ZOOKEEPER-1045 Test Report

Last Updated: 10/27/2016

Reference: Test Plan; Feature Document, Previous Test Report

Summary:

- No regressions found. Major functional cases passed.
- Two issues found and documented in the report.
- Not all test cases have been executed due to time constraints.

Target:

- ZOOKEEPER-1045 latest (10-24) patch.

Test Cluster Setup:

- A 3 node ZooKeeper ensemble with server 1, 2, and 3.
- KDC installed on ZooKeeper server 1.
- Environment: Ubuntu-14.04, Kerberos 5 version 1.12, Java version "1.7.0_101".

Build Verification / Sanity Check **(PASS)**:

- Verify source builds with JDK 1.6, 1.7 and 1.8 on Linux and OSX.
- Run all Java and C unit tests with default configuration (server to server mutual authentication disabled).
- Verify a ZooKeeper ensemble with size 3 is able to start with default configurations.
- Verify a ZooKeeper ensemble with size 3 is able to start with server to server mutual auth enabled.
- The following procedures performed to verify the soundness of an ensemble:
    a. Use zkCli to connect to each server in ensemble, executed commands.
    b. Server logs were examined, no errors found.
    c. No observable slowness on leader election.

Functional Cases:

- jaas.conf (Replace 1 with 2 and 3 respective to get config for server 2 and 3.).

```
QuorumServer {
  com.sun.security.auth.module.Krb5LoginModule required
  debug=true
```

```
  useKeyTab=true
  keyTab="/root/zookeeper/conf/zkquorum-hanm-zk-kerberos-1.keytab"
  storeKey=true
  useTicketCache=false

principal="zkquorum/hanm-zk-kerberos-1.vpc.cloudera.com@VPC.CLOUDERA.C
OM";
};
QuorumLearner {
  com.sun.security.auth.module.Krb5LoginModule required
  debug=true
  useKeyTab=true
  keyTab="/root/zookeeper/conf/learner-hanm-zk-kerberos-1.keytab"
  storeKey=true
  useTicketCache=false

principal="learner/hanm-zk-kerberos-1.vpc.cloudera.com@VPC.CLOUDERA.CO
M";
};
```

- zoo.cfg

```
tickTime=2000
initLimit=10t
syncLimit=5
dataDir=/tmp/zookeeper
clientPort=2181

server.1=hanm-zk-kerberos-1.vpc.cloudera.com:2888:3888
server.2=hanm-zk-kerberos-2.vpc.cloudera.com:2888:3888
server.3=hanm-zk-kerberos-3.vpc.cloudera.com:2888:3888

quorum.auth.enableSasl=true
quorum.auth.learnerRequireSasl=true
quorum.auth.serverRequireSasl=true

quorum.auth.kerberos.servicePrincipal=zkquorum/_HOST
```

- **PASS**: SASL server to server mutual authentication is enabled on every server and every server has deployed valid Kerberos credentials with principal name in the form of name/host@REALM.COM. Verified quorum form and zkCli works.
- **FAIL**: Similar to previous case setup, except that the Kerberos principal name is changed to name@REALM.COM (remove the ‘host’ part). We expected this to work, but this form of principal name turns out to be malformed and not supported. What actually got is ZK server fail to parse zoo.cfg with call stack:

```
[myid:3] - ERROR
[QuorumConnectionThread-[myid=3]-3:SecurityUtils@249] - server principal
name/hostname determination error:
java.lang.StringIndexOutOfBoundsException: String index out of range: -1
        at java.lang.String.substring(String.java:1911)
        at
org.apache.zookeeper.util.SecurityUtils.createSaslServer(SecurityUtils.java:173)
        at
org.apache.zookeeper.server.quorum.auth.SaslQuorumAuthServer.authenticate(SaslQ
uorumAuthServer.java:93)
        at
org.apache.zookeeper.server.quorum.QuorumCnxManager.handleConnection(Quorum
CnxManager.java:451)
        at
org.apache.zookeeper.server.quorum.QuorumCnxManager.receiveConnection(Quoru
mCnxManager.java:371)
```

Analysis:

The SASL server creation logic reuses the previous logic introduced in ZOOKEEPER-938, which requires that a principal name must be in the form of 'name/host'. The reason we want to support name@REALM.COM principal name is to satisfy the use case where every ZK node will share the same Kerberos principal, which simplify deployment and management of Kerberos principal. In this use case, it is obvious to see that the 'host' portion does not matter, so we omit it in the principal name.

