OCSP Peer Verification

Metadata	Value
Date	2023-06-20
Author	@tbeets
Status	Implemented
Tags	server, security

Release History

Revision	Date	Description
1	2023-06-20	Initial release

Context and Problem Statement

Many users of NATS are highly invested in X.509 certificates to identify applications, certificate authority tooling and policies, and ultimately TLS handshake to authenticate applications in their environment (solely or in combination with NATS user credentials). OCSP Peer adds the option for NATS Server to OCSP verify an *external peer* against the peer's own certificate authority (or authorities) at the time of TLS negotiation and before ultimately accepting or rejecting the TLS connection. External peers are NATS client applications establishing mutual TLS (mTLS) connections with NATS Server (MQTT, WebSocket, and NATS protocols) and NATS Leaf connections (over mTLS and TLS) between two NATS Servers.

OCSP Peer allows an operator to allow or revoke NATS connectivity at either a fine-grain (leaf certificate) or coarse-grain level (intermediate CA certificate) using their CA tools and CA OCSP responder capabilities.

Adding dependency on peer-specified CA OCSP responder services for client connection necessarily adds a single point of failure (SPOF) from the NATS Server point of view and will in any case slow overall connection time. To mitigate, OCSP Peer is paired with a local OCSP response cache whose main purpose is to minimize expensive network calls to external services, but also to provide some connection resilience (in the happy-path) when OCSP responder services are offline or not reachable.

This feature is intended to comply with the following standards:

Standard	Description
RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP	OCSP Responder specification (Sections 2.1, 2.2)
RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile	Authority Information Access (AIA) extension (Section 4.2.2.1)

Prior Work

The OCSP Stapling (Server 2.3+) feature enables NATS Server to pre-fetch and "staple" its own CA verification (OCSP response) to be used in identity exchange with an inbound TLS client during handshake if so-requested by the TLS client.

A NATS Server so-configured also validates the staple provided by an *internal peer* NATS Server of the same cluster (ROUTE connections) or of the same supercluster (GATEWAY connections) in handshake negotiations that it initiates as a TLS client.

> Note: OCSP Peer applies to CLIENT and LEAF connections only and does not overlap or supersede the OCSP Stapling feature.

Design

The OCSP Peer feature has four main elements:

OCSP:

- **Verification check** during TLS handshake after trust-store verification
- **Eligibility check** based on CA's AIA assertion in trust-chain certificates
- **Callout to CA** responder service for eligible certificates
- **Response cache** to minimize callouts (in an expiry period set by the CA)

Configuration

Configuring OCSP peer verification

In the NATS Server configuration file, the `ocsp_peer` configuration option may be added to the respective `tls` configuration map of the following client and leaf connection types:

Client connection type	TLS map of (configuration)	During TLS handshake, OCSP verify of	TLS verify (mTLS) required
Inbound NATS	*Root*	TLS client	Yes
Inbound MQTT	`mqtt`	TLS client	Yes
Inbound WebSocket	`websocket`	TLS client	Yes
Inbound Leaf (hub)	`leafnodes`	TLS client	Yes
Outbound Leaf (spoke)	*Leafnode* `remote`	TLS server	No

> OCSP verification check will be made during TLS handshake **after** trust-chain verification is successful, i.e. peer's leaf certificate chains to server's trusted CA certificate(s) specified in `ca_file` or the operating system's default trust store (when unset).

Defaults, short, and long form

The `ocsp_peer` configuration option may be specified in short or long forms.

Short form

The short form is a boolean value:

`ocsp_peer`	OCSP peer verification for the TLS map
`true`	Is enabled; equivalent to long form with `verify: true` and otherwise defaults

`ocsp_peer`	**OCSP peer verification for the TLS map**
`false` (default, unset)	Is not enabled

Here is an example NATS Server configuration snippet for Inbound NATS connections:

```
    port: 4222
    tls {
        cert_file: "configs/certs/ocsp_peer/mini-ca/server1/TestServer1_bundle.pem"
        key_file: "configs/certs/ocsp_peer/mini-
ca/server1/private/TestServer1_keypair.pem"
        ca_file: "configs/certs/ocsp_peer/mini-ca/root/root_cert.pem"
        timeout: 5
        verify: true
        ocsp_peer: true
    }
```

Long form

The long form is a map of customization options:

```
    port: 4222
    tls: {
        cert_file: "configs/certs/ocsp_peer/mini-ca/server1/TestServer1_bundle.pem"
        key_file: "configs/certs/ocsp_peer/mini-
ca/server1/private/TestServer1_keypair.pem"
        ca_file: "configs/certs/ocsp_peer/mini-ca/root/root_cert.pem"
        timeout: 5
        verify: true
        ocsp_peer: {
           verify: true
           ca_timeout: 2
           allowed_clockskew: 30
           warn_only: false
           unknown_is_good: false
           allow_when_ca_unreachable: false
           cache_ttl_when_next_update_unset: 3600
        }
    }
```

Customization options

Option	Description	Type	Default
`verify`	Enable OCSP peer validation	bool	`false`
`ca_timeout`	OCSP responder timeout in seconds (may be fractional)	float64	`2`
`allowed_clockskew`	Allowed skew between server and OCSP responder time in seconds (may be fractional)	float64	`30`

Option	Description	Type	Default
`warn_only`	Warn-only and never reject connections	bool	`false`
`unknown_is_good`	Treat response *Unknown* status as valid certificate	bool	`false`
`allow_when_ca_unreachable`	Warn-only if no CA response can be obtained and no cached revocation exists	bool	`false`
`cache_ttl_when_next_update_unset`	If response *NextUpdate* unset by CA, set a default cache TTL in seconds (may be fractional) from *ThisUpdate*	float64	`3600`

Configuring OCSP response cache

In the NATS Server configuration file, the `ocsp_cache` configuration option may be used to explicitly enable a server-scoped OCSP response cache. Such cache will be used for all TLS listeners enabled for OCSP Peer Verification (as above).

