



# **security commands**

## ONTAP commands

NetApp  
February 04, 2026

This PDF was generated from <https://docs.netapp.com/us-en/ontap-cli-93/security-snmpusers.html> on February 04, 2026. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# Table of Contents

security commands .....	1
security snmpusers .....	1
Description .....	1
Parameters .....	1
Examples .....	2
security audit commands .....	2
security audit modify .....	2
security audit show .....	3
security audit log show .....	4
security certificate commands .....	4
security certificate create .....	4
security certificate delete .....	7
security certificate generate-csr .....	8
security certificate install .....	10
security certificate show .....	13
security certificate sign .....	16
security certificate ca-issued revoke .....	20
security certificate ca-issued show .....	20
security config commands .....	23
security config modify .....	23
security config show .....	24
security config ocsp disable .....	26
security config ocsp enable .....	27
security config ocsp show .....	29
security config status show .....	30
security key-manager commands .....	31
security key-manager add .....	31
security key-manager create-key .....	32
security key-manager delete-key-database .....	34
security key-manager delete-kmp-config .....	34
security key-manager delete .....	35
security key-manager prepare-to-downgrade .....	36
security key-manager query .....	36
security key-manager restore .....	39
security key-manager setup .....	42
security key-manager show .....	44
security key-manager update-passphrase .....	45
security key-manager backup show .....	46
security key-manager certificate update .....	47
security key-manager external boot-interfaces modify .....	48
security key-manager external boot-interfaces show .....	50
security key-manager key show .....	52
security login commands .....	55

security login create . . . . .	55
security login delete . . . . .	58
security login expire-password . . . . .	60
security login lock . . . . .	61
security login modify . . . . .	62
security login password-prepare-to-downgrade . . . . .	64
security login password . . . . .	65
security login show . . . . .	66
security login unlock . . . . .	69
security login whoami . . . . .	70
security login banner modify . . . . .	70
security login banner show . . . . .	71
security login domain-tunnel create . . . . .	72
security login domain-tunnel delete . . . . .	73
security login domain-tunnel modify . . . . .	73
security login domain-tunnel show . . . . .	74
security login motd modify . . . . .	75
security login motd show . . . . .	76
security login publickey create . . . . .	77
security login publickey delete . . . . .	78
security login publickey load-from-uri . . . . .	79
security login publickey modify . . . . .	80
security login publickey show . . . . .	81
security login role create . . . . .	82
security login role delete . . . . .	83
security login role modify . . . . .	84
security login role prepare-to-downgrade . . . . .	85
security login role show-ontapi . . . . .	85
security login role show . . . . .	87
security login role config modify . . . . .	88
security login role config reset . . . . .	90
security login role config show . . . . .	91
security protocol commands . . . . .	94
security protocol modify . . . . .	94
security protocol show . . . . .	95
security protocol ssh modify . . . . .	96
security protocol ssh show . . . . .	96
security saml-sp commands . . . . .	97
security saml-sp create . . . . .	97
security saml-sp delete . . . . .	98
security saml-sp modify . . . . .	99
security saml-sp repair . . . . .	100
security saml-sp show . . . . .	100
security saml-sp status show . . . . .	101
security session commands . . . . .	102

security session kill-cli .....	102
security session show .....	105
security session limit create .....	109
security session limit delete .....	110
security session limit modify .....	110
security session limit show .....	111
security session limit application create .....	112
security session limit application delete .....	112
security session limit application modify .....	113
security session limit application show .....	114
security session limit location create .....	114
security session limit location delete .....	115
security session limit location modify .....	116
security session limit location show .....	116
security session limit request create .....	117
security session limit request delete .....	118
security session limit request modify .....	119
security session limit request show .....	119
security session limit user create .....	120
security session limit user delete .....	121
security session limit user modify .....	122
security session limit user show .....	122
security session limit vserver create .....	123
security session limit vserver delete .....	124
security session limit vserver modify .....	125
security session limit vserver show .....	125
security session request-statistics show-by-application .....	126
security session request-statistics show-by-location .....	130
security session request-statistics show-by-request .....	133
security session request-statistics show-by-user .....	135
security session request-statistics show-by-vserver .....	139
security ssh commands .....	142
security ssh add .....	142
security ssh modify .....	143
security ssh prepare-to-downgrade .....	144
security ssh remove .....	145
security ssh show .....	146
security ssl commands .....	148
security ssl modify .....	148
security ssl show .....	151

# security commands

## security snmpusers

Show SNMP users

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

### Description

The `security snmpusers` displays the following information about SNMP users:

- User name
- Authentication method
- Hexadecimal engine ID
- Authentication protocol
- Privacy protocol
- Security group

### 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 ] }

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

**[-vserver <Vserver Name>] - Vserver**

If this parameter is specified, the command displays information only about the SNMP user or users that belong to the specified Vserver.

**[-username <text>] - User Name**

If this parameter is specified, the command displays information only about the SNMP user with the specified user name.

**[-authmethod <text>] - Authentication Method**

If this parameter is specified, the command displays information only about the SNMP user or users that use the specified authentication method. Possible values include the following:

- community-SNMP community strings
- usm-SNMP user security model

**[-remote-switch-ipaddress <IP Address>] - Remote Switch IP Address**

If this parameter is specified, the command displays information only about the remote SNMP user or users that belong to the specified remote switch.

#### **[-engineid <Hex String>] - Engine Id**

If this parameter is specified, the command displays information only about the SNMP user or users that use the specified engine ID, specified in hexadecimal format.

#### **[-authprotocol <text>] - Authentication Protocol**

If this parameter is specified, the command displays information only about the SNMP user or users that use the specified authentication protocol.

#### **[-privprotocol <text>] - Privacy Protocol**

If this parameter is specified, the command displays information only about the SNMP user or users that use the specified privacy protocol.

#### **[-securitygroup <text>] - Security Group**

If this parameter is specified, the command displays information only about the SNMP user or users that belong to the specified security group.

## **Examples**

The following example displays information about all SNMP users:

```
cluster1::> security snmpusers
Protocols Security Remote
Vserver  UserName  AuthMethod EngineId  Auth  Priv Group  Switch
IP
-----
-----
cluster1 comm1    community  8000031504312d38302d313233343536  -  -  -  readwrite
cluster1 private  community  8000031504312d38302d313233343536  -  -  -  readwrite
cluster1 snmpuser1 usm      80000634b2100000533296869      -  -  -  readwrite
172.2.20.91
vs1      snmpuser2  community  8000031504312d38302d31323334353632  -  -  -  readwrite
vs1      snmpuser3  usm      8000031504312d38302d31323334353632  -  -  -  readwrite
```

## **security audit commands**

### **security audit modify**

Set administrative audit logging settings

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

## Description

The `security audit modify` command modifies the following audit-logging settings for the management interface:

- Whether get requests for the CLI are audited
- Whether get requests for the Data ONTAP API (ONTAPI) are audited

## Parameters

### **`[-cliget {on|off}]` - Enable auditing of CLI get operations**

This specifies whether get requests for the CLI are audited. The default setting is `off`.

### **`[-ontapiget {on|off}]` - Enable auditing of Data ONTAP API get operations**

This specifies whether get requests for the Data ONTAP API (ONTAPI) interface are audited. The default setting is `off`.

## Examples

The following example turns off auditing of get requests for the CLI interface:

```
cluster1::> security audit modify -cliget off
```

## **security audit show**

Show administrative audit logging settings

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

## Description

The `security audit show` command displays the following audit-logging settings for the management interface:

- Whether get requests for the CLI are audited
- Whether get requests for the Data ONTAP API (ONTAPI) are audited

Audit log entries are written to the 'audit' log, viewable via the 'security audit log show' command.

## Examples

The following example displays the audit-logging settings for the management interface:

```
cluster1::> security audit show
          Auditing State for
          Get Requests:
          -----
          CLI:      off
          ONTAPI:   on
```

## security audit log show

Display audit entries merged from multiple nodes in the cluster

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

### Description

The `security audit log show` command displays cluster-wide audit log messages. Messages from each node are interleaved in chronological order.

### 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 ] }**

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

**[-timestamp <Date>] - Log Entry Timestamp**

Selects the entries that match the specified input for timestamp. This will be in the local timezone.

**[-node {<nodename>|local}] - Node**

Selects the entries that match the specified input for node.

**[-entry <text>] - Log Message Entry**

Selects the entries that match the specified input for entry.

## security certificate commands

### security certificate create

Create and Install a Self-Signed Digital Certificate

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `security certificate create` command creates and installs a self-signed digital certificate, which can be used for server authentication, for signing other certificates by acting as a certificate authority (CA), or

for Data ONTAP as an SSL client. The certificate function is selected by the `-type` field. Self-signed digital certificates are not as secure as certificates signed by a CA. Therefore, they are not recommended in a production environment.

## Parameters

### **-vserver <Vserver Name>** - Name of Vserver

This specifies the name of the Vserver on which the certificate will exist.

### **-common-name <FQDN or Custom Common Name>** - FQDN or Custom Common Name

This specifies the desired certificate name as a fully qualified domain name (FQDN) or custom common name or the name of a person. The supported characters, which are a subset of the ASCII character set, are as follows:

- Letters a through z, A through Z
- Numbers 0 through 9
- Asterisk (\*), period (.), underscore (\_) and hyphen (-)

The common name must not start or end with a "-" or a ". ". The maximum length is 253 characters.

### **-type <type of certificate>** - Type of Certificate

This specifies the certificate type. Valid values are the following:

- *server* - creates and installs a self-signed digital certificate and intermediate certificates to be used for server authentication
- *root-ca* - creates and installs a self-signed digital certificate to sign other certificates by acting as a certificate authority (CA)
- *client* - includes a self-signed digital certificate and private key to be used for Data ONTAP as an SSL client

### **[-subtype <kmip-cert>]** - Certificate Subtype

This specifies a certificate subtype. This optional parameter can have an empty value (the default). The only valid value is as follows:

- *kmip-cert* - this is a Key Management Interoperability Protocol (KMIP) certificate

### **-size <size of requested certificate in bits>** - Size of Requested Certificate in Bits

This specifies the number of bits in the private key. The larger the value, the more secure is the key. The default is 2048. Possible values include 512, 1024, 1536, 2048 and 3072 when the "FIPS Mode" in "security config" is false. When the "FIPS Mode" is true, the possible values are 2048 and 3072.

### **-country <text>** - Country Name

This specifies the country where the Vserver resides. The country name is a two-letter code. The default is US. Here is the list of country codes:

[Country Codes](#)

### **-state <text>** - State or Province Name

This specifies the state or province where the Vserver resides.

**-locality <text> - Locality Name**

This specifies the locality where the Vserver resides. For example, the name of a city.

**-organization <text> - Organization Name**

This specifies the organization where the Vserver resides. For example, the name of a company.

**-unit <text> - Organization Unit**

This specifies the unit where the Vserver resides. For example, the name of a section or a department within a company.

**-email-addr <mail address> - Contact Administrator's Email Address**

This specifies the email address of the contact administrator for the Vserver.

**-expire-days <integer> - Number of Days until Expiration**

This specifies the number of days until the certificate expires. The default value is 365 days. Possible values are between 1 and 3652.

**-protocol <protocol> - Protocol**

This specifies the protocol type. This parameter currently supports only the SSL protocol type. The default is SSL.

**-hash-function <hashing function> - Hashing Function**

This specifies the cryptographic hashing function for signing the certificate. The default is SHA256. Possible values include *SHA1* , *SHA256* , *MD5* , *SHA224* , *SHA384* and *SHA512* when the "FIPS Mode" in "security config" is false. When the "FIPS Mode" is true, the possible values are *SHA224* , *SHA256* , *SHA384* and *SHA512*

## Examples

This example creates a server type, self-signed digital certificate for a Vserver named vs0 at a company whose custom common name is *www.example.com* and whose Vserver name is vs0.

```
cluster1::> security certificate create -vserver vs0 -common-name  
www.example.com -type server
```

This example creates a root-ca type, self-signed digital certificate with a 2048-bit private key generated by the SHA256 hashing function that will expire in 365 days for a Vserver named vs0 for use by the Software group in IT at a company whose custom common name is *www.example.com* , located in Sunnyvale, California, USA. The email address of the contact administrator who manages the Vserver is *web@example.com* .

```
cluster1::> security certificate create -vserver vs0 -common-name  
www.example.com -type root-ca -size 2048 -country US -state California  
-locality Sunnyvale -organization IT -unit Software -email-addr  
web@example.com -expire-days 365 -hash-function SHA256
```

This example creates a client type of self-signed digital certificate for a Vserver named vs0 at a company that uses Data ONTAP as an SSL client. The company's custom common name is *www.example.com* and its

Vserver name is vs0.

```
cluster1::> security certificate create -vserver vs0 -common-name
www.example.com -type client -size 2048 -country US -state California
-locality Sunnyvale -organization IT -unit Software -email-addr
web@example.com -expire-days 365 -hash-function SHA256
```

## security certificate delete

### Delete an Installed Digital Certificate

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

This command deletes an installed digital security certificate.

#### Parameters

##### **-vserver <Vserver Name>** - Name of Vserver

This specifies the Vserver that contains the certificate.

##### **-common-name <FQDN or Custom Common Name>** - FQDN or Custom Common Name

This specifies the desired certificate name as a fully qualified domain name (FQDN) or custom common name or the name of a person. The supported characters, which are a subset of the ASCII character set, are as follows:

- Letters a through z, A through Z
- Numbers 0 through 9
- Asterisk (\*), period (.), underscore (\_) and hyphen (-)

The common name must not start or end with a "-" or a ". ". The maximum length is 253 characters.

##### **[-serial <text>]** - Serial Number of Certificate

This specifies the certificate serial number.

##### **-ca <text>** - Certificate Authority

This specifies the certificate authority (CA).

##### **-type <type of certificate>** - Type of Certificate

This specifies the certificate type. Valid values are the following:

- *server* - includes server certificates and intermediate certificates
- *root-ca* - includes a self-signed digital certificate to sign other certificates by acting as a certificate authority (CA)
- *client-ca* - includes the public key certificate for the root CA of the SSL client. If this client-ca certificate is created as part of a root-ca, it will be deleted along with the corresponding deletion of the root-ca.

- *server-ca* - includes the public key certificate for the root CA of the SSL server to which Data ONTAP is a client. If this server-ca certificate is created as part of a root-ca, it will be deleted along with the corresponding deletion of the root-ca.
- *client* - includes a public key certificate and private key to be used for Data ONTAP as an SSL client

#### **[-subtype <kmip-cert>] - Certificate Subtype**

This specifies a certificate subtype. This optional parameter can have an empty value (the default). The only valid value is as follows:

- *kmip-cert* - this is a Key Management Interoperability Protocol (KMIP) certificate

#### **Examples**

This example deletes a root-ca type digital certificate for a Vserver named vs0 in a company named *www.example.com* with serial number 4F57D3D1.

```
cluster1::> security certificate delete -vserver vs0 -common-name
www.example.com -ca www.example.com -type root-ca -serial 4F57D3D1
```

## **security certificate generate-csr**

### Generate a Digital Certificate Signing Request

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### **Description**

This command generates a digital certificate signing request and displays it on the console. A certificate signing request (CSR or certification request) is a message sent securely to a certificate authority (CA) via any electronic media, to apply for a digital identity certificate.

#### **Parameters**

##### **-common-name <FQDN or Custom Common Name> - FQDN or Custom Common Name**

This specifies the desired certificate name as a fully qualified domain name (FQDN) or custom common name or the name of a person. The supported characters, which are a subset of the ASCII character set, are as follows:

- Letters a through z, A through Z
- Numbers 0 through 9
- Asterisk (\*), period (.), underscore (\_) and hyphen (-)

The common name must not start or end with a "-" or a ".". The maximum length is 253 characters.

##### **[-size <size of requested certificate in bits>] - Size of Requested Certificate in Bits**

This specifies the number of bits in the private key. The higher the value, the more secure is the key. The default is 2048. Possible values include 512, 1024, 1536 and 2048.

#### **[-country <text>] - Country Name**

This specifies the country where the Vserver resides. The country name is a two-letter code. The default is US. Here is the list of country codes:

[Country Codes](#)

#### **[-state <text>] - State or Province Name**

This specifies the state or province where the Vserver resides.

#### **[-locality <text>] - Locality Name**

This specifies the locality where the Vserver resides. For example, the name of a city.

#### **[-organization <text>] - Organization Name**

This specifies the organization where the Vserver resides. For example, the name of a company.

#### **[-unit <text>] - Organization Unit**

This specifies the unit where the Vserver resides. For example, the name of a section or a department within a company.

#### **[-email-addr <mail address>] - Contact Administrator's Email Address**

This specifies the email address of the contact administrator for the Vserver.

#### **[-hash-function <hashing function>] - Hashing Function**

This specifies the cryptographic hashing function for signing the certificate. The default is SHA256. Possible values include *SHA1* , *SHA256* and *MD5* .

### **Examples**

This example creates a certificate-signing request with a 2048-bit private key generated by the SHA256 hashing function for use by the Software group in IT at a company whose custom common name is *www.example.com* , located in Sunnyvale, California, USA. The email address of the contact administrator who manages the Vserver is *web@example.com* .

```
cluster1::> security certificate generate-csr -common-name www.example.com
-size 2048 -country US -state California -locality Sunnyvale -organization
IT -unit Software
-email-addr web@example.com -hash-function SHA256
```

Certificate Signing Request :

-----BEGIN CERTIFICATE REQUEST-----

```
MIIBGjCBxQIBADBgMRQwEgYDVQQDEwtleGftcGx1LmNvbTELMAkGA1UEBhMCVVMx
CTAHBgNVBAgTADEJMAcGA1UEBxMAMQkwBwYDVQQKEwAxCTAHBgNVBAsTADEPMA0G
CSqGSIB3DQEJARYAMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAPXFanNoJApT1nzS
xOcxixqImRRGZCR7tVmTYyqPSuTvfhVtwDJbmXuj6U3a1woUsb13wfEvQnHVFNci
2ninsJ8CAwEAAaAAMA0GCSqGSIB3DQEBCwUAA0EA6EagLfs05+4g+ejirKKTUPQO
UqOUEoKuvxhOvPC2w7b//fNSFsFHvXloqEOhYEcN/NX9h8mbphCoM5YZ4OfnKw==
```

-----END CERTIFICATE REQUEST-----

Private Key :

-----BEGIN RSA PRIVATE KEY-----

```
MIIBAwIBAAJBAPXFanNoJApT1nzSxOcxixqImRRGZCR7tVmTYyqPSuTvfhVtwDJb
mXuj6U3a1woUsb13wfEvQnHVFNci2ninsJ8CAwEAAQJAWt2AO+bW3FKezEuIrQlu
KoMyRYK455wtMk8BrOyJfhYsB20B28eifjJvRWdTOBEav99M7cEzgPv+p5kaZTTM
gQIhAPsp+j1hrUXSRj979LIJY0sNez397i7ViFXWQScx/ehAiEA+oDbOooWlVvu
xj4aitxVBu6ByVckYU8LbsfeRNsZwD8CIQCbZ1/ENvmlJ/P7N9Exj2NCtEYxd0Q5
cwBZ5NfZeMBpwQIhAPk0KWQSLadGfsKO077itF+h9FGFNHbtuNTrVq4vPW3nAiAA
peMBQgEv28y2r8D4dkYzxcXmjzJluUSZSZ9c/wS6fA==
```

-----END RSA PRIVATE KEY-----

Note: Please keep a copy of your certificate request and private key for future reference.

## security certificate install

### Install a Digital Certificate

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The *security certificate install* command installs digital security certificates signed by a certificate authority (CA) and the public key certificate of the root CA. Digital security certificates also include the intermediate certificates to construct the chain for server certificates (the *server* type), client-side root CA certificates (the *client-ca* type), or server-side root CA certificates (the *server-ca* type). with FIPS enabled, the following restrictions apply to the certificate getting installed. *server/client/server-ca/client-ca*: Key size >= 2048, *server/client*: Hash function (No MD-5, No SHA-1), *server-ca/client-ca*: (Intermediate CA), Hash Function (No MD-5, No SHA-1), *server-ca/client-ca*: (Root CA), Hash Function (No MD-5)

## Parameters

### **-vserver <Vserver Name> - Name of Vserver**

This specifies the Vserver that contains the certificate.

### **-type <type of certificate> - Type of Certificate**

This specifies the certificate type. Valid values are the following:

- *server* - includes server certificates and intermediate certificates.
- *client-ca* - includes the public key certificate for the root CA of the SSL client
- *server-ca* - includes the public key certificate for the root CA of the SSL server to which Data ONTAP is a client
- *client* - includes a self-signed or CA-signed digital certificate and private key to be used for Data ONTAP as an SSL client

### **[-subtype <kmip-cert>] - Certificate Subtype**

This specifies a certificate subtype. This optional parameter can have an empty value (the default). The only valid value is as follows:

- *kmip-cert* - this is a Key Management Interoperability Protocol (KMIP) certificate

### **[-kmip-server-ip <IP Address>] - (DEPRECATED)-IPv4 and IPv6 address**



This parameter is deprecated and might be removed in the future releases of Data ONTAP.

This parameter is applicable only to the ``\_kmip-cert\_`` subtype. It specifies the IP address of the KMIP server.

