Certificate Store

Metadata	Value
Date	2023-06-22
Author	@tbeets
Status	Implemented
Tags	server, security

Problem Statement

Some users need to source the NATS Server's TLS identity from a *credential store* rather than a file. This may be for either client (mTLS) or server identity use cases. Need is driven either by an insitu credential store and/or organizational policy that disallows deploying secrets as an operating system file.

Edge computing scenarios involving large "fleets" of managed servers may especially require credential store integration.

Context

A credential store may be offered as a supported operating system service such as the Microsoft Software Key Storage Provider, or by a 3rd-party trusted platform module (TPM) provider. Some credential store providers may implement a standards-based interface such as PKCS #11 Cryptographic Token Interface.

NATS Server requires a configuration options interface for operators to specify a specific credential store provider, in a TLS configuration block (or TLS Map), with provider-specific identity parameters.

Design

The following configuration properties are added to the TLS map:

Properties in TLS Map

Property	Description	Default	Example Value
cert_store	a supported credential store provider (see Enabled Providers)		"WindowsCurrentUser"
cert_match_by	provider-specific identity search/lookup option	"Subject"	"Subject"
cert_match	identity search/lookup term		"example.com"

If the `cert_store` configuration properties are used in a given TLS map, it logically takes the place of `cert_file` and `key_file` properties.

If the operator specifies both `cert_store` and `cert_file` properties in the same TLS map, the server will error on startup with message `'cert_file' and 'cert_store' may not both be configured`.

For a given TLS map, if `cert_store` is configured, the `key_file` property, if present, will be ignored.

> Note: provider name is case-insensitive

If a `cert_store` provider unknown to the NATS Server (and the specific operating system build) is configured, the server will error at startup with message `cert store type not implemented`.

Enabled Cert Store Providers

Provider Name	Description	Operating System
`WindowsCurrentUser`	Microsoft Software Key Storage Provider, local "MY" (Personal) store of Current User	Windows
`WindowsLocalMachine`	Microsoft Software Key Storage Provider, local "MY" (Personal) store of Local Machine	Windows

Windows Providers

The Microsoft Software Key Storage Provider (KSP) is accessed by NATS Server at startup and on-demand when TLS-negotiation signatures are required.

The two providers differ only by the operating system access and permissions scope required and the specific "store" of credentials (certificates and related private key) that are sourced from the provider:

- `WindowsCurrentUser` - The "MY" store associated with the *operating system user* of the NATS Server process.
- `WindowsLocalMachine` - The "MY" store associated with the *local machine*. The NATS Server process user must have the necessary Windows entitlement to access the local machine's certificate store.

> Note: The "MY" store on Windows is what appears as Personal->Certificates in the Microsoft Management Console (Certificates snap-in).

The Windows-build of NATS Server has been enhanced to directly leverage the Windows Security & Identity library functions. APIs from libraries `ncrypt.dll` and `crypt32.dll` are invoked to find and retrieve public certificates at startup and perform signatures during TLS negotiation.

Inclusion of Intermediate CA Certificates

When a leaf certificate is matched (see below Example Configurations), NATS server will attempt to source a valid trust chain of certificates from the local Windows machine's trust store, i.e. a valid chain from the leaf to a trusted self-signed certificate in the store (typically a CA root).

If at least one valid chain is found, the first valid chain is selected and NATS server will form a final certificate as the matched leaf certificate plus non-self signed intermediate certificates that may be present in the valid chain.

If no valid trust chain is found in the local Windows machine's trust store, the NATS server will form the final certificate as the matched leaf certificate only, no intermediate chained certs will be included.

Validation Policy

Note that CRL, OCSP, explicit role validation (TLS server or TLS client) and other policy features are specifically avoided in certificate match (and intermediate population) against the Windows KSP, as these are ultimately provided by the eventual trust validator in TLS negotiation, i.e.this provider implements identity lookup and identity signature but is not itself the trust/policy validator of its own identity claims.

Identity Lookup Options

`cert_match_by` may be one of the following:

- `Subject` - the KSP will compare the `cert_match` property value to each of the certificate's Subject RDN values and return the first match. See also: CERT_FIND_SUBJECT_STR
- `Issuer` - the KSP will compare the `cert_match` property value to each of the certificate's Issuer RDN values and return the first match. See also: CERT_FIND_ISSUER_STR

If the configured `cert_match_by` does not match an available provider option, the server will error with message `cert match by type not implemented`.

Example Configurations

Given a certificate provisioned to MY store with the following Issuer and Subject distinguished names:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            7c:b1:37:8c:1a:70:1a:99:4e:50:37:29:6f:12:2c:bd:12:27:0c:64
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = Synadia Communications Inc., OU = NATS.io, CN = localhost
        Validity
            Not Before: Feb  4 19:51:00 2019 GMT
            Not After : Feb  3 19:51:00 2024 GMT
        Subject: OU = NATS.io, CN = example.com
        Subject Public Key Info:
...
```

and TLS Map in a server configuration in format:

```
tls {
    cert_store: <WindowsCurrentUser|WindowsLocalMachine>
    cert_match_by: <Subject|Issuer>
    cert_match: <Lookup Value>
    ...
}
```

cert_match_by	cert_match	Result

cert_match_by	cert_match	Result
"Subject"	"example.com"	Success (found)
"Subject"	"NATS.io"	Success (found)
"Subject"	"OU = NATS.io, CN = example.com"	Fail (not found)
"Subject"	"CN = example.com"	Fail (not found)
"Issuer"	"localhost"	Success (found)
"Issuer"	"Synadia Communications Inc."	Success (found)
"Issuer"	"O = Synadia Communications Inc., OU = NATS.io, CN = localhost"	Fail (not found)
"Issuer"	"CN = localhost"	Fail (not found)

> Note: To avoid TLS negotiation failure caused by return of the wrong certificate, it's recommended to lookup by the Subject value representing the unique name of your NATS Server's certificate identity, e.g. the CN value as in the "example.com" case above.

Futures

The certificate store interface and new TLS Map configuration entries are intended to be extensible to future provider interfaces that NATS Server may implement.