> Note: If `ocsp_cache` is configured, but no TLS listeners are enabled for OCSP Peer Verification, the NATS Server will not initialize a cache. If `ocsp_cache` is absent, but one or more TLS listeners are enabled for OCSP Peer Verification, the NATS Server *will* initialize a local cache with default settings. This is equivalent to `ocsp_cache: true`.

Defaults, short, and long form

The `ocsp_cache` configuration option may be specified in short or long forms.

Short form

The short form is a boolean value:

`ocsp_cache`	OCSP cache behavior
`true` (default, unset)	Is enabled; equivalent to long form with `type: local` and otherwise defaults
`false`	Is disabled; equivalent to long form with `type: none`

Here is an example NATS Server configuration snippet with short form configuration:

```
    port: 4222
    ocsp_cache: true
    tls {
        cert_file: "configs/certs/ocsp_peer/mini-ca/server1/TestServer1_bundle.pem"
        key_file: "configs/certs/ocsp_peer/mini-
ca/server1/private/TestServer1_keypair.pem"
        ca_file: "configs/certs/ocsp_peer/mini-ca/root/root_cert.pem"
        timeout: 5
        verify: true
        ocsp_peer: true
    }
```

Long form

The long form is a map of cache customization options:

```
    port: 4222
    ocsp_cache: {
        type: local
        local_store: "_rc_"
        preserve_revoked: false
        save_interval: 300
    }
    tls: {
        cert_file: "configs/certs/ocsp_peer/mini-ca/server1/TestServer1_bundle.pem"
        key_file: "configs/certs/ocsp_peer/mini-
ca/server1/private/TestServer1_keypair.pem"
        ca_file: "configs/certs/ocsp_peer/mini-ca/root/root_cert.pem"
        timeout: 5
        verify: true
        ocsp_peer: true
    }
```

Customization options

Option	Description	Type	Default
`type`	Sets the cache implementation: `local` or `none`	string	`local`
`local_store`	Sets the directory where the local cache will persist `cache.json`. Relative paths will be relative to current working directory of the NATS Server executable.	string	`_rc_`
`preserve_revoked`	When set to `true` the local cache implementation will ignore commands to delete cached responses of status *Revoke*. See also OCSP Peer setting `allow_when_ca_unreachable`.	bool	`false`
`save_interval`	Set how often the in-memory `local` cache is persisted to disk (in seconds). The default value is 5 minute interval saves (every 300 seconds). A minimum value of 1 second is enforced.	float64	`300`

Peer OCSP verification

Trust-chain pre-requisite

Peer OCSP verification occurs during TLS handshake cycle, only AFTER successful trust-chain verification. Peer connections are immediately rejected if trust-chain verification fails.

Peer rejection

If a peer connection is rejected due to failed OCSP verification, the peer will receive a summary TLS handshake error from the NATS Server as:

Handshake reject	Connection type
`client not OCSP valid`	NATS, WebSocket, and MQTT client connections. Inbound Leaf (hub) connections.

Handshake reject	Connection type
`server not OCSP valid`	Outbound Leaf (spoke) connections.

The connection is then terminated.

Log entries

Certificate's that fail OCSP verification - which could be a peer leaf certificate or an Intermediate CA certificate - will be logged at **warning** level.

A rejected peer connection will be logged at **error** level (the same whether OCSP verification is enabled or not).

```
[6980] 2023/06/20 12:50:07.444055 [WRN] OCSP verify fail for
[CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US] with CA status [revoked]
[6980] 2023/06/20 12:50:07.444125 [ERR] 127.0.0.1:57312 - cid:7 - TLS handshake error:
client not OCSP valid
```

Advisory system events

The NATS Server will also emit Advisory system events corresponding to the log entries above:

Event type	Event subject	Event frequency
`io.nats.server.advisory.v1.ocsp_peer_reject`	`$SYS.SERVER.<server>.OCSP.PEER.CONN.REJECT`	1 per rejected connection
`io.nats.server.advisory.v1.ocsp_peer_link_invalid`	`$SYS.SERVER.<server>.OCSP.PEER.LINK.INVALID`	1 per link evaluated and invalid

See below in this document for event payload examples.

Peer rejected event

If a peer connection is rejected due to failed OCSP verification, the NATS Server will emit an advisory system event. This event carries information about the peer's leaf certificate to aid operators in diagnosing a configuration issue or attempted exploit that is preventing successful connections.

> Note: This advisory event does not imply that the peer's leaf certificate directly failed OCSP verification. The leaf certificate (e.g. Subject field) is used as top-level peer identification as rejection takes place *before* NATS Authorization and binding to a NATS User/Account.

Peer link invalid event

Whenever a certificate's OCSP response is obtained and the CA has asserted not "Good", the NATS Server will emit an advisory system event. The event carries information about the certificate's (Subject) identity as well as the certificate identity of the corresponding peer's leaf certificate. This event aids operators in understanding the root cause of a

peer's connection rejection, i.e. the specific certificate that is OCSP valid which could be the leaf certificate of the peer or an Intermediate CA certificate.

> Note: In the typical case, there will be one peer link invalid event per peer rejection, i.e. the peer's single trust-chain OCSP invalidated immediately upon finding a single invalid link; however, *if the peer forms multiple trust-chains*, there may be multiple peer link invalid events at time of connection, and the peer may ultimately be allowed or rejected.