## Examples

This example installs a CA-signed certificate (along with intermediate certificates) for a Vserver named vs0.

```
cluster1::> security certificate install -vserver vs0 -type server
Please enter Certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIIB8TCCA ZugAwIBAwIBADANBgkqhkiG9w0BAQQFADBfMRMwEQYDVQQDEwpuZXRh
cHAuY29tMQswCQYDVQQGEwJVUzEJMAcGA1UECBMAMQkwBwYDVQQHEwAxCTAHBgNV
BAoTADEJMAcGA1UECxMAMQ8wDQYJKoZIhvcNAQkBFgAwHhcNMTAwNDI2MTk0OTI4
WhcNMTAwNTI2MTk0OTI4WjBfMRMwEQYDVQQDEwpuZXRhchAuY29tMQswCQYDVQQG
EwJVUzEJMAcGA1UECBMAMQkwBwYDVQQHEwAxCTAHBgNVBAoTADEJMAcGA1UECxMA
MQ8wDQYJKoZIhvcNAQkBFgAwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAyXrK2sry
-----END CERTIFICATE-----
Please enter Private Key: Press <Enter> when done
-----BEGIN RSA PRIVATE KEY-----
MIIBPAIBAAJBAM16ytrK8nQj82UsWeHOeT8gk0BPX+Y5MLycsUdXA7hXhumHNpvF
C61X2G32Sx8VEa1th94tx+vOEzq+UaqHlt0CAwEAAQJBAMZjDWlgmlm3qIr/n8VT
```

```
PFnnZnbVcXVM70tbUsgPKw+QCCh9dF1jmuQKeDr+wUMWkn1DeGrfhILpzfJGhrLJ
z7UCIQDr8d3g0G71UyX+BbFmo/N0uAKjS2cvUU+Y8a8pDxGLLwIhANqa99SuS18U
DiPvdaKTj6+EcGuXfcXz+G0rfgtZK8uzAiEAR1mnrfYC8KwE9k7A0ylRzBLdUwK9
AvuJDn+/z+H1Bd0CIQDD93P/xpaJETNz53Au49VE5Jba/Jugckrbosd/1Sd7nQIg
aEMAz6qHHT4mndi8Bo8sDGedG2SKx6Qbn2IpuNZ7rc=
-----END RSA PRIVATE KEY-----
```

```
Do you want to continue entering root and/or intermediate certificates
{y|n}: y
```

```
Please enter Intermediate Certificate: Press <Enter> when done
```

```
-----BEGIN CERTIFICATE-----
```

```
MIIE+zCCBGSgAwIBAgICAQ0wDQYJKoZIhvcNAQEFBQAwgbsxJDAiBgnVBAcTG1Zh
bG1DZXJ0IFZhbg1kYXRpb24gTmV0d29yazEXMBUGA1UEChMOVmFsaUN1cnQsIElu
Yy4xNTAzBgnVBAstLFZhbg1DZXJ0IENSYXNzIDIgUG9saWN5IFZhbg1kYXRpb24g
QXV0aG9yaXR5MSEwHwYDVQQDExhodHRwOi8vd3d3LnZhbg1jZXJ0LmNvbS8xIDAe
BgkqhkiG9w0BCQEWEWluZm9AdmFsaWN1cnQuY29tMB4XDTA0MDYyOTE3MDYyMFoX
DTI0MDYyOTE3MDYyMFowYzELMAkGA1UEBhMCVVMxITAfBgNVBAoTGFRoZSBHbyBE
YWRkeSBHcm91cCwgSW5jLjExMC8GA1UECxMoR28gRGFkZHkgQ2xhc3MgMiBDZXJ0
-----END CERTIFICATE-----
```

```
Do you want to continue entering root and/or intermediate certificates
{y|n}: y
```

```
Please enter Intermediate Certificate: Press <Enter> when done
```

```
-----BEGIN CERTIFICATE-----
```

```
MIIC5zCCAlACAQEWdQYJKoZIhvcNAQEFBQAwgbsxJDAiBgnVBAcTG1Zhbg1DZXJ0
IFZhbg1kYXRpb24gTmV0d29yazEXMBUGA1UEChMOVmFsaUN1cnQsIEluYy4xNTAz
BgnVBAstLFZhbg1DZXJ0IENSYXNzIDIgUG9saWN5IFZhbg1kYXRpb24gQXV0aG9y
aXR5MSEwHwYDVQQDExhodHRwOi8vd3d3LnZhbg1jZXJ0LmNvbS8xIDAeBgkqhkiG
9w0BCQEWEWluZm9AdmFsaWN1cnQuY29tMB4XDTk5MDYyNjAwMTk1NFoXDTE5MDYy
NjAwMTk1NFowgbsxJDAiBgnVBAcTG1Zhbg1DZXJ0IFZhbg1kYXRpb24gTmV0d29y
azEXMBUGA1UEChMOVmFsaUN1cnQsIEluYy4xNTAzBgnVBAstLFZhbg1DZXJ0IENS
YXNzIDIgUG9saWN5IFZhbg1kYXRpb24gQXV0aG9yaXR5MSEwHwYDVQQDExhodHRw
-----END CERTIFICATE-----
```

```
Do you want to continue entering root and/or intermediate certificates
{y|n}: n
```

```
You should keep a copy of the private key and the CA-signed digital
certificate
for future reference.
```

This example installs a CA certificate for client authentication for a Vserver named vs0.

```
cluster1::> security certificate install -vserver vs0 -type client-ca
```

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----

```
MIIDNjCCAp+gAwIBAgIQNhIilsXjOKUgodJfTNCJVDANBgkqhkiG9w0BAQUFADCB
zjELMAkGA1UEBhMCWkExFTATBgnVBAgTDFd1c3R1cm4gQ2FwZTESMBAGA1UEBxMJ
Q2FwZSBUb3duMR0wGwYDVQQKExRUaGF3dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UE
CxMfQ2VydGlmaWNhdGlvbiBTZXJ2aN1cyBEaXZpc21vbjEhMB8GA1UEAxMYVGhh
d3R1IFByZW1pdW0gU2VydmVyIENBMSgwJgYJKoZIhvcNAQkBFh1wcmVtaXvtLXN1
cnZlckB0aGF3dGUuY29tMB4XDTk2MDgwMTAwMDAwMFoXDTIxMDEwMTIzNTk1OVow
gc4xCzAJBgnVBAYTAlpBMRUwEwYDVQQIEwxZXN0ZXJuIENhcGUxEjAQBgnVBAct
```

-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for future reference.

This example installs a CA certificate for server authentication for a Vserver named vs0. In this case, Data ONTAP acts as an SSL client.

```
cluster1::> security certificate install -vserver vs0 -type server-ca
```

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----

```
MIIDNjCCAp+gAwIBAgIQNhIilsXjOKUgodJfTNCJVDANBgkqhkiG9w0BAQUFADCB
zjELMAkGA1UEBhMCWkExFTATBgnVBAgTDFd1c3R1cm4gQ2FwZTESMBAGA1UEBxMJ
Q2FwZSBUb3duMR0wGwYDVQQKExRUaGF3dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UE
CxMfQ2VydGlmaWNhdGlvbiBTZXJ2aN1cyBEaXZpc21vbjEhMB8GA1UEAxMYVGhh
d3R1IFByZW1pdW0gU2VydmVyIENBMSgwJgYJKoZIhvcNAQkBFh1wcmVtaXvtLXN1
cnZlckB0aGF3dGUuY29tMB4XDTk2MDgwMTAwMDAwMFoXDTIxMDEwMTIzNTk1OVow
gc4xCzAJBgnVBAYTAlpBMRUwEwYDVQQIEwxZXN0ZXJuIENhcGUxEjAQBgnVBAct
```

-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for future reference.

## security certificate show

### Display Installed Digital Certificates

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

This command displays information about the installed digital certificates. Some details are displayed only when you use the command with the *-instance* parameter.

## 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 ] }

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

**[-vserver <Vserver Name>]** - Name of Vserver

Selects the Vserver whose digital certificates you want to display.

**[-common-name <FQDN or Custom Common Name>]** - FQDN or Custom Common Name

Selects the certificates that match this parameter value.

**[-serial <text>]** - Serial Number of Certificate

Selects the certificates that match this parameter value.

**[-ca <text>]** - Certificate Authority

Selects the certificates that match this parameter value.

**[-type <type of certificate>]** - Type of Certificate

Selects the certificates that match this parameter value.

**[-subtype <kmip-cert>]** - Certificate Subtype

Selects the certificate subtype that matches the specified value. The valid values are as follows:

- *kmip-cert* - this is a Key Management Interoperability Protocol (KMIP) certificate

**[-size <size of requested certificate in bits>]** - Size of Requested Certificate in Bits

Selects the certificates that match this parameter value.

**[-start <Date>]** - Certificate Start Date

Selects the certificates that match this parameter value.

**[-expiration <Date>]** - Certificate Expiration Date

Selects the certificates that match this parameter value.

**[-public-cert <certificate>]** - Public Key Certificate

Selects the certificates that match this parameter value.

**[-country <text>]** - Country Name

Selects the certificates that match this parameter value.

**[-state <text>]** - State or Province Name

Selects the certificates that match this parameter value.

**[-locality <text>]** - Locality Name

Selects the certificates that match this parameter value.

**[-organization <text>] - Organization Name**

Selects the certificates that match this parameter value.

**[-unit <text>] - Organization Unit**

Selects the certificates that match this parameter value.

**[-email-addr <mail address>] - Contact Administrator's Email Address**

Selects the certificates that match this parameter value.

**[-protocol <protocol>] - Protocol**

Selects the certificates that match this parameter value.

**[-hash-function <hashing function>] - Hashing Function**

Selects the certificates that match this parameter value.

**[-self-signed {true|false}] - Self-Signed Certificate**

Selects the certificates that match this parameter value.

## Examples

The examples below display information about digital certificates.

```
cluster1::> security certificate show

Vserver      Serial Number      Common Name          Type
-----  -----  -----  -----
-----  -----
vs0          4F4E4D7B          www.example.com
server
  Certificate Authority: www.example.com
  Expiration Date: Thu Feb 28 16:08:28 2013
```

```

cluster1::> security certificate show -instance
                           Vserver: vs0
                           FQDN or Custom Common Name: www.example.com
                           Serial Number of Certificate: 4F4E4D7B
                           Certificate Authority: www.example.com
                           Type of Certificate: server
                           Size of Requested Certificate(bits): 2048
                           Certificate Start Date: Fri Apr 30 14:14:46 2010
                           Certificate Expiration Date: Sat Apr 30 14:14:46 2011
                           Public Key Certificate: -----BEGIN CERTIFICATE-----

MIIDftCCAmWgAwIBAwIBADANBgkqhkiG9w0BAQsFADBgMRQwEgYDVQQDEwtsYWIu
YWJjLmNvbTELMAkGAlUEBhMCVVMxCTAHBgNVBAgTADEJMAcGAlUEBxMAMQkwBwYD
VQQKEwAxCTAHBgNVBAsTADEPMA0GCSqGSIb3DQEJARYAMB4XDTEwMDQzMDE4MTQ0
BgNVHQ8BAf8EBAMCAQYwHQYDVR0OBBYEFCVG7dYGe51akE14ecaCdL+LOAxUMA0G
CSqGSIb3DQEBCwUAA4IBAQBJ1E51pkDY3ZpsSrQeMOoWLteIR+1H0wKZOM1Bhy6Q
+gsE3XEtN07AE4npjIT0eVP0nI9QIJAbP0uPKaCGAVBSBMoM2mOwbfsWI7aJoEh
+XuEoNr0GOz+mltnfhgv11fT6Ms+xzd3LGZYQTworus2
                           -----END CERTIFICATE-----
                           Country Name (2 letter code): US
                           State or Province Name (full name): California
                           Locality Name (e.g. city): Sunnyvale
                           Organization Name (e.g. company): example
                           Organization Unit (e.g. section): IT
                           Email Address (Contact Name): web@example.com
                           Protocol: SSL
                           Hashing Function: SHA256

```

## security certificate sign

Sign a Digital Certificate using Self-Signed Root CA

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

This command signs a digital certificate signing request and generates a certificate using a Self-Signed Root CA certificate in either PEM or PKCS12 format. You can use the [security certificate generate-csr](#) command to generate a digital certificate signing request.

## Parameters

### **-vserver <Vserver Name>** - Name of Vserver

This specifies the name of the Vserver on which the signed certificate will exist.

### **-ca <text>** - Certificate Authority to Sign

This specifies the name of the Certificate Authority that will sign the certificate.

### **-ca-serial <text>** - Serial Number of CA Certificate

This specifies the serial number of the Certificate Authority that will sign the certificate.

### **[-expire-days <integer>]** - Number of Days until Expiration

This specifies the number of days until the signed certificate expires. The default value is 365 days.

Possible values are between 1 and 3652 .

### **[-format <certificate format>]** - Certificate Format

This specifies the format of signed certificate. The default value is PEM. Possible values include *PEM* and *PKCS12* .

### **[-destination { (ftp|http)://(hostname|IPv4 Address|['IPv6 Address'])... }]** - Where to Send File

This specifies the destination to upload the signed certificate. This option can only be used when the format is PKCS12.

### **[-hash-function <hashing function>]** - Hashing Function

This specifies the cryptographic hashing function for the self-signed certificate. The default value is SHA256. Possible values include *SHA1* , *SHA256* and *MD5* .

## Examples

This example signs a digital certificate for a Vserver named vs0 using a Certificate Authority certificate that has a ca of *www.ca.com* and a ca-serial of 4F4EB629 in PEM format using the SHA256 hashing function.

```

cluster1::> security certificate sign -vserver vs0 -ca www.ca.com -ca
-serial 4F4EB629 -expire-days 36 -format PEM -hash-function SHA256

Please enter Certificate Signing Request(CSR): Press <Enter> when done
-----BEGIN CERTIFICATE REQUEST-----
MIIBGjCBxQIBADBgMRQwEgYDVQQDEwtleGFTcGx1LmNvbTELMAkGA1UEBhMCVVMx
CTAHBgNVBAgTADEJMAcGA1UEBxMAMQkwBwYDVQQKEwAxCTAHBgNVBAsTADEPMA0G
CSqGSIB3DQEJARYAMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAPXFanNoJApT1nzS
xOcxixqImRRGZCR7tVmTYyqPSuTvfhVtwDJbmXuj6U3a1woUsb13wfEvQnHVFNci
2ninsJ8CAwEAAaAAMA0GCSqGSIB3DQEBCwUAA0EA6EagLfso5+4g+ejirRKKTUPQO
UqOUEoKuvxhOvPC2w7b//fNSFsFHvXloqEOhYECn/NX9h8mbphCoM5YZ4OfnKw==
-----END CERTIFICATE REQUEST-----
Signed Certificate: :
-----BEGIN CERTIFICATE-----
MIICwDCCAAigAwIBAgIET1oskDANBgkqhkiG9w0BAQsFADBdMREwDwYDVQQDEwh2
czAuY2VydDELMAkGA1UEBhMCVVMxCTAHBgNVBAgTADEJMAcGA1UEBxMAMQkwBwYD
VQQKEwAxCTAHBgNVBAsTADEPMA0GCSqGSIB3DQEJARYAMB4XDTEyMDMwOTE2MTUX
M1oXDTEyMDQxNDE2MTUxM1owYDEUMBIGA1UEAxMLZXhhbXBsZS5jb20xCzAJBgNV
BAYTA1VTMQkwBwYDVQQIEwAxCTAHBgNVBAcTADEJMAcGA1UEChMAMQkwBwYDVQQL
EwAxDzANBgkqhkiG9w0BCQEWAADBcMA0GCSqGSIB3DQEBAQUAA0sAMEgCQQD1xWpz
-----END CERTIFICATE-----

```

This example signs and exports a digital certificate to destination <ftp://10.98.1.1/u/sam/sign.pfx> for a Vserver named vs0 using a Certificate Authority certificate that expires in 36 days and has a ca value of `www.ca.com` and a ca-serial value of `4F4EB629` in PKCS12 format by the MD5 hashing function.

```

cluster1::> security certificate sign -vserver vs0 -ca www.ca.com -ca
-serial 4F4EB629
-expire-days 36 -format PKCS12 -destination
ftp://10.98.1.1//u/sam/sign.pfx -hash-function MD5

Please enter Certificate Signing Request (CSR): Press <Enter> when done
-----BEGIN CERTIFICATE REQUEST-----
MIIBGjCBxQIBADBgMRQwEgYDVQQDEwtleGFTcGx1LmNvbTELMAkGA1UEBhMCVVMx
CTAHBgNVBAgTADEJMAcGA1UEBxMAMQkwBwYDVQQKEwAxCTAHBgNVBAsTADEPMA0G
CSqGSIB3DQEJARYAMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAPXFanNoJApT1nzS
xOcxixqImRRGZCR7tVmTYyqPSuTvfhVtwDJbmXuj6U3a1woUsb13wfEvQnHVFNci
2ninsJ8CAwEAAaAAMA0GCSqGSIB3DQEBCwUAA0EA6EagLfso5+4g+ejirKKTUPQO
UqOUEoKuvxhOvPC2w7b//fNSFsFHvXloqEOhYECn/NX9h8mbphCoM5YZ4OfnKw==
-----END CERTIFICATE REQUEST-----
Signed Certificate: :
-----BEGIN CERTIFICATE-----
MIICwDCCAAigAwIBAgIET1ot8jANBgkqhkiG9w0BAQsFADBdMREwDwYDVQQDEwh2
czAuY2VydDELMAkGA1UEBhMCVVMxCTAHBgNVBAgTADEJMAcGA1UEBxMAMQkwBwYD
VQQKEwAxCTAHBgNVBAsTADEPMA0GCSqGSIB3DQEJARYAMB4XDTEyMDMwOTE2MjEw
N1oXDTEyMDQxNDE2MjEwN1oWYDEUMBIGA1UEAxMLZXhhBxBsZS5jb20xCzAJBgNV
BAYTA1VTMQkwBwYDVQQIEwAxCTAHBgNVBAcTADEJMAcGA1UEChMAMQkwBwYDVQQL
EwAxDzANBgkqhkiG9w0BCQEWAADBcMA0GCSqGSIB3DQEBAQUAA0sAMEgCQQD1xWpz
oarXHSyDzv3T5QIxBGRJ0ActgdjJuqtuAdmnKvKfLS1o4C90
-----END CERTIFICATE-----
Please enter Private Key: Press <Enter> when done
-----BEGIN RSA PRIVATE KEY-----
MIIBAwIBAAJBAPXFanNoJApT1nzSxOcxixqImRRGZCR7tVmTYyqPSuTvfhVtwDJb
mXuj6U3a1woUsb13wfEvQnHVFNci2ninsJ8CAwEAAQJAWt2AO+bW3FKezEuIrQlu
KoMyRYK455wtMk8BrOyJfhYsB20B28eifjJvRWdTOBEav99M7cEzgPv+p5kaZTTM
gQIhAPsp+j1hrUXSRj979LIJJY0sNez397i7ViFXWQScx/ehAiEA+oDbOooWlVvu
xj4aitxVBu6ByVckYU8LbsfeRNsZwD8CIQCbZ1/ENvmlJ/P7N9Exj2NCtEYxd0Q5
cwBZ5NfZeMBpwQIhApk0KWQSLadGfsKO077itF+h9FGFNHbtuNTrVq4vPW3nAiAA
peMBQgEv28y2r8D4dkYzxcXmjzJluUSZSZ9c/wS6fA==
-----END RSA PRIVATE KEY-----
Please enter a password for pkcs12 file:
Please enter it again:

Enter User for Destination URI: sam
Enter Password:

```

## Related Links

- [security certificate generate-csr](#)

## security certificate ca-issued revoke

### Revoke a Digital Certificate

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

This command revokes a digital certificate signed by a Self-Signed Root CA.

#### Parameters

##### **-vserver <Vserver Name> - Vserver**

This specifies the name of the Vserver on which the certificate is stored.

##### **-serial <text> - Serial Number of Certificate**

This specifies the serial number of the certificate.

##### **-ca <text> - Certificate Authority**

This specifies the name of the Certificate Authority whose certificate will be revoked.

##### **-ca-serial <text> - Serial Number of CA Certificate**

This specifies the serial number of Certificate Authority.

##### **[-common-name <FQDN or Custom Common Name>] - FQDN or Custom Common Name**

This specifies a fully qualified domain name (FQDN) or custom common name or the name of a person.

This field is optional if ca-serial is specified.

#### Examples

This example revokes a signed digital certificate for a Vserver named vs0 with serial as 4F5A2DF2 for a Certificate Authority certificate that has a ca of [www.ca.com](http://www.ca.com) and a ca-serial of 4F4EB629.

```
cluster1::> security certificate ca-issued revoke -vserver vs0 -serial  
4F5A2DF2 -ca www.ca.com -ca-serial 4F4EB629
```

## security certificate ca-issued show

### Display CA-Issued Digital Certificates

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

This command displays the following information about the digital certificates issued by the self-signed root-ca:

- Vserver
- Serial number of certificate
- FQDN or custom common name or the name of a person

- Serial number of CA certificate
- Status (active, revoked )
- Certificate Authority
- Expiration date
- Revocation date

To display more details, run the command with the `-instance` parameter. This will add the following information:

- Country name
- State or province name
- Locality name
- Organization name
- Organization unit
- Contact administrator's email address

## 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 ] }**

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

**[-vserver <Vserver Name>] - Vserver**

Selects the certificates that match this parameter value.

**[-serial <text>] - Serial Number of Certificate**

Selects the certificates that match this parameter value.

**[-ca <text>] - Certificate Authority**

Selects the certificates that match this parameter value.

**[-ca-serial <text>] - Serial Number of CA Certificate**

Selects the certificates that match this parameter value.

**[-common-name <FQDN or Custom Common Name>] - FQDN or Custom Common Name**

Selects the certificates that match this parameter value.

**[-status <status of certificate>] - Status of Certificate**

Selects the certificates that match this parameter value. Possible values include active and revoked.

**[-expiration <Date>] - Certificate Expiration Date**

Selects the certificates that match this parameter value.

**[-revocation <Date>] - Certificate Revocation Date**

Selects the certificates that match this parameter value.

**[-country <text>] - Country Name (2 letter code)**

Selects the certificates that match this parameter value.

**[-state <text>] - State or Province Name (full name)**

Selects the certificates that match this parameter value.

**[-locality <text>] - Locality Name (e.g. city)**

Selects the certificates that match this parameter value.

**[-organization <text>] - Organization Name (e.g. company)**

Selects the certificates that match this parameter value.

**[-unit <text>] - Organization Unit (e.g. section)**

Selects the certificates that match this parameter value.

**[-email-addr <mail address>] - Email Address (Contact Name)**

Selects the certificates that match this parameter value.

## Examples

The examples below display information about CA issued digital certificates.

```
cluster1::> security certificate ca-issued show
Serial Number of
Vserver      Serial Number      Common Name          CA's Certificate
Status
-----
-----
vs0          4F5A2C90          example.com          4F4EB629
active
    Certificate Authority: vs0.cert
    Expiration Date: Sat Apr 14 16:15:13 2012
    Revocation Date: -
vs0          4F5A2DF2          example.com          4F4EB629
revoked
    Certificate Authority: vs0.cert
    Expiration Date: Sat Apr 14 16:21:06 2012
    Revocation Date: Fri Mar 09 17:08:30 2012
2 entries were displayed.
```

```

cluster1::> security certificate ca-issued show -instance
Vserver: vs0
    Serial Number of Certificate: 4F5A2C90
    Certificate Authority: vs0.cert
    Serial Number of CA Certificate: 4F4EB629
    FQDN or Custom Common Name: example.com
    Status of Certificate: active
    Certificate Expiration Date: Sat Apr 14 16:15:13 2012
    Certificate Revocation Date: -
    Country Name (2 letter code): US
    State or Province Name (full name): California
    Locality Name (e.g. city): Sunnyvale
    Organization Name (e.g. company): example
    Organization Unit (e.g. section): IT
    Email Address (Contact Name): web@example.com

```

## 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 TLSv1.2 and TLSv1.1). 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 Data ONTAP supports the TLSv1.2, TLSv1.1 and TLSv1 protocols. For backward compatibility, Data ONTAP supports adding SSLv3 to the supported-protocols list when FIPS mode is disabled. Use the `-supported-ciphers` parameter to configure only AES, or AES and 3DES, or disable weak ciphers such as RC4 by specifying `!RC4`. By default the supported-cipher setting is `ALL:!LOW:!aNULL:!EXP:!eNULL`. This setting means that all supported cipher suites for the protocols are enabled, except the ones with no authentication, no encryption, no exports, and low encryption cipher suites (currently those using 64-bit or 56-bit encryption algorithms). Select a cipher suite which is available with the corresponding selected protocol. An invalid configuration may cause some functionality to fail to operate properly. Refer to "<https://www.openssl.org/docs/apps/ciphers.html>" published by the OpenSSL software foundation for the correct cipher string syntax. After modifying the security configuration, reboot all the nodes manually.

#### Parameters

##### **-interface <SSL> - FIPS-Compliant Interface (privilege: advanced)**

Selects the FIPS-compliant interface. Default is `SSI`.

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

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

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

Selects the supported protocols for the selected interface. Default is *TLSv1.2, TLSv1.1, TLSv1*

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

Selects the supported cipher suites for the selected interface. Default is

*ALL:!LOW:!aNULL:!EXP:!eNULL*.

## Examples

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

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

The following command modifies supported protocols to *TLSv1.2* and *TLSv1.1* in the cluster. (Default setting for supported protocols is *TLSv1.2, TLSv1.1, TLSv1*)

```
cluster1::*> security config modify -interface SSL -supported-protocols
TLSv1.2, TLSv1.1
```

The following command modifies supported ciphers to *ALL:!LOW:!aNULL:!EXP:!eNULL:!RC4* in the cluster. (Default setting for supported ciphers is *ALL:!LOW:!aNULL:!EXP:!eNULL*)

```
cluster1::*> security config modify -interface SSL -supported-ciphers
ALL:!LOW:!aNULL:!EXP:!eNULL:!RC4
```

## 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: *TLSv1.2, TLSv1.1, TLSv1*
- Supported ciphers: *ALL:!LOW:!aNULL:!EXP:!eNULL*

The default cipher suites represent all suites for the listed protocols except those that have no authentication, no encryption, no exports, and low encryption (below 64 or 56 bit).

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 ciphers that the cluster will support. When all the nodes in the cluster are updated with the modified settings, the cluster security config ready value will be shown as yes .

## 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 ] }

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

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

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 {TLSv1.2|TLSv1.1|TLSv1|SSLv3}] - Supported Protocols (privilege: advanced)**

Displays configurations that match the specified protocols.

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

Displays the configurations that match the specified supported ciphers.

