Wallet

Secure Data Distribution and Management

2008 Update

Russ Allbery

May 13, 2008

Contents

- What is the Wallet?
- The Goal
- The Architecture
- Example Wallet Object
- Wallet Object Types
- Wallet ACL Verifiers
- Next Steps: 1.0...
- Next Steps: ...and Beyond

What is the Wallet?

- Manages any type of secure data
  - Keytabs and simple files implemented
  - WebAuth keyrings and X.509 CA planned
  - Extensible system for adding new types
- Rich ACL mechanism
  - Fine-grained access control for operations
  - Simple identity and NetDB implemented
  - Extensible system for adding more verifiers
  - PTS, LDAP, and nested groups planned
- Built on remctl, but server and client can be replaced with any other authenticated RPC layer (SOAP, REST, etc.) without major server changes

The Goal

- All secure data goes into the wallet
- All non-secure system information goes into Puppet
- Fully automated server deployment except for initial keying (and maybe initial keying as well)
- Chained permissions: key the server and the server key can download other required secure data
- Central management of secure data
- Allows automated rekeying where appropriate
- Unchanging support for generated objects

The Architecture

- Authorization and privacy via remctl protocol
- C client with simple command passthrough, handling of file creation and some special keytab logic
- Server wrapper that interprets remctld authentication
- Wallet::Server handles ACL checking and high-level API
- Separate ACLs for show/get/store/destroy and owner
- Wallet::Object::* implements each wallet type
- Wallet::ACL::* (will be renamed) implements ACL types
- Basic support for local policy and object autocreation

Example Wallet Object

```
           Type: keytab
           Name: host/windlord.stanford.edu
          Owner: host/windlord.stanford.edu
       Enctypes: aes256-cts
     Created by: rra/root@stanford.edu
   Created from: windlord.Stanford.EDU
     Created on: 2007-12-06 16:55:13
  Downloaded by: rra/root@stanford.edu
Downloaded from: windlord.Stanford.EDU
  Downloaded on: 2008-02-08 13:38:56

Members of ACL host/windlord.stanford.edu (id: 2) are:
  krb5 host/windlord.stanford.edu@stanford.edu
  netdb-root windlord.stanford.edu
```

Wallet Object Types

- Support create, destroy, get, store
- Can hook into flag settings
- Can support arbitrary per-type attributes (example: enctypes)
- Currently implemented:
  - Simple file objects (opaque data chunks)
  - Kerberos keytabs

Wallet ACL Verifiers

- Initialize method to create persistant resources
- Check method to check an identity against an ACL value
- Currently implemented:
  - krb5 (simple identity comparison)
  - NetDB roles (Stanford's GPL'd host management software)
- Nested groups will require some special handling to prevent recursion

Next Steps: 1.0...

- Better history support for deleted objects
- Better reporting and search
- Heimdal support for the client (and maybe server)
- Upgrade support for the database
- LDAP and PTS ACL verifiers
- WebAuth keyring object type
- Tests, tests, tests

Next Steps: ...and Beyond

- X.509 and ssh keypair object type support
- Rekeying
- remctl server fixes to allow data containing nuls
- Better object templating for autocreation
- Even more documentation: conventions, naming, replacing the protocol
- More native Perl support for kadmin and Kerberos
- CGI and REST proof of concept