Security Guide On Sqoop 2

Title: Authentication Guide in SQOOP 2

JIRA : SQOOP-1525 and its sub tickets

Security Guide On Sqoop 2

Most Hadoop components, such as HDFS, Yarn, Hive, etc., have security frameworks, which support Simple, Kerberos and LDAP authentication. currently Sqoop 2 provides 2 types of authentication: simple and kerberos. The authentication module is pluggable, so more authentication types can be added.

Simple Authentication

Configuration

Modify Sqoop configuration file, normally in <Sqoop Folder>/server/config/sqoop.properties.

```
org.apache.sqoop.authentication.type=SIMPLE
org.apache.sqoop.authentication.handler=org.apache.sqoop.security.SimpleAuthenticationHandler
org.apache.sqoop.anonymous=true
```

- Simple authentication is used by default. Commenting out authentication configuration will yield the use of simple authentication.

Run command

Start Sqoop server as usual.

```
<Sqoop Folder>/bin/sqoop.sh server start
```

Start Sqoop client as usual.

```
<Sqoop Folder>/bin/sqoop.sh client
```

Kerberos Authentication

Kerberos is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.

Dependency

Set up a KDC server. Skip this step if KDC server exists. It's difficult to cover every way Kerberos can be setup (ie: there are cross realm setups and multi-trust environments). This section will describe how to setup the sqoop principals with a local deployment of MIT kerberos.

- All components which are Kerberos authenticated need one KDC server. If current Hadoop cluster uses Kerberos authentication, there should be a KDC server.
- If there is no KDC server, follow http://web.mit.edu/kerberos/krb5-devel/doc/admin/install_kdc.html to set up one.

Configure Hadoop cluster to use Kerberos authentication.

- Authentication type should be cluster level. All components must have the same authentication type: use Kerberos or not. In other words, Sqoop with Kerberos authentication could not communicate with other Hadoop components, such as HDFS, Yarn, Hive, etc., without Kerberos authentication, and vice versa.
- How to set up a Hadoop cluster with Kerberos authentication is out of the scope of this document. Follow the related links like https://hadoop.apache.org/docs/r2.5.0/hadoop-project-dist/hadoop-common/SecureMode.html

Create keytab and principal for Sqoop 2 via kadmin in command line.

```
addprinc -randkey HTTP/<FQDN>@<REALM>
addprinc -randkey sqoop/<FQDN>@<REALM>
xst -k /home/kerberos/sqoop.keytab HTTP/<FQDN>@<REALM>
xst -k /home/kerberos/sqoop.keytab sqoop/<FQDN>@<REALM>
```

- The <FQDN> should be replaced by the FQDN of the server, which could be found via “hostname -f” in command line.
- The <REALM> should be replaced by the realm name in krb5.conf file generated when installing the KDC server in the former step.
- The principal HTTP/<FQDN>@<REALM> is used in communication between Sqoop client and Sqoop server. Since Sqoop server is an http server, so the HTTP principal is a must during SPNEGO process, and it is case sensitive.
- Http request could be sent from other client like browser, wget or curl with SPNEGO support.
- The principal sqoop/<FQDN>@<REALM> is used in communication between Sqoop server and Hdfs/Yarn as the credential of Sqoop server.

Configuration

Modify Sqoop configuration file, normally in <Sqoop Folder>/server/config/sqoop.properties.

```
org.apache.sqoop.authentication.type=KERBEROS
org.apache.sqoop.authentication.handler=org.apache.sqoop.security.KerberosAuthenticationHandler
org.apache.sqoop.authentication.kerberos.principal=sqoop/_HOST@<REALM>
org.apache.sqoop.authentication.kerberos.keytab=/home/kerberos/sqoop.keytab
org.apache.sqoop.authentication.kerberos.http.principal=HTTP/_HOST@<REALM>
org.apache.sqoop.authentication.kerberos.http.keytab=/home/kerberos/sqoop.keytab
org.apache.sqoop.authentication.kerberos.proxyuser=true
```

- When _HOST is used as FQDN in principal, it will be replaced by the real FQDN. https://issues.apache.org/jira/browse/HADOOP-6632
- If parameter proxyuser is set true, Sqoop server will use proxy user mode (sqoop delegate real client user) to run Yarn job. If false, Sqoop server will use sqoop user to run Yarn job.

Run command

Set SQOOP2_HOST to FQDN.

```
export SQOOP2_HOST=$(hostname -f)
```

- The <FQDN> should be replaced by the FQDN of the server, which could be found via “hostname -f” in command line.

Start Sqoop server using sqoop user.

```
sudo –u sqoop <Sqoop Folder>/bin/sqoop.sh server start
```

Run kinit to generate ticket cache.

```
kinit HTTP/<FQDN>@<REALM> -kt /home/kerberos/sqoop.keytab
```

Start Sqoop client.

```
<Sqoop Folder>/bin/sqoop.sh client
```

Verify

If the Sqoop server has started successfully with Kerberos authentication, the following line will be in <@LOGDIR>/sqoop.log:

```
2014-12-04 15:02:58,038 INFO security.KerberosAuthenticationHandler [org.apache.sqoop.security.
KerberosAuthenticationHandler.secureLogin(KerberosAuthenticationHandler.java:84)] Using Kerberos
authentication, principal [sqoop/_HOST@HADOOP.COM] keytab [/home/kerberos/sqoop.keytab]
```

If the Sqoop client was able to communicate with the Sqoop server, the following will be in <Sqoop Folder>/server/log/catalina.out:

```
Refreshing Kerberos configuration
Acquire TGT from Cache
Principal is HTTP/<FQDN>@HADOOP.COM
null credentials from Ticket Cache
principal is HTTP/<FQDN>@HADOOP.COM
Will use keytab
Commit Succeeded
```

Customized Authentication

Users can create their own authentication modules. By performing the following steps:

- Create customized authentication handler extends abstract class AuthenticationHandler.
- Implement abstract function doInitialize and secureLogin in AuthenticationHandler.

```
public class MyAuthenticationHandler extends AuthenticationHandler {
        private static final Logger LOG = Logger.getLogger(MyAuthenticationHandler.class);
        public void doInitialize() {
        securityEnabled = true;
        }
        public void secureLogin() {
        LOG.info("Using customized authentication.");
        }
}
```

- Modify configuration org.apache.sqoop.authentication.handler in <Sqoop Folder>/server/config/sqoop.properties and set it to the customized authentication handler class name.
- Restart the Sqoop server.

Design Details

https://issues.apache.org/jira/secure/attachment/12671414/SQOOP-1525.pdf