## Examples

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

```
cluster1::> security config show
          Cluster          Cluster
          Security
          Interface FIPS Mode  Supported Protocols      Supported Ciphers Config
          Ready
          -----
          -----
          SSL      false      TLSv1.2, TLSv1.1, TLSv1  ALL:!LOW:      yes
          !aNULL:!EXP:
          !eNULL
```

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

```

cluster1::> security config show
      Cluster                               Cluster
  Security
Interface FIPS Mode  Supported Protocols      Supported Ciphers Config
Ready
-----
-----
SSL      true      TLSv1.2, TLSv1.1      ALL:!LOW:
                                         !aNULL:!EXP:
                                         !eNULL:!RC4
                                         yes

```

## 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 <Application supporting SSL/TLS protocol>,... - 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
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
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 <Application supporting SSL/TLS protocol>,... - 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
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
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 ] }

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

[-application <Application supporting SSL/TLS protocol>] - 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)

[-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
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

### Display Security Configuration Status

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

#### Description

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 [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 ] }**

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](#)

## **security key-manager commands**

### **security key-manager add**

Add a key management server

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

#### **Description**

This command adds a key management server at the indicated IP address to its list of four possible active key management servers. The command fails if there are already four key management servers configured. This command is not supported when onboard key management is enabled.

## Parameters

### **-address <IP Address> - IP Address**

This parameter specifies the IP address of the key management server you want to use to store keys.

### **[-server-port <integer>] - Server TCP Port**

This parameter specifies the TCP port on which the key management server will listen for incoming connections.

## Examples

The following example adds the key management server with address 10.233.1.98, listening for incoming connections on the default TCP port 5696, to the list of key management servers used by the external key manager:

```
cluster-1::> security key-manager add -address 10.233.1.198
```

The following example adds the key management server with address 10.233.1.98, listening for incoming connections on TCP port 15696, to the list of key management servers used by the external key manager:

```
cluster-1::> security key-manager add -address 10.233.1.198 -server-port  
15696
```

## security key-manager create-key

### Create a new authentication key

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

## Description

This command creates a new authentication key (AK) and stores it on the configured key management servers. The command fails if the configured key management servers are already storing more than 128 AKs. If command fails due to more than 128 keys in cluster, delete unused keys on your key management servers and try the command again. This command is not supported when onboard key management is enabled.

## Parameters

### **[-key-tag <text>] - Key Tag**

This parameter specifies the key tag that you want to associate with the new authentication key (AK). The default value is the node name. This parameter can be used to help identify created authentication keys (AKs). For example, the key-manager query command key-tag parameter can be used to query for a specific key-tag value.

### **[-prompt-for-key {true|false}] - Prompt for Authentication Passphrase**

If you specify this parameter as true, the command prompts you to enter an authentication passphrase manually instead of generating it automatically. For security reasons, the authentication passphrase you entered is not displayed at the command prompt. You must enter the authentication passphrase a second time for verification. To avoid errors, copy and paste authentication passphrases electronically instead of

entering them manually. Data ONTAP saves the resulting authentication key/key ID pair automatically on the configured key management servers.

## Examples

The following example creates an authentication key with the node name as the default key-tag value:

```
cluster-1::> security key-manager create-key

Verifying requirements...

Node: node1
Creating authentication key...
Authentication key creation successful.
Key ID: 000000000000000020000000000100D0F7C2462D626B739FE81B89F29A092F.

Node: node2
Key manager restore operation initialized.
Successfully restored key information.
```

The following example creates an authentication key with key-tag "disk1-key":

```
cluster-1::> security key-manager create-key -key-tag disk1-key

Verifying requirements...

Node: node1
Creating authentication key...
Authentication key creation successful.
Key ID: 000000000000000020000000000100B8297A6189BC24B9B84C1916ED576857.

Node: node2
Key manager restore operation initialized.
Successfully restored key information.
```

The following example creates an authentication key with a user-specified authentication passphrase:

```
cluster-1::> security key-manager create-key -prompt-for-key true

Enter a new passphrase::

Reenter the passphrase::

Verifying requirements...

Node: node1
Creating authentication key...
Authentication key creation successful.
Key ID: 00000000000000002000000000001006268333F870860128FBE17D393E5083B.

Node: node2
Key manager restore operation initialized.
Successfully restored key information.
```

## security key-manager delete-key-database

Deletes the key hierarchy for onboard key manager

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

### Description

The `security key-manager delete-key-database` command permanently deletes the onboard key-management configuration from all nodes of the cluster.

### Examples

The following example deletes the onboard key-management configuration from all nodes of the cluster:

```
cluster-1::*> security key-manager delete-key-database

Warning: This command will permanently delete all keys from onboard key
management.
Do you want to continue? {y|n}: y
```

## security key-manager delete-kmip-config

Deletes the KMIP configuration

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

## Description

The `security key-manager delete-kmip-config` command permanently deletes the Key Management Interoperability Protocol (KMIP) server configuration from all nodes of the cluster.



The keys stored by the external KMIP servers cannot be deleted by Data ONTAP, and must be deleted by using external tools.

## Examples

The following example deletes the KMIP-server configuration from all nodes of the cluster:

```
cluster-1::*> security key-manager delete-kmip-config

Warning: This command will permanently delete the KMIP-server
configuration
from all nodes of the cluster.
Do you want to continue? {y|n}: y
The KMIP-server configuration has been successfully deleted from all nodes
of the
cluster. The keys stored by the external KMIP servers cannot be deleted by
Data ONTAP,
and must be deleted by using external tools.
```

## security key-manager delete

### Delete a key management server

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

#### Description

This command removes the key management server at the indicated IP address from the list of active key management servers. If the indicated key management server is the sole storage location for any key that is in use by Data ONTAP, you will be unable to remove the key server. This command is not supported when onboard key management is enabled.

#### Parameters

##### **-address <IP Address> - IP Address**

This parameter specifies the IP address of the key management server you want to remove from use.

## Examples

The following example removes the key server at IP address 10.233.1.198 from the set of configured key management servers:

```
cluster-1::> security key-manager delete -address 10.233.1.198
```

## security key-manager prepare-to-downgrade

Disables onboard keymanagement features for unsupported versions

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

### Description

The security key-manager prepare-to-downgrade command disables the onboard key management features that are not supported in releases prior to ONTAP 9.1.0. The features that are disabled are onboard key management support for Metrocluster configurations, and Volume Encryption (VE).

### Examples

The following example disables the onboard key management support for Metrocluster configurations and Volume Encryption (VE):

```
cluster1::*> security key-manager prepare-to-downgrade
```

## security key-manager query

Displays the key IDs stored in a key management server.

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

### Description

This command displays the IDs of the keys that are stored on the key management servers. This command does not update the key tables on the node. To refresh the key tables on the nodes with the key management server key tables, run the [security key-manager restore](#) command. This command is not supported when onboard key management is enabled.

### 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 ] }**

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

**[-node {<nodename>|local}] - Node**

This parameter specifies the name of the node that queries the specified key management servers. If this parameter is not specified, then all nodes will query the specified key management servers.

**[-address <IP Address>] - IP Address**

This parameter specifies the IP address of the key management server that you want to query.

**[-key-id <text>] - Key ID**

If you specify this parameter, then the command displays only the key IDs that match the specified value.

**[-key-tag <text>] - Key Tag**

If you specify this parameter, then the command displays only the key IDs that match the specified value. The key-tag for Volume Encryption Keys (VEKs) is set to the UUID of the encrypted volume.

**[-key-type <Key Usage Type>] - Key Type**

If you specify this parameter, then the command displays only the key IDs that match the specified value.

**[-count <integer>] - (DEPRECATED)-Key Server's Total Key Count**

The value *count* is deprecated and may be removed in a future release of Data ONTAP. This parameter specifies the total number of keys stored in the key management servers. If you specify this parameter, then the command displays only the key IDs retrieved from the key management servers whose total key count matches the specified count number.

**[-restored {yes|no}] - Key/Key ID Pair Present in Node's Key Table?**

This parameter specifies whether the key corresponding to the displayed key ID is present in the specified node's internal key table. If you specify 'yes' for this parameter, then the command displays the key IDs of only those keys that are present in the system's internal key table. If you specify 'no' for this parameter, then the command displays the key IDs of only those keys that are not present in the system's internal key table.

**[-key-manager-server-status {available|not-responding|unknown}] - Command Error Code**

This parameter specifies the connectivity status of the key management server. If you specify this parameter, then the command displays only the key IDs retrieved from the key management servers with specified status.

## Examples

The following example shows all the keys on all configured key servers, and whether those keys have been restored for all nodes in the cluster:

The following example shows all keys stored on the key server with address "10.0.0.10" from node "node1" with key-tag "node1":

```
cluster-1::> security key-manager query -address 10.0.0.10 -node node1
-key-tag node1
Node: node1
  Key Manager: 10.0.0.10
  Server Status: available

Key Tag          Key Type  Restored
-----
node1           NSE-AK    yes

  Key ID:
0000000000000000000020000000000001001d71f3b2468d7e16a6e6972d3e66452000000000000
000000

If any listed keys have "no" in the "Restored" column, run "security key-
manager
restore" to restore those keys.
```

The following example shows the Volume Encryption Key (VEK) with key-tag (i.e., volume UUID) "301a4e57-9efb-11e7-b2bc-0050569c227f" on nodes where that key has not been restored:

## Related Links

- security key-manager restore

## security key-manager restore

Restore the authentication key and key ID pairs from the key management servers.

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

## Description

This command retrieves and restores any current unrestored keys associated with the storage controller from the specified key management servers. This command is not supported when onboard key management is enabled.

## 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 ] }

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

**[-node {<nodename>|local}] - Node**

This parameter specifies the name of the node that is to load the key IDs into its internal key table. If not specified, all nodes retrieve keys into their internal key table.

**[-address <IP Address>] - IP Address**

If this parameter is specified, the command restores only from key management server at the specified IP address. If not specified the command restores from all available key management servers.

**[-key-tag <text>] - Key Tag**

This parameter specifies the value associated with the key ID pair at the time of their creation. If specified, restore only key ID pairs associated with the specified key tag. If not specified, all key ID pairs for the cluster are retrieved.

**[-key-ids <text>,...] - Authentication Key ID**

If this parameter is specified, the command restores only the specified key IDs.

**[-count <integer>] - AK/Key ID Pair Count**

The value *count* is deprecated and may be removed in a future release of Data ONTAP. This parameter specifies the total number of keys stored in the key management servers. If this parameter is specified, then the command displays only the key IDs retrieved from the key management servers whose total key count matches the specified count number.

**[-key-manager-server-status {available|not-responding|unknown}] - Command Error Code**

This parameter specifies the connectivity status of the key management server. If you specify this parameter the command displays only the key IDs retrieved from key management servers with specified status.

## Examples

The following command restores keys that are currently on a key server but are not stored within the key tables on the cluster:

The following loads any keys that exist on the key servers with IP address 10.0.0.10 with key-tag "node1" that are not currently stored in key tables of the nodes in the cluster. In this example, a key with that key-tag was missing from two nodes in the cluster:

```
cluster-1::> security key-manager restore -address 10.0.0.10 -key-tag
node1
Node: node1
    Key Manager: 10.0.0.10
    Server Status: available

Key IDs
-----
0000000000000000000020000000000001001d71f3b2468d7e16a6e6972d3e66452000000000000
000000
Node: node2
    Key Manager: 10.0.0.10
    Server Status: available

Key IDs
-----
0000000000000000000020000000000001001d71f3b2468d7e16a6e6972d3e66452000000000000
000000
```

## security key-manager setup

Configure key manager connectivity

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

### Description

The `security key-manager setup` command enables you to configure key management. Data ONTAP supports two mutually exclusive key management methods: external via one or more key management interoperability protocol (KMIP) servers, or internal via an onboard key manager. This command is used to configure an external or internal key manager. When configuring an external key management server, this command records networking information on all node that is used during the boot process to retrieve keys needed for booting from the KMIP servers. For onboard key management, this command prompts you to configure a passphrase to protect internal keys in encrypted form.

This command can also be used to refresh missing onboard keys. For example, if you add a node to a cluster that has onboard key management configured, you will run this command to refresh the missing keys.

For onboard key management in a MetroCluster configuration, if the [security key-manager update-passphrase](#) command is used to update the passphrase on one site, then run the `security key-manager setup` command with the new passphrase on the partner site before proceeding with any key-manager operations.

### Parameters

#### **[`-node <nodename>`] - Node Name**

This parameter is used only with onboard key management when a refresh operation is required (see command description). This parameter is ignored when configuring external key management and during the initial setup of onboard key management.

### Examples

The following example creates a configuration for external key management:

```
cluster-1::> security key-manager setup
Welcome to the key manager setup wizard, which will lead you through
the steps to add boot information.
```

Enter the following commands at any time  
"help" or "?" if you want to have a question clarified,  
"back" if you want to change your answers to previous questions, and  
"exit" if you want to quit the key manager setup wizard. Any changes  
you made before typing "exit" will be applied.

Restart the key manager setup wizard with "security key-manager setup". To  
accept a default or omit a question, do not enter a value.

```
Would you like to configure onboard key management? {yes, no} [yes]: no
Would you like to configure the KMIP server environment? {yes, no} [yes]:
yes
```

The following example creates a configuration for onboard key management:

```
cluster-1::> security key-manager setup
Welcome to the key manager setup wizard, which will lead you through
the steps to add boot information.
```

Enter the following commands at any time  
"help" or "?" if you want to have a question clarified,  
"back" if you want to change your answers to previous questions, and  
"exit" if you want to quit the key manager setup wizard. Any changes  
you made before typing "exit" will be applied.

Restart the key manager setup wizard with "security key-manager setup". To  
accept a default or omit a question, do not enter a value.

```
Would you like to configure onboard key management? {yes, no} [yes]: yes
Enter the cluster-wide passphrase for onboard key management. To continue
the
configuration, enter the passphrase, otherwise type "exit":
Re-enter the cluster-wide passphrase:
After configuring onboard key management, save the encrypted configuration
data
in a safe location so that you can use it if you need to perform a manual
recovery
operation. To view the data, use the "security key-manager backup show"
command.
```

## Related Links

- [security key-manager update-passphrase](#)

## security key-manager show

Display key management servers

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

### Description

This command displays the key management servers configured on the cluster. This command is not supported when onboard key management is enabled.

### 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.

**| [-status ]**

If you specify this parameter, the command displays the status of each key management server.

**| [-instance ] }**

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

**[-node {<nodename>|local}] - Node**

This parameter specifies the name of the node that you want to retrieve key management server status for. If parameter is not specified, all nodes will retrieve the key management servers status.

**[-address <IP Address>] - IP Address**

Shows only a key management server registered with the input address. It is also possible to show multiple key management servers.

**[-server-port <integer>] - Server TCP Port**

If you specify this parameter, the command displays only key servers listening on this port.

### Examples

The following example lists all configured key management servers:

```
cluster-1::> security key-manager show

Node           Registered Key Manager
-----
node1          10.225.89.33
node2          10.225.89.33
```

The following example lists all configured key management servers, the TCP port on which those servers are expected to listen for incoming KMIP connections, and their server status:

```
cluster-1::> security key-manager show -status

Node          Port  Registered Key Manager      Status
-----  -----  -----
node1        5696  10.225.89.33      available
node2        5696  10.225.89.33      available
```

## **security key-manager update-passphrase**

Update cluster-wide passphrase

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

### **Description**

The `security key-manager update-passphrase` command provides a way to update the cluster-wide passphrase, created initially by running the [security key-manager setup](#) command, that is used for onboard key management. This command prompts for the existing passphrase, and if that passphrase is correct then the command prompts for a new passphrase.

When the `security key-manager update-passphrase` command is executed in a MetroCluster configuration, then run the [security key-manager setup](#) command with the new passphrase on the partner site before proceeding with any key-manager operations. This allows the updated passphrase to be replicated to the partner site.

### **Examples**

The following example updates the cluster-wide passphrase used for onboard key management:

```
cluster-1::*> security key-manager update-passphrase

Warning: This command will reconfigure the cluster passphrase for onboard
key-management.
Do you want to continue? {y|n}: y

Enter current passphrase:

Enter new passphrase:

Reenter the new passphrase:
Update passphrase has completed. Save the new encrypted configuration data
in
a safe location so that you can use it if you need to perform a manual
recovery
operation. To view the data, use the "security key-manager backup show"
command.
```

## Related Links

- [security key-manager setup](#)

## security key-manager backup show

Show salt and wrapped keys as a hex dump

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

## Description

This command displays the backup information for onboard key management, which would be used to recover the cluster in case of catastrophic situations. The information displayed is for the cluster as a whole (not individual nodes). This command is not supported for an external key management configuration.

## Examples

The following example displays the onboard key management backup data for the cluster:

## security key-manager certificate update

(DEPRECATED)-Update key manager SSL certificates

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

### Description



This command is deprecated and might be removed in a future release of Data ONTAP.

This command updates the SSL/TLS certificate in-place without requiring the original SSL/TLS certificate to be

deleted. This command is not supported when onboard key management is enabled.

## Parameters

### **-type {client|server} - (DEPRECATED)-SSL Certificate Type**

This parameter is either "client" or "server". If "client", the internal client certificate is replaced. If "server", the internal server certificate is replaced.

### **[-address <IP Address>] - (DEPRECATED)-Key Manager IP Address**

This parameter updates the key manager server certificate for a particular key management server at the given IP address.

## Examples

The following example is for updating a server certificate:

```
cluster-1::> security key-manager certificate update -type server -address
10.232.186.8

Node: node1
Key manager 10.232.186.8 certificate-authority certificate will be
updated.
Update successful.

Node: node2
Key manager 10.232.186.8 certificate-authority certificate will be
updated.
Update successful.
```

The following example is for updating a client certificate:

```
cluster-1::> security key-manager certificate update -type client

Node: node1
The system client certificate registered with key manager will be updated.
Update successful.

Node: node2
The system client certificate registered with key manager will be updated.
Update successful.
```

## **security key-manager external boot-interfaces modify**

Modify external key manager logical interfaces

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

## Description

This command enables cluster administrators to modify the IP address and route information that the external key manager uses at boot time to restore keys from external key servers.

## Parameters

### **-node {<nodename>|local} - Node (privilege: advanced)**

Use this parameter to modify information on the node that you specify.

### **-address-type {ipv4|ipv6|ipv6z} - Address Type (privilege: advanced)**

Use this parameter to modify information for the address-type that you specify.

### **[-address <IP Address>] - Local Interface Address (privilege: advanced)**

Use this parameter to modify the IP address that the system will use at boot time to restore keys from external key servers. This parameter implies `-override-default true`.

### **{ [-netmask <IP Address>] - Network Mask (privilege: advanced)}**

Use this parameter to modify the IP netmask that the system will use at boot time to restore keys from external key servers. This parameter can be used only with address-type ipv4. This parameter implies `-override-default true`.

### **| [-netmask-length <integer>] - Bits in Network Mask (privilege: advanced) }**

Use this parameter to modify the IP netmask length that the system will use at boot time to restore keys from external key servers. This parameter implies `-override-default true`.

### **[-gateway <IP Address>] - Gateway (privilege: advanced)**

Use this parameter to modify the IP gateway that the system will use at boot time to restore keys from external key servers. This parameter implies `-override-default true`.

### **[-port {<netport>|<ifgrp>}] - Network Port (privilege: advanced)**

Use this parameter to modify the port that the system will use at boot time to restore keys from external key servers. The value that you specify cannot be a vlan or ifgrp port. This parameter implies `-override-default true`.

### **[-override-default {true|false}] - Override Default Setting? (privilege: advanced)**

Use this parameter to modify the system's selection of boot time IP address and route information. When this value is `false`, the system will use the information associated with a node management LIF. When this value is `true`, then the administrator has chosen to override the defaults.

## Examples

The following shows how to modify the port used by node "node2" at boot time to restore keys from external IPv4 key servers. In the example, IPv6 is not enabled in the cluster, so the `-address-type` parameter defaults to `ipv4`.

```
cluster-1::>*> security key-manager external boot-interfaces modify -node node2 -port e0d
```

The following example shows how to modify the IP address and gateway parameters used by node "node1" at boot time to restore keys from external IPv6 key servers.

```
cluster-1::*> security key-manager external boot-interfaces modify -node node1 -address-type ipv6 -address fd20:8b1e:b255:814e:749e:11a3:3bff:5820 -gateway fd20:8b1e:b255:814e::1
```

## **security key-manager external boot-interfaces show**

Show external key manager logical interfaces

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

### **Description**

This command enables cluster administrators to view the IP address and route information that the external key manager uses at boot time to restore keys from external key servers.

### **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 ] }**

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

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

Use this parameter to display information only about boot-time IP address and route information for the node that you specify.

**[-address-type {ipv4|ipv6|ipv6z}] - Address Type (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the address-type that you specify.

**[-address <IP Address>] - Local Interface Address (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the IP address that you specify.

**[-netmask <IP Address>] - Network Mask (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the network mask that you specify.

**[-netmask-length <integer>] - Bits in Network Mask (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the network mask length that you specify.

**[-gateway <IP Address>] - Gateway (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the gateway that you specify.

**[-port {<netport>|<ifgrp>}] - Network Port (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information for the port that you specify.

**[-override-default {true|false}] - Override Default Setting? (privilege: advanced)**

Use this parameter to display information only about boot-time IP address and route information with the override-default setting that you specify.

## Examples

The following example shows how to display the IP address and route information that the external key manager uses at boot time to restore keys. In the example, IPv6 is not enabled in the cluster and, as a result, the command displays information for only the IPv4 address-type. The override-default value is false for all rows, which indicates that the system automatically configured the values based on the node management LIF configuration on the nodes.

```
cluster-1::*> security key-manager external boot-interfaces show
      Address Network
      Address/Mask          Override
      Node     Type      Gateway      Port   Default?
-----
node1
  ipv4    10.224.113.159/24  10.224.113.1    e0M   false
node2
  ipv4    10.224.113.160/24  10.224.113.1    e0M   false
2 entries were displayed.
```

The following example shows how to display the IP address and route information that the external key manager uses at boot time to restore keys. In the example, IPv6 is enabled in the cluster and, as a result, the command displays information for both the IPv4 and IPv6 address-types. The override-default value is false for most rows, which indicates that the system automatically configured the values based on the node management LIF configuration on the nodes. The override-default value for node1 and address-type ipv4 is true, which indicates an administrator has used the [security key-manager external boot-interfaces modify](#) command to override one or more fields, and that the values may differ from the corresponding node management LIF.

```

cluster-1::*> security key-manager external boot-interfaces show
      Address Network                                Override
      Node     Type     Address/Mask      Gateway      Port  Default?
-----
node1
  ipv4     10.224.113.159/24  10.224.113.1    e0d   true
  ipv6     fd20:8b1e:b255:814e:32bd:f35c:832c:5a09/64
          fd20:8b1e:b255:814e::1
          e0M   false
node2
  ipv4     10.224.113.160/24  10.224.113.1    e0M   false
  ipv6     fd20:8b1e:b255:814e:749e:11a3:3bff:5820/64
          fd20:8b1e:b255:814e::1
          e0M   false
4 entries were displayed.

```

## Related Links

- [security key-manager external boot-interfaces modify](#)

## security key-manager key show

### Display Encryption Key IDs

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

### Description

This command displays the key IDs of the authentication keys (NSE-AK) and vserver keys (SVM-KEK) that are available in onboard key management. This command is not supported for an external key management configuration.

### 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.

| [-detail ]

If this parameter is specified, the command displays additional details about the key IDs.

| [-instance ] }

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

**[-node {<nodename>|local}] - Node**

If this parameter is specified, the command displays information only about key IDs that are located on the specified storage system.

**[-key-store <Key Store>] - Key Store**

If this parameter is specified, the command displays information only about key IDs that are managed by the specified key management. For example, use *onboard* for onboard key management.

**[-key-id <text>] - Key Identifier**

If this parameter is specified, the command displays information only about the specified key IDs.

**[-key-tag <text>] - Key Tag**

If this parameter is specified, the command displays information only about key IDs that have the specified key tags.

**[-key-location <text>] - Key Location**

If this parameter is specified, the command displays information only about key IDs that are located on the specified key location. For example, use *local-cluster* for onboard key management.

**[-used-by <Key Usage Type>] - Used By**

If this parameter is specified, the command displays information only about key IDs that are associated with the specified application usage of the keys. For example, "NSE-AK" would display key IDs only for NSE drives.

**[-restored {yes|no}] - Restored**

If this parameter is specified, the command displays information only about key IDs that have the specified value of restored keys. If restored is *yes*, then the corresponding key is available (normal). If restored is *no*, use the [security key-manager setup](#) command to restore the key. See the man page for [security key-manager setup](#) for details.