OCSP Peer verification criteria

OCSP verified

Peer with:

1. A self-signed certificate
2. At least one chain with *zero* OCSP-eligible links
3. At least one chain with *one or more* OCSP-eligible links having a "Good" OCSP response for all eligible links

OCSP NOT verified

Peer with: 4. None of the above ([1],[2],[3]) true

Criteria modifiers

Non-default configuration settings modify above criterion as follows:

- If `unknown_is_good` is `true` then a CA response of *Unknown* status is considered the same as *Good* status as it applies to [3].
- If `allow_when_ca_unreachable` is `true` then a non-response is considered *Good* status as it applies to [3].

> Note: When `allow_when_ca_unreachable` is `true`, if a *Revoked* CA response entry is found in cache, even if "expired" (in respect to NextUpdate), the corresponding chain is NOT verified in respect to [3].

Peer OCSP eligibility

After trust is determined, there is *at least one* verified trust chain that connects the leaf certificate to the NATS Server's trust-anchor. Each chain is evaluated for links (certificates) that are OCSP eligible. A certificate is considered OCSP eligible if the certificate's issueing CA declares an OCSP responder web URI (http or https) in the certificate's **Authority Information Access (AIA) extension**. Non-web URI schemes are NOT supported and are ignored.

> Note: In practice, CA OCSP Responders usually reside at non-TLS web endpoints (http) as their OCSP Responses are intentionally public and digitally signed. Hosting CA OCSP Responders at TLS web endpoints (https) may create ambiguity in certificate verification. NATS Server will attempt to use https endpoints if encountered; the server host's default trust store will be used to verify the web server.

If the link is the trust-anchor, i.e. *explicitly* trusted by the NATS Server, then the link is not evaluated for OCSP eligiblity.

> Note: A trust "chain" may consist of just one link, the leaf certificate. This is self-signed trust (there is no CA). In this case, the leaf certificate is a trust-anchor and is not OCSP eligibile.

Certificate example

In the following OpenSSL-style "pretty print" of certificate extensions for a sample client certificate, the CA's declared Authority Information Access (AIA) web URI is shown:

```
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                AF:4B:3E:F2:BE:A1:F2:E5:7E:0B:31:CC:BB:A5:5F:83:7F:42:B3:94
            X509v3 Authority Key Identifier:
                7B:14:FB:1B:B4:A0:09:30:C8:81:BC:E1:01:32:67:D0:68:A8:A3:D1
            X509v3 Basic Constraints: critical
                CA:FALSE
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, E-mail Protection
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:http://crl.tinghus.net/intermediate_crl.der
            Authority Information Access:
                OCSP - URI:http://ocsp.tinghus.net/
            X509v3 Subject Alternative Name:
                email:UserA1@user.net
```

OCSP responder callout

When evaluating an eligible trust-chain certificate for OCSP validity, the OCSP response cache will always be checked first. If no existing OCSP response entry is found in cache, or a found entry is not in an effective time window, then the NATS Server will make a synchronous call to the CA OCSP responder's web endpoint.

> Note: The NATS Server must have network access to the CA OCSP responder's web endpoint as well as DNS access to resolve a URI expressed as a hostname and domain.

NATS Server will wait for (default) 2 seconds for an HTTP response from the OCSP responder. The timeout is configurable as the `ca_timeout` option. If no response, or a non-HTTP 200 response is received, the NATS Server will log an error and consider the certificate not OCSP valid for purposes of peer evaluation. As the CA's actual intent is ambiguous, no advisory system event will be emitted. If a successful HTTP response is received, the response payload will be parsed as an OCSP Response. If the response fails to parse than an error will be logged and the certificate will be considered not OCSP valid for evaluation purposes; no advisory system event will be emitted.

If the response parses, the CA's OCSP Response will be evaluated to determine:

- Valid digital signature of the OCSP Response, either the issuing CA or a signing delegate entitled by the issuing CA
- Valid effectivity time window, i.e. "now" after **ThisUpdate** and before **NextUpdate**
- Certificate's status in set **Good**, **Revoked**, or **Unknown**

Successfully obtained and valid OCSP Responses will be cached for future use.

OCSP response cache

There are two implementation types of OCSP response cache:

Cache Type	Description
`none`	A "no-op" cache implementation. No OCSP responses are cached.
`local`	A server-scoped in-memory cache with periodic snapshot to disk.

The default cache type is `local`. The `none` cache type exists for testing purposes or an operating environment where there is a mandated OCSP check of peer certificates at every connection.

Local cache

The `local` cache type is a server-scoped in-memory cache with periodic snapshot to disk. The persistent snapshot is a JSON document in a file named `cache.json`. The `local_store` cache configuration option is used to tell NATS Server where to find `cache.json` on startup/reload (if it exists) and where to write the latest snapshot periodically (every 5 minutes by default) and at server shutdown. The default `local_store` value if unset is relative directory path `_rc_`. Snapshot frequency may be configured with the `save_interval` option (value in seconds).

> Note: Setting a fully qualified directory path for `local_store` is recommended

Eviction of expired OCSP responses from cache is "passive" in the sense that cache entries are only evicted when the respective certificate is evaluated again as constituent of a peer connection attempt. If the cached entry is found to be expired at that time, it is evicted. Note that the cache option `preserve_revoked` can be enabled such that cached responses that represent certificate revocations are never evicted (although they can be replaced by a newer response).

Format

The persisted format is essentially a map of certificates (keyed by certificate hash) to obtained CA OCSP responses `resp`. Responses are stored as base64 encoding of the raw bytes returned by the CA OCSP responder.

Additional fields `subject`, `resp_status`, and `resp_expires` are extracted and stored in human-readable format for operator convenience and debugging purposes, but are "non-normative" for runtime OCSP verification.

> Note: Whether a CA OCSP Response is obtained from cache or directly from web call, identical response parsing and validation is performed at runtime.

