Lustre Test Plan for Security Features

Author	Date	Description of Document Change	Client Approval By	Client Approval Date
Jian Yu	05/13/2008	First draft.		
Jian Yu	06/02/2008	Second draft. Update the test plan with the suggestions from Eric Mei.		
Jian Yu	06/03/2008	Update the “Test Plan Approval” section.	Eric Mei	06/03/2008

I. Test Plan Overview

Executive Summary

- Statement of the problem trying to solve: testing Lustre security features to make them production ready in Lustre 2.0.0 (based on HEAD branch)
- Required inputs: cvs tag name of HEAD branch or security feature development branch name, Bugzilla tracking ticket
- Hardware to be used: Washie test cluster
- Software to be used: acceptance-small test suite (which contains security test utilities) in Lustre 2.0.0 and I/O benchmarks (IOR and mdsrate)
- Expected outputs: a functional state of the security features (regressions, new defects detected in the testing, etc.), a performance report for comparing different security flavors
- Collaborators: Eric Mei and Fan Yong (Lustre Owl team)

Problem Statement

Security features in the current HEAD branch are not production ready yet. Many functions need to be tested further, such as capability feature, shifting among all supported security flavors, how the security features work with CMD (Clustered MetaData) configuration, etc. The test cases QE team used for testing against security features were insufficient in the past. More test cases and test scenarios need to be designed and performed to make the security codes stable and production worthy.

Goal

The main testing goal is to find out the defects in the Lustre security features and drive the Lustre security codes to a stable and production ready state. It contains:

- complete acc-sm run with shifting among different security flavors (null, krb5i and krb5p) between tests
- complete I/O benchmark (IOR and mdsrate) run against all of the supported security flavors (null, plain, krb5n, krb5a, krb5i, and krb5p), gather and analyze the performance data

Success Factors

All of the tests need to be run with non-CMD (1 MDT) and CMD (4MDTs) configurations on at least three nodes (1 MDS/KDC, 1 OSS and 1 Client). Remote ACL testing needs two clients (1 local client and 1 remote client).

If no defect, functional and performance regression occur during testing, then the feature testing is regarded as successful.

References

- MIT Kerberos: http://web.mit.edu/Kerberos/
- Kerberos Lustre Setup: http://wiki.lustre.org/index.php?title=Kerb_Lustre

Testing Plan

Define the setup steps that need to happen for the hardware to be ready? Who is responsible for these tests?
Supported distributions: As per the lustre/ChangeLog in the current HEAD branch, the following distributions are supported for Lustre 2.0.0 servers: RHEL 5 and SLES 10. Clients are all patchless and work with kernels 2.6.16 and greater. For SLES 10, patchless client could not work with GSS/Krb5 enabled now because SLES 10 (up to SP2) vendor kernel doesn't have CONFIG_KEYS enabled. The next SP 3 may enable it. So, before SLES 10 SP 3 is available, we need setup Lustre client with patched kernel and packages. Supported architectures: x86_64, i686, ia64 According to the supported distros and archs, tests need to be run against the following platforms: RHEL5/x86_64, RHEL5/i686, RHEL5/ia64, SLES10/x86_64, SLES10/i686 Testing environment setup steps are as follows: 1) submit build requests (respective for patched and patchless) on LBATS system for building Lustre packages against HEAD branch or security feature development branch with GSS/Krb5 enabled (configuring Lustre with "--enable-gss" option) 2) reserve Washie test cluster time through Cluster Scheduler 3) setup the test cluster by running setup_nodes.sh on lts-head node 4) setup Kerberos environment on the test cluster by running setup_kerberos.sh on one of the test nodes After testing environment is setup as per the above steps, tests against Lustre security features can be started. All of the test results need be reported to Buffalo by running send_report.pl on lts-head node. QE team in Lustre group is responsible for setting up the test environment, running the tests, vetting and reporting the test results.

Specify the date these tests will start, and length of time that these test will take to complete.
The tests will start in the following situations: 1) regular HEAD branch testing 2) security feature development branch testing It would take one QE about three days for setting up the test environment on one distro/arch, running the tests with one configuration (non-CMD or CMD), vetting and reporting the test results. For supported architectures, i686 and ia64 are less important than x86_64. QE needs focus more effort on the core combinations (SLES10/x86_64 and RHEL5/x86_64).

Specify (at a high level) what tests will be completed?
The following tests would be completed:

1) Functional tests: acceptance small test suite 2) Performance tests: IOR and mdsrate

Specify how you will restore the hardware (if required) and notify the client your testing is done.
All of the test results would be reported to Buffalo according to the vetting and reporting process. Status summary and performance report would be updated in Bugzilla tracking tickets.

Test Cases

Functional Test Cases Besides the existing test cases in the current acceptance small test suite (including **sanity-gss.sh** and **sanity-sec.sh**, which are security feature specific) on HEAD branch, the following security feature specific test cases also need be executed:
1. Remote ACL: local user setfacl for remote user/group
2. Remote ACL: remote user setfacl for user/group on the same remote client
3. Remote ACL: remote user setfacl for user/group on other clients
4. Remote ACL: remote user setfacl for user/group on the same remote client who is not mapped dynamically on server
5. Remote ACL: remote user use "lfs {l,r}getfacl"
6. Remote ACL: local user use "lfs {l,r}{s,g}etfacl"
7. Remote ACL: non-owner use "lfs rgetfacl"
8. Remote ACL: server config "rmtacl/normtacl" permission
9. GSS/Krb5 authentication: security server failure (one master and one slave KDC need be setup before running this test case)

Performance Test Cases The following tests would be run for comparing the changes in performance while using different security flavors (null, plain, krb5n, krb5a, krb5i, and krb5p):
1. I/O performance (IOR)
2. Metadata performance (mdsrate)

Benchmarking

There are no previous performance results for Lustre security features on Washie test cluster. QE team need gather the performance data for comparing overhead among different security flavors and finding any performance defects.

II. Test Plan Approval

- Review date for the Test Plan review with the client
  - 05/31/2008 – reviewed by Eric Mei
  - 06/03/2008 – reviewed by Eric Mei
- Date the Test Plan was approved by the client (and by whom)
  - 06/03/2008 – approved by Eric Mei
- Date(s) agreed to by the client to conduct testing
  - 06/03/2008 – security feature testing could be started on regular HEAD branch testing