## Examples

The following example shows all keys stored in the onboard key manager:

```
cluster-1::> security key-manager key show

Node: node1
Key Store: onboard
Key ID                                     Used By
-----
00000000000000002000000000001001BC4C708E2A89A312E14B6CE6D4D49D4 NSE-AK
00000000000000002000000000001005E89099721F8817E65E3AEB68BE1BFCA NSE-AK
0000000000000000200000000000A0046DF92864D4CECE662B93BEB7F536610 SVM-KEK

Node: node2
Key Store: onboard
Key ID                                     Used By
-----
00000000000000002000000000001001BC4C708E2A89A312E14B6CE6D4D49D4 NSE-AK
00000000000000002000000000001005E89099721F8817E65E3AEB68BE1BFCA NSE-AK
0000000000000000200000000000A0046DF92864D4CECE662B93BEB7F536610 SVM-KEK
6 entries were displayed.
```

The following example shows a detailed view of all keys stored in the onboard key manager:

```

cluster-1::> security key-manager key show -detail

Node: node1
Key Store: onboard
Key ID Key Tag          Used By      Stored In
Restored
-----
-----
00000000000000002000000000001001BC4C708E2A89A312E14B6CE6D4D49D4
      -          NSE-AK      local-cluster      yes
00000000000000002000000000001005E89099721F8817E65E3AEB68BE1BFCA
      -          NSE-AK      local-cluster      yes
0000000000000000200000000000A0046DF92864D4CECE662B93BEB7F536610
      -          SVM-KEK     local-cluster      yes

Node: node2
Key Store: onboard
Key ID Key Tag          Used By      Stored In
Restored
-----
-----
00000000000000002000000000001001BC4C708E2A89A312E14B6CE6D4D49D4
      -          NSE-AK      local-cluster      yes
00000000000000002000000000001005E89099721F8817E65E3AEB68BE1BFCA
      -          NSE-AK      local-cluster      yes
0000000000000000200000000000A0046DF92864D4CECE662B93BEB7F536610
      -          SVM-KEK     local-cluster      yes
6 entries were displayed.

```

## Related Links

- [security key-manager setup](#)

## security login commands

### security login create

Add a login method

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

#### Description

The `security login create` command creates a login method for the management utility. A login method consists of a user name, an application (access method), and an authentication method. A user name can be associated with multiple applications. It can optionally include an access-control role name. If an Active

Directory, LDAP, or NIS group name is used, then the login method gives access to users belonging to the specified group. If the user is a member of multiple groups provisioned in the security login table, then the user will get access to a combined list of the commands authorized for the individual groups.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This specifies the Vserver name of the login method.

### **-user-or-group-name <text> - User Name or Group Name**

This specifies the user name or Active Directory, LDAP, or NIS group name of the login method. The Active Directory, LDAP, or NIS group name can be specified only with the *domain* or *nsswitch* authentication method and *ontapi* and *ssh* application. If the user is a member of multiple groups provisioned in the security login table, then the user will get access to a combined list of the commands authorized for the individual groups.

### **-application <text> - Application**

This specifies the application of the login method. Possible values include *console*, *http*, *ontapi*, *rsh*, *snmp*, *service-processor*, *ssh*, and *telnet*.

Setting this parameter to *service-processor* grants the user access to the Service Processor (SP). Because the SP supports only password authentication, when you set this parameter to *service-processor*, you must also set the *-authentication-method* parameter to *password*. Vserver user accounts cannot access the SP. Therefore, you cannot use the *-vserver* parameter when you set this parameter to *service-processor*.

### **-authentication-method <text> - Authentication Method**

This specifies the authentication method for login. Possible values include the following:

- cert - SSL certificate authentication
- community - SNMP community strings
- domain - Active Directory authentication
- nsswitch - LDAP or NIS authentication
- password - Password
- publickey - Public-key authentication
- usm - SNMP user security model
- saml - SAML authentication

### **[-remote-switch-ipaddress <IP Address>] - Remote Switch IP Address**

This specifies the IP address of the remote switch. The remote switch could be a cluster switch monitored by cluster switch health monitor (CSHM) or a Fibre Channel (FC) switch monitored by MetroCluster health monitor (MCC-HM). This parameter is applicable only when the application is *snmp* and authentication method is *usm* (SNMP user security model).

### **-role <text> - Role Name**

This specifies an access-control role name for the login method.

#### **[-comment <text>] - Comment Text**

This specifies comment text for the user account, for example, "Guest account". The maximum length is 128 characters.

#### **[-is-ns-switch-group {yes|no}] - Whether Ns-switch Group**

This specifies whether *user-or-group-name* is an LDAP or NIS group. Possible values are yes or no. Default value is no.

#### **[-second-authentication-method {none|publickey|password}] - Second Authentication Method2**

This specifies the authentication method for the login. It will be used as the second factor for authentication. Possible values include the following:

- password - Password
- publickey - Public-key authentication
- none - default value

### **Examples**

The following example illustrates how to create a login that has the user name *monitor*, the application *ssh*, the authentication method *password*, and the access-control role *guest* for Vserver *vs*:

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application ssh -authentication-method password -role guest
```

The following example illustrates how to create a login that has the user name *monitor*, the application *ontapi*, the authentication method *password*, and the access-control role *vsadmin* for Vserver *vs*:

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application ontapi -authentication-method password -role vsadmin
```

The following example illustrates how to create a login that has the user name *monitor*, the application *ssh*, the authentication method *publickey*, and the access-control role *guest* for Vserver *vs*:

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application ssh -authentication-method publickey -role guest
```

The following example illustrates how to create a login that has the user name *monitor*, the application *http*, the authentication method *cert*, and the access-control role *admin* for Vserver *vs*:

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application http -authentication-method cert -role admin
```

The following example illustrates how to create a login that has the Active Directory group name *adgroup* in

`DOMAIN1` , the application `ssh` , the authentication method `domain` , and the access-control role `vsadmin` for Vserver `vs` :

```
cluster1::> security login create -vserver vs -user-or-group-name  
DOMAIN1\adgroup -application ssh -authentication-method domain -role  
vsadmin
```

The following example illustrates how to create a login that has a group name `nssgroup` in the LDAP or NIS server, the application `ontapi` , the authentication method `nsswitch` , and the access-control role `vsadmin` for Vserver `vs` . Here `is-ns-switch-group` must be set to `yes` :

```
cluster1::> security login create -vserver vs -user-or-group-name nssgroup  
-application ontapi -authentication-method nsswitch -role vsadmin -is-ns  
-switch-group yes
```

The following example illustrates how to create a login that has the user name `monitor` , the application `ssh` , the authentication method `password` , the second authentication method `publickey` and the access-control role `vsadmin` for Vserver `vs` :

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application ssh -authentication-method password -second-authentication  
-method publickey -role vsadmin
```

The following example illustrates how to create a login that has the user name `monitor` , the application `ssh` , the authentication method `password` , the second authentication method `none` and the access-control role `vsadmin` for Vserver `vs` :

```
cluster1::> security login create -vserver vs -user-or-group-name monitor  
-application ssh -authentication-method password -second-authentication  
-method none -role vsadmin
```

## security login delete

Delete a login method

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

### Description

The `security login delete` command deletes a login method.

### Parameters

**-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver name of the login method.

**-user-or-group-name <text> - User Name or Group Name**

This specifies the user name or Active Directory, LDAP, or NIS group name of the login method that is to be deleted. A user name can be associated with multiple applications.

**-application <text> - Application**

This specifies the application of the login method. Possible values include console, http, ontapi, rsh, snmp, service-processor, ssh, and telnet.

**-authentication-method <text> - Authentication Method**

This specifies the authentication method of the login method. Possible values include the following:

- cert - SSL certificate authentication
- community - SNMP community strings
- domain - Active Directory authentication
- nsswitch - LDAP or NIS authentication
- password - Password
- publickey - Public-key authentication
- usm - SNMP user security model
- saml - SAML authentication

**[-remote-switch-ipaddress <IP Address>] - Remote Switch IP Address**

This specifies the IP address of the remote switch. The remote switch could be a cluster switch monitored by cluster switch health monitor (CSHM) or a Fibre Channel (FC) switch monitored by MetroCluster health monitor (MCC-HM). This parameter is applicable only when the application is *snmp* and authentication method is *usm* (SNMP user security model).

## Examples

The following example illustrates how to delete a login that has the username *guest* , the application *ssh* , and the authentication method *password* for Vserver *vs* :

```
cluster1::> security login delete -user-or-group-name guest -application
ssh -authentication-method password -vserver vs
```

The following example illustrates how to delete a login that has the username *guest* , the application *ontapi* , and the authentication method *cert* for Vserver *vs* :

```
cluster1::> security login delete -user-or-group-name guest -application
ontapi -authentication-method cert -vserver vs
```

The following example illustrates how to delete a login that has the Active Directory group name *adgroup* in *DOMAIN1* , the application *ssh* , and the authentication method *domain* for Vserver *vs* :

```
cluster1::> security login delete -user-or-group-name DOMAIN1\adgroup  
-application ssh -authentication-method domain -vserver vs
```

The following example illustrates how to delete a login that has a group name *nssgroup* in the LDAP or NIS server, the application *ontapi* , and the authentication method *nsswitch* for Vserver *vs* :

```
cluster1::> security login delete -user-or-group-name nssgroup  
-application ontapi -authentication-method nsswitch -vserver vs
```

## **security login expire-password**

Expire user's password

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

### **Description**

The *security login expire-password* command expires a specified user account password, forcing the user to change the password upon next login.

### **Parameters**

#### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver to which the user account belongs.

#### **-username <text> - Username**

This specifies the user name of the account whose password you want to expire.

#### **[-hash-function {sha512|sha256}] - Password Hash Function**

This optionally specifies the password-hashing algorithm used for encrypting the passwords that you want to expire. The supported values include are as follows:

- sha512 - Secure hash algorithm (512 bits)
- sha256 - Secure hash algorithm (256 bits)
- md5 - Message digest algorithm (128 bits)

#### **[-lock-after <integer>] - Lock User Account After N days (privilege: advanced)**

This optionally specifies the number of days after which the new password hash policy will be enforced. The enforcement will lock all user accounts that are still compliant with the provided hash algorithm using *-hash-function* parameter.

### **Examples**

The following command expires the password of the 'jdoe' user account which belongs to the 'vs1' Vserver.

```
cluster1::> security login expire-password -vserver vs1 -username jdoe
```

The following command expires all user account passwords that are encrypted with the MD5 hash function.

```
cluster1::> security login expire-password -vserver * -username * -hash  
-function md5
```

The following command expires the password of any Vserver's user account named 'jdoe' that is encrypted with the MD5 hash function.

```
cluster1::> security login expire-password -vserver * -username jdoe -hash  
-function md5
```

The following command expires the password of the 'vs1' Vserver user account named 'jdoe' that is encrypted with the MD5 hash function.

```
cluster1::> security login expire-password -vserver vs1 -username jdoe  
-hash-function md5
```

The following command expires all user account passwords that are encrypted with the MD5 hash function and enforce the new password hash policy after 180 days.

```
cluster1::> security login expire-password -vserver * -username * -hash  
-function md5 -lock-after 180
```

## security login lock

Lock a user account with password authentication method

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

### Description

The `security login lock` command locks a specified account, preventing it from accessing the management interface.

### Parameters

**-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver to which the user account belongs.

**-username <text> - Username**

This specifies the user name of the account that is to be locked.

## Examples

The following example locks a user account named 'jdoe' which belongs to the Vserver 'vs1'.

```
cluster1::> security login lock -vserver vs1 -username jdoe
```

## security login modify

Modify a login method

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

### Description

The *security login modify* command modifies the access-control role name of a login method. If the user is a member of multiple groups provisioned in the security login table, then the user will get access to a combined list of the commands authorized for the individual groups.

### Parameters

#### **-vserver <Vserver Name> - Vserver**

This specifies the Vserver name of the login method.

#### **-user-or-group-name <text> - User Name or Group Name**

This specifies the user name, Active Directory, LDAP, or NIS group name of the login method that is to be modified. A user name can be associated with multiple applications. If the user is a member of multiple groups provisioned in the security login table, then the user will get access to a combined list of the commands authorized for the individual groups.

#### **-application <text> - Application**

This specifies the application of the login method. Possible values include console, http, ontapi, rsh, snmp, service-processor, ssh, and telnet.

#### **-authentication-method <text> - Authentication Method**

This specifies the authentication method of the login method. Possible values include the following:

- cert - SSL certificate authentication
- community - SNMP community strings
- domain - Active Directory authentication
- nsswitch - LDAP or NIS authentication
- password - Password
- publickey - Public-key authentication
- usm - SNMP user security model
- saml - SAML authentication

#### **[-remote-switch-ipaddress <IP Address>] - Remote Switch IP Address**

This specifies the IP address of the remote switch. The remote switch could be a cluster switch monitored

by cluster switch health monitor (CSHM) or a Fibre Channel (FC) switch monitored by MetroCluster health monitor (MCC-HM). This parameter is applicable only when the application is *snmp* and authentication method is *usm* (SNMP user security model).

#### **[-role <text>] - Role Name**

This modifies the access-control role name for the login method.

#### **[-comment <text>] - Comment Text**

This specifies comment text for the user account, for example, "Guest account". The maximum length is 128 characters.

#### **[-is-ns-switch-group {yes|no}] - Whether Ns-switch Group**

This specifies if *user-or-group-name* is an LDAP or NIS group. Possible values are yes or no. Default value is no.

#### **[-second-authentication-method {none|publickey|password}] - Second Authentication Method2**

This specifies the authentication method for the login method. It will be used as the second factor for authentication. Possible values include the following:

- password - Password
- publickey - Public-key authentication
- none - default value

### **Examples**

The following example illustrates how to modify a login method that has the user name *guest* , the application *ontapi* , and the authentication method *password* to use the access-control role *guest* for Vserver *vs* :

```
cluster1::> security login modify -user-or-group-name guest -application
ontapi -authentication-method password -role guest -vserver vs
```

The following example illustrates how to modify a login method that has the user name *guest* , the application *ssh* , and the authentication method *publickey* to use the access-control role *vsadmin* for Vserver *vs* :

```
cluster1::> security login modify -user-or-group-name guest -application
ssh -authentication-method publickey -role vsadmin -vserver vs
```

The following example illustrates how to modify a login method that has the group name *nssgroup* , the application *ontapi* , and the authentication method *nsswitch* to use the access-control role *readonly* for Vserver *vs* . Here *is-ns-switch-group* must be set to *yes* :

```
cluster1::> security login modify -user-or-group-name nssgroup
-application ontapi -authentication-method nsswitch -role readonly
-vserver vs -is-ns-switch-group yes
```

The following example illustrates how to modify a login method that has the user name *guest* , the application *ssh* , and the authentication method *publickey* to use the second-authentication-method *password* for Vserver *vs* :

```
cluster1::> security login modify -user-or-group-name guest -application
ssh -authentication-method publickey -second-authentication-method
password -vserver vs
```

The following example illustrates how to modify a login method to have individual authentication methods that have the user name *guest* , the application *ssh* , and the authentication method *publickey* to use the second-authentication-method *none* for Vserver *vs* :

```
cluster1::> security login modify -user-or-group-name guest -application
ssh -authentication-method publickey -second-authentication-method none
-vserver vs
```

## **security login password-prepare-to-downgrade**

Reset password features introduced in the Data ONTAP version

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

### **Description**

If the password of the system administrator is not encrypted with an encryption type supported by releases earlier than ONTAP 9.0, this command prompts the administrator for a new password and encrypt it using a supported encryption type on each cluster or at each site in a MetroCluster configuration. In a MetroCluster configuration, this command must be run on both sites. The password for all other users are marked as "expired". This causes them to be re-encrypted using a compatible encryption type. The expired passwords are changed with an internally generated password. The administrator must change the passwords for all users before the users can login. The users are prompted to change their password upon login. This command disables the logging of unsuccessful login attempts. The command must be run by a user with the cluster admin role from a clustershell session on the console device. This user must be unlocked. If you fail to run this command, the revert process fails.

### **Parameters**

**-disable-feature-set <downgrade version>** - Data ONTAP Version (privilege: advanced)

This parameter specifies the Data ONTAP version that introduced the password feature set.

### **Examples**

The following command disables the logging of unsuccessful login attempts.

```
cluster1::*> security login password prepare-to-downgrade -disable-feature
-set 8.3.1

Warning: This command will disable the MOTD feature that prints
unsuccessful login attempts.

Do you want to continue? {y|n}: y
cluster1::*>
```

The following command prompts system administrator to enter password and encrypt it with the hashing algorithm supported by releases earlier than Data ONTAP 9.0.

```
cluster1::*> security login password prepare-to-downgrade -disable-feature
-set 9.0.0
Warning: If your password is not encrypted with an encryption type
supported by
                           releases earlier than Data ONTAP 9.0.0, this command will
prompt you
                           for a new password and encrypt it using a supported
encryption type on
                           each cluster or at each site in a MetroCluster configuration. In a
                           MetroCluster configuration, this command must be run on both sites.
                           The password for all other users are marked as "expired" and
                           changed to an internally generated password. The administrator must
change
                           the passwords for all users before the users can login. The users are
                           prompted to change their password upon login.

Do you want to continue? {y|n}:

Enter a new password:
Enter it again:

cluster1::*>
```

## security login password

### Modify a password for a user

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The *security login password* command resets the password for a specified user. The command prompts you for the user's old and new password.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver name of the login method.

### **-username <text> - Username**

This optionally specifies the user name whose password is to be changed. If you do not specify a user, the command defaults to the user name you are currently using.

## Examples

The following command initiates a password change for the 'admin' user account of the 'vs' Vserver.

```
cluster1::> security login password -username admin -vserver vs
```

The following command initiates a password change for the 'vs' Vserver user account named 'admin'. The new password will be encrypted by using the SHA512 password-hashing algorithm.

```
cluster1::*> security login password -username admin -vserver vs -hash
-function sha512
```

The following command initiates a password change for the 'vs' Vserver user account named 'admin'. The new password will be encrypted by using the SHA256 password-hashing encryption algorithm.

```
cluster1::*> security login password -username admin -vserver vs -hash
-function sha256
```

## security login show

Show user login methods

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

## Description

The `security login show` command displays the following information about user login methods:

- User name
- Application (console, http, ontapi, rsh, snmp, service-processor, ssh, or telnet)
- Authentication method (community, password, publickey, or usm)
- Role name
- Whether the account is locked
- Whether the user name refers to *nsswitch* group
- Password hash function

## 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 ] }

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

**[-vserver <Vserver Name>] - Vserver**

Displays the login methods that match the specified Vserver name.

**[-user-or-group-name <text>] - User Name or Group Name**

Displays the login methods that match this parameter value. Value can be a user name or Active Directory, LDAP, or NIS group name.

**[-application <text>] - Application**

Displays the login methods that match the specified application type. Possible values include console, http, ontapi, rsh, snmp, service-processor, ssh, and telnet.

**[-authentication-method <text>] - Authentication Method**

Displays the login methods that match the specified authentication method. Possible values include the following:

- cert - SSL certificate authentication
- community - SNMP community strings
- domain - Active Directory authentication
- nsswitch - LDAP or NIS authentication
- password - Password
- publickey - Public-key authentication
- usm - SNMP user security model
- saml - SAML authentication

**[-remote-switch-ipaddress <IP Address>] - Remote Switch IP Address**

Displays the login methods that match the specified IP address of the remote switch. The remote switch could be a cluster switch monitored by cluster switch health monitor (CSHM) or a Fibre Channel (FC) switch monitored by MetroCluster health monitor (MCC-HM). This parameter is applicable only when the application is `snmp` and authentication method is `usm` (SNMP user security model).

**[-role <text>] - Role Name**

Displays the login methods that match the specified role.

**[-is-account-locked {yes|no}] - Account Locked**

Displays the login methods that match the specified account lock status.

**[-comment <text>] - Comment Text**

Displays the login methods that match the specified comment text.

**[-is-ns-switch-group {yes|no}] - Whether Ns-switch Group**

This specifies whether *user-or-group-name* is an LDAP or NIS group. Possible values are yes or no.

**[-hash-function {sha512|sha256}] - Password Hash Function (privilege: advanced)**

Displays the login methods that match the specified password-hashing algorithm. Possible values are:

- sha512 - Secure hash algorithm (512 bits)
- sha256 - Secure hash algorithm (256 bits)
- md5 - Message digest algorithm (128 bits)

**[-second-authentication-method {none|publickey|password}] - Second Authentication Method2**

Displays the login methods that match the specified authentication method to be used as the second factor. Possible values include the following:

- password - Password
- publickey - Public-key authentication
- none - default value

## Examples

The example below illustrates how to display information about all user login methods:

```
cluster1::> security login show
```

Vserver: cluster1

User/Group	Authentication			Acct	
Authentication		Name	Application Method	Role Name	Locked Method
-----		-----	-----	-----	-----
admin	console	password	admin	no	none
admin	http	password	admin	no	none
admin	ontapi	password	admin	no	none
admin	service-processor		admin	no	none
admin	ssh	password	admin	no	none
autosupport	console	password	autosupport	no	none

Vserver: vs1.netapp.com

User/Group	Authentication			Acct	
Authentication		Name	Application Method	Role Name	Locked Method
-----		-----	-----	-----	-----
vsadmin	http	password	vsadmin	yes	none
vsadmin	ontapi	password	vsadmin	yes	none
vsadmin	ssh	password	vsadmin	yes	none

9 entries were displayed.

## security login unlock

Unlock a user account with password authentication method

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

### Description

The *security login unlock* command unlocks a specified account, enabling it to access the management interface.

### Parameters

#### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver to which the user account belongs.

#### **-username <text> - Username**

This specifies the user name of the account that is to be unlocked.

### **Examples**

The following command unlocks a user account named jdoe which belongs to the Vserver vs1.

```
cluster1::> security login unlock -vserver vs1 -username jdoe
```

## **security login whoami**

Show the current user and role of this session

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### **Description**

The *security login whoami* command displays the name and role of the user logged in at the current console session. It takes no options or other parameters.

### **Examples**

The following example shows that the current session is logged in by using the 'admin' user account:

```
cluster1::> whoami
              (security login whoami)
User: admin
Role: admin
```

## **security login banner modify**

Modify the login banner message

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### **Description**

The *security login banner modify* command modifies the login banner. The login banner is printed just before the authentication step during the SSH and console device login process.

### **Parameters**

#### **-vserver <Vserver Name> - Vserver Name**

Use this parameter to specify the Vserver whose banner will be modified. Use the name of the cluster admin Vserver to modify the cluster-level message. The cluster-level message is used as the default for data Vservers that do not have a message defined.

## { [-message <text>] - Login Banner Message

This optional parameter can be used to specify a login banner message. If the cluster has a login banner message set, the cluster login banner will be used by all data Vservers as well. Setting a data Vserver's login banner will override the display of the cluster login banner. To reset a data Vserver's login banner to use the cluster login banner, use this parameter with the value "-".

If you use this parameter, the login banner cannot contain newlines (also known as end of lines (EOLs) or line breaks). To enter a login banner message with newlines, do not specify any parameter. You will be prompted to enter the message interactively. Messages entered interactively can contain newlines.

Non-ASCII characters must be provided as Unicode UTF-8.

## { [-uri { (ftp|http)://(hostname|IPv4 Address|['IPv6 Address'])...} ] - Download URI for the Banner Message }

Use this parameter to specify the URI from where the login banner will be downloaded. Note that the message must not exceed 2048 bytes in length. Non-ASCII characters must be provided as Unicode UTF-8.

## Examples

This example shows how to enter a login banner interactively:

```
cluster1::> security login banner modify
Enter the login banner for Vserver "cluster1".
Max size: 2048. Enter a blank line to terminate input. Press Ctrl-C to
abort.
0           1           2           3           4           5           6           7
8
12345678901234567890123456789012345678901234567890123456789012345678901234
567890
Authorized users only!
cluster1::>
```

## security login banner show

Display the login banner message

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `security login banner show` command displays the login banner.

### 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 ] ]**

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

**[ -vserver <Vserver Name> ] - Vserver Name**

Selects login banners that match the specified value. Use the name of the admin Vserver to specify the cluster-level login banner.

**[ -message <text> ] - Login Banner Message**

Selects login banners that match the specified value. By default, this command will not display unconfigured, or empty, login banners. To display all banners, specify `-message *`.

## Examples

The following shows sample output from this command:

```
cluster1::> security login banner show
Message
-----
-----
Authorized users only!
cluster1::>
```

## security login domain-tunnel create

Add authentication tunnel Vserver for administrative Vserver

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

### Description

This command establishes a gateway (tunnel) for authenticating Windows Active Directory (AD) domain users' access to the cluster.

Before using this command to establish the tunnel, the following must take place:

- You must use the [security login create](#) command to create one or more AD domain user accounts that will be granted access to the cluster.
- The `-authmethod` parameter of the [security login create](#) command must be set to 'domain'.
- The `-username` parameter of the [security login create](#) command must be set to a valid AD domain user account that is defined in a Windows Domain Controller's Active Directory. The user account must be specified in the format of `<domainname>\<username>`, where "domainname" is the name of the CIFS domain server.
- You must identify or create a CIFS-enabled data Vserver that will be used for Windows authentication with the Active Directory server. This Vserver is the tunnel Vserver, and it must be running for this command to succeed.

Only one Vserver can be used as the tunnel. If you attempt to specify more than one Vserver for the tunnel, Data ONTAP returns an error. If the tunnel Vserver is stopped or deleted, AD domain users' authentication requests to the cluster will fail.

## Parameters

### **-vserver <vserver> - Authentication Tunnel Vserver**

This parameter specifies a data Vserver that has been configured with CIFS. This Vserver will be used as the tunnel for authenticating AD domain users' access to the cluster.

## Examples

The following commands create an Active Directory domain user account ('DOMAIN1\Administrator') for the 'cluster1' cluster, create a data Vserver ('vs'), create a CIFS server ('vscifs') for the Vserver, and specify 'vs' as the tunnel for authenticating the domain user access to the cluster.

