CASSANDRA-1567

rnirmal

2010-10-01

Contents

Chapter 1

Root issue CASSANDRA-1567

1.1 Summary

Provide configurable encryption support for internode communication

1.2 Description

Provide the option to encrypt internode communication. The initial thought is to use JSSE (http://download.oracle.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html) to wrap the existing ServerSocket & Sockets. This will only be an optional configuration and not enabled by default. The defaults would be TLS V1, RSA 1024-bit keys for handshake and SSL_RSA_WITH_RC4_128_MD5 as the cipher suite. Although this can be made configurable if the need arises.

1.3 Commits

No related commits

1.4 Comments

1. **rnirmal:** working code. Need to update the configuration for keystores

2. **stuhood:** * For 0001, I would really like to see an {{A(bstract)StreamableSocket}} rather than complete duplication of the Stream classes
   * Rather than a boolean, the {{internode_encryption}} setting should probably be an enum, to leave room to add conditional encryption based on zones returned by the snitch
   * The SSL settings in JVM_OPTS should be disabled by default, and need a comment linking to a place to get more information about the keystore and truststore files (probably the 'Creating Keystores' section of the link in the description)

   Sorry for the long delayed review: Thanks a ton for tackling this!

3. **stuhood:** Also, can we add a startup message that indicates the encryption mode being used?

4. **rnirmal:** bq. For 0001, I would really like to see an A(bstract)StreamableSocket rather than complete duplication of the Stream classes
   Done
   bq. Rather than a boolean, the internode_encryption setting should probably be an enum,

to leave room to add conditional encryption based on zones returned by the snitch
Updated to use an enum, just (all, none) for now.
bq. The SSL settings in JVM_OPTS should be disabled by default, and need a comment linking to a place to get more information about the keystore and truststore files (probably the 'Creating Keystores' section of the link in the description)
Having those properties in should not be a problem. We can provide a wiki page on how to get everything setup.

5. **rnirmal:** There's one more change I'm going to add, and will hopefully have it out soon, rebased and all. I'm going to make the default cipher suite we use to AES_128/256 with SHA.

6. **stuhood:** Nirmal mentioned that he was going to do a bit more refactoring of this one before calling it reviewable again.

7. **rnirmal:** So for this I think we'll go with just all internode encryption with AES_128/256 in an either/or situation. Either all your cluster node transfers is encrypted or not. Based on if there's demand to have just cross DC encrypted we can update it at that point and if users want to configure encryption options.

8. **jbellis:** Where did patch 0001 go? Was it committed separately?

9. **rnirmal:** During the last update I combined 0001 & 0002. So everything is in 0002 and 0003 is a sample keystore and env config. yet to upload the latest changes, but what's there works

10. **jbellis:** bq. make the default cipher suite we use to AES_128/256 with SHA

    This looks like all that needs to be done to close out this ticket. That and probably a fairly hairy rebase. :)

11. **rnirmal:** Attaching rebased versions (-V2) with the latest updates. I've tested it a bit and seems to work fine. Would be nice to test it out a little more.

12. **xedin:** Nirmal Ranganathan: Can the latest 002 and 003 patches be considered as complete solution?

13. **rnirmal:** Yes it can be considered as patch ready. Pavel it will be great if you can review it too.

14. **xedin:** Great! I will review it too. Can you please change status to Patch Available?

15. **xedin:** Minor notes:

    1. Clean code style of the method definitions (e.g. DatabaseDescriptor methods size-MemtableThroughput, sizeMemtableOperations)
    2. remove space in line "bytesRead += buf. limit();" of SSLIncomingStreamReader class

    Everything else looks good.

16. **gdusbabek:** Am I missing something big, or does this only encrypt stream communications?

    EDIT: nm. Stu pointed out the parts that I missed.

17. **stuhood:** Nirmal, thanks again for your work here: this will really be a killer feature.

    * Could we add a class level javadoc to SSLFileStreamTask to indicate that it exists because FileStreamTask uses sendFile?

    +1 aside from that.

18. **rnirmal:** Pavel: I've updated based on your notes, btw sizeMemtableThroughput and sizeMemtableOperations shouldn't be there in the first place, I think it got left behind in one of my rebases, removed it now.

    StuHood: Thanks for looking through it, I've added comments to explain why SSLSocket/SSLServerSocket cannot encrypt data transferred using FileChannel.transferTo/transferFrom. Gary: As StuHood mentioned that's the reason the additional SSL versions. But all internode data is getting encrypted.

19. **gdusbabek:** * setReuseAddress happens before bind.
    * added link to keytool docs in cassandra.yaml.

20. **rnirmal:** Updated the patch to cleanly apply on 0.7 branch and trunk

21. **gdusbabek:** Moving to 0.8 based on the "shorter release schedule" thread on -dev.

22. **gdusbabek:** committed. excellent work!

23. **hudson:** Integrated in Cassandra #678 (See [https://hudson.apache.org/hudson/job/Cassandra/678/]) configurable internode encryption. patch by rnirmal, reviewed by gdusbabek. CASSANDRA-1567

24. **jbellis:** where did we document how to use this?

25. **rnirmal:** Haven't documented yet, just the info in the conf file for now, since it was moved to 0.8 release. I haven't looked at the wiki recently, if we have sections or docs for 0.8 release, I'll add this with a note.

1.5 Pull requests

No pull requests