Example `cache.json` file with three cached OCSP responses:


```
{
 "0aJpXCPoRO6ZTxfmOlhuXlEM25YBWGUjiZzQFu9Y0/Q=": {
  "subject": "CN=UserA1,O=Tinghus,L=Tacoma,ST=WA,C=US",
  "cached_at": "2023-06-05T23:13:15Z",
  "resp_status": "good",
  "resp_expires": "2023-06-05T23:14:15Z",
  "resp":
"/wYAAFMyc1R3TwBOBQBOTsRv0gzUMIIGTgoBAKCCBkcwggZDBgkrBgEFBQcwAQEEggY0MIIGMDCB5KFYMFYxCzA
JBgNVBAYTAlVTEQ0gCAwCV0ExDzANAQ0wBwwGVGFjb21hMRAwDgERNAoMB1RpbmdodXMxFzAVARLwdgMMDk9DU1A
gUmVzcG9uZGVyGA8yMDIzMDYwNTIzMTMxNVowdzB1ME0wCQYFKw4DAhoFAAQUYj2aszQjKY7QXbC4y+QQljxp/20
EFHsU+xu0oAkwyIG84QEyZ9BoqKPRAhRpl4uSS8bBa7SX0U8Biv0HaHpKgIAAQmYABKARMhMAADQBefQ9AQ0GCSq
GSIb3DQEBCwUAA4IBAQAgTe7D6y7jSpQf5o7U0ZK6cfQNMH3bYaVAHsVZKLfcS9jImaKOOuEmXaHQZeZntMRA8As
7sndd48leOV3u4EZ5fP2Uuwra/GT/K20uvNhrVkOVKypQvk98oWkx92HW2MO1qNRae/vBlVk5zrEY/snJjq94MF1
WXvX0C04HnEYF2GjLuDIOLhk0dDcuJ+x4G0fXMkGf/QRixGkT1suaJVpBoeVQPphjYNskjWfu33QqAx6WLdESvPJ
C6eVCriLqxEWHJPnnbHtdXEp0+rj9LU+o3zZxI+VeVxPMQ0pnbpFv+JczGtB2ZLjPjLRVrxGom2W49nIkX+nuTKE
Gsu2dzoCJoIIEMTCCBC0wggQpMIIDEaADAgECAhQXEaeCXcregyeqhdBXzFyrMhgHNjo/AQQwV1Ej1jACCBgwFkk
```

```
woA9JbnRlcm1lZGlhdGUgQ0EwHhcNMjMwNDE3MjE1MDU0WhcNMjQwNDE2DQ9RqRUASAwwggEiLiMCAAEFAPRAAQ8
AMIIBCgKCAQEA0NEVy8NMyVY71RBZvSw02fhp3MerRQaFM6pvXZgqYD5CzBfuEE1Mn+mYx3l1wRPcK+xAjQJvT3K
HdzOVYZtAIM0t231R+TdLI+VEsW6j70kWazJbfYqswPfYLoaYjpmfgffd2XlmCwm9wdQMUCtAgxwnu8rZef24CkB
L9TPOpqu5kNNRXWSTTsmcLsZ6EfQfyXujurX1/HHlv2ebU126QlMKoJ+CS0mPPDiS/Rpv7QEyLlaHuEfcsTOWKZn
S9vVYQqdXY8Qc3UKk38E/c5PNeOkaV5+5hIhEmE+ouQSnttpYSXIblZUFg1HG/A/Yq1yFjCuDlSYHNmnxMsPDhz5
zjwIDAQABo4HtMIHqMB0GA1UdDgQWBBTtTAyYBkUuCwF7OJHb2P0XDaVH+TAfBgNVHSMEGDAWgFLHAzQwDAYDVR0
TAQH/BAIwAIVMBB0PAQ4QBAMCB4BFGgQdJQEQEAwwCgYIibGAAwkwPAYDVR0fBDUwMzAxoC+gLYYraHR0cDovL2N
ybC50iYkULm5ldC9pXVMkX2NybC5kZXIwNBFKHAEBBCgwJjAkERAMMAGGGA1JDG9jc3AySgAuEgKRNfD/hM0u+Ha
L3XCwZPiY5b2sdUQcAKJVQCeUHGhYjn5DU1ROv5euXF33+/TwnBbYuFnT7x6r1qAfiZvOQkrViOJVFYYcMITLOUW
5RJac2GOhiSpfcFgHN36VuL3qxdGXVSmtCC5J/uqLvs10algRKtoAcmAHV1MbwndnjS8/mIesw0oueJgbYI+GNb2
O3+acdQuv6jZonK/7ZeHkGeMgumMOBTQ0RKtkmzDDp4xIAsDctTQCZf3MlJF8pQVfBOE92oZIA5b2rAg5YoGoy8K
4ZAT26NBuaUEVgaC0+zc9FIOlrzyqgNF43A/wl9nj0sAX0n3uGZBKVtRxR2sUeL/EUqW4HQ==
```



```
 },
 "6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9QBBbY=": {
  "subject": "CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US",
  "cached_at": "2023-06-05T23:14:54Z",
  "resp_status": "revoked",
  "resp_expires": "2023-06-05T23:15:53Z",
  "resp":
"/wYAAFMyc1R3TwBlBQBm6fX86gzUMIIGZgoBAKCCBl8wggZbBgkrBgEFBQcwAQEEggZMMIIGSDCB/KFYMFYxCzA
JBgNVBAYTAlVTEQ0gCAwCV0ExDzANAQ0wBwwGVGFjb21hMRAwDgERNAoMB1RpbmdodXMxFzAVARLweAMMDk9DU1A
gUmVzcG9uZGVyGA8yMDIzMDYwNTIzMTQ1M1owgY4wgYswTTAJBgUrDgMCGgUABBRiPZqzNCMpjtBdsLjL5BCWPGn
/bQQUexT7G7SgCTDIgbzhATJn0Gioo9ECFEW+adELDY2oBZMwjEsvsrzk65oloRYNaDg0MTgwNjE0MDdaoAMKAQE
NFhl+BKARMhMAADUBkfQ9AQ0GCSqGSIb3DQEBCwUAA4IBAQBFoDY3eZOOv4jmm812XNCdn/tWsPm1tSwxOFFyk2D
uSTiu64L8QTPktws2b7Ls9JvEomhgremeytV3XqxsuNo1VlKRDclTy9t63RY7axCcW2X2qB7SRsMll2XgSWpITGU
MmXLF4Tq8SRCcsEzDVDz9V3z25W/kE9eG2E4pmEjL0LU8FdkNW7Zm6F4xBy30LhZnjcY1Ic1KiKat9xjAm8fx18/
KwUn+fqm/pGWlkFzaIEuuzH1zVQmfW56gahLu/PFibgoDemjHVbdMJEDu8ODfXqSOkyJtD0cKEDVvapyjkltcX1A
4qRT1v58IcGNyWuD6Yk/NYcVcr687cT51tOGAoIIEMTCCBC0wggQpMIIDEaADAgECAhQXEaeCXcregyeqhdBXzFy
