security config commands

Command reference

NetApp
February 02, 2026

Table of Contents

security config commands

security config modify

Modify Security Configuration Options

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

The `security config modify` command modifies the existing cluster-wide security configuration. If you enable FIPS-compliant mode, the cluster will automatically select only compliant TLS protocols (currently TLSv1.3 and TLSv1.2). Non-compliant protocols are not enabled when FIPS-compliant mode is disabled. Use the `-supported-protocols` parameter to include or exclude TLS protocols independently from the FIPS mode. All protocols at or above the lowest version specified will be enabled, even those not explicitly specified. By default, FIPS mode is disabled, and ONTAP supports the TLSv1.3 and TLSv1.2 protocols. For backward compatibility, ONTAP supports adding SSLv3 and TLSv1 to the supported-protocols list when FIPS mode is disabled. Use the `-supported-cipher-suites` parameter to control which TLS cipher suites are permitted by the system. By default the supported-cipher-suites setting is *TLS_RSA_WITH_AES_128_CCM, TLS_RSA_WITH_AES_128_CCM_8, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CCM, TLS_RSA_WITH_AES_256_CCM_8, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA256, TLS_RSA_WITH_ARIA_128_GCM_SHA256, TLS_RSA_WITH_ARIA_256_GCM_SHA384, TLS_RSA_WITH_CAMELLIA_128_CBC_SHA, TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256, TLS_RSA_WITH_CAMELLIA_256_CBC_SHA, TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256, TLS_DHE_DSS_WITH_AES_128_GCM_SHA256, TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DHE_DSS_WITH_AES_128_CBC_SHA256, TLS_DHE_DSS_WITH_AES_256_GCM_SHA384, TLS_DHE_DSS_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA256, TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256, TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384, TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA, TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256, TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA, TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256, TLS_DHE_PSK_WITH_AES_128_CBC_SHA, TLS_DHE_PSK_WITH_AES_128_CBC_SHA256, TLS_DHE_PSK_WITH_AES_128_CCM, TLS_PSK_DHE_WITH_AES_128_CCM_8, TLS_DHE_PSK_WITH_AES_128_GCM_SHA256, TLS_DHE_PSK_WITH_AES_256_CBC_SHA, TLS_DHE_PSK_WITH_AES_256_CBC_SHA384, TLS_DHE_PSK_WITH_AES_256_CCM, TLS_PSK_DHE_WITH_AES_256_CCM_8, TLS_DHE_PSK_WITH_AES_256_GCM_SHA384, TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256, TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384, TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256, TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384, TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256, TLS_DHE_RSA_WITH_AES_128_CCM, TLS_DHE_RSA_WITH_AES_128_CCM_8, TLS_DHE_RSA_WITH_AES_128_GCM_SHA256, TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA256, TLS_DHE_RSA_WITH_AES_256_CCM, TLS_DHE_RSA_WITH_AES_256_CCM_8, TLS_DHE_RSA_WITH_AES_256_GCM_SHA384, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA256, TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256, TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384, TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA, TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256, TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA, TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256, TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384,*