```
cluster1::> security login create -vserver cluster1 -username
DOMAIN1\Administrator -application ssh -authmethod domain -role admin
cluster1::> vserver create -vserver vs -rootvolume vol -aggregate aggr
-rootvolume-security-style mixed
cluster1::> vserver cifs create -vserver vs -cifs-server vscifs
-domain companynname.example.com -ou CN=Computers
cluster1::> security login domain-tunnel create -vserver vs
```

## Related Links

- [security login create](#)

## **security login domain-tunnel delete**

Delete authentication tunnel Vserver for administrative Vserver

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

## Description

The `security login domain-tunnel delete` command deletes the tunnel established by the [security login domain-tunnel create](#) command. An error message will be generated if no tunnel exists.

## Examples

The following command deletes the tunnel established by [security login domain-tunnel create](#) .

```
cluster1::> security login domain-tunnel delete
```

## Related Links

- [security login domain-tunnel create](#)

## **security login domain-tunnel modify**

Modify authentication tunnel Vserver for administrative Vserver

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

## Description

The `security login domain-tunnel modify` command modifies or replaces the tunnel Vserver. If a tunnel Vserver is not already specified, it sets the current tunnel Vserver with this Vserver, otherwise, it replaces the current tunnel Vserver with the Vserver that you specify. If the tunnel Vserver is changed, authentication requests via previous Vserver will fail. See [security login domain-tunnel create](#) for more information.

## Parameters

### **[-vserver <vserver>] - Authentication Tunnel Vserver**

This parameter specifies a Vserver that has been configured with CIFS and is associated with a Windows Domain Controller's Active Directory authentication. This Vserver will be used as an authentication tunnel for login accounts so that they can be used with administrative Vservers.

## Examples

The following command modifies the tunnel Vserver for administrative Vserver.

```
cluster1::> security login domain-tunnel modify -vserver vs
```

## Related Links

- [security login domain-tunnel create](#)

## **security login domain-tunnel show**

Show authentication tunnel Vserver for administrative Vserver

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

## Description

The `security login domain-tunnel show` command shows the tunnel Vserver that was specified by the [security login domain-tunnel create](#) or [security login domain-tunnel modify](#) command.

## Examples

The example below shows the tunnel Vserver, `vs`, that is currently used as an authentication tunnel. The output informs you that the table is currently empty if tunnel Vserver has not been specified.

```
cluster1::> security login domain-tunnel show
Tunnel Vserver: vs
```

## Related Links

- [security login domain-tunnel create](#)

- [security login domain-tunnel modify](#)

## security login motd modify

Modify the message of the day

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `security login motd modify` command updates the message of the day (MOTD).

There are two categories of MOTDs: the cluster-level MOTD and the data Vserver-level MOTD. A user logging in to a data Vserver's clustershell will potentially see two messages: the cluster-level MOTD followed by the Vserver-level MOTD for that Vserver. The cluster administrator can enable or disable the cluster-level MOTD on a per-Vserver basis. If the cluster administrator disables the cluster-level MOTD for a Vserver, a user logging into the Vserver will not see the cluster-level message. Only a cluster administrator can enable or disable the cluster-level message.

### Parameters

#### **-vserver <Vserver Name> - Vserver Name**

Use this parameter to specify the Vserver whose MOTD will be modified. Use the name of the cluster admin Vserver to modify the cluster-level message.

#### **{ [-message <text>] - Message of the Day (MOTD)**

This optional parameter can be used to specify a message. If you use this parameter, the MOTD cannot contain newlines (also known as end of lines (EOLs) or line breaks). If you do not specify any parameter other than the `-vserver` parameter, you will be prompted to enter the message interactively. Messages entered interactively can contain newlines. Non-ASCII characters must be provided as Unicode UTF-8.

The message may contain dynamically generated content using the following escape sequences:

- `\ \` - A single backlash character.
- `\b` - No output: supported for compatibility with Linux only.
- `\C` - Cluster name.
- `\d` - Current date as set on the login node.
- `\t` - Current time as set on the login node.
- `\I` - Incoming LIF IP address (prints 'console' for a console login).
- `\l` - Login device name (prints 'console' for a console login).
- `\L` - Last login for the user on any node in the cluster.
- `\m` - Machine architecture.
- `\n` - Node or data Vserver name.
- `\N` - Name of user logging in.
- `\o` - Same as \O. Provided for Linux compatibility.

- `\o` - DNS domain name of the node. Note that the output is dependent on the network configuration and may be empty.
- `\r` - Software release number.
- `\s` - Operating system name.
- `\u` - Number of active clustershell sessions on the local node. For the cluster admin: all clustershell users. For the data Vserver admin: only active sessions for that data Vserver.
- `\U` - Same as `\u`, but has 'user' or 'users' appended.
- `\v` - Effective cluster version string.
- `\w` - Active sessions across the cluster for the user logging in ('who').

A backslash followed by any other character is emitted as entered.

**`[-uri { (ftp|http)://(hostname|IPv4 Address|['IPv6 Address'])...}]` - Download URI for the MOTD }**

Use this parameter to specify the URI from where the message of the day will be downloaded. Note that the message must not exceed 2048 bytes in length. Non-ASCII characters must be provided as Unicode UTF-8.

**`[-is-cluster-message-enabled {true|false}]` - Is Cluster-level Message Enabled?**

Use this parameter to enable or disable the display of the cluster-level MOTD for the specified Vserver.

## Examples

This example shows how to enter a MOTD interactively:

```
cluster1::> security login motd modify -vserver vs0

Enter the message of the day for Vserver "vs0".
Max size: 2048. Enter a blank line to terminate input. Press Ctrl-C to
abort.
0           1           2           3           4           5           6           7
8
12345678901234567890123456789012345678901234567890123456789012345678901234
567890
Welcome to the Vserver!
cluster1::>
```

## security login motd show

Display the message of the day

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `security login motd show` command displays information about the cluster-level and data Vserver

clustershell message of the day (MOTD).

## 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 ] }

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

**[-vserver <Vserver Name>] - Vserver Name**

Selects the message of the day entries that match this parameter value. Use the name of the cluster admin Vserver to see the cluster-level MOTD.

**[-message <text>] - Message of the Day (MOTD)**

Selects the message of the day entries that match this parameter value.

**[-is-cluster-message-enabled {true|false}] - Is Cluster-level Message Enabled?**

Selects the message of the day entries that match this parameter value.

## Examples

The following example displays all message of the day entries:

```
cluster1::> security login motd show
Vserver: cluster1
Is the Cluster MOTD Displayed?: true
Message
-----
-----
The cluster is running normally.

Vserver: vs0
Is the Cluster MOTD Displayed?: true
Message
-----
-----
Welcome to the Vserver!

2 entries were displayed.
```

## security login publickey create

Add a new public key

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `security login publickey create` associates an existing public key with a user account. This command requires that you enter a valid OpenSSH-formatted public key, a user name, index number, and optionally, a comment.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This parameter optionally specifies the Vserver of the user for whom you are adding the public key.

### **-username <text> - Username**

This parameter specifies the name of the user for whom you are adding the public key. If you do not specify a user, the user named admin is specified by default.

### **[-index <integer>] - Index**

This parameter specifies an index number for the public key. The default value is the next available index value, starting with zero if it is the first public key created for the user.

### **-publickey <certificate> - Public Key**

This specifies the OpenSSH public key, which must be enclosed in double quotation marks.

### **[-comment <text>] - Comment**

This optionally specifies comment text for the public key. Note that comment text should be enclosed in quotation marks.

## Examples

The following command associates a public key with a user named tsmith for Vserver vs1. The public key is assigned index number 5 and the comment text is "This is a new key".

```
cluster1::> security login publickey create -vserver vs1 -username tsmith
-index 5 -publickey
"ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAspH64CYbUsDQCdW22JnK6J
/vU9upnKzd2zAk9C1f7YaWRUAFNs2Qe51UmQ3ldi8AD0Vfbr5T6HZPCixNAIza
FcIDy7hgnmdj9eNGedGr/JNrftQbLD1hZybX+72DpQB0tYWBhe6eDJ1oPLob
ZBGfM1PXh8VjeU44i7W4+s0hG0E=tsmith@publickey.example.com"
-comment "This is a new key"
```

## **security login publickey delete**

### Delete a public key

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `security login publickey delete` command deletes a public key for a specific user. To delete a public key, you must specify a user name and index number.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This parameter optionally specifies the Vserver of the user for whom you are adding the public key.

### **-username <text> - Username**

This parameter specifies the name of the user for whom you are deleting a public key. If you do not specify a user, the user named admin is specified by default.

### **-index <integer> - Index**

This parameter specifies an index number for the public key.

## Examples

The following command deletes the public key for the user named tsmith with the index number 5.

```
cluster1::> security login publickey delete -username tsmith -index 5
```

## **security login publickey load-from-uri**

Load one or more public keys from a URI

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `security login publickey load-from-uri` command loads one or more public keys from a Universal Resource Identifier (URI). To load public keys from a URI, you must specify a user name, the URI from which to load them, and optionally, whether you want to overwrite the existing public keys.

## Parameters

### **-vserver <vserver name> - Vserver**

This parameter optionally specifies the Vserver for the user associated with the public keys.

### **-username <text> - Username**

This parameter specifies the username for the public keys. If you do not specify a username, the username "admin" is used by default.

### **-uri { (ftp|http)://(hostname|IPv4 Address|['IPv6 Address']) ... } - URI to load from**

This parameter specifies the URI from which the public keys will be loaded.

### **-overwrite {true|false} - Overwrite Entries**

This parameter optionally specifies whether you want to overwrite existing public keys. The default value for this parameter is *false*. If the value is *true* and you confirm to overwrite, then the existing public keys are overwritten with the new public keys. If you use the value *false* or do not confirm the overwrite, then newly loaded public keys are appended to the list of existing public keys using the next available index.

## Examples

The following command shows how to load public keys for the user named tsmith from the URI <ftp://ftp.example.com/identity.pub>. This user's existing public keys are not overwritten.

```
cluster1::> security login publickey load-from-uri -username tsmith  
-uri ftp://ftp.example.com/identity.pub -overwrite false
```

The following command shows how to load public keys for the user named tsmith from the URI <ftp://ftp.example.com/identity.pub>. This user's existing public keys are overwritten if user entered the option 'y' or 'Y'. The user's existing public keys are not overwritten if user entered the option 'n' or 'N' and the newly loaded public keys are appended to the list of existing public keys using the next available index. The user and password credentials that you provide when you use this command are the credentials to access the server specified by the URI.

```
cluster1::> security login publickey load-from-uri -username  
tsmith -uri ftp://ftp.example.com/identity.pub -overwrite true -vserver  
vs0
```

Enter User:

Enter Password:

Warning: You are about to overwrite the existing publickeys for the user "tsmith" in Vserver "vs0". Do you want to proceed? {y|n}:

## security login publickey modify

### Modify a public key

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The *security login publickey modify* command modifies a public key and optionally its comment text.

#### Parameters

##### **-vserver <Vserver Name> - Vserver**

Specifies the Vserver for the user associated with the public key.

##### **-username <text> - Username**

Specifies the username for the public key. If you do not specify a username, the username 'admin' is used by default.

##### **-index <integer> - Index**

Specifies the index number of the public key. The index number of the public key can be found by using the [security login publickey show](#) command.

### **[-publickey <certificate>] - Public Key**

Specifies the new public key. You must enclose the new public key in double quotation marks.

### **[-comment <text>] - Comment**

Specifies the new comment text for the public key.

## **Examples**

The following command modifies the public key at index number 10 for the user named tsmith of Vserver vs1.

```
cluster1::> security login publickey modify -vserver vs1 -username tsmith
-index 10 -publickey
"ssh-rsa
AAAAAB3NzaC1yc2EAAAQABAAQDDDD+pFzFgV/2dlowKRFgym9K910H/u+BVTGitCtHteHy
o8thmaXT
1GLCzaoc/12+XXiYKMRhJ00S9Svo4QQKUXHdCPXFSGR5PnAs39set39ECCLzmduplJnkWtX96p
QH/bg2g3upFcdC6z9
c37uqFtNVPfv8As1Si/9WDQmEJ2mRtJudJeU5GZwZw5ybgTaN1jxDWus9SO2C43F/vmoCKVT52
9Uht4/ePcaaHOGTiQ
O8+Qmm59uTgcfnpg53zYkpeAQV8RdYtMdWlRr44neh1WZrmW7x5N4nXNvtEzr9cvb9sJyqTX1C
kQGfDOdb+7T7y3X7M
if/qKQY6FsovjvfZD"
```

## **Related Links**

- [security login publickey show](#)

## **security login publickey show**

Display public keys

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### **Description**

The *security login publickey show* command displays information about public keys.

### **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 ] }**

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

**[-vserver <Vserver Name>] - Vserver**

Selects the public keys that match this parameter value.

**[-username <text>] - Username**

Selects the public keys that match this parameter value.

**[-index <integer>] - Index**

Selects the public keys that match this parameter value.

**[-publickey <certificate>] - Public Key**

Selects the public keys that match this parameter value.

**[-fingerprint <text>] - Hex Fingerprint**

Selects the public keys that match this parameter value.

**[-bubblebabble <text>] - Bubblebabble Fingerprint**

Selects the public keys that match this parameter value.

**[-comment <text>] - Comment**

Selects the public keys that match this parameter value.

## Examples

The example below displays public key information for the user named tsmith.

```
cluster1::> security login publickey show -username tsmith
UserName: tsmith Index: 5
Public Key:
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAspH64CYbUsDQCdW22JnK6J
/vU9upnKzd2zAk9C1f7YaWRUAFNs2Qe5lUmQ3ldi8AD0Vfbr5T6HZPCixNAIza
FciDy7hgnmdj9eNGedGr/JNrftQbLD1hZybX+72DpQB0tYWBhe6eDJ1oPLob
ZBGfM1PXh8VjeU44i7W4+s0hG0E=tsmith@publickey.example.com
Fingerprint:
07:b4:27:52:ce:7f:35:81:5a:f2:07:cf:c1:87:91:97
Bubblebabble fingerprint:
xuzom-nelug-bisih-nihyr-metig-kemal-puhut-somyd-mumuh-zomis-syxex
Comment:
This is a new key
```

## security login role create

Add an access control role

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

## Description

The security login role create command creates an access-control role. An access-control role consists of a role name and a command or directory to which the role has access. It optionally includes an access level (none, readonly, or all) and a query that applies to the specified command or command directory. After you create an access-control role, you can apply it to a management-utility login account by using the [security login modify](#) or [security login create](#) commands.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver name associated with the role.

### **-role <text> - Role Name**

This specifies the role that is to be created.

### **-cmddirname <text> - Command / Directory**

This specifies the command or command directory to which the role has access. To specify the default setting, use the special value "``DEFAULT``".

### **[-access <Access>] - Access Level**

This optionally specifies an access level for the role. Possible access level settings are none, readonly, and all. The default setting is all .

### **[-query <query>] - Query**

This optionally specifies the object that the role is allowed to access. The query object must be applicable to the command or directory name specified by -cmddirname. The query object must be enclosed in double quotation marks (""), and it must be a valid field name.

## Examples

The following command creates an access-control role named "admin" for the vs1.example.com Vserver. The role has all access to the "volume" command but only within the "aggr0" aggregate.

```
cluster1::> security login role create -role admin -cmddirname volume
          -query "-aggr aggr0" -access all -vserver vs1.example.com
```

## Related Links

- [security login modify](#)
- [security login create](#)

## **security login role delete**

Delete an access control role

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

## Description

The security login role delete command deletes an access-control role.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver name associated with the role.

### **-role <text> - Role Name**

This specifies the role that is to be deleted.

### **-cmddirname <text> - Command / Directory**

This specifies the command or command directory to which the role has access. To specify the default setting, use the special value "DEFAULT".

## Examples

The following command deletes an access-control role with the role name readonly and the command access "volume" for Vserver vs.example.com.

```
cluster1::> security login role delete -role readonly -cmddirname volume  
-vserver vs.example.com
```

## security login role modify

Modify an access control role

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

## Description

The security login role modify command modifies an access-control role.

## Parameters

### **-vserver <Vserver Name> - Vserver**

This optionally specifies the Vserver name associated with the role.

### **-role <text> - Role Name**

This specifies the role that is to be modified.

### **-cmddirname <text> - Command / Directory**

This specifies the command or command directory to which the role has access. To specify the default setting for a role, use the special value "DEFAULT". This value can be modified only for the roles created for the admin Vserver.

### **[-access <Access>] - Access Level**

This optionally specifies a new access level for the role. Possible access level settings are none, readonly,

and all. The default setting is all .

### **[-query <query>] - Query**

This optionally specifies the object that the role is allowed to access. The query object must be applicable to the command or directory name specified by -cmddirname. The query object must be enclosed in double quotation marks (""), and it must be a valid field name.

## **Examples**

The following command modifies an access-control role with the role name readonly and the command access "volume" to have the access level readonly for Vserver vs.example.com:

```
cluster1::> security login role modify -role readonly -cmddirname volume  
-access readonly -vserver vs.example.com
```

## **security login role prepare-to-downgrade**

Update role configurations so that they are compatible with earlier releases of Data ONTAP

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

## **Description**

The *security login role prepare-to-downgrade* command restores predefined roles of all Vservers earlier than Data ONTAP 8.3.2. You must run this command in advanced privilege mode when prompted to do so during the release downgrade.

## **Examples**

The following command restores predefined roles of all Vservers earlier than Data ONTAP 8.3.2.

```
cluster1::*> security login role prepare-to-downgrade
```

## **security login role show-ontapi**

Display the mapping between Data ONTAP APIs and CLI commands

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## **Description**

The *security login role show-ontapi* command displays Data ONTAP APIs (ONTAPIs) and the CLI commands that they are mapped to.

## **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 ] }

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

**[-ontapi <text>] - ONTAPI Name**

Use this parameter to view the corresponding CLI command for the specified API.

**[-command <text>] - CLI Command**

Use this parameter to view the corresponding API or APIs for the specified CLI command.

## Examples

The following command displays all Data ONTAP APIs and their mapped CLI commands:

```
cluster1::> security login role show-ontapi
ONTAPI                               Command
-----
-----
aggr-add                         storage aggregate add-disks
aggr-check-spare-low               storage aggregate check_spare_low
aggr-create                        storage aggregate create
aggr-destroy                        storage aggregate delete
aggr-get-filer-info                aggr
aggr-get-iter                       storage aggregate show-view
aggr-offline                        storage aggregate offline
aggr-online                         storage aggregate online
aggr-options-list-info              storage aggregate show
aggr-rename                          storage aggregate rename
aggr-restrict                        storage aggregate restrict
aggr-set-option                     storage aggregate modify
autosupport-budget-get             system node autosupport budget show
autosupport-budget-get-iter        system node autosupport budget show
autosupport-budget-get-total-records
                                         system node autosupport budget show
autosupport-budget-modify         system node autosupport budget modify
autosupport-config-get             system node autosupport show
autosupport-config-get-iter        system node autosupport show
autosupport-config-get-total-records
                                         system node autosupport show
autosupport-config-modify         system node autosupport modify
Press <space> to page down, <return> for next line, or 'q' to quit...
```

The following example displays all Data ONTAP APIs which are mapped to the specified CLI command:

```
cluster1::> security login role show-ontapi -command version
ONTAPI                               Command
-----
-----
system-get-ontapi-version  version
system-get-version        version
2 entries were displayed.
```

The following example displays the CLI command that is mapped to the specified Data ONTAPI API:

```
cluster1::> security login role show-ontapi -ontapi aggr-create
ONTAPI Name: aggr-create
          Command: storage aggregate create
```

## **security login role show**

Show access control roles

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

### **Description**

The `security login role show` command displays the following information about access-control roles:

- Role name
- Command or command directory to which the role has access
- Access level (none, read-only, or all)
- Query (detailed view only)

### **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 ] }

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

**[-vserver <Vserver Name>] - Vserver**

Selects the roles that match this parameter value.

**[-role <text>] - Role Name**

Selects the roles that match this parameter value. If this parameter and the `-cmddirname` parameter are both used, the command displays detailed information about the specified access-control role.

### **[-cmddirname <text>] - Command / Directory**

Selects the roles that match this parameter value. If this parameter and the `-role` parameter are both used, the command displays detailed information about the specified access-control role.

### **[-access <Access>] - Access Level**

Selects the roles that match this parameter value.

### **[-query <query>] - Query**

Selects the roles that match this parameter value.

## **Examples**

The example below displays information about all access-control roles:

```
cluster1::> security login role show

Vserver      RoleName      Command/Directory      Query
AccessLevel

-----
-----
vs          vsadmin        DEFAULT                  none
vs          vsadmin        dashboard health vserver  readonly
vs          vsadmin        job                      readonly
vs          vsadmin        job schedule             none
vs          vsadmin        lun                      all
vs          vsadmin        network connections      readonly
cluster1    admin          DEFAULT                  all
cluster1    readonly      DEFAULT                  readonly
cluster1    readonly      volume                  none
```

## **security login role config modify**

Modify local user account restrictions

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

### **Description**

The `security login role config modify` command modifies user account and password restrictions.

For the password character restrictions documented below (uppercase, lowercase, digits, etc.), the term "characters" refers to ASCII-range characters only - not extended characters.

### **Parameters**

#### **-vserver <vserver name> - Vserver**

This specifies the Vserver name associated with the profile configuration.

**`-role <text>` - Role Name**

This specifies the role whose account restrictions are to be modified.

**`[-username-minlength <integer>]` - Minimum Username Length Required**

This specifies the required minimum length of the user name. Supported values are 3 to 16 characters. The default setting is 3 characters.

**`[-username-alphanum {enabled|disabled}]` - Username Alpha-Numeric**

This specifies whether a mix of alphabetic and numeric characters are required in the user name. If this parameter is enabled, a user name must contain at least one letter and one number. The default setting is *disabled*.

**`[-passwd-minlength <integer>]` - Minimum Password Length Required**

This specifies the required minimum length of a password. Supported values are 3 to 64 characters. The default setting is 8 characters.

**`[-passwd-alphanum {enabled|disabled}]` - Password Alpha-Numeric**

This specifies whether a mix of alphabetic and numeric characters is required in the password. If this parameter is enabled, a password must contain at least one letter and one number. The default setting is *enabled*.

**`[-passwd-min-special-chars <integer>]` - Minimum Number of Special Characters Required in the Password**

This specifies the minimum number of special characters required in a password. Supported values are from 0 to 64 special characters. The default setting is 0, which requires no special characters.

**`[-passwd-expiry-time <unsigned32_or_unlimited>]` - Password Expires In (Days)**

This specifies password expiration in days. A value of 0 means all passwords associated with the accounts in the role expire now. The default setting is *unlimited*, which means the passwords never expire.

**`[-require-initial-passwd-update {enabled|disabled}]` - Require Initial Password Update on First Login**

This specifies whether users must change their passwords when logging in for the first time. Initial password changes can be done only through SSH or serial-console connections. The default setting is *disabled*.

**`[-max-failed-login-attempts <integer>]` - Maximum Number of Failed Attempts**

This specifies the allowed maximum number of consecutive invalid login attempts. When the failed login attempts reach the specified maximum, the account is automatically locked. The default is 0, which means failed login attempts do not cause an account to be locked.

**`[-lockout-duration <integer>]` - Maximum Lockout Period (Days)**

This specifies the number of days for which an account is locked if the failed login attempts reach the allowed maximum. The default is 0, which means the accounts will be locked for 1 day.

**`[-disallowed-reuse <integer>]` - Disallow Last 'N' Passwords**

This specifies the number of previous passwords that are disallowed for reuse. The default setting is six, meaning that the user cannot reuse any of their last six passwords. The minimum allowed value is 6.

**[-change-delay <integer>] - Delay Between Password Changes (Days)**

This specifies the number of days that must pass between password changes. The default setting is *0*.

**[-delay-after-failed-login <integer>] - Delay after Each Failed Login Attempt (Secs)**

This specifies the amount of delay observed by the system in seconds upon invalid login attempts. The default setting is *4* seconds.

**[-passwd-min-lowercase-chars <integer>] - Minimum Number of Lowercase Alphabetic Characters Required in the Password**

This specifies the minimum number of lowercase characters required in a password. Supported values are from *0* to *64* lowercase characters. The default setting is *0*, which requires no lowercase characters.

**[-passwd-min-uppercase-chars <integer>] - Minimum Number of Uppercase Alphabetic Characters Required in the Password**

This specifies the minimum number of uppercase characters required in a password. Supported values are from *0* to *64* uppercase characters. The default setting is *0*, which requires no uppercase characters.

**[-passwd-min-digits <integer>] - Minimum Number of Digits Required in the Password**

This specifies the minimum number of digits required in a password. Supported values are from *0* to *64* digits characters. The default setting is *0*, which requires no digits.

**[-passwd-expiry-warn-time <unsigned32\_or\_unlimited>] - Display Warning Message Days Prior to Password Expiry (Days)**

This specifies the warning period for password expiry in days. A value of *0* means warn user about password expiry upon every successful login. The default setting is *unlimited*, which means never warn about password expiry.

**[-account-expiry-time <unsigned32\_or\_unlimited>] - Account Expires in (Days)**

This specifies account expiration in days. The default setting is *unlimited*, which means the accounts never expire. The account expiry time must be greater than account inactive limit.

**[-account-inactive-limit <unsigned32\_or\_unlimited>] - Maximum Duration of Inactivity before Account Expiration (Days)**

This specifies inactive account expiry limit in days. The default setting is *unlimited*, which means the inactive accounts never expire. The account inactive limit must be less than account expiry time.

## Examples

The following command modifies the user-account restrictions for an account with the role name *admin* for a Vserver named *vs*. The minimum size of the password is set to *12* characters.

