Scope Statement
For
Shared Key Authentication and Encryption in Lustre 2.X

Revision History

Date	Revision	Author
2012-07-10	Created	Andrew Korty
2012-11-10	Version 2	Stephen Simms

Scope Statement

Table of Contents

Introduction **2**

Problem Statement **2**

Project Goals **2**

In-Scope **2**

Out of Scope **2**

Project Constraints **2**

Project Assumptions **2**

Key Deliverables **2**

Key Milestones **2**

Scope Statement

Introduction

The following scope statement describes a project to add shared-key authentication and encryption to the UID/GID mapping project within the SFS-DEV-002 contract/SOW beginning 10/23/2012. The new functionality will provide node-based authentication (rather than user-based authentication) and will use Lustre's existing support for the Generic Security Services Application Program Interface (GSSAPI), which provides access to security services. The GSSAPI is an IETF standard that allows arbitrary security services to be developed in the form of plug-ins called *mechanisms*.

Problem Statement

Currently, client-level strong authentication and encryption in Lustre is handled by Kerberos via the GSSAPI. While a robust and well-respected scheme, Kerberos is not an existing part of every site's infrastructure. Those sites that don't run Kerberos may be hesitant to implement it for policy or resource reasons. Lightweight authentication and encryption mechanisms are needed in Lustre to allow rapid deployment in such environments.

Project Goals

1. Implement a null mechanism for GSSAPI.
2. Implement the null mechanism security flavor within Lustre 2.X to provide a method for testing.
3. Implement a shared-key mechanism for GSSAPI.
4. Implement the shared-key authentication and encryption security flavors for Lustre 2.X.
5. Implement userspace tools to create shared keys, and methods to make the keys available to Lustre without exposing them.

In-Scope

- Develop null and shared-key GSSAPI mechanisms.
- Develop shared-key authentication and encryption security flavor code on a Lustre project branch that will perform comparably with the Kerberos security flavors.
- Provide performance data for shared-key authentication and encryption.
- Observe best practices for cipher selection and key generation and management. Ciphers and algorithms will be chosen based on their abilities to resist cryptanalysis, including known-plaintext attacks.
- Provide appropriate explanation and/or documentation of design decisions

Scope Statement

- Provide unit tests for null and shared key mechanisms to be included in the Lustre test directory.
- Preserve interoperability with clients of one minor revision prior to release.
- Ensure co-existence of shared-key GSSAPI mechanism and security flavors with Kerberos mechanisms and security flavors.
- Produce documentation in the form of additions or creation of man pages detailing new features.
- Produce documentation in the form of additions to the Lustre 2.X manual.

Out of Scope

- Dynamic GSSAPI mechanism negotiation. Authentication mechanisms must be configured statically.
- Tools to transport shared keys among multiple Lustre sites or perform any other key management outside of Lustre.
- Removal of the GSSAPI's build dependency on Kerberos libraries. Separating the GSSAPI from Kerberos in the autoconf code is non-trivial and would require more time than allotted for the project. Kerberos libraries will therefore need to be present at build time, even for systems not using Kerberos authentication.

Project Constraints

None known.

Project Assumptions

- Landing is dependent on availability of Lustre tree maintainers to assist with integrating code into the release tree.
- Test hardware on the OpenSFS Test Cluster will be made available.

Key Deliverables

The key deliverables for this project are:

- Signed Milestone documents for project phases:
  - Solution Architecture
  - High-Level Design
- Test Plan
- Source code that meets feature requirements and runs with Lustre 2.X
- Source code for new test cases
- Shared-key authentication code landed in Master WC-Lustre 2.X
- Performance evaluation of encrypted vs unencrypted traffic

Key Milestones

- **Start + 2 weeks:** Scope statement completed
- **Start + 4 weeks:** Solution architecture documented completed

Scope Statement

- **Start + 6 weeks:** High-level design document completed
- **Start + 12 weeks:** Solution implemented
- **Start + 16 weeks:** Code tested and found problems fixed
- **Start + 20 weeks:** Demonstration prepared and solution delivered

Glossary

- GSSAPI: Generic Security Services Application Program Interface – a generic API that provides high level functions for authentication and encrpytion.
- HMAC: Hashed Machine Authentication Code – a cryptographic method that produces a signature with the content and a shared key.
- Kerberos: Security infrastructure that uses a trusted third party to authenticate and encrypt/decrypt data.
- Shared Key: Symmetic encryption where both the sender and the receiver shared a common secret to produce signatures and encrypt/decrypt data.