TLS_ECDHE_ECDSA_WITH_AES_128_CCM, TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_CCM,
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384,
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA, TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA, TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256,
TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384,
TLS_ECDHE_PSK_WITH_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256, TLS_PSK_WITH_AES_128_CBC_SHA,
TLS_PSK_WITH_AES_128_CBC_SHA256, TLS_PSK_WITH_AES_128_CCM,
TLS_PSK_WITH_AES_128_CCM_8, TLS_PSK_WITH_AES_128_GCM_SHA256,
TLS_PSK_WITH_AES_256_CBC_SHA, TLS_PSK_WITH_AES_256_CBC_SHA384,
TLS_PSK_WITH_AES_256_CCM, TLS_PSK_WITH_AES_256_CCM_8,
TLS_PSK_WITH_AES_256_GCM_SHA384, TLS_PSK_WITH_ARIA_128_GCM_SHA256,
TLS_PSK_WITH_ARIA_256_GCM_SHA384, TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256,
TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384, TLS_PSK_WITH_CHACHA20_POLY1305_SHA256,
TLS_RSA_PSK_WITH_AES_128_CBC_SHA, TLS_RSA_PSK_WITH_AES_128_CBC_SHA256,
TLS_RSA_PSK_WITH_AES_128_GCM_SHA256, TLS_RSA_PSK_WITH_AES_256_CBC_SHA,
TLS_RSA_PSK_WITH_AES_256_CBC_SHA384, TLS_RSA_PSK_WITH_AES_256_GCM_SHA384,
TLS_RSA_PSK_WITH_ARIA_128_GCM_SHA256, TLS_RSA_PSK_WITH_ARIA_256_GCM_SHA384,
TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256,
TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384,
TLS_RSA_PSK_WITH_CHACHA20_POLY1305_SHA256, TLS_SRP_SHA_WITH_AES_128_CBC_SHA,
TLS_SRP_SHA_WITH_AES_256_CBC_SHA, TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA,
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA, TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA,
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA, TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256 . Select a cipher suite which is available with the corresponding selected protocol. An invalid configuration may cause some functionality to fail to operate properly. Valid values for supported-cipher-suites are listed at "https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml" published by IANA.

Parameters

`-interface <SSL>` - (DEPRECATED)-FIPS-Compliant Interface (privilege: advanced)

This parameter has been deprecated in ONTAP 9.8 and may be removed in a future release of ONTAP.

```
Selects the FIPS-compliant interface. The only valid value is ``_SSL_``
.
```

[-is-fips-enabled {true|false}] - FIPS Mode (privilege: advanced)

Enables or disables FIPS-compliant mode for the entire cluster. Default is *false* .

[-supported-protocols {TLSv1.3|TLSv1.2|TLSv1.1|TLSv1|SSLv3}] - Supported Protocols (privilege: advanced)

Selects the supported protocols for the selected interface. Default is *TLSv1.3,TLSv1.2*

[-supported-ciphers <Cipher String>] - (DEPRECATED)-Supported Ciphers (privilege: advanced)

This parameter has been deprecated in ONTAP 9.8 and may be removed in a future release of ONTAP. Use the supported-ciphers-suites parameter instead.

```
Selects the supported cipher suites for the selected interface. Default
is ``_ALL:!LOW:!aNULL:!EXP:!eNULL_`` .
```

[-supported-cipher-suites <Cipher String>,…] - Supported Cipher Suites (privilege: advanced)

Selects the supported cipher suites for the selected interface.

Examples

The following command enables FIPS mode in the cluster. (Default setting for FIPS mode is *false*)

```
cluster1::> security config modify * -is-fips-enabled true
```

The following command limits the supported protocols to just TLSv1.3 in the cluster. (Default setting for supported protocols is *TLSv1.3,TLSv1.2*)

```
cluster1::*> security config modify * -supported-protocols TLSv1.3
```

The following command limits the supported cipher suites in the cluster to the listed ciphers.

```
cluster1::*> security config modify * -supported-cipher-suites
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_AES_256_GCM_SHA384
```

security config show

Display Security Configuration Options

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

The `security config show` command displays the security configurations of the cluster in advanced privilege mode.

Default values are as follows:

- SSL FIPS mode: disabled
- Supported protocols: TLSv1.3,TLSv1.2
- Supported cipher suites: All suites for the listed protocols except those that have no authentication, low encryption strength (less than 56 bits), or use 3DES or static DH key exchange.

Enabling FIPS mode will cause the entire cluster to use FIPS-compliant crypto operations only.

Use the security config modify command to change the protocols and cipher suites that the cluster will support.

