KIP-294 - Enable TLS hostname verification by default

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *Adopted*

Discussion thread: *here*

JIRA: *KAFKA-3665*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Apache Kafka is increasingly used in environments where security is critical. TLS can be used a security protocol with Kafka to enable server authentication, client authentication and encryption. Even though Kafka supports server hostname verification and the documentation talks about setting hostnames in server certificates, hostname verification is disabled by default. This in an insecure default value since hostname verification is required to prevent man-in-the-middle attacks.

JSSE docs says:

> *When using raw `SSLSocket` and `SSLEngine` classes, you should always check the peer's credentials before sending any data. The `SSLSocket` a nd `SSLEngine` classes do not automatically verify that the host name in a URL matches the host name in the peer's credentials. An application could be exploited with URL spoofing if the host name is not verified.*

RFC-2818 talks about endpoint identification required to establish server identity for HTTP over TLS:

> *In general, HTTP/TLS requests are generated by dereferencing a URI. As a consequence, the hostname for the server is known to the client. If the hostname is available, the client MUST check it against the server's identity as presented in the server's Certificate message, in order to prevent man-in-the-middle attacks.*
>
> *If the client has external information as to the expected identity of the server, the hostname check MAY be omitted. (For instance, a client may be connecting to a machine whose address and hostname are dynamic but the client knows the certificate that the server will present.) In such cases, it is important to narrow the scope of acceptable certificates as much as possible in order to prevent man in the middle attacks. In special cases, it may be appropriate for the client to simply ignore the server's identity, but it must be understood that this leaves the connection open to active attack.*

http://www.cs.utexas.edu/~shmat/shmat_ccs12.pdf gives a good explanation of the risk of using TLS without hostname verification.

This KIP proposes to enable hostname verification by default for Kafka client connections to prevent man-in-the-middle attacks.

Public Interfaces

The default value of `ssl.endpoint.identification.algorithm` will be changed to `HTTPS`. Hostname verification can be disabled if required by setting the config to an empty string. The default will apply to both client configs (producer/consumer/admin client etc.) and inter-broker connections. The config can be set at listener-level.

Disabling endpoint identification in Java clients

Create produce/consumer/admin client etc. using a properties or map that sets endpoint identification to an empty String. For example:

```
Properties props = new Properties();
props.put(SslConfigs.SSL_ENDPOINT_IDENTIFICATION_ALGORITHM_CONFIG, "");
```

Disabling endpoint identification in server.properties

Set `ssl.endpoint.identification.algorithm` in server.properties to an empty value. Listener prefix may be added if required. For example:

- `ssl.endpoint.identification.algorithm=`
- `listener.name.internal.ssl.endpoint.identification.algorithm=`

Disabling endpoint identification using dynamic config update

Endpoint validation may be disabled when creating listeners using dynamic config update by specifying an empty value to `kafka-configs.sh`. For example:

- *`bin/kafka-configs.sh --bootstrap-server localhost:9093 --entity-type brokers --entity-name 0 --alter --add-config "listener.name.internal.ssl.endpoint.identification.algorithm="`*

Proposed Changes

This will be a simple change of the default value of `ssl.endpoint.identification.algorithm` to `HTTPS`. Any tests that rely on this config to be disabled will be updated to explicitly set the config to an empty string. Any test which doesn't care about the value of this config, but generates certificates without hostnames will be updated to set hostnames in certificates. All system tests are currently run wth `ssl.endpoint.identification.algorithm=HTTPS` anyway, so no changes are required.

Compatibility, Deprecation, and Migration Plan

Even though hostname verification should be enabled to protect against man-in-the-middle attacks, there may be environments where certificates are generated without hostnames and endpoint identification is performed in other ways. It is possible that some of these installations are currently using the default config. They will need to explicitly set `ssl.endpoint.identification.algorithm` to an empty string when upgrading. This will be documented in upgrade notes.

Rejected Alternatives

Disable hostname verification for SASL_SSL with Kerberos by default

Kerberos performs hostname verification, so we could leave default value as-is, i.e. disabled for client connections with SASL_SSL using GSSAPI as the mechanism. But it is safer to use the same default value for TLS across all security protocols and SASL mechanisms. Hence this KIP proposes to simply use a secure default, leaving it up to the user to disable if required.