rMhgHNjo/AQQwV1E71kgCCBgwFklIoA9JbnRlcm1lZGlhdGUgQ0EwHhcNMjMwNDE3MjE1MDU0WhcNMjQwNDE2DQ9
RwRUASAwwggEiLiMCAAEFAPRAAQ8AMIIBCgKCAQEA0NEVy8NMyVY71RBZvSw02fhp3MerRQaFM6pvXZgqYD5CzBf
uEE1Mn+mYx3l1wRPcK+xAjQJvT3KHdzOVYZtAIM0t231R+TdLI+VEsW6j70kWazJbfYqswPfYLoaYjpmfgffd2Xl
mCwm9wdQMUCtAgxwnu8rZef24CkBL9TPOpqu5kNNRXWSTTsmcLsZ6EfQfyXujurX1/HHlv2ebU126QlMKoJ+CS0m
PPDiS/Rpv7QEyLlaHuEfcsTOWKZnS9vVYQqdXY8Qc3UKk38E/c5PNeOkaV5+5hIhEmE+ouQSnttpYSXIblZUFg1H
G/A/Yq1yFjCuDlSYHNmnxMsPDhz5zjwIDAQABo4HtMIHqMB0GA1UdDgQWBBTtTAyYBkUuCwF7OJHb2P0XDaVH+TA
fBgNVHSMEGDAWgFLdA1AwDAYDVR0TAQH/BAIwADAOBgNVHQ8BDhAEAwIHgEUaBB0lARAQDDAKBgiJyYADCTA8BgN
VHR8ENTAzMDGgL6AthitodHRwOi8vY3JsLnSJoRQubmV0L2ldUyRfY3JsLmRlcjA0EUocAQEEKDAmMCQREAwwAYY
YDUkMb2NzcDJKAC4SApE18P+EzS74dovdcLBk+Jjlvax1RBwAolVAJ5QcaFiOfkNTVE6/l65cXff79PCcFti4WdP
vHqvWoB+Jm85CStWI4lUVhhwwhMs5RblElpzYY6GJKl9wWAc3fpW4verF0ZdVKa0ILkn+6ou+zXRqWBEq2gByYAd
XUxvCd2eNLz+Yh6zDSi54mBtgj4Y1vY7f5px1C6/qNmicr/tl4eQZ4yC6Yw4FNDREq2SbMMOnjEgCwNy1NAJl/cy
UkXylBV8E4T3ahkgDlvasCDligajLwrhkBPbo0G5pQRWBoLT7Nz0Ug6WvPKqA0XjcD/CX2ePSwBfSfe4ZkEpW1HF
HaxR4v8RSpbgd"
 },
 "L5KmmDWaZ7JRPuQU+5+6qPS+QIZiHcbAUn5cYmLaZAI=": {
  "subject": "CN=Intermediate CA,O=Tinghus,L=Tacoma,ST=WA,C=US",
  "cached_at": "2023-06-05T23:13:15Z",
  "resp_status": "good",
  "resp_expires": "2023-06-05T23:14:15Z",
  "resp":
"/wYAAFMyc1R3TwBIBQC6qBY3ygzUMIIGRgoBAKCCBj8wggY7BgkrBgEFBQcwAQEEggYsMIIGKDCB56FbMFkxCzA
JBgNVBAYTAlVTEQ0gCAwCV0ExDzANAQ0wBwwGVGFjb21hMRAwDgERNAoMB1RpbmdodXMxGjAYARLweQMMEUNBIE9
DU1AgUmVzcG9uZGVyGA8yMDIzMDYwNTIzMTMxNVowdzB1ME0wCQYFKw4DAhoFAAQU1ulMFVfdg9oIN8Cm4P8Xbp9
KX8kEFM8miAT6OeIHPz6mep0OF7amfxBZAhR05CcEcq0fyUn2DUu67bUK8IKPa4AAQmYABKARMnkAADQBefQ9AQ0
GCSqGSIb3DQEBCwUAA4IBAQDh976LKdW5Ahy3lS1WzyW/J63/Abb2ZprBJVSF/B6zx89VwvYYXWkivMVGD42u1HE
zmrgW5kZEYPcUQnv1fOL5lIoOHHyGkitiz1fmRah68P/TUTGxa3le087yKMaZvPC3se/2UG5wfI1yejtJtUxDXeb
GJ2JeM5mhiHlZhbyv2Q/xN4OB0GOdUeEJxBjjcphPQWgc7JBhmGOnITum8KaTJyDKMIBY1ksFpbBMUc1XlcDXBOM
1k4vsVwhwJdr1IF0YO5B8ATQ9ZlM2el0wAfxNAsumS4W0+RaftQuiTkE3pGanhWf5S2FpajgI3WNVE5SKx8hGIOy
DsVUkDEE2pmsyoIIEJjCCBCIwggQeMIIDBqADAgECAhRzwvLtr5lfhJo/zMDyBaVvAMWs1Do/AQQwT1Em1jMCTUV
```

```
0AwwHUm9vdCBDQTAeFw0yMzA0MTcyMjM2NDFaFw0zAQ8ANA0AUaQVAEsMMIIBIi4eAgABBQD0QAEPADCCAQoCggE
BAPFecS6VD9uWs391mirSF2ZvtVRQQM2TGmlPJC6nUDfpizT7vvdqye2U3Yqiv5D++UEijYlUGCB5Ufb6GDUv0EB
XMP+sN9O88ZXTpZoNd1dy4x9uSfDm/eP5olsR98b+G1BfDFU+94jHP+6bMifp4ONeYCRz2RzlqfBjr3OBW/CxSl7
jlqJtkEH480KGMh8VpfF12Vi3O/Yg1Impr9IabI6CZW78ua302epo6wFett2LgStDYqIw49RklnHFHcXBRkkfoCi
j5ybpmyoblOJB0k6YgXV05oKMS6ewnH6SgShVTdfnFqWBjTu6RSvk4uwmuwvz0p69IvAiqcIFW4MA1ucCAwEAAaO
B5zCB5DAdBgNVHQ4EFgQUmmQFckE8vZEKxkWDa7os3EPMci8wHwYDVR0jBBgwFoBSwgM0MAwGA1UdEwEB/wQCMAC
FSgQdDwEOGAQDAgeAMBYBHgAlARAQDDAKBgiJr4ADCTA0BgNVHR8ELTArMCmgJ6AlhiNodHRwOi8vY3JsLnSJh0g
ubmV0L3Jvb3RfY3JsLmRlcjA2EUIcAQEEKjAoMCYRUgwwAYYaEUEQYW9jc3AyRAAuDAKRKvD/5k6KPs+YbDzJ39Y
ZONiYEwlqsgeo1XjXfSW/pcOcjSYMrbTmxLlVzlJEoDFHfmQ38OG1+oAez22tz0SfNhnSNpUGMng6MvLsq0i9r58
5PzFwrMyjusi8t1/vxoSWuaaSwI3iqxokLJ/ReaPztoAt2yZUO3uZNp2btJP00J5KQq9TtL+QGgcODzRASyvChxj
6drClmMdAsSaeCDxUx4pUyvpbSkr7RFlNVRZTzOqAvXwVBgzbpuDGKURdIlWgvo6+t9GSeWMtVRSS79BqZ2AWZ0l
blQ7T4VHElY2tRYyoYPoJ/64aFeUMIPKnbA0Kd6k/lB2cYIa88bSQbh1lecSOrw=="
 }
}
```