Parameters

`[-fields <fieldname>,…]`

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| `[-instance ]` (privilege: advanced) }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

`[-interface <SSL>]` - (DEPRECATED)-FIPS-Compliant Interface (privilege: advanced)

This parameter has been deprecated in ONTAP 9.8 and may be removed in a future release of ONTAP. As there only ever existed one valid value for this parameter, filtering on it has never altered the results.

```
Displays configurations that match the specified value for the
interface.
```

`[-is-fips-enabled {true|false}]` - FIPS Mode (privilege: advanced)

Display configurations that match the specified value for FIPS mode.

`[-supported-protocols {TLSv1.3|TLSv1.2|TLSv1.1|TLSv1|SSLv3}]` - Supported Protocols (privilege: advanced)

Displays configurations that match the specified protocols.

`[-supported-ciphers <Cipher String>]` - (DEPRECATED)-Supported Ciphers (privilege: advanced)

This parameter has been deprecated in ONTAP 9.8 and may be removed in a future release of ONTAP. Use the supported-cipher-suites parameter instead.

```
Displays the configurations that match the specified supported ciphers.
```

[-supported-cipher-suites <Cipher String>,…] - Supported Cipher Suites (privilege: advanced)

Displays the configurations that match the specified supported cipher suites.

Examples

The following example shows the default security configurations for a cluster.

```
cluster1::> security config show
Cluster    Supported
FIPS Mode  Protocols Supported Cipher Suites
---------- ---------
-----------------------------------------------------------
false      TLSv1.3,  TLS_RSA_WITH_AES_128_CCM, TLS_RSA_WITH_AES_128_CCM_8,
           TLSv1.2   TLS_RSA_WITH_AES_128_GCM_SHA256,
                     TLS_RSA_WITH_AES_128_CBC_SHA,
                     TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CCM,
                     TLS_RSA_WITH_AES_256_CCM_8,
                     TLS_RSA_WITH_AES_256_GCM_SHA384,
                     TLS_RSA_WITH_AES_256_CBC_SHA,
                     TLS_RSA_WITH_AES_256_CBC_SHA256,
                     TLS_RSA_WITH_ARIA_128_GCM_SHA256,
                     TLS_RSA_WITH_ARIA_256_GCM_SHA384,
                     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA,
                     TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256,
                     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA,
                     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
                     TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,
                     TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,
                     TLS_DHE_DSS_WITH_AES_256_CBC_SHA,
                     TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
                     TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256,
                     TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384,
                     TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA,
                     TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256,
                     TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA,
                     TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256,
                     TLS_DHE_PSK_WITH_AES_128_CBC_SHA,
                     TLS_DHE_PSK_WITH_AES_128_CBC_SHA256,
                     TLS_DHE_PSK_WITH_AES_128_CCM,
```

```
                    TLS_PSK_DHE_WITH_AES_128_CCM_8,
                    TLS_DHE_PSK_WITH_AES_128_GCM_SHA256,
                    TLS_DHE_PSK_WITH_AES_256_CBC_SHA,
                    TLS_DHE_PSK_WITH_AES_256_CBC_SHA384,
                    TLS_DHE_PSK_WITH_AES_256_CCM,
                    TLS_PSK_DHE_WITH_AES_256_CCM_8,
                    TLS_DHE_PSK_WITH_AES_256_GCM_SHA384,
                    TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256,
                    TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384,
                    TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_DHE_RSA_WITH_AES_128_CCM,
                    TLS_DHE_RSA_WITH_AES_128_CCM_8,
                    TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
                    TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
                    TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
                    TLS_DHE_RSA_WITH_AES_256_CCM,
                    TLS_DHE_RSA_WITH_AES_256_CCM_8,
                    TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
                    TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
                    TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,
                    TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256,
                    TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384,
                    TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA,
                    TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA,
                    TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256,
                    TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256,
                    TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CCM,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8,
                    TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CCM,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8,
                    TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
                    TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256,
                    TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384,
                    TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
```

```
                    TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384,
                    TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_ECDHE_PSK_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
                    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
                    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
                    TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_PSK_WITH_AES_128_CBC_SHA,
                    TLS_PSK_WITH_AES_128_CBC_SHA256,
TLS_PSK_WITH_AES_128_CCM,
                    TLS_PSK_WITH_AES_128_CCM_8,
                    TLS_PSK_WITH_AES_128_GCM_SHA256,
                    TLS_PSK_WITH_AES_256_CBC_SHA,
                    TLS_PSK_WITH_AES_256_CBC_SHA384,
TLS_PSK_WITH_AES_256_CCM,
                    TLS_PSK_WITH_AES_256_CCM_8,
                    TLS_PSK_WITH_AES_256_GCM_SHA384,
                    TLS_PSK_WITH_ARIA_128_GCM_SHA256,
                    TLS_PSK_WITH_ARIA_256_GCM_SHA384,
                    TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_PSK_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_RSA_PSK_WITH_AES_128_CBC_SHA,
                    TLS_RSA_PSK_WITH_AES_128_CBC_SHA256,
                    TLS_RSA_PSK_WITH_AES_128_GCM_SHA256,
                    TLS_RSA_PSK_WITH_AES_256_CBC_SHA,
                    TLS_RSA_PSK_WITH_AES_256_CBC_SHA384,
                    TLS_RSA_PSK_WITH_AES_256_GCM_SHA384,
                    TLS_RSA_PSK_WITH_ARIA_128_GCM_SHA256,
                    TLS_RSA_PSK_WITH_ARIA_256_GCM_SHA384,
                    TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256,
                    TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384,
                    TLS_RSA_PSK_WITH_CHACHA20_POLY1305_SHA256,
                    TLS_SRP_SHA_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_WITH_AES_256_CBC_SHA,
                    TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA,
```

```
                    TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA,
                    TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384,
                    TLS_CHACHA20_POLY1305_SHA256
```

