STITCH FIX

How to Secure Apache Spark?

Neelesh Srinivas Salian

About Me

Stitch Fix – Data Platform

Cloudera – YARN, Spark, Kafka

Apache Software Foundation – Flink, Beam and Spark

Agenda

1. Spark Security
2. Issues
3. Configurations
4. Questions

Security in Spark

org.apache.spark.SecurityManager.scala

1. Authentication
2. Authorization
3. UI Security
4. Encryption

Authentication

Authentication can be defined as "The process of evaluating a user and their presented credentials to verify if their identity for interaction with a system or service".

- Spark currently uses the HashLoginService to authenticate using DIGEST-MD5 via a single user and the shared secret.

- Authentication can be configured to be on via the ***'spark.authenticate'*** configuration

- Tools to help enable Authentication

1. Kerberos / SPNEGO - Network layer
2. LDAP - Cluster service level

Security in Spark

org.apache.spark.SecurityManager.scala

1. Authentication
2. Authorization
3. UI Security
4. Encryption

Authorization

Authorization is the process of validating an authenticated user has effective rights to access a service, or data within a service.

Tools to help achieve Authorization

1. Hadoop User/Group Authorization
2. LDAP Authorization

Security in Spark

org.apache.spark.SecurityManager.scala

1. Authentication
2. Authorization
3. UI Security
4. Encryption

Secure UI

- *spark.acls.enable, spark.ui.view.acls and*
  spark.ui.view.acls.groups control the behavior of the ACLs.

- Secured using Javax servlet filters
  spark.ui.filters

--Dspark.ui.filters=com.test.filter1
--Dspark.com.test.filter1.params='param1=foo,param2=testing'

Security in Spark

org.apache.spark.SecurityManager.scala

1. Authentication
2. Authorization
3. UI Security
4. Encryption

Encryption

Spark supports SSL for HTTP protocols. SASL encryption is supported for the block transfer service and the RPC endpoints.

For SSL:

spark.ssl.fs, spark.ssl.ui, spark.ssl.standalone, spark.ssl.historyServerHistory

For SASL

spark.authenticate.enableSaslEncryption

spark.network.sasl.serverAlwaysEncrypt

HDFS Data at Rest Encryption

HDFS supports transparent end-to-end encryption of data read from and written to HDFS without requiring to change the user's application code

- Data can only be encrypted and decrypted by the client
- Encryption Zones
- Hadoop Key Management Server (KMS)

Issues

Incorrect permissions for the keytab directory

Symptom:

.. javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided Mechanism level: Failed to find any Kerberos TGT]

Solution:

1. Change the access mode of the Spark Server Keytab file:

   `chmod 600 spark-server.keytab`

2. Then change ownership of the file:

   `chown spark:spark spark-server.keytab`

3. Then Verify the keytab is accurate:

   `kinit username@MYDOMAIN.COM -k -t spark-server.keytab`

Spark Applications Intermittently Fail

Symptom: A Connection Refused error may often occur.

Cause:
Spark assigns a random port number to a group of configuration properties that are used for communication between the client and the cluster on a Spark on YARN installation. If any of those random port number assignments fall outside of the range of open ports at the time when the application is submitted, it fails.

Restriction on the Range of Ports

spark-submit --class org.apache.spark.examples.SparkPi --conf spark.driver.port=54521 --conf spark.executor.port=54522 --conf spark.fileserver.port=54523 --conf spark.broadcast.port=54524 --conf spark.replClassServer.port=54525 --conf spark.blockManager.port=54526 /opt/example/parcels/CDH/jars/spark-example*.jar 2

Long running applications Fail – Token Expiry

In *yarn-site.xml*

- Set *yarn.resourcemanager.proxy-user-privileges.enabled=true*

In *core-site.xml*

- Set *hadoop.proxyuser.yarn.hosts=**
- Set *hadoop.proxyuser.yarn.groups=**

Configurations

EMR or Local Setup

→ *spark-defaults.conf*

spark.master=yarn

spark.history.ui.port=18080

spark.driver.memory=2g

spark.authenticate=true

spark.authenticate.secret=history

spark.history.fs.update.interval=60s

spark.ssl.enabled=true

spark.ssl.protocol=TLSv1

spark.ssl.historyServer=true

Contact

Blog: http://multithreaded.stitchfix.com/algorithms/

Email: neeleshssalian@gmail.com

LinkedIn: Neelesh Srinivas Salian

Twitter: @NeelS7

Thank You