To address this issue, there are at least two choices:

a. Instead of supporting name@REALM.COM, we require the shared principal name must contain a host section, and this section must be 'localhost'. The existing principal name parsing logic will pass against name/localhost@REALM.COM . In authorization check logic, we just skip authorization check when we have a match of 'localhost' as the host section.
b. Alternatively, we can update existing principal parsing logic so ZooKeeper does not expect a host section in the principal name.

- **PASS**: SASL server to server mutual authentication is enabled on server 1 and 2 which has also deployed with valid Kerberos credentials. Server 3 has SASL server to server authentication disabled (quorum.auth.enableSasl=false). Verified quorum still form (between server 1 and 2), zkCli works (when connecting to either server 1 or 2), but server 3 can't join quorum.

- **PASS**: SASL server to server authentication is enabled on all three servers. Server 1 and 2 have valid Kerberos keytabs deployed, server 3 has an invalid Kerberos credential. Verified that server 3 fail to pass auth check and fail to join quorum.
- **FAIL?**: Similar as previous case, except that server 3 now have server 1's credential (both jaas.conf and keytabs are the same between server 1 and server 3.). In other words, server 3 is now impersonating server 1. We expect server 3 fail to authenticate and fail to join quorum. However, server 3 can actually join quorum. How?

  Analysis:

  This is an interesting case because the credential server 3 has is valid, in the context of Kerberos, but not valid, in the context of the server 3 as we expect server 3 has and only has its own specific Kerberos credential. In other words, we would treat impersonating as a failure because that is what authorization (as opposed to authentication) supposed to do.

  What actually happened is (thanks **Rakesh** for the analysis): server 3 is not accepting any (1 & 2) learner's auth request because when server 1&2 Learner sending AuthPacket to 3, these servers will find server3 host from ensemble details, and replace _HOST with hanm-zk-kerberos-3.vpc.cloudera.com. and form the principal as zkquorum/hanm-zk-kerberos-3.vpc.cloudera.com@VPC.CLOUDERA.COM. But QuorumAuthServer of 3 has registered his principal as zkquorum/hanm-zk-kerberos-1.vpc.cloudera.com@VPC.CLOUDERA.COM , so an auth error will throw inside Kerberos GSSAPI for the mismatch of the principal names (checksum failed).

  But interesting thing is, 3 can act as a valid Learner, because he can replace _HOST of 1 & 2 properly.

Stress/Scaling:

- **PASS:** Internal Jenkins job to run unit tests at stress (i.e. run UT every hour).

Test pass rate is **95**%, failed build caused by known flaky tests (the most active flaky test is org.apache.zookeeper.server.zxidrollovertest).

- **WIP:** Set up a 3 node ensemble that contains 2 servers a, b with valid credential & configuration and 1 server c with invalid credential. Run the ensemble for one week. Verify the ensemble is in health state: leader statistic look normal in terms of memory / sockets consumption. Now grant server c with valid credential and verify server c can connect to ensemble immediately without obvious delay.

Fault Injection:

- **PASS:** Shutdown leader and verify quorum converge with no obvious slow down when SASL server to server auth is enabled.
- **PASS:** Do a rolling restart of the ensemble and verify as previous test case.

Performance:

- No observable slow down on leader election / quorum converge when mutual auth is enabled on a 3 node ensemble.

Open Items:

Due to time constraints and priority these tests are not performed / not extensively executed:

- Rolling upgrade.
- SASL Digest-MD5 scheme.
- Stress and performance tests.