The following example shows the security configuration after FIPS mode has been enabled.

```
cluster1::> security config show
Cluster    Supported
FIPS Mode  Protocols Supported Cipher Suites
---------- ---------
------------------------------------------------------------
true       TLSv1.3,  TLS_RSA_WITH_AES_128_CCM, TLS_RSA_WITH_AES_128_CCM_8,
           TLSv1.2   TLS_RSA_WITH_AES_128_GCM_SHA256,
                     TLS_RSA_WITH_AES_128_CBC_SHA,
                     TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CCM,
                     TLS_RSA_WITH_AES_256_CCM_8,
                     TLS_RSA_WITH_AES_256_GCM_SHA384,
                     TLS_RSA_WITH_AES_256_CBC_SHA,
                     TLS_RSA_WITH_AES_256_CBC_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,
                     TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
                     TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,
                     TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,
                     TLS_DHE_DSS_WITH_AES_256_CBC_SHA,
                     TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
                     TLS_DHE_PSK_WITH_AES_128_CBC_SHA,
                     TLS_DHE_PSK_WITH_AES_128_CBC_SHA256,
                     TLS_DHE_PSK_WITH_AES_128_CCM,
                     TLS_PSK_DHE_WITH_AES_128_CCM_8,
                     TLS_DHE_PSK_WITH_AES_128_GCM_SHA256,
                     TLS_DHE_PSK_WITH_AES_256_CBC_SHA,
                     TLS_DHE_PSK_WITH_AES_256_CBC_SHA384,
                     TLS_DHE_PSK_WITH_AES_256_CCM,
                     TLS_PSK_DHE_WITH_AES_256_CCM_8,
                     TLS_DHE_PSK_WITH_AES_256_GCM_SHA384,
                     TLS_DHE_RSA_WITH_AES_128_CCM,
                     TLS_DHE_RSA_WITH_AES_128_CCM_8,
                     TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
                     TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
                     TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
                     TLS_DHE_RSA_WITH_AES_256_CCM,
                     TLS_DHE_RSA_WITH_AES_256_CCM_8,
                     TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
```

```
                    TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
                    TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CCM,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8,
                    TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CCM,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8,
                    TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
                    TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384,
                    TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
                    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
                    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
                    TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
                    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
                    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
                    TLS_PSK_WITH_AES_128_CBC_SHA,
                    TLS_PSK_WITH_AES_128_CBC_SHA256,
TLS_PSK_WITH_AES_128_CCM,
                    TLS_PSK_WITH_AES_128_CCM_8,
                    TLS_PSK_WITH_AES_128_GCM_SHA256,
                    TLS_PSK_WITH_AES_256_CBC_SHA,
                    TLS_PSK_WITH_AES_256_CBC_SHA384,
TLS_PSK_WITH_AES_256_CCM,
                    TLS_PSK_WITH_AES_256_CCM_8,
                    TLS_PSK_WITH_AES_256_GCM_SHA384,
                    TLS_RSA_PSK_WITH_AES_128_CBC_SHA,
                    TLS_RSA_PSK_WITH_AES_128_CBC_SHA256,
                    TLS_RSA_PSK_WITH_AES_128_GCM_SHA256,
                    TLS_RSA_PSK_WITH_AES_256_CBC_SHA,
                    TLS_RSA_PSK_WITH_AES_256_CBC_SHA384,
                    TLS_RSA_PSK_WITH_AES_256_GCM_SHA384,
                    TLS_SRP_SHA_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_WITH_AES_256_CBC_SHA,
                    TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA,
                    TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA,
                    TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA,
                    TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384,
                    TLS_CHACHA20_POLY1305_SHA256
```

