Test Plan

SELinux support on client side

version: 3
date: 2015/08/07
author: Sebastien Buisson – sebastien.buisson@atos.net

Introduction

While the Lustre Manual still states that SELinux must be disabled prior to running Lustre, since LU-506 (Lustre 2.3) and the replacement of *ll_d_add()* with *d_add()* (which internally calls security_d_instantiate()), SELinux does not prevent Lustre from working properly anymore.
However, work done in LU-506 is not enough to be able to enforce SELinux security policy on Lustre.

The goal of this project is to support SELinux on the Lustre client side, which means being able to enforce SELinux security policy on the Lustre client.
This work is tracked under Jira ticket LU-5560.

Tests

Non regression tests

The full set of Auster test suites will be run to verify there are no functional regression.

Functional tests

The Auster test suite is enhanced with a new script called sanity-selinux.sh, designed to run on an SELinux-enable Lustre client. It contains the following test cases:

- test 1: create a file
  Check that security.selinux xattr is set on MDT.
- test 2: create a directory
  Check that security.selinux xattr is set on MDT
  - when directory is created with mkdir;
  - when directory is created with 'lfs mkdir' or 'lfs setdirstripe',
- test 3: SELinux user « unconfined »
  Check that user has transparent access to Lustre filesystem.
- test 4: Specific SELinux user
  Check that:
  - with no SELinux access rights: all accesses denied;
  - with SELinux access rights: corresponding accesses granted by policy

- test 5: security context retrieval from MDT xattr
  Check that after purging client's cache, security context is correctly retrieved from MDT xattr.
- test 10: [consistency] security context change from another client
  Initial node can see security context change.
- test 20: [atomicity] concurrent access from another client
  Second client sees no file, or the file with correct security.selinux xattr, but not a file with wrong security.selinux xattr.

Performances

The performance impact of SELinux will be evaluated, in terms of metadata performance.

Upgrade and downgrade

Upgrade and downgrade of the Lustre client will be verified.
Upgrade and downgrade of the Lustre server will be verified.

Results

Non regression tests

Complete Auster test suite has been triggered by Maloo when patch from LU-55660 was submitted.
All tests passed.

Functional tests

Functional tests have been carried out on the following in-house test system:

- software
  - RHEL 6
  - Lustre 2.5 + patches
- hardware:
  - 1 MGS
  - 1 MDS+OSS } ramdisk storage
  - 1 client
    - 12 cores
    - 24 GB RAM
  - Interconnect: Infiniband QDR

All test cases in sanity-selinux.sh pass, with the following exceptions:
- test 10: consistency
- test 20: atomicity
These limitations have already been identified in LU-5560.

Performances

To evaluate SELinux impact over metadata performance, mdtest benchmark was launched on an SELinux-enabled client on the in-house test system detailed above.

With the version of the patch from LU-5560 that bypasses Lustre xattr cache for security.selinux extended attribute, we show that there is a drop in performance of about 35% in file create, 35% in file stat, and nearly no impact over file remove.

For complete details and explanations on the performance figures, please see the presentation given at LAD'14:
http://www.eofs.eu/fileadmin/lad2014/slides/10_Sebastien_Buisson_Lustre_SELinux_v2.pdf

Upgrade and downgrade

Upgrade and downgrade tests have not been run yet.