KIP-76 Enable getting password from executable rather than passing as plaintext in config files

Status

Current state: DISCUSS

Discussion thread:

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

POC: WIP PR

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

Motivation

Currently there are a couple of options to pass passwords, like, SSL passwords, to Kafka, i.e., via properties file or via command line argument. Both of these are not recommended security practices.

- A password on a command line is a no-no: it's trivial to see that password just by using the 'ps' utility.
- Putting a password into a file, and then passing the location to that file, is the next best option. The access to the file will be governed by unix access permissions which we all know and love. The downside is that the password is still just sitting there in a file, and those who have access can still see it trivially.
- The most general, secure solution is to provide a layer of abstraction: provide functionality to get the password from "somewhere else". The most flexible and generic way to do this is to simply call an executable which returns the desired password.
    - The executable is again protected with normal file system privileges
    - The simplest form, a script that looks like "echo 'my-password'", devolves back to putting the password in a file
    - A more interesting implementation could open up a local encrypted password store and extract the password from it
    - A maximally secure implementation could contact an external secret manager with centralized control and audit functionality.
    - In short: getting the password as the output of a script/executable is maximally generic and enables both simple and complex use cases.

Goals

- Allow password retrieval from executables.
- Have the behavior configurable and turned off by default.

Proposed Changes

Introduce a new config, `executable.password.enable`, for controlling password retrieval. By default, values of password configs, like, `ssl.key.password`, `ssl.keystore.password`, etc., will be taken as plaintext passwords. However, when `executable.password.enable` is set, values of password configs will be executed to retrieve actual passwords.

When `executable.password.enable` is set, password values can have executable path followed by arguments, delimited by ' '. For instance, "`echo test-password`", will take `test-password` as a password.

Public Interfaces

The proposal includes a new configuration, `executable.password.enable`, which will be set to `False` by default.

When `executable.password.enable` is set, values of password configs will be executed to retrieve actual passwords for those configs. If the executable specified dies, or is not an executable, or does not exist, kafka server or kafka client will exit with a `ConfigException`.

Compatibility, Deprecation, and Migration Plan

The proposed change is backwards compatible.

Test Plan

Following system test will be added.

- Start a broker with SSL turned on and `executable.password.enable` set.
- Run producer and consumer with `executable.password.enable set to false.`
- Run producer and consumer with `executable.password.enable set to true.`

Q&A

1. Are there other projects that follow the proposed way of getting passwords?

Hadoop implemented something called the CredentialProvider specifically for the purpose of encrypting passwords. See HADOOP-10607. This functionality is now supported by other projects, including Hive, HBase, etc.

Rejected Alternatives

Password generator config per password related config

The approach creates a new generator config for each of the password related configs, like, `ssl.key.password.generator` for `ssl.key.password` config. When a password config is not set, check for password.generator config for that config and execute it's value to get the password.

This allows to have a subset of password configs to have executables as password, while others have plaintext as password. However, this approach is rejected for following reasons.

1. The only use case someone would want to use executables as password is to be compliant of some security standard and for more security, leaving some of the passwords as plaintext, takes away both.
2. There will a lot of new configs, and every time there is a new password config added, a generator config will also have to be added.