Monitoring additions

As a visual indication for operators, a new field will appear in `varz` JSON output wherever `ocsp_peer` has been enabled in a TLS map:

```
...
"tls_ocsp_peer_verify": true,
...
```

If `ocsp_cache` is enabled (implicitly or explicitly) `varz` will reflect the current cache type and provide updated cache statistics to help the operator understand cache effectiveness:

```
"ocsp_peer_cache": {
   "cache_type": "local",
   "cache_misses": 2,
   "cached_responses": 3,
   "cached_revoked_responses": 1,
   "cached_good_responses": 2
}
```

Debug logging

The following debug-enabled log output shows log entries example for: server startup, a rejected peer connection due to a revoked certificate, and server shutdown.

```
...
[6638] [DBG] Starting OCSP peer cache
[6638] [DBG] Loading OCSP peer cache [/home/todd/lab/mtls-ocsp/test/_rc_/cache.json]
[6638] [DBG] No OCSP peer cache found, starting with empty cache
[6638] [INF] OCSP peer cache online, type [local]
[6638] [INF] Server is ready
...
[6638] [DBG] 127.0.0.1:55140 - cid:5 - Client connection created
[6638] [DBG] 127.0.0.1:55140 - cid:5 - Starting TLS client connection handshake
[6638] [DBG] Peer OCSP enabled: 1 TLS client chain(s) will be evaluated
```

```
[6638] [DBG] Chain [0]: 3 total link(s)
[6638] [DBG] Chain [0] has 2 OCSP eligible link(s)
[6638] [DBG] Checking OCSP peer cache for [CN=UserA1,O=Testnats,L=Tacoma,ST=WA,C=US],
key [5xL/SuHl6JN0OmxrNMpzVMTA73JVYcRfGX8+HvJinEI=]
[6638] [DBG] OCSP peer cache miss for key [5xL/SuHl6JN0OmxrNMpzVMTA73JVYcRfGX8+HvJinEI=]
[6638] [DBG] Trying OCSP responder url [http://127.0.0.1:18888/]
[6638] [DBG] Caching OCSP response for [CN=UserA1,O=Testnats,L=Tacoma,ST=WA,C=US], key
[5xL/SuHl6JN0OmxrNMpzVMTA73JVYcRfGX8+HvJinEI=]
[6638] [DBG] OCSP response compression ratio: [0.851943]
[6638] [WRN] OCSP verify fail for [CN=UserA1,O=Testnats,L=Tacoma,ST=WA,C=US] with CA
status [revoked]
[6638] [DBG] Invalid OCSP response status: revoked
[6638] [DBG] No OCSP valid chains, thus peer is invalid
[6638] [ERR] 127.0.0.1:55140 - cid:5 - TLS handshake error: client not OCSP valid
[6638] [DBG] 127.0.0.1:55140 - cid:5 - Client connection closed: TLS Handshake Failure
...
[6638] [INF] Initiating Shutdown...
[6638] [DBG] Client accept loop exiting..
[6638] [DBG] SYSTEM - System connection closed: Client Closed
[6638] [INF] Server Exiting..
[6638] [DBG] Stopping OCSP peer cache
[6638] [DBG] OCSP peer cache is dirty, saving
[6638] [DBG] Saving OCSP peer cache [/home/todd/lab/mtls-ocsp/test/_rc_/cache.json]
[6638] [DBG] Saved OCSP peer cache successfully (2080 bytes)
...
```