```
cluster1::> security login role config modify -role admin -vserver vs  
-passwd-minlength 12
```

## security login role config reset

Reset RBAC characteristics supported on releases later than Data ONTAP 8.1.2

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

## Description

The `security login role config reset` command resets the following role based access control (RBAC) characteristics to their default values. The system prompts you to run this command if you revert to Data ONTAP 8.1.2 or earlier. If you do not reset these characteristics, the revert process will fail.

- Minimum number of special characters required in password ("0")
- Password-expiration time, in days ("unlimited")
- Whether the password must be changed at the initial login ("disabled")
- Maximum number of failed login attempts permitted before the account is locked out ("0")
- Number of days that the user account is locked out after the maximum number of failed login attempts is reached ("0")

## Examples

The following command resets the above mentioned RBAC characteristics of all cluster and Vserver roles to their default values.

```
cluster1::> security login role config reset
```

## security login role config show

Show local user account restrictions

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

## Description

The `security login role config show` command displays the following information about account restrictions for management-utility user accounts:

- Role name `-role`
- Minimum size of the password, in characters `-passwd-minlength`
- Whether the password requires alphanumeric characters `-passwd-alphanum`
- Number of previous passwords that cannot be reused `-disallowed-reuse`
- Minimum number of days that must elapse before users can change their passwords `-change-delay`

You can display detailed information about the restrictions on a specific account by specifying the `-role` parameter. This adds the following information:

- Minimum length of the user name, in characters `-username-minlength`
- Whether the user name requires alphanumeric characters `-username-alphanum`
- Minimum length of the password, in characters `-passwd-minlength`
- Whether the password requires alphanumeric characters `-passwd-alphanum`
- Minimum number of special characters required in password `-passwd-min-special-chars`

- Minimum number of lowercase characters required in password -passwd-min-lowercase-chars
- Minimum number of uppercase characters required in password -passwd-min-uppercase-chars
- Minimum number of digits required in password -passwd-min-digits
- Minimum number of days that must elapse before users can change their passwords -change-delay
- Whether the password must be changed at the initial login -require-initial-passwd-update
- Password-expiration time, in days -passwd-expiry-time
- Display warning message days prior to password expiry -passwd-expiry-warn-time
- Number of previous passwords that cannot be reused -disallowed-reuse
- Maximum number of failed login attempts permitted before the account is locked out -max-failed-login-attempts
- Number of days for which the user account is locked after the maximum number of failed login attempts is reached -lockout-duration
- Account-expiration time, in days -account-expiry-time
- Maximum duration of inactivity before account expiration, in days -account-inactive-limit
- Delay after each failed login attempt, in secs -delay-after-failed-login

## 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 ] }

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

**[-vserver <vserver name>] - Vserver**

Selects the profile configurations that match this parameter value

**[-role <text>] - Role Name**

If this parameter is specified, the command displays detailed information about restrictions for the specified user account.

**[-username-minlength <integer>] - Minimum Username Length Required**

Selects the profile configurations that match this parameter value.

**[-username-alphanum {enabled|disabled}] - Username Alpha-Numeric**

Selects the profile configurations that match this parameter value. Enabled means a user name must contain both letters and numbers.

**[-passwd-minlength <integer>] - Minimum Password Length Required**

Selects the profile configurations that match this parameter value.

**[-passwd-alphanum {enabled|disabled}] - Password Alpha-Numeric**

Selects the profile configurations that match this parameter value. Enabled means a password must contain both letters and numbers.

**[-passwd-min-special-chars <integer>] - Minimum Number of Special Characters Required in the Password**

Selects the profile configurations that match this parameter value.

**[-passwd-expiry-time <unsigned32\_or\_unlimited>] - Password Expires In (Days)**

Selects the profile configurations that match this parameter value.

**[-require-initial-passwd-update {enabled|disabled}] - Require Initial Password Update on First Login**

Selects the profile configurations that match this parameter value.

**[-max-failed-login-attempts <integer>] - Maximum Number of Failed Attempts**

Selects the profile configurations that match this parameter value.

**[-lockout-duration <integer>] - Maximum Lockout Period (Days)**

Selects the profile configurations that match this parameter value.

**[-disallowed-reuse <integer>] - Disallow Last 'N' Passwords**

Selects the profile configurations that match this parameter value.

**[-change-delay <integer>] - Delay Between Password Changes (Days)**

Selects the profile configurations that match this parameter value.

**[-delay-after-failed-login <integer>] - Delay after Each Failed Login Attempt (Secs)**

Selects the profile configurations that match this parameter value.

**[-passwd-min-lowercase-chars <integer>] - Minimum Number of Lowercase Alphabetic Characters Required in the Password**

Selects the profile configurations that match this parameter value.

**[-passwd-min-uppercase-chars <integer>] - Minimum Number of Uppercase Alphabetic Characters Required in the Password**

Selects the profile configurations that match this parameter value.

**[-passwd-min-digits <integer>] - Minimum Number of Digits Required in the Password**

Selects the profile configurations that match this parameter value.

**[-passwd-expiry-warn-time <unsigned32\_or\_unlimited>] - Display Warning Message Days Prior to Password Expiry (Days)**

Selects the profile configurations that match this parameter value.

**[-account-expiry-time <unsigned32\_or\_unlimited>] - Account Expires in (Days)**

Selects the profile configurations that match this parameter value.

**[-account-inactive-limit <unsigned32\_or\_unlimited>] - Maximum Duration of Inactivity before Account Expiration (Days)**

Selects the profile configurations that match this parameter value.

## Examples

The example below displays restriction information about all user accounts:

```
cluster1::> security login role config show
              ----- Password Restrictions -----
Vserver      RoleName      Size AlphaNum NoReuse ChangeDelay
-----
vs           vsadmin        8   enabled      6   0 days
vs           vsadmin-protocol 8   enabled      6   0 days
vs           vsadmin-readonly 8   enabled      6   0 days
vs           vsadmin-volume   8   enabled      6   0 days
cluster1     admin          6   enabled      6   0 days
cluster1     readonly       6   enabled      6   0 days
```

# security protocol commands

## security protocol modify

Modify application configuration options

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

### Description

The `security protocol modify` command modifies the existing cluster-wide configuration of RSH and Telnet. Enable RSH and Telnet in the cluster by setting the `enabled` field as *true*.

### Parameters

**-application <text> - application (privilege: advanced)**

Selects the application. Supported values are *rsh* and *telnet*.

**[-enabled {true|false}] - enabled (privilege: advanced)**

Enables or disables the corresponding application. The default value is *false*.

## Examples

The following command enables RSH in the cluster. The default setting for RSH is *false*:

```
cluster1::> security protocol modify -application rsh -enabled true
```

The following command enables Telnet in the cluster. The default setting for Telnet is *false*:

```
cluster1::> security protocol modify -application telnet -enabled true
```

## security protocol show

Show application configuration options

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

### Description

The `security protocol show` command displays the cluster-wide configuration of RSH and Telnet in the cluster in advanced privilege mode. RSH and Telnet are disabled by default. Use the [security protocol modify](#) command to change the RSH and Telnet configuration that the cluster supports.

### 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 ] }

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

**[-application <text>] - application (privilege: advanced)**

Displays the insecure applications in the cluster.

**[-enabled {true|false}] - enabled (privilege: advanced)**

Displays whether the application is enabled or disabled in the cluster.

### Examples

The following example shows the default security protocol configurations for a cluster:

```
cluster1::> security protocol show
```

Application	Enabled
rsh	false
telnet	false

The following example shows the security protocol configuration after RSH and Telnet have been enabled:

```
cluster1::> security protocol show
Application      Enabled
-----
rsh            true
telnet          true
```

## Related Links

- [security protocol modify](#)

## security protocol ssh modify

Modify the SSH configuration

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

### Description

The `security protocol ssh modify` command modifies the existing cluster-wide configuration of SSH

### Parameters

**`[-per-source-limit <integer>]` - Per-Source Limit (privilege: advanced)**

Modifies the maximum number of SSH instances per source IP address on a per-node basis.

**`[-max-instances <integer>]` - Maximum Number of Instances (privilege: advanced)**

Modifies the maximum number of SSH instances that can be handled on a per-node basis.

**`[-connections-per-second <integer>]` - Connections Per Second (privilege: advanced)**

Modifies the maximum number of SSH connections per second on a per-node basis.

### Examples

The following example modifies cluster-wide SSH configuration:

```
cluster1::*> security protocol ssh modify -per-source-limit 30 -max
-instances 60 -connections-per-second 5
```

## security protocol ssh show

Show the SSH configuration

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

### Description

The `security protocol ssh show` command displays the cluster-wide SSH configuration in advanced privilege mode. Use the [security protocol ssh modify](#) command to change the SSH configuration that the

cluster supports.

## Examples

The following example displays cluster-wide SSH configuration:

```
cluster1::*> security protocol ssh show
Per-Source Limit: 32
Maximum Number of Instances: 64
Connections Per Second: 10
```

## Related Links

- [security protocol ssh modify](#)

# security saml-sp commands

## security saml-sp create

Configure SAML service provider for authentication

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

### Description

The `security saml-sp create` command configures ONTAP with Security Assertion Markup Language (SAML) Service Provider (SP) for single sign-on authentication. This command does not enable SAML SP, it just configures it. Configuring and enabling SAML SP is a two-step process:

- Create a SAML SP configuration using `security saml-sp create` command.
- Enable SAML SP by using `security saml-sp modify-is-enabled true`

After the SAML SP configuration is created, it cannot be modified. It must be deleted and created again to change any settings.



This restarts the web server. Any HTTP/S connections that are active will be disrupted.

### Parameters

**`-idp-uri { (ftp|http)://(hostname|IPv4 Address|['IPv6 Address']) ... } - Identity Provider (IdP) Metadata Location`**

This is the URI of the desired identity provider's (IdP) metadata.

**`[-sp-host <Remote InetAddress>] - SAML Service Provider Host`**

This specifies the SAML service provider host IP address.

**`{ -cert-ca <text> - Server Certificate Issuing CA`**

This specifies the service provider's certificate issuing CA.

**`-cert-serial <text>` - Server Certificate Serial Number**

This specifies the service provider's certificate's serial number.

**`[-cert-common-name <FQDN or Custom Common Name>]` - Server Certificate Common Name }**

This specifies the service provider certificate's common name.

**`[-verify-metadata-server {true|false}]` - Verify IdP Metadata Server Identity**

When the IdP metadata is downloaded, the identity of the server hosting the metadata is verified using transport layer security (TLS), validating the server's X.509 certificate against the list of certificate authorities (CAs) in Data ONTAP, and verifying that the host in the server certificate matches the host in the URI (the `idp-uri` field). This verification can be bypassed by setting this field to `false`. Bypassing the server verification is not recommended as the server can not be trusted that way, but will be necessary to use non-TLS URLs, e.g. with the "http" scheme, or when the server certificates are self-signed. If the server's certificate was signed by a CA that is not installed in Data ONTAP, the [security certificate install -type server-ca](#) command can be used to install it.

**`[-foreground {true|false}]` - Foreground Process**

When this parameter is set to `false` the command runs in the background as a job. The default is `true`, which causes the command to return after the operation completes.

## Examples

The following example configures ONTAP with SAML SP IdP information:

```
cluster1::> security saml-sp create -idp-uri http://public-idp-uri -sp
-host 1.1.1.1
[Job 9] Job succeeded.
cluster1::>
```

## Related Links

- [security saml-sp modify](#)
- [security certificate install](#)

## **security saml-sp delete**

Delete SAML service provider for authentication

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

### Description

The `security saml-sp delete` command is used to remove the Security Access Markup Language (SAML) Service Provider (SP). Running this command frees resources used by the SP. SAML SP services will no longer be available after the SP is removed.

If the SAML SP is currently enabled, it is necessary to first use `security saml-sp modify -is-enabled false` prior to `security saml-sp delete`. The `security saml-sp modify -is-enabled false` command must be issued by a password authenticated console application user or from a SAML authenticated command

interface.



This restarts the web server. Any HTTP/S connections that are active will be disrupted.

## Examples

The following example unconfigures SAML SP:

```
cluster1::> security saml-sp delete
cluster1::>
```

## Related Links

- [security saml-sp modify](#)

## security saml-sp modify

Modify SAML service provider authentication

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

## Description

The `security saml-sp modify` command modifies the Security Assertion Markup Language (SAML) Service Provider (SP) configuration for single sign-on authentication. This command is used to enable or disable an existing SAML SP, `security saml-sp modify-is-enabled true` or `false` respectively.

This command will check the validity of the current SAML SP configuration before enabling the SP. Also, it is necessary to use this command with the `-is-enabled false` parameter prior to deleting an existing SAML SP configuration. SAML SP can only be disabled in this way by a password authenticated console application user or from a SAML authenticated command interface. The `delete` command must be used if the SAML configuration settings are to be changed, as only the `is-enabled` parameter can be modified.



This may restart the web server. Any HTTP/S connections that are active may be disrupted.

## Parameters

### **`[-is-enabled {true|false}]` - SAML Service Provider Enabled**

Use this parameter to enable or disable the SAML SP.

## Examples

The following example enables SAML SP:

```
cluster1::> security saml-sp modify -is-enabled true
cluster1::>
```

## security saml-sp repair

Repair a failed SAML SP configuration

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

### Description

The `security saml-sp repair` command attempts to repair a failed SAML SP configuration on a given node. The status of the individual nodes can be viewed using the [security saml-sp status show](#) command.



This restarts the web server. Any active HTTP/S requests to the web server will be disrupted.

### Parameters

**-node {<nodename>|local} - Node (privilege: advanced)**

This identifies a single node that matches the input. The repair job will run on this node.

**[-foreground {true|false}] - Foreground Process (privilege: advanced)**

When this parameter is set to *false* the command runs in the background as a job. The default is *true*, which causes the command to return after the operation completes.

### Examples

The following example repairs a failed SAML SP configuration:

```
cluster1:> security saml-sp repair -node node-2
Warning: This restarts the web server. Any active HTTP/S requests to the
web
server will be disrupted
Do you want to continue? {y|n}: y
[Job 1321] Job succeeded.
cluster1:>
```

### Related Links

- [security saml-sp status show](#)

## security saml-sp show

Display SAML service provider for authentication

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

### Description

The `security saml-sp show` command displays the Security Assertion Markup Language (SAML) Service Provider (SP) configuration.

The Identity Provider (IdP) URI indicates the URI of the desired IdP's metadata.

The Service Provider (SP) host indicates the IP address containing SAML SP metadata.

The Certificate Common Name indicates the SAML SP certificate's common name.

The Certificate Serial indicates the SAML SP certificate's serial number.

## Examples

The following example displays the SAML SP configuration:

```
cluster1::> security saml-sp show
Identity Provider URI: https://www.my.idp.com
  Service Provider Host: 1.1.1.1
    Certificate Name: mycert
    Certificate Serial: 1234abcd
    Is SAML Enabled: false
```

## security saml-sp status show

Display SAML service provider configuration status

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

### Description

The `security saml-sp status show` command displays the SAML Service Provider (SP) status for all nodes in the cluster.

### 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 ] }

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

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

This identifies the node in the cluster.

**[-status {not-configured|config-in-progress|config-failed|config-success}] - Update Status (privilege: advanced)**

This identifies the SAML SP status on the specified node.

**[-error-text <text>] - Error Text (privilege: advanced)**

This identifies the error text associated with the latest saml SP update for this node.

## **[-is-enabled {true|false}] - SAML Service Provider Enabled (privilege: advanced)**

When this parameter is set to *true* it indicates that the SAML SP is enabled on this node. Similarly, when this parameter is set to *false*, it indicates that the SAML SP is not enabled on this node.

### **Examples**

The following example displays the SAML SP status information for all nodes in the cluster.

```
cluster:::security saml-sp status> show
Node           SAML SP Status    Enabled
-----
cluster-node1  not-configured   false
cluster-node2  not-configured   false
2 entries were displayed.

cluster:::*>
```

## **security session commands**

### **security session kill-cli**

Kill an active CLI session

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

#### **Description**

The `security session kill-cli` command is used to terminate active CLI sessions. If the session being killed is actively processing a non-read command, the kill will wait until the command is complete before terminating the session. If the session being killed is actively processing a read (show) command, the kill will wait until the current row is returned before terminating the session.

#### **Parameters**

##### **-node {<nodename>|local} - Node**

Selects the sessions that match this parameter value. This identifies the node that is processing the session.

##### **[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that is processing the session.

##### **[-start-time <MM/DD HH:MM:SS>] - Start Time**

Selects the sessions that match this parameter value. This identifies the start time of the current active session.

##### **-session-id <integer> - Session ID**

Selects the sessions that match this parameter value. This number uniquely identifies a management

session within a given node.

#### **[-vserver <vserver>] - Vserver**

Selects the sessions that match this parameter value. This identifies the Vserver associated with this management session.

#### **[-username <text>] - Username**

Selects the sessions that match this parameter value. This identifies the authenticated user associated with this management session.

#### **[-application <text>] - Client Application**

Selects the sessions that match this parameter value. This identifies the calling application by name.

#### **[-location <text>] - Client Location**

Selects the sessions that match this parameter value. This identifies the location of the calling client application. This is typically the IP address of the calling client, or "console" or "localhost" for console or localhost connections.

#### **[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When a session is not actively executing a command request (the session is idle), this indicates the time (in seconds) since the last request completed.

#### **[-state {pending|active|idle}] - Session State**

Selects the sessions that match this parameter value. This identifies the state (pending, active, or idle) of the session. The state is "pending" if it hit a session limit and the session is waiting for another session to end. The state is "idle" for CLI sessions that are waiting at the command prompt. The state is "active" if the session is actively working on a request.

#### **[-request <text>] - Active Command**

Selects the sessions that match this parameter value. This identifies the request (command) that is currently being handled by the session.

### **Examples**

The following example illustrates killing a CLI session by specifying the node and the session id.

```

cluster1::> security session show -node node1

Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 16:58:13 1358      console      console      cluster1 admin
-
      Active Seconds: 0  Request: security session show
03/27 16:58:17 1359      ssh          10.98.16.164      cluster1 admin
650
2 entries were displayed.

cluster1::>

cluster1::> security session kill-cli -node node1 -session-id 1359
1 entry was acted on.

cluster1::> security session show -node node1

Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 16:58:13 1358      console      console      cluster1 admin
-
      Active Seconds: 0  Request: security session show

```

The following example illustrates killing a CLI session by specifying the node and specifying a query on idle-seconds.

```
cluster1::> security session show -node node1

Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 16:58:13 1358      console      console      cluster1 admin
-
      Active Seconds: 0  Request: security session show
03/27 17:13:36 1479      ssh          10.98.16.164      cluster1 admin
83
2 entries were displayed.
```

```
cluster1::> security session kill-cli -node node1 -session-id * -idle
-seconds > 80
1 entry was acted on.
```

```
cluster1::> security session show
```

```
Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 16:58:13 1358      console      console      cluster1 admin
-
      Active Seconds: 0  Request: security session show
```

```
cluster1::>
```

## security session show

Show active CLI & ONTAPI sessions

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

### Description

The `security session show` command displays all active management sessions across the cluster.

### 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 ] }

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

**[-node {<nodename>|local}] - Node**

Selects the sessions that match this parameter value. This identifies the node that is processing the session.

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that is processing the session.

**[-start-time <MM/DD HH:MM:SS>] - Start Time**

Selects the sessions that match this parameter value. This identifies the start time of the current active session.

**[-session-id <integer>] - Session ID**

Selects the sessions that match this parameter value. This number uniquely identifies a management session within a given node.

**[-vserver <vserver>] - Vserver**

Selects the sessions that match this parameter value. This identifies the Vserver associated with this management session.

**[-username <text>] - Username**

Selects the sessions that match this parameter value. This identifies the authenticated user associated with this management session.

**[-application <text>] - Client Application**

Selects the sessions that match this parameter value. This identifies the calling application by name.

**[-location <text>] - Client Location**

Selects the sessions that match this parameter value. This identifies the location of the calling client application. This is typically the IP address of the calling client, or "console" or "localhost" for console or localhost connections.

**[-ipspace <IPspace>] - IPspace of Location**

Selects the sessions that match this parameter value. This identifies the IPspace of the client location.

**[-total <integer>] - Total Requests**

Selects the sessions that match this parameter value. This identifies the total number of requests that have been made thus far in the active session. The following commands are not counted: top, up, cd, rows, history, exit.

**[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that have failed for any reason (including if they were blocked by configured limits).

**[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took for this session.

**[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took for this session.

**[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that has been taken by all completed requests for the current session; it does not include session idle time.

**[-state {pending|active|idle}] - Session State**

Selects the sessions that match this parameter value. This identifies the state (pending, active, or idle) of the session. The state is "pending" if it hit a session limit and the session is waiting for another session to end. The state is "idle" for CLI sessions that are waiting at the command prompt. The state is "active" if the session is actively working on a request.

**[-request <text>] - Request Input**

Selects the sessions that match this parameter value. This identifies the request (command) that is currently being handled by the session.

**[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When a session is not actively executing a command request (the session is idle), this indicates the time (in seconds) since the last request completed.

**[-active-seconds <integer>] - Active Seconds**

Selects the sessions that match this parameter value. When a session is actively executing a command request, this indicates the time (in seconds) since the current request started.

## Examples

The following example illustrates displaying all active sessions across the cluster. In this example, we see one active session on node *node2* from the *console* application. We also see three active sessions on node *node1*. One is from the *console* application and two are from the *ssh* application. Also one of the *ssh* sessions is from user *diag* and the other *ssh* session is from user *admin*.