Related Links

- security config modify

security config ocsp disable

Disable OCSP for one or more selected applications

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

The `security config ocsp disable` command disables the OCSP-based certificate status check for applications supporting SSL/TLS communications. For more information about the OCSP-based certificate status check for applications supporting SSL/TLS communications, see the security config ocsp show command.

Parameters

`-application <SSL/TLS Application supporting OCSP>,…` - Application Name (privilege: advanced)

Use this parameter to specify the application to disable the OCSP support. To disable all applications, the value 'all' can be used. Note: You cannot specify the value 'all' with other applications.

Examples

The following example disables the OCSP support for AutoSupport and EMS applications:

```
cluster1::*> security config ocsp disable -application autosupport,ems

cluster1::> security config ocsp show
Application          OCSP Enabled?
-------------------- -------------
autosupport          false
audit_log            true
fabricpool           true
ems                  false
kmip                 true
ldap                 true
ssh                  true
6 entries were displayed.
```

The following example disables the OCSP support for all applications:

```
cluster1::*> security config ocsp disable -application all
Warning: OCSP will be disabled for all applications. Any previous
modifications
         will be ignored.
         Do you want to continue? {y|n}: y

cluster1::*> security config ocsp show
Application          OCSP Enabled?
-------------------- -------------
autosupport          false
audit_log            false
fabricpool           false
ems                  false
kmip                 false
ldap                 false
ssh                  false
6 entries were displayed.
```

Related Links

- security config ocsp show

security config ocsp enable

Enable OCSP for one or more selected applications

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

The `security config ocsp enable` command enables the OCSP-based certificate status check for applications supporting SSL/TLS communications. For more information about the OCSP-based certificate status check for applications supporting SSL/TLS communications, see the security config ocsp show command.

Parameters

`-application <SSL/TLS Application supporting OCSP>,…` - List of Applications (privilege: advanced)

Use this parameter to specify the application to enable the OCSP support. To enable all applications, the value 'all' can be used. Note: You cannot specify the value 'all' with other applications.