Advisor system events (examples)

Example when a "bad" peer attempts client connection:

```
23:22:10 Subscribing on $SYS.SERVER.*.OCSP.>
[#1] Received on
"$SYS.SERVER.NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDOKL64W4W4.OCSP.PEER.LINK.INV
ALID"
{"type":"io.nats.server.advisory.v1.ocsp_peer_link_invalid","id":"cDlWM74JVKNnaAQmqC10mT
","timestamp":"2023-06-20T06:23:13.659116379Z","link":
{"subject":"CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US","issuer":"CN=Intermediate
CA,O=Tinghus,L=Tacoma,ST=WA,C=US","fingerprint":"6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9Q
BBbY=","raw":"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMA
kGA1UEBhMCVVMxCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBA
MMD0ludGVybWVkaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQ
swCQYDVQQIDAJXQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQT
EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsx
cAWsZhVcFOaTxjLtkYVPQ19dddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYa
x98RuFr/acUgfkAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWE
BQVNB5Mcd5yjMbFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKed
ZQ/kXgmd+ySDBYn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIU
Edkl8wHwYDVR0jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBA
MCBaAwDgYDVR0PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6
AthitodHRwOi8vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBg
grBgEFBQcwAYYYaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBg
kqhkiG9w0BAQsFAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezC
IIabLTC5pnCwm9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3u
Q7LB/kXxbpiNaUwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1C
CaZCZzWiTKkylNOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVx
```

```
AviGTpX78TVA=="},"peer":
{"subject":"CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US","issuer":"CN=Intermediate
CA,O=Tinghus,L=Tacoma,ST=WA,C=US","fingerprint":"6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9Q
BBbY=","raw":"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMA
kGA1UEBhMCVVMxCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBA
MMD0ludGVybWVkaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQ
swCQYDVQQIDAJXQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQT
EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsx
cAWsZhVcFOaTxjLtkYVPQ19dddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYa
x98RuFr/acUgfkAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWE
BQVNB5Mcd5yjMbFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKed
ZQ/kXgmd+ySDBYn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIU
Edkl8wHwYDVR0jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBA
MCBaAwDgYDVR0PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6
AthitodHRwOi8vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBg
grBgEFBQcwAYYYaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBg
kqhkiG9w0BAQsFAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezC
IIabLTC5pnCwm9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3u
Q7LB/kXxbpiNaUwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1C
CaZCZzWiTKkylNOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVx
AviGTpX78TVA=="},"server":
{"name":"tester","host":"0.0.0.0","id":"NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDO
KL64W4W4","ver":"2.10.0-beta.41","seq":31,"jetstream":true,"time":"2023-06-
20T06:23:13.659211768Z"},"reason":"Invalid OCSP response status: revoked"}

[#2] Received on
"$SYS.SERVER.NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDOKL64W4W4.OCSP.PEER.CONN.REJ
ECT"
{"type":"io.nats.server.advisory.v1.ocsp_peer_reject","id":"cDlWM74JVKNnaAQmqC10pL","tim
estamp":"2023-06-20T06:23:13.659151705Z","kind":"Client","peer":
{"subject":"CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US","issuer":"CN=Intermediate
CA,O=Tinghus,L=Tacoma,ST=WA,C=US","fingerprint":"6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9Q
BBbY=","raw":"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMA
kGA1UEBhMCVVMxCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBA
MMD0ludGVybWVkaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQ
swCQYDVQQIDAJXQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQT
EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsx
cAWsZhVcFOaTxjLtkYVPQ19dddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYa
x98RuFr/acUgfkAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWE
BQVNB5Mcd5yjMbFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKed
ZQ/kXgmd+ySDBYn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIU
Edkl8wHwYDVR0jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBA
MCBaAwDgYDVR0PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6
AthitodHRwOi8vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBg
grBgEFBQcwAYYYaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBg
kqhkiG9w0BAQsFAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezC
IIabLTC5pnCwm9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3u
Q7LB/kXxbpiNaUwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1C
CaZCZzWiTKkylNOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVx
AviGTpX78TVA=="},"server":
{"name":"tester","host":"0.0.0.0","id":"NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDO
KL64W4W4","ver":"2.10.0-beta.41","seq":32,"jetstream":true,"time":"2023-06-
20T06:23:13.659317657Z"},"reason":"client not OCSP valid"}
```


