KIP-235: Add DNS alias support for secured connection

Status

Current state : accepted

Discussion thread : https://lists.apache.org/thread.html/c29f6744b3e87b8e691de8cf5feb31f33dd8c2a5e07b77f957255a4e@%3Cdev.kafka.apache.org%3E

Voting thread : https://lists.apache.org/thread.html/931b9be025c537b6837fe19918f16bb8446c70e70c477b7454066780@%3Cdev.kafka.apache.org%3E

JIRA : ⚠ **Unable to render Jira issues macro, execution error.**

Released: 2.1.0

Motivation

When specifying a dns alias in bootstrap.server, the Java client API doesn't resolve all the CNAMES behind it.

This breaks kerberos based SASL authentication and therefore clients are unable to connect to a secured cluster.

Using an alias in bootstrap.servers along with SASL auth results in the following error :

javax.security.sasl.SaslException: An error: (java.security.PrivilegedActionException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Fail to create credential. (63) - No service creds)]) occurred when evaluating SASL token received from the Kafka Broker. Kafka Client will go to AUTH_FAILED state. [Caused by javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Fail to create credential. (63) - No service creds)]]

This is due to the following :

- When using SASL/Kerberos authentication, the kafka server principal is of the form kafka@kafka/broker1.hostname.com@EXAMPLE.COM
- Kerberos requires that the hosts can be resolved by their FQDNs.
- During SASL handshake, the client creates a SASL token and then sends it to kafka for auth.
  But to create a SASL token the client first needs to be able to validate that the broker's kerberos is a valid one.

The kafka server principal doesn't match the hostname referenced by the client (as the SaslAuthenticator will compare the alias' FQDN with the kafka broker hostname).
This fails the client broker kerberos validation and results in SASL authentication failure.

Public Interfaces

org.apache.kafka.clients

Proposed Changes

Client code change :

Change parseAndValidateAddresses() in ClientUtils to allow full dns resolution which will result in adding all underlying hosts as kafka nodes. This will allow using an alias in bootstrap.servers
Forcing this behaviour down on existing users isn't desirable since it could break SSL authentication. This should therefore be an optional feature.

Code snippets in the JIRA.

Client configuration

Proposed parameter : client.dns.lookup

Implemented with a ClientDnsLookup enum including values :

RESOLVE_CANONICAL_BOOTSTRAP_SERVERS_ONLY("resolve_canonical_bootstrap_servers_only")
DEFAULT("default")

This enum can be further extended to support new behaviours (potentially KIP-302 ?)

The default value for this parameter is false, there will be no backwards compatibility issue.
Setting the parameter to true will have the client perform the reverse lookup regardless of which security.protocol is specified.

This parameter shouldn't be set to true when using SSL authentication as it can break SSL hostname verification depending on the name contained in the certificate.

Security considerations

This doesn't change the underlying SASL authentication mechanism.
If the principal sent by the broker doesn't match any hostname in bootstrap.servers, the authentication will fail.

Rejected alternatives

Other option considered :

Changing default behaviour.

Modifying the existing code path to perform reverse dns lookup will break SSL authentication if kafka users use brokers IP addresses in bootstrap.servers and in the SubjectAlternativeName field of the certificates.

In this case, parseAndValidateAddresses() will perform the lookup and replace the IP addresses with hostnames, which will be matched against the IPs in the certificates, so the SSL handshake will fail.
This mismatch won't be obvious to users, as both bootstrap.servers and the certificates are consistent.
Changing default behaviour would mean breaking a valid use case.