Examples

The following example enables the OCSP support for AutoSupport and EMS applications:

```
cluster1::*> security config ocsp enable -application autosupport,ems

cluster1::> security config ocsp show
Application          OCSP Enabled?
-------------------- -------------
autosupport          true
audit_log            false
fabricpool           false
ems                  true
kmip                 false
ldap                 false
ssh                  true
6 entries were displayed.
```

The following example enables the OCSP support for all applications:

```
cluster1::*> security config ocsp enable -application all
Warning: OCSP will be enabled for all applications. Any previous
modifications
         will be ignored.
         Do you want to continue? {y|n}: y

cluster1::*> security config ocsp show
Application          OCSP Enabled?
-------------------- -------------
autosupport          true
audit_log            true
fabricpool           true
ems                  true
kmip                 true
ldap                 true
ssh                  true
6 entries were displayed.
```

Related Links

- security config ocsp show

security config ocsp show

Show Online Certificate Status Protocol (OCSP) settings

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

The `security config ocsp show` command displays the support status of the OCSP-based certificate status check for applications supporting SSL/TLS communications. If the OCSP support is enabled for an application, this check is done in addition to the certificate chain validation as part of the SSL handshake process. The OCSP-based certificate status check is done for all the certificates in the chain, provided that each certificate has the OCSP URI access points mentioned in them. If no access points are specified, the OCSP-based certificate revocation status check is ignored for that certificate and checking continues for the rest of the certificates in the chain.

Parameters

`[-fields <fieldname>,…]`

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| `[-instance ]` (privilege: advanced) }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

`[-application <SSL/TLS Application supporting OCSP>]` - Application Name (privilege: advanced)

Selects the application that matches this parameter value. Applications include:

- autosupport - AutoSupport
- audit_log - Audit Logging
- fabricpool - External capacity tiers
- ems - Event Management System
- kmip - Key Management Interoperability Protocol
- ldap_ad - Lightweight Directory Access Protocol - Active Directory (query and modify items in Active Directory)
- ldap_nis_namemap - Lightweight Directory Access Protocol - NIS and Name Mapping (query Unix user, group, netgroup and name mapping information)
- ssh - Secure Shell

`[-is-ocsp-enabled {true|false}]` - Is OCSP-based Certificate Status Check Enabled? (privilege: advanced)

Selects the application that matches this parameter value.

Examples

The following example displays the OCSP support for the applications supporting SSL/TLS communications:

```
cluster1::> security config ocsp show
Application          OCSP Enabled?
-------------------- -------------
autosupport          true
audit_log            false
fabricpool           false
ems                  true
kmip                 false
ldap                 false
ssh                  false
6 entries were displayed.
```

The following example displays the OCSP support for AutoSupport:

```
cluster1::*> security config ocsp show -application autosupport
Application Name: autosupport
Is OCSP-based Certificate Status Check Enabled?: true
```

security config status show

(DEPRECATED)-Display Security Configuration Status

Availability: This command is available to *cluster* administrators at the *advanced* privilege level.

Description

> This command has been deprecated as of 9.9 and may be removed in a future release of ONTAP. Reboot is no longer required to apply the security configuration, so it now always displays false.

```
The `security config status show` command displays the required reboot
status of the nodes in the cluster after security configuration settings
have been modified using the xref:{relative_path}security-config-
modify.html[security config modify] command. Use this command to monitor
the status of the required reboot process. When all nodes have rebooted,
the cluster is ready to use the new security configuration settings.
```

Parameters

[-fields <fieldname>,…]

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

| [-instance] (privilege: advanced) }

If you specify the `-instance` parameter, the command displays detailed information about all fields.

[-node {<nodename>|local}] - Node Name (privilege: advanced)

Select the node whose reboot-status you want to display.

[-reboot-needed {true|false}] - Reboot Needed (privilege: advanced)

reboot-needed status of the node that tells if the node requires a reboot for security configuration to take effect.

Examples

The following example displays the status of a configuration change in a four-node cluster.

```
cluster1::> security config status show
Nodes in Cluster      Reboot Needed
--------------------- ------------------
node1                 true
node2                 true
node3                 false
node4                 false
4 entries were displayed.
```

The following example shows the output of the command after the cluster reboot process is complete.

```
cluster1::> security config status show
Nodes in Cluster      Reboot Needed
--------------------- ------------------
node1                 false
node2                 false
node3                 false
node4                 false
4 entries were displayed.
```

Related Links

- security config modify

Copyright information

Trademark information