```

cluster1::> security session show

Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 16:58:13 1358      console      console      cluster1 admin
-
      Active Seconds: 0  Request: security session show
03/27 17:17:04 1514      ssh          10.98.16.164    cluster1 admin
139
03/27 17:17:29 1515      ssh          10.98.16.164    cluster1 diag
115

Node: node2           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 17:18:54 1509      console      console      cluster1 admin
23
4 entries were displayed.

cluster1::>

```

The following example illustrates displaying all active sessions that have been idle for longer than 500 seconds.

```

cluster1::> security session show -idle-seconds > 500

Node: node1           Interface: cli
Idle
Start Time      Sess ID Application Location      Vserver Username
Seconds

-----
-----
03/27 17:17:04 1514      ssh          10.98.16.164      cluster1 admin
607
03/27 17:17:29 1515      ssh          10.98.16.164      cluster1 diag
583
2 entries were displayed.

cluster1::>

```

## security session limit create

Create default session limit

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

### Description

This command allows creation of a default management session limit that does not yet exist. The default limits can be overridden for specific values within each category by using advanced privilege level commands.

### Parameters

#### **-interface {cli|ontapi} - Interface**

The interface (CLI or ONTAPI) to which the limit applies.

#### **-category {application|location|request|user|vserver} - Category**

The session type for this default limit. The following categories are supported: application, location, request, user, Vserver.

#### **-max-active-limit <integer> - Max-Active Limit**

The maximum number of concurrent sessions allowed for this interface and category.

### Examples

The following example illustrates creating a default limit for management sessions using the same application.

```

cluster1::> security session limit create -interface ontapi -category
application -max-active-limit 8

```

## security session limit delete

Delete default session limit

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

### Description

This command allows deletion of a default management session limit.

### Parameters

#### **-interface {cli|ontapi} - Interface**

The interface (CLI or ONTAPI) to which the limit applies.

#### **-category {application|location|request|user|vserver} - Category**

The session type for this default limit. The following categories are supported: application, location, request, user, Vserver.

### Examples

The following example illustrates deleting all default limits for CLI management sessions.

```
cluster1::> security session limit delete -interface cli -category *
3 entries were deleted.
```

## security session limit modify

Modify default session limit

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

### Description

This command allows modification of a default management session limit.

### Parameters

#### **-interface {cli|ontapi} - Interface**

The interface (CLI or ONTAPI) to which the limit applies.

#### **-category {application|location|request|user|vserver} - Category**

The session type for this default limit. The following categories are supported: application, location, request, user, Vserver.

#### **[-max-active-limit <integer>] - Max-Active Limit**

The maximum number of concurrent sessions allowed for this interface and category.

## Examples

The following example illustrates modifying the default limit for CLI management sessions from the same location.

```
cluster1::> security session limit modify -interface cli -category
location -max-active-limit 4
```

## security session limit show

Show default session limits

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

### Description

This command shows the default management session limits that have been configured for each interface and category.

### 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 ] }

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

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

**[-category {application|location|request|user|vserver}] - Category**

Selects the sessions that match this parameter value. This identifies the category for the limit. The following categories are supported: application, location, request, user, and Vserver.

**[-max-active-limit <integer>] - Max-Active Limit**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying the default limits for management sessions.

```
cluster1::> security session limit show
Interface Category      Max-Active
-----
cli      user            2
cli      vserver          4
ontapi   vserver          2
3 entries were displayed.
```

## security session limit application create

Create per-application session limit

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

### Description

This command allows creation of a per-application management session limit that does not yet exist.

### Parameters

**-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

**-application <text> - Application (privilege: advanced)**

The specified application to which this limit applies. The limit with the application name *-default-* is the limit used for any application without a specific configured limit.

**-max-active-limit <integer> - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and application.

### Examples

The following example illustrates creating a limit for management sessions from a custom application.

```
cluster1::*> security session limit application create -interface ontapi
               -application "custom_app" -max-active-limit 8
```

## security session limit application delete

Delete per-application session limit

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

### Description

This command allows deletion of a per-application management session limit.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-application <text> - Application (privilege: advanced)**

The specified application to which this limit applies. The limit with the application name `-default-` is the limit used for any application without a specific configured limit.

## Examples

The following example illustrates deleting a limit for management sessions from a custom application.

```
cluster1::*> security session limit application delete -interface ontapi  
-application "custom_app"
```

## security session limit application modify

Modify per-application session limit

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

## Description

This command allows modification of a per-application management session limit.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-application <text> - Application (privilege: advanced)**

The specified application to which this limit applies. The limit with the application name `-default-` is the limit used for any application without a specific configured limit.

### **[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and application.

## Examples

The following example illustrates modifying management session limits for some custom applications.

```
cluster1::*> security session limit application modify -interface ontapi  
-application custom* -max-active-limit 4  
2 entries were modified.
```

## security session limit application show

Show per-application session limits

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

### Description

This command shows the per-application management session limits that have been configured for each interface and application.

### Parameters

**{ [-fields <fieldname>,...]**

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

**| [-instance ] }**

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

**[-interface {cli|ontapi}] - Interface (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

**[-application <text>] - Application (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the application for the limit. The limit with the application name `-default-` is the limit used for any application without a specific configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

### Examples

The following example illustrates displaying the per-application limits for ONTAPI management sessions.

```
cluster1::*> security session limit application show -interface ontapi
Interface Application      Max-Active
----- -----
ontapi      -default-      5
ontapi      custom_app     10
2 entries were displayed.
```

## security session limit location create

Create per-location session limit

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

## Description

This command allows creation of a per-location management session limit that does not yet exist.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-location <text> - Location (privilege: advanced)**

The specified location to which this limit applies. The limit with the location name *-default-* (in the *Default* IPspace) is the limit used for any location (in any IPspace) without a specific configured limit.

### **[-ipspace <IPspace>] - IPspace of Location (privilege: advanced)**

This identifies the IPspace of the client location. If not specified, changes are made in the *Default* IPspace.

### **-max-active-limit <integer> - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and location.

## Examples

The following example illustrates creating a CLI limit for specific location.

```
cluster1::*> security session limit location create -interface cli  
-location 10.98.16.164 -max-active-limit 1
```

## **security session limit location delete**

### Delete per-location session limit

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

## Description

This command allows deletion of a per-location management session limit.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-location <text> - Location (privilege: advanced)**

The specified location to which this limit applies. The limit with the location name *-default-* (in the *Default* IPspace) is the limit used for any location (in any IPspace) without a specific configured limit.

### **[-ipspace <IPspace>] - IPspace of Location (privilege: advanced)**

This identifies the IPspace of the client location. If not specified, changes are made in the *Default* IPspace.

## Examples

The following example illustrates deleting limits for management sessions from a specific set of locations.

```
cluster1::*> security session limit location delete -interface * -location
10.98.1
3 entries were deleted.
```

## security session limit location modify

Modify per-location session limit

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

### Description

This command allows modification of a per-location management session limit.

### Parameters

#### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

#### **-location <text> - Location (privilege: advanced)**

The specified location to which this limit applies. The limit with the location name *-default-* (in the *Default* IPspace) is the limit used for any location (in any IPspace) without a specific configured limit.

#### **[-ipspace <IPspace>] - IPspace of Location (privilege: advanced)**

This identifies the IPspace of the client location. If not specified, changes are made in the *Default* IPspace.

#### **[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and location.

## Examples

The following example illustrates modifying management sessions limits for specific locations.

```
cluster1::*> security session limit location modify -interface * -location
10.98.1 -max-active-limit 2
3 entries were modified.
```

## security session limit location show

Show per-location session limits

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

## Description

This command shows the per-location management session limits that have been configured for each interface and location.

## Parameters

{ [-fields <fieldname>,...]

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

| [-instance ] }

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

[-interface {cli|ontapi}] - Interface (privilege: advanced)

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

[-location <text>] - Location (privilege: advanced)

Selects the sessions that match this parameter value. This identifies the location for the limit. The limit with the location name `-default-` (only in the *Default* IPspace) is the limit used for any location (in any IPspace) without a specific configured limit.

[-ipspace <IPspace>] - IPspace of Location (privilege: advanced)

Selects the sessions that match this parameter value. This identifies the IPspace of the client location. The default IPspace is *Default*.

[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying the per-location limits for management sessions.

```
cluster1::*> security session limit location show
Interface Location          IPspace      Max-Active
----- -----
cli      -default-          Default      16
cli      10.98.16.164        Default      0
ontapi   -default-          Default      6
ontapi   10.98.16.164        Default      0
4 entries were displayed.
```

## security session limit request create

Create per-request session limit

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

## Description

This command allows creation of a per-request management session limit that does not yet exist.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-request <text> - Request Name (privilege: advanced)**

The specified request to which this limit applies. The limit with the request name *-default-* is the limit used for any request without a specific configured limit.

### **-max-active-limit <integer> - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and request.

## Examples

The following example illustrates creating a limit for number of clients executing a specific API.

```
cluster1::*> security session limit request create -interface ontapi
             -request storage-disk-get-iter -max-active-limit 2
```

## **security session limit request delete**

Delete per-request session limit

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

## Description

This command allows deletion of a per-request management session limit.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-request <text> - Request Name (privilege: advanced)**

The specified request to which this limit applies. The limit with the request name *-default-* is the limit used for any request without a specific configured limit.

## Examples

The following example illustrates deleting custom limits for that were configured for the volume commands and APIs.

```
cluster1::*> security session limit request delete -interface * -request
volume*
4 entries were deleted.
```

## security session limit request modify

Modify per-request session limit

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

### Description

This command allows modification of a per-request management session limit.

### Parameters

**-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

**-request <text> - Request Name (privilege: advanced)**

The specified request to which this limit applies. The limit with the request name *-default* is the limit used for any request without a specific configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and request.

### Examples

The following example illustrates modifying the limit of the number of clients simultaneously executing a specific API.

```
cluster1::*> security session limit request modify -interface ontapi
-request storage-disk-get-iter -max-active-limit 4
```

## security session limit request show

Show per-request session limits

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

### Description

This command shows the per-request management session limits that have been configured for each interface and request.

### 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 ] }

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

**[-interface {cli|ontapi}] - Interface (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

**[-request <text>] - Request Name (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the request (command or API) for the limit. The limit with the request name -default- is the limit used for any request without a specific configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying the per-request limits for management sessions.

```
cluster1::*> security session limit request show
Interface Request           Max-Active
-----
cli      -default-          10
ontapi   -default-          5
ontapi   storage-disk-get-iter  2
3 entries were displayed.
```

## security session limit user create

Create per-user session limit

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

### Description

This command allows creation of a per-user management session limit that does not yet exist.

### Parameters

**-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

### **-user <text> - User (privilege: advanced)**

The specified user to which this limit applies. The limit with the user name *-default-* is the limit used for any user without a specific configured limit.

### **-max-active-limit <integer> - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface, Vserver, and user.

## **Examples**

The following example illustrates creating a per-user limit override for ONTAPI requests for the *admin* user in the admin Vserver.

```
cluster1::*> security session limit user create -interface ontapi -vserver
cluster1 -username admin -max-active-limit 16
```

## **security session limit user delete**

### **Delete per-user session limit**

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

## **Description**

This command allows deletion of a per-user management session limit.

## **Parameters**

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

### **-user <text> - User (privilege: advanced)**

The specified user to which this limit applies. The limit with the user name *-default-* is the limit used for any user without a specific configured limit.

## **Examples**

The following example illustrates deleting all user-specific limits for CLI management sessions.

```
cluster1::*> security session limit user delete -interface cli -user !"-  
default"  
2 entries were deleted.
```

## security session limit user modify

Modify per-user session limit

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

### Description

This command allows modification of a per-user management session limit.

### Parameters

#### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

#### **-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

#### **-user <text> - User (privilege: advanced)**

The specified user to which this limit applies. The limit with the user name *-default-* is the limit used for any user without a specific configured limit.

#### **[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface, Vserver, and user.

### Examples

The following example illustrates modifying the admin user's limit for CLI management sessions.

```
cluster1::*> security session limit user modify -interface cli -vserver  
cluster1 -username admin -max-active-limit 30
```

## security session limit user show

Show per-user session limits

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

### Description

This command shows the per-user management session limits that have been configured for each interface, Vserver, and user.

## 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 ] }

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

**[-interface {cli|ontapi}] - Interface (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

**[-vserver <vserver>] - Vserver (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the Vserver for the limit. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

**[-user <text>] - User (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the user for the limit. The limit with the user name `-default-` is the limit used for any user without a specific configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying the per-user limits for CLI management sessions. In this example, there is a default limit of 4 sessions for each user. That limit is expanded to 8 for the admin Vserver. That limit is further expanded to 20 for the `admin` user in the admin Vserver.

```
cluster1::*> security session limit user show -interface cli
Interface Vserver           User           Max-Active
-----
cli      Cluster           -default-        4
cli      cluster1          -default-        8
cli      cluster1          admin           20
3 entries were displayed.
```

## security session limit vserver create

Create per-vserver session limit

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

### Description

This command allows creation of a per-Vserver management session limit that does not yet exist.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

### **-max-active-limit <integer> - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and Vserver.

## Examples

The following example illustrates creating a per-Vserver limit override for ONTAPI requests on the admin Vserver.

```
cluster1::*> security session limit vserver create -interface ontapi  
-vserver cluster1 -max-active-limit 4
```

## **security session limit vserver delete**

Delete per-vserver session limit

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

## Description

This command allows deletion of a per-Vserver management session limit. The "Cluster" vserver is used when the specific Vserver doesn't have a configured limit.

## Parameters

### **-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

### **-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

## Examples

The following example illustrates deleting all per-Vserver limits for management sessions except the default limit.

```
cluster1::*> security session limit vserver delete -interface * -vserver  
!Cluster  
1 entries was deleted.
```

## security session limit vserver modify

Modify per-vserver session limit

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

### Description

This command allows modification of a per-Vserver management session limit.

### Parameters

**-interface {cli|ontapi} - Interface (privilege: advanced)**

The interface (CLI or ONTAPI) to which the limit applies.

**-vserver <vserver> - Vserver (privilege: advanced)**

The specified Vserver to which this limit applies. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

The maximum number of concurrent sessions allowed for this interface and Vserver.

### Examples

The following example illustrates modifying the admin Vserver's limit for CLI management sessions.

```
cluster1::*> security session limit vserver modify -interface cli -vserver
cluster1 -max-active-limit 40
```

## security session limit vserver show

Show per-vserver session limits

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

### Description

This command shows the per-Vserver management session limits that have been configured for each interface and Vserver.

### 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 ] }**

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

**[-interface {cli|ontapi}] - Interface (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) to which the limit applies.

**[-vserver <vserver>] - Vserver (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the Vserver for the limit. The "Cluster" Vserver is used to limit Vservers that do not have a configured limit.

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying the per-Vserver limits for management sessions.

```
cluster1::*> security session limit vserver show
Interface Vserver          Max-Active
-----
cli      Cluster           4
ontapi   Cluster           2
ontapi   cluster1          16
3 entries were displayed.
```

## security session request-statistics show-by-application

Show session request statistics by application

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

### Description

The *security session request-statistics show-by-application* command shows historical statistics for management session activity, categorized by application name. CLI sessions connections will have an application name based on the connection method, i.e.: *ssh*, *telnet*, *rsh*, *console*, or *ngsh*. ONTAPI sessions will extract the application name from the ZAPI request. ONTAP looks for the application name in the following three locations, in the following order of precedence:

1. The "X-Dot-Client-App" HTTP header;
2. The "app-name" attribute of the "netapp" element, within the ZAPI XML request;
3. The "User-Agent" HTTP header.

### 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 ] }**

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

**[-node {<nodename>|local}] - Node**

Selects the sessions that match this parameter value. This identifies the node that processed the session.

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that processed the session.

**[-application <text>] - Application**

Selects the sessions that match this parameter value. This identifies the calling application by name.

**[-total <integer>] - Total Requests**

Selects the sessions that match this parameter value. This identifies the total number of requests that have been made on a session. The following commands are not counted: top, up, cd, rows, history, exit.

**[-blocked <integer>] - Blocked Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that were blocked due to configured limits.

**[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took.

**[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took.

**[-active <integer>] - Number Active Now**

Selects the sessions that match this parameter value. This identifies the number of currently active sessions.

**[-max-active <integer>] - Max Number Active**

Selects the sessions that match this parameter value. This identifies the maximum number of concurrently active sessions.

**[-last-active-seconds <integer>] - Seconds Since Last Session Start**

Selects the sessions that match this parameter value. When a session is active, this indicates the time (in seconds) since the last session started.

**[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When no sessions are active, this indicates the time (in seconds) since the last session ended.

**[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that have been taken by all completed requests; it does not include session idle time.

**[-average-time <integer>] - Average Time (ms)**

Selects the sessions that match this parameter value. This identifies the mean time spent processing requests.

**[-success-percent <percent>] - Success Percent**

Selects the sessions that match this parameter value. This identifies the percentage of successful requests.

**[-blocked-percent <percent>] - Blocked Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that were blocked due to configured limits.

**[-failed-percent <percent>] - Failed Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying historical statistics for all management session activity across the cluster, categorized by application name.

```

cluster1::> security session request-statistics show-by-application

Node: node1           Interface: cli           Idle   Total
Application          Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)               

-----
-----


console              2126  0   6  95%  96       68      361
170
ssh                  6    2   3 100%  0        -       794
132444

Node: node1           Interface: ontapi        Idle   Total
Application          Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)               

-----
-----


api_test             2    0   1 100%  0        13      0
18

Node: node2           Interface: cli           Idle   Total
Application          Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)               

-----
-----


console              2090  0   6  95%  96       90      655
313

4 entries were displayed.

cluster1::>

```

The following example illustrates displaying historical statistics for management session activity on a specific node and for a specific application.

```

cluster1::> security session request-statistics show-by-application -node
node1 -application api_test

Node: node1           Interface: ontapi           Idle   Total
Application          Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----
api_test              2    0    1 100%    0        102        0
18

cluster1::>

```

## security session request-statistics show-by-location

Show session request statistics by location

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

### Description

The `security session request-statistics show-by-location` command shows historical statistics for management session activity, categorized by client location.

### 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 ] }**

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

**[-node {<nodename>|local}] - Node**

Selects the sessions that match this parameter value. This identifies the node that processed the session.

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that processed the session.

**[-location <text>] - Client Location**

Selects the sessions that match this parameter value. This identifies the location of the calling client application. This is typically the IP address of the calling client, or "console" or "localhost" for console or localhost connections.

**[-ipspace <IPspace>] - IPspace of Location**

Selects the sessions that match this parameter value. This identifies the IPspace of the client location.

**[-total <integer>] - Total Requests**

Selects the sessions that match this parameter value. This identifies the total number of requests that have been made on a session. The following commands are not counted: top, up, cd, rows, history, exit.

**[-blocked <integer>] - Blocked Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that were blocked due to configured limits.

**[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took.

**[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took.

**[-active <integer>] - Number Active Now**

Selects the sessions that match this parameter value. This identifies the number of currently active sessions.

**[-max-active <integer>] - Max Number Active**

Selects the sessions that match this parameter value. This identifies the maximum number of concurrently active sessions.

**[-last-active-seconds <integer>] - Seconds Since Last Session Start**

Selects the sessions that match this parameter value. When a session is active, this indicates the time (in seconds) since the last session started.

**[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When no sessions are active, this indicates the time (in seconds) since the last session ended.

**[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that have been taken by all completed requests; it does not include session idle time.

**[-average-time <integer>] - Average Time (ms)**

Selects the sessions that match this parameter value. This identifies the mean time spent processing requests.

**[-success-percent <percent>] - Success Percent**

Selects the sessions that match this parameter value. This identifies the percentage of successful requests.

**[-blocked-percent <percent>] - Blocked Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that were blocked due to configured limits.

**[-failed-percent <percent>] - Failed Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## Examples

The following example illustrates displaying historical statistics for all management session activity across the cluster, categorized by location.

```
cluster1::> security session request-statistics show-by-location

Node: node1          Interface: cli          Idle      Total
Location      IPspace      Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----
console      Default      21   1   1 100%   0          -        127
6063
localhost    Default      2523  0   5 95%   115        20        280
111

Node: node1          Interface: ontapi        Idle      Total
Location      IPspace      Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----
10.98.17.254  Default      2   0   1 100%   0        2419        0
18

Node: node2          Interface: cli          Idle      Total
Location      IPspace      Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----
console      Default      6   0   1 83%    1        2941      423
70557
localhost    Default      2502  0   5 95%   114        41        277
110
7 entries were displayed.

cluster1::>
```

The following example illustrates displaying historical statistics for management session activity on a specific

node and for a specific location.

```
cluster1::> security session request-statistics show-by-location -node
node2 -location localhost

Node: node2           Interface: cli           Idle   Total
Location      IPspace   Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----  -----
localhost      Default    2524   0   5  95%  115        30      279
110

cluster1::>
```

## security session request-statistics show-by-request

Show session request statistics by request name

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

### Description

The *security session request-statistics show-by-request* command shows historical statistics for management session activity, categorized by request (command or API name).

### 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 ] }

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

[-node {<nodename>|local}] - Node

Selects the sessions that match this parameter value. This identifies the node that processed the session.

[-interface {cli|ontapi}] - Interface

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that processed the session.

[-request <text>] - Request Name

Selects the sessions that match this parameter value. This identifies the command associated with these requests.

[-total <integer>] - Total Requests

Selects the sessions that match this parameter value. This identifies the total number of requests that have

been made on a session. The following commands are not counted: top, up, cd, rows, history, exit.

#### **[-blocked <integer>] - Blocked Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that were blocked due to configured limits.

#### **[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that failed for any reason (including if they were blocked by configured limits).

#### **[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took.

#### **[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took.

#### **[-active <integer>] - Number Active Now**

Selects the sessions that match this parameter value. This identifies the number of currently active requests.

#### **[-max-active <integer>] - Max Number Active**

Selects the sessions that match this parameter value. This identifies the maximum number of concurrently active requests.

#### **[-last-active-seconds <integer>] - Seconds Since Last Request Start**

Selects the sessions that match this parameter value. When requests are active, this indicates the time (in seconds) since the last request started.

#### **[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When no requests are active, this indicates the time (in seconds) since the last request ended.

#### **[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that have been taken by all completed requests; it does not include session idle time.

#### **[-average-time <integer>] - Average Time (ms)**

Selects the sessions that match this parameter value. This identifies the mean time spent processing requests.

#### **[-success-percent <percent>] - Success Percent**

Selects the sessions that match this parameter value. This identifies the percentage of successful requests.

#### **[-blocked-percent <percent>] - Blocked Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that were blocked due to configured limits.

### **[-failed-percent <percent>] - Failed Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that failed for any reason (including if they were blocked by configured limits).

### **[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## **Examples**

The following example illustrates displaying historical statistics for all management session activity on a specific node, with a specific request query.

