone tests your code

Lessons - technical

- Avoid the raw Kerberos API
- Avoid implementing IDs in their early years
- Runtime loading is a double edged sword
- Using CyrusSASL is a good thing (on Linux)

Dangerous Assumptions

- Every mechanism only needs one round trip
- If one mechanism fails, they all will
- We'll always need a password
- Security layers - what's that?

Questions?

simon@sxw.org.uk

... or catch me in the bar!