Event: io.nats.server.advisory.v1.ocsp_peer_link_invalid

```
{
  "type": "io.nats.server.advisory.v1.ocsp_peer_link_invalid",
  "id": "cDlWM74JVKNnaAQmqC10mT",
  "timestamp": "2023-06-20T06:23:13.659116379Z",
  "link": {
    "subject": "CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "issuer": "CN=Intermediate CA,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "fingerprint": "6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9QBBbY=",
    "raw":
"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMCVVM
xCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBAMMD0ludGVybWV
kaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJ
XQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQTEwggEiMA0GCSq
GSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsxcAWsZhVcFOaTx
jLtkYVPQ19dddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYax98RuFr/acUgf
kAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWEBQVNB5Mcd5yjM
bFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKedZQ/kXgmd+ySDB
Yn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIUEdkl8wHwYDVR0
jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBAMCBaAwDgYDVR0
PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6AthitodHRwOi8
vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYY
YaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBgkqhkiG9w0BAQs
FAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezCIIabLTC5pnCwm
9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3uQ7LB/kXxbpiNa
UwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1CCaZCZzWiTKkyl
NOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVxAviGTpX78TVA=
="
  },
  "peer": {
    "subject": "CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "issuer": "CN=Intermediate CA,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "fingerprint": "6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9QBBbY=",
    "raw":
"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMCVVM
xCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBAMMD0ludGVybWV
kaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJ
XQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQTEwggEiMA0GCSq
GSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsxcAWsZhVcFOaTx
jLtkYVPQ19dddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYax98RuFr/acUgf
kAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWEBQVNB5Mcd5yjM
bFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKedZQ/kXgmd+ySDB
Yn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIUEdkl8wHwYDVR0
jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBAMCBaAwDgYDVR0
PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6AthitodHRwOi8
vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYY
YaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBgkqhkiG9w0BAQs
FAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezCIIabLTC5pnCwm
9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3uQ7LB/kXxbpiNa
UwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1CCaZCZzWiTKkyl
NOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVxAviGTpX78TVA=
="
  },
  "server": {
    "name": "tester",
    "host": "0.0.0.0",
    "id": "NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDOKL64W4W4",
```

```
    "ver": "2.10.0-beta.41",
    "seq": 31,
    "jetstream": true,
    "time": "2023-06-20T06:23:13.659211768Z"
  },
  "reason": "Invalid OCSP response status: revoked"
}
```


Event: io.nats.server.advisory.v1.ocsp_peer_reject


```
{
  "type": "io.nats.server.advisory.v1.ocsp_peer_reject",
  "id": "cDlWM74JVKNnaAQmqC10pL",
  "timestamp": "2023-06-20T06:23:13.659151705Z",
  "kind": "Client",
  "peer": {
    "subject": "CN=BadUserA1,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "issuer": "CN=Intermediate CA,O=Tinghus,L=Tacoma,ST=WA,C=US",
    "fingerprint": "6QS2jCKv9hRrgLR0/2VTuNSVmtWa+/j1jEumc9QBBbY=",
    "raw":
"MIIEXDCCA0SgAwIBAgIURb5p0QsNjagFkzCMSy+yvOTrmiUwDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMCVVM
xCzAJBgNVBAgMAldBMQ8wDQYDVQQHDAZUYWNvbWExEDAOBgNVBAoMB1RpbmdodXMxGDAWBgNVBAMMD0ludGVybWV
kaWF0ZSBDQTAeFw0yMzA0MTcyMzA2NTJaFw0yNDA0MTYyMzA2NTJaMFExCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJ
XQTEPMA0GA1UEBwwGVGFjb21hMRAwDgYDVQQKDAdUaW5naHVzMRIwEAYDVQQDDAlCYWRVc2VyQTEwggEiMA0GCSq
GSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCrSWveLaNeL6KzHwNuIXku4OsDgX9ys5eW/7mNENRcsxcAWsZhVcFOaTx
jLtkYVPQ19ddpTADZCg3W2BIB6vZQixwRggB+xC1GyOQFFuCspAv+mrnLsX/bTo72LJCmZSqYax98RuFr/acUgf
kAtmaA0xLlauZnAWRZpLMkGMzRKJCo28+XZbzm+Y1Jd0BoMO5+vNtXqZr2Fq5F+NsLPda73BZWEBQVNB5Mcd5yjM
bFZ4KAovwk7ShvzmST94cPoLrWzTm/iGM7lnHjkNjfMKMi8AY+mwdpknr4n6CWCavvGnyrHHKedZQ/kXgmd+ySDB
Yn9h76I5GG5Trs8U6LRovAgMBAAGjggEkMIIBIDAdBgNVHQ4EFgQUEOaMMHDtJiReYXSfDjMZIUEdkl8wHwYDVR0
jBBgwFoAUexT7G7SgCTDIgbzhATJn0Gioo9EwDAYDVR0TAQH/BAIwADARBglghkgBhvhCAQEEBAMCBaAwDgYDVR0
PAQH/BAQDAgXgMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDBDA8BgNVHR8ENTAzMDGgL6AthitodHRwOi8
vY3JsLnRpbmdodXMubmV0L2ludGVybWVkaWF0ZV9jcmwuZGVyMDQGCCsGAQUFBwEBBCgwJjAkBggrBgEFBQcwAYY
YaHR0cDovL29jc3AudGluZ2h1cy5uZXQvMBoGA1UdEQQTMBGBD1VzZXJBMUB1c2VyLm5ldDANBgkqhkiG9w0BAQs
FAAOCAQEADgTil110Tc4dn09Gww4L6CjriTWpFh0syc+cpZ+QF/BbQE1p/UtwPfYE/Vg+COUezCIIabLTC5pnCwm
9S34X7ieRjCGmkMY26QmrP6VzSdFF9lD45Q4O9YDUqsZMmIKy9XEG1qOR4qUGb+ODmheUMhKj3uQ7LB/kXxbpiNa
UwQVbIFX83wh3jNbI8rHACRpQm5Dk81tKh01WGrHE3g1Ic8VgDH9Hr8yTgaesCIwpz3InbX0A1CCaZCZzWiTKkyl
NOxdn5e1O46SdHT30pFEHc1tpPDHucZKyNJAqlB/Eb+uHS5QaYqg2crWFA/npVk4eQCbiCYmQVxAviGTpX78TVA=
="
  },
  "server": {
    "name": "tester",
    "host": "0.0.0.0",
    "id": "NAXQD6DG5FVZANGJTOB7BM2H3PYDEHSYOZDHNBEJZARWOPDOKL64W4W4",
    "ver": "2.10.0-beta.41",
    "seq": 32,
    "jetstream": true,
    "time": "2023-06-20T06:23:13.659317657Z"
  },
  "reason": "client not OCSP valid"
}
```