```
cluster1::> security session request-statistics show-by-request -node
node1 -request network*
Node: node1                                Interface: cli
Request Name                               Total  Now  Max  Pass  Fail  Idle   Total
(ms)                                         Total    Avg
-----  -----  -----  -----  -----  -----  -----  -----
-----  -----
network interface create          2      0      1  100%      0    2556      0
485
network interface modify          1      0      1  100%      0    2518      0
34
network interface show           8      0      1  100%      0    2152     12
1614
network route create            1      0      1  100%      0    2135      0
45
network route show              2      0      1  100%      0    2145      0
17
5 entries were displayed.

cluster1::>
```

## **security session request-statistics show-by-user**

Show session request statistics by username

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

## **Description**

The *security session request-statistics show-by-user* command shows historical statistics for management session activity, categorized by username. Entries for username 'autosupport' reflect commands that are executed by the AutoSupport OnDemand feature.

## 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 ] }

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

**[-node {<nodename>|local}] - Node**

Selects the sessions that match this parameter value. This identifies the node that processed the session.

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that processed the session.

**[-vserver <vserver>] - Vserver**

Selects the sessions that match this parameter value. This identifies the Vserver associated with this management session.

**[-username <text>] - Username**

Selects the sessions that match this parameter value. This identifies the authenticated user associated with this management session.

**[-total <integer>] - Total Requests**

Selects the sessions that match this parameter value. This identifies the total number of requests that have been made on a session. The following commands are not counted: top, up, cd, rows, history, exit.

**[-blocked <integer>] - Blocked Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that were blocked due to configured limits.

**[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took.

**[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took.

**[-active <integer>] - Number Active Now**

Selects the sessions that match this parameter value. This identifies the number of currently active sessions.

**[-max-active <integer>] - Max Number Active**

Selects the sessions that match this parameter value. This identifies the maximum number of concurrently

active sessions.

#### **[-last-active-seconds <integer>] - Seconds Since Last Session Start**

Selects the sessions that match this parameter value. When a session is active, this indicates the time (in seconds) since the last session started.

#### **[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When no sessions are active, this indicates the time (in seconds) since the last session ended.

#### **[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that have been taken by all completed requests; it does not include session idle time.

#### **[-average-time <integer>] - Average Time (ms)**

Selects the sessions that match this parameter value. This identifies the mean time spent processing requests.

#### **[-success-percent <percent>] - Success Percent**

Selects the sessions that match this parameter value. This identifies the percentage of successful requests.

#### **[-blocked-percent <percent>] - Blocked Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that were blocked due to configured limits.

#### **[-failed-percent <percent>] - Failed Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that failed for any reason (including if they were blocked by configured limits).

#### **[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

## **Examples**

The following example illustrates displaying historical statistics for all management session activity across the cluster, categorized by username.

```

cluster1::> security session request-statistics show-by-user

Node: node1           Interface: cli           Idle   Total
Vserver      Username   Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)

-----
----- cluster1      admin      81   1   3  80%   16      -      1228
15171
1511958      diag       1   0   1 100%   0      1982      1511
17
17      autosupport   4   0   1 100%   0      -      0

Node: node1           Interface: ontapi          Idle   Total
Vserver      Username   Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)

-----
----- cluster1      admin      2   0   1 100%   0      2585      0
18

Node: node2           Interface: cli           Idle   Total
Vserver      Username   Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)

-----
----- cluster1      admin      6   1   1  83%   1      3106      423
70557
4 entries were displayed.

cluster1::>

```

The following example illustrates displaying historical statistics for management session activity on a specific node and for a specific username.

```

cluster1::> security session request-statistics show-by-user -node node1
-username diag

Node: node1           Interface: cli           Idle   Total
Vserver      Username   Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
-----  -----
cluster1      diag       1   0   1 100%   0        -      1511
1511958

cluster1::>

```

## security session request-statistics show-by-vserver

Show session request statistics by Vserver

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

### Description

The `security session request-statistics show-by-vserver` command shows historical statistics for management session activity, categorized by vserver.

### 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 ] }**

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

**[-node {<nodename>|local}] - Node**

Selects the sessions that match this parameter value. This identifies the node that processed the session.

**[-interface {cli|ontapi}] - Interface**

Selects the sessions that match this parameter value. This identifies the interface (CLI or ONTAPI) that processed the session.

**[-vserver <vserver>] - Vserver**

Selects the sessions that match this parameter value. This identifies the Vserver associated with this management session.

**[-total <integer>] - Total Requests**

Selects the sessions that match this parameter value. This identifies the total number of requests that have been made on a session. The following commands are not counted: top, up, cd, rows, history, exit.

**[-blocked <integer>] - Blocked Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that were blocked due to configured limits.

**[-failed <integer>] - Failed Requests**

Selects the sessions that match this parameter value. This identifies the number of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-time <integer>] - Maximum Time (ms)**

Selects the sessions that match this parameter value. This identifies the maximum amount of time (in milliseconds) that any request took.

**[-last-time <integer>] - Last Time (ms)**

Selects the sessions that match this parameter value. This identifies the amount of time (in milliseconds) that the last request took.

**[-active <integer>] - Number Active Now**

Selects the sessions that match this parameter value. This identifies the number of currently active sessions.

**[-max-active <integer>] - Max Number Active**

Selects the sessions that match this parameter value. This identifies the maximum number of concurrently active sessions.

**[-last-active-seconds <integer>] - Seconds Since Last Session Start**

Selects the sessions that match this parameter value. When a session is active, this indicates the time (in seconds) since the last session started.

**[-idle-seconds <integer>] - Idle Seconds**

Selects the sessions that match this parameter value. When no sessions are active, this indicates the time (in seconds) since the last session ended.

**[-total-seconds <integer>] - Total Seconds**

Selects the sessions that match this parameter value. This identifies the total time (in seconds) that have been taken by all completed requests; it does not include session idle time.

**[-average-time <integer>] - Average Time (ms)**

Selects the sessions that match this parameter value. This identifies the mean time spent processing requests.

**[-success-percent <percent>] - Success Percent**

Selects the sessions that match this parameter value. This identifies the percentage of successful requests.

**[-blocked-percent <percent>] - Blocked Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that were blocked due to configured limits.

**[-failed-percent <percent>] - Failed Percent**

Selects the sessions that match this parameter value. This identifies the percentage of requests that failed for any reason (including if they were blocked by configured limits).

**[-max-active-limit <integer>] - Max-Active Limit (privilege: advanced)**

Selects the sessions that match this parameter value. This identifies the configured limit that is used to throttle or reject requests.

**Examples**

The following example illustrates displaying historical statistics for all management session activity across the cluster, categorized by Vserver.

```
cluster1::> security session request-statistics show-by-vserver

Node: node1           Interface: cli           Idle   Total
Vserver                  Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
----- cluster1           2725   1   8  94%  146      -    3052
1120

Node: node1           Interface: ontapi          Idle   Total
Vserver                  Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
----- cluster1           2     0   1 100%   0      2742      0
18

Node: node2           Interface: cli           Idle   Total
Vserver                  Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
----- cluster1           2552   1   6  95%  117      -    705
276
3 entries were displayed.

cluster1::>
```

The following example illustrates displaying historical statistics for management session activity on a specific node, for a specific Vserver.

```

cluster1::> security session request-statistics show-by-vserver -node
node1 -vserver cluster1

Node: node1           Interface: cli           Idle   Total
Vserver                      Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
----- cluster1           2747   1   8  94%  147      -    3055
1112

Node: node1           Interface: ontapi        Idle   Total
Vserver                      Total Now Max Pass Fail  Seconds  Seconds Avg
(ms)
-----
----- cluster1           2     0   1 100%   0      2902      0
18
2 entries were displayed.

cluster1::>

```

## security ssh commands

### security ssh add

Add SSH configuration options

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The ` security ssh add ` command adds additional SSH key exchange algorithms or ciphers or MAC algorithms to the existing configurations of the cluster or a Vserver. The added algorithms or ciphers or MAC algorithms are enabled on the cluster or Vserver. If you change the cluster configuration settings, it is used as the default for all newly created Vservers. The existing SSH key exchange algorithms, ciphers, and MAC algorithms remain unchanged in the configuration. If the SSH key exchange algorithms or ciphers or MAC algorithms are already enabled in the current configuration, the command will not fail. Data ONTAP supports the *diffie-hellman-group-exchange-sha256* key exchange algorithm for SHA-2. Data ONTAP also supports the *diffie-hellman-group-exchange-sha1* , *diffie-hellman-group14-sha1* , and *diffie-hellman-group1-sha1* SSH key exchange algorithms for SHA-1. The SHA-2 key exchange algorithm is more secure than the SHA-1 key exchange algorithms. Data ONTAP also supports *ecdh-sha2-nistp256* , *ecdh-sha2-nistp384* , *ecdh-sha2-nistp521* , and *curve25519-sha256* . Data ONTAP also supports the AES and 3DES symmetric encryptions (also known as ciphers) of the following types: *aes256-ctr* , *aes192-ctr* , *aes128-ctr* , *aes256-cbc* , *aes192-cbc* , *aes128-cbc* , *aes128-gcm* , *aes256-gcm* , and *3des-cbc* . Data ONTAP supports MAC algorithms of the following types: *hmac-sha1* , *hmac-sha1-96* , *hmac-md5* , *hmac-md5-96* , *hmac-ripemd160* , *umac-64* , *umac-64* , *umac-128* , *hmac-sha2-256* , *hmac-sha2-512* , *hmac-sha1-etm* , *hmac-sha1-96-etm* , *hmac-sha2-256-etm* , *hmac-*

*sha2-512-etm*, *hmac-md5-etm*, *hmac-md5-96-etm*, *hmac-ripemd160-etm*, *umac-64-etm*, and *umac-128-etm*.

## Parameters

### **-vserver <Vserver Name> - Vserver**

Identifies the Vserver to which you want to add additional SSH key exchange algorithms or ciphers.

### **[-key-exchange-algorithms <algorithm name>, ...] - List of SSH Key Exchange Algorithms to Add**

Adds the specified SSH key exchange algorithm or algorithms to the Vserver.

### **[-ciphers <cipher name>, ...] - List of SSH Ciphers to Add**

Adds the specified cipher or ciphers to the Vserver.

### **[-mac-algorithms <MAC name>, ...] - List of SSH MAC Algorithms to Add**

Adds the specified MAC algorithm or algorithms to the Vserver.

## Examples

The following command adds the *diffie-hellman-group-exchange-sha256* and *diffie-hellman-group-exchange-sha1* key exchange algorithms for the cluster1 Vserver. It also adds the *aes256-cbc* and *aes192-cbc* ciphers and the *hmac-sha1* and *hmac-sha2-256* MAC algorithms to the cluster1 Vserver.

```
cluster1::> security ssh add -vserver cluster1 -key-exchange-algorithms  
diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1  
-ciphers aes256-cbc,aes192-cbc -mac-algorithms hmac-sha1,hmac-sha2-256
```

## security ssh modify

### Modify SSH configuration options

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The ` security ssh modify` command replaces the existing configurations of the SSH key exchange algorithms or ciphers or MAC algorithms for the cluster or a Vserver with the configuration settings you specify. If you modify the cluster configuration settings, it will be used as the default for all newly created Vservers. Data ONTAP supports the *diffie-hellman-group-exchange-sha256* key exchange algorithm for SHA-2. Data ONTAP also supports the *diffie-hellman-group-exchange-sha1*, *diffie-hellman-group14-sha1*, and *diffie-hellman-group1-sha1* SSH key exchange algorithms for SHA-1. The SHA-2 key exchange algorithm is more secure than the SHA-1 key exchange algorithms. Data ONTAP also supports the AES and 3DES symmetric encryptions (also known as ciphers) of the following types: *aes256-ctr*, *aes192-ctr*, *aes128-ctr*, *aes256-cbc*, *aes192-cbc*, *aes128-cbc*, *aes128-gcm*, *aes256-gcm*, and *3des-cbc*. Data ONTAP supports MAC algorithms of the following types: *hmac-sha1*, *hmac-sha1-96*, *hmac-md5*, *hmac-md5-96*, *hmac-ripemd160*, *umac-64*, *umac-64*, *umac-128*, *hmac-sha2-256*, *hmac-sha2-512*, *hmac-sha1-etm*, *hmac-sha1-96-etm*, *hmac-sha2-256-etm*, *hmac-sha2-512-etm*, *hmac-md5-etm*, *hmac-md5-96-etm*, *hmac-ripemd160-etm*, *umac-64-etm*, and *umac-128-etm*.

## Parameters

### **-vserver <Vserver Name> - Vserver**

Identifies the Vserver for which you want to replace the existing SSH key exchange algorithm and cipher configurations.

### **[-key-exchange-algorithms <algorithm name>, ...] - Key Exchange Algorithms**

Enables the specified SSH key exchange algorithm or algorithms for the Vserver. This parameter also replaces all existing SSH key exchange algorithms with the specified settings.

### **[-ciphers <cipher name>, ...] - Ciphers**

Enables the specified cipher or ciphers for the Vserver. This parameter also replaces all existing ciphers with the specified settings.

### **[-mac-algorithms <MAC name>, ...] - MAC Algorithms**

Enables the specified MAC algorithm or algorithms for the Vserver. This parameter also replaces all existing MAC algorithms with the specified settings.

### **[-max-authentication-retry-count <integer>] - Max Authentication Retry Count**

Modifies the maximum number of authentication retry count for the Vserver.

## Examples

The following command enables the *diffie-hellman-group-exchange-sha256* and *diffie-hellman-group14-sha1* key exchange algorithms for the cluster1 Vserver. It also enables the *aes256-ctr*, *aes192-ctr* and *aes128-ctr* ciphers, *hmac-sha1* and *hmac-sha2-256* MAC algorithms for the cluster1 Vserver. It also modifies the maximum authentication retry count to 3 for the cluster1 Vserver:

```
cluster1::> security ssh modify -vserver cluster1 -key-exchange-algorithms diffie-hellman-group-exchange-sha256,diffie-hellman-group14-sha1 -ciphers aes256-ctr,aes192-ctr,aes128-ctr -mac-algorithms hmac-sha1,hmac-sha2-256 -max-authentication-retry-count 3
```

## **security ssh prepare-to-downgrade**

Downgrade the SSH configuration to be compatible with releases earlier than Data ONTAP 9.2.0.

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

## Description

This command downgrades the SSH configurations of all Vservers and the cluster to settings compatible with releases earlier than Data ONTAP 9.2.0. This command also disables the max-authentication-retry feature. You must run this command in advanced privilege mode when prompted to do so during the release downgrade. Otherwise, the release downgrade process will fail.

## Examples

The following command downgrades the SSH security configurations of all Vservers and the cluster to settings compatible with releases earlier than Data ONTAP 9.2.0.

```
cluster1::>*> security ssh prepare-to-downgrade
```

## security ssh remove

Remove SSH configuration options

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `security ssh remove` command removes the specified SSH key exchange algorithms or ciphers from the existing configurations of the cluster or a Vserver. The removed algorithms or ciphers are disabled on the cluster or Vserver. If you changed the cluster configuration settings, it will be used as the default for all newly created Vservers. If the SSH key exchange algorithms or ciphers that you specify with this command are not currently enabled, the command will not fail. Data ONTAP supports the *diffie-hellman-group-exchange-sha256* key exchange algorithm for SHA-2. Data ONTAP also supports the *diffie-hellman-group-exchange-sha1*, *diffie-hellman-group14-sha1*, and *diffie-hellman-group1-sha1* SSH key exchange algorithms for SHA-1. The SHA-2 key exchange algorithm is more secure than the SHA-1 key exchange algorithms. Data ONTAP also supports *ecdh-sha2-nistp256*, *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521*, and *curve25519-sha256*. Data ONTAP also supports the AES and 3DES symmetric encryption (also known as ciphers) of the following types: *aes256-ctr*, *aes192-ctr*, *aes128-ctr*, *aes256-cbc*, *aes192-cbc*, *aes128-cbc*, *aes128-gcm*, *aes256-gcm* and *3des-cbc*. Data ONTAP supports MAC algorithms of the following types: *hmac-sha1*, *hmac-sha1-96*, *hmac-md5*, *hmac-md5-96*, *hmac-ripemd160*, *umac-64*, *umac-64*, *umac-128*, *hmac-sha2-256*, *hmac-sha2-512*, *hmac-sha1-etm*, *hmac-sha1-96-etm*, *hmac-sha2-256-etm*, *hmac-sha2-512-etm*, *hmac-md5-etm*, *hmac-md5-96-etm*, *hmac-ripemd160-etm*, *umac-64-etm*, and *umac-128-etm*.

### Parameters

#### **-vserver <Vserver Name>** - Vserver

Identifies the Vserver from which you want to remove the SSH key exchange algorithms or ciphers.

#### **[-key-exchange-algorithms <algorithm name>, ...]** - List of SSH Key Exchange Algorithms to Remove

Removes the specified key exchange algorithm or algorithms from the Vserver.

#### **[-ciphers <cipher name>, ...]** - List of SSH Ciphers to Remove

Removes the specified cipher or ciphers from the Vserver.

#### **[-mac-algorithms <MAC name>, ...]** - List of SSH MAC algorithms to Remove

Removes the specified MAC algorithm or algorithms from the Vserver.

## Examples

The following command removes the *diffie-hellman-group1-sha1* and *diffie-hellman-group-*

`exchange-sha1` key exchange algorithms from the cluster1 Vserver. It also removes the `aes128-cbc` and `3des-cbc` ciphers and the `hmac-sha1-96` and `hmac-sha2-256` MAC algorithms from the cluster1 Vserver.

```
cluster1::> security ssh remove -vserver cluster1 -key-exchange-algorithms
diffie-hellman-group1-sha1,diffie-hellman-group-exchange-sha1 -ciphers
aes128-cbc,3des-cbc -mac-algorithms hmac-sha1-96,hmac-sha2-256
```

## security ssh show

### Display SSH configuration options

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `security ssh show` command displays the configurations of the SSH key exchange algorithms, ciphers, MAC algorithms and maximum authentication retry count for the cluster and Vservers. The SSH protocol uses a Diffie-Hellman based key exchange method to establish a shared secret key during the SSH negotiation phrase. The key exchange method specifies how one-time session keys are generated for encryption and authentication and how the server authentication takes place. Data ONTAP supports the `diffie-hellman-group-exchange-sha256` key exchange algorithm for SHA-2. Data ONTAP also supports the `diffie-hellman-group-exchange-sha1`, `diffie-hellman-group14-sha1`, and `diffie-hellman-group1-sha1` key exchange algorithms for SHA-1. Data ONTAP also supports `ecdh-sha2-nistp256`, `ecdh-sha2-nistp384`, `ecdh-sha2-nistp521`, `curve25519-sha256`. Data ONTAP also supports the AES and 3DES symmetric encryptions (also known as ciphers) of the following types: `aes256-ctr`, `aes192-ctr`, `aes128-ctr`, `aes256-cbc`, `aes192-cbc`, `aes128-cbc`, `aes128-gcm`, `aes256-gcm` and `3des-cbc`. Data ONTAP supports MAC algorithms of the following types: `hmac-sha1`, `hmac-sha1-96`, `hmac-md5`, `hmac-md5-96`, `hmac-ripemd160`, `umac-64`, `umac-64`, `umac-128`, `hmac-sha2-256`, `hmac-sha2-512`, `hmac-sha1-etm`, `hmac-sha1-96-etm`, `hmac-sha2-256-etm`, `hmac-sha2-512-etm`, `hmac-md5-etm`, `hmac-md5-96-etm`, `hmac-ripemd160-etm`, `umac-64-etm`, `umac-128-etm`.

#### 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 ] }**

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

**[-vserver <Vserver Name>] - Vserver**

Identifies the Vserver for which you want to display the SSH key exchange algorithm, cipher, and MAC algorithm configurations.

**[-key-exchange-algorithms <algorithm name>, ...] - Key Exchange Algorithms**

Displays the Vserver or Vservers that have the specified key exchange algorithms enabled.

**[-ciphers <cipher name>, ...] - Ciphers**

Displays the Vserver or Vservers that have the specified ciphers enabled.

**[-mac-algorithms <MAC name>, ...] - MAC Algorithms**

Displays the Vserver or Vservers that have the specified MAC algorithm or algorithms.

**[-max-authentication-retry-count <integer>] - Max Authentication Retry Count**

Displays Vservers with a matching maximum authentication retry count value.

## Examples

The following command displays the enabled SSH key exchange algorithms, ciphers, MAC algorithms and maximum number of authentication retry count for the cluster and all Vservers. The cluster settings are used as the default for all newly created Vservers:

```

cluster-1::> security ssh show
                                         Key Exchange   MAC          Max
                                         Authentication
                                         Vserver      Ciphers      Algorithms      Algorithms      Retry
                                         Count
-----
-----          -----
cluster-1      3des-cbc      diffie-      hmac-sha1
4
                                         hellman-
                                         group-
                                         exchange-
                                         sha256
vs1           aes256-       diffie-      hmac-sha1,
6
                                         ctr,         hellman-
                                         aes192-      group-
                                         ctr,         exchange-
                                         aes128-      sha256,      hmac-sha1-etm,
                                         ctr,         diffie-
                                         aes256-      hellman-
                                         cbc,         group-
                                         aes192-      exchange-
                                         cbc,         sha1,       hmac-sha2-512-
                                         aes128-      diffie-
                                         cbc,         hellman-
                                         3des-cbc,    group14-
                                         aes128-      sha1,       ripemd160,
                                         gcm,         ecdh-sha2-
                                         aes256-gcm  nistp256,   umac-64,
                                         ecdh-sha2-
                                         nistp384,   umac-128,
                                         ecdh-sha2-
                                         nistp521,   hmac-md5-96-
                                         curve25519-
                                         sha256      ripemd160-etm,
                                         umac-64-etm,
                                         umac-128-etm

2 entries were displayed.

```

## security ssl commands

### security ssl modify

Modify the SSL configuration for HTTP servers

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

This command modifies the configuration of encrypted HTTP (SSL) for Vservers in the cluster. Depending on the requirements of the individual node's or cluster's web services (displayed by the [vserver services web show](#) command), this encryption might or might not be used. If the Vserver does not have a certificate associated with it, SSL will not be available.

## Parameters

### **-vserver <Vserver Name>** - Vserver

Identifies a Vserver for hosting SSL-encrypted web services.

### **[-ca <text>]** - Server Certificate Issuing CA

Identifies a Certificate Authority (CA) of a certificate to be associated with the instance of a given Vserver. If this parameter, along with serial, is omitted during modification, a self-signed SSL certificate can be optionally generated for that Vserver.

### **[-serial <text>]** - Server Certificate Serial Number

Identifies a serial number of a certificate to be associated with the instance of a given Vserver. If this parameter, along with ca, is omitted during modification, a self-signed SSL certificate can be optionally generated for that Vserver.

### **[-common-name <FQDN or Custom Common Name>]** - Server Certificate Common Name

Identifies the common name (CN) of a certificate to be associated with the instance of a given Vserver. This parameter becomes optional if serial and ca are specified. You can use the [security certificate create](#) and [security certificate install](#) commands to add new certificates to Vservers.



The use of self-signed SSL certificates exposes users to man-in-the-middle security attacks. Where possible, obtain a certificate that is signed by a reputable certificate authority (CA) and use the [security certificate install](#) command to configure it before enabling SSL on a Vserver.

### **[-server-enabled {true|false}]** - SSL Server Authentication Enabled

Defines the working condition of SSL server authentication in an instance of the Vserver. Any Vserver with a valid certificate of type server is server-enabled.

### **[-client-enabled {true|false}]** - SSL Client Authentication Enabled

Defines the working condition of SSL client authentication in an instance of the Vserver. Any Vserver with a valid certificate of type client-ca is client-enabled. It can only be enabled if server-enabled is true.

### **[-ocsp-enabled {true|false}]** - Online Certificate Status Protocol Validation Enabled

This parameter enables OCSP validation of the client certificate chain. When this parameter is enabled, certificates in the certificate chain of the client will be validated against an OCSP responder after normal verification (including CRL checks) has occurred. The OCSP responder used for validation process is either extracted from the certificate itself, or it is derived by configuration.

### **[-ocsp-default-responder <text>]** - URI of the Default Responder for OCSP Validation

This parameter sets the default OCSP responder to use. If this parameter is not enabled, the URI given will be used only if no responder URI is specified in the certificate that are being verified.

## **[-ocsp-override-responder {true|false}] - Force the Use of the Default Responder URI for OCSP Validation**

This parameter forces the configured default OCSP responder to be used during OCSP certificate validation, even if the certificate that is being validated references an OCSP responder.

## **[-ocsp-responder-timeout <[<integer>h] [<integer>m] [<integer>s]>] - Timeout for OCSP Queries**

Use this parameter to specify the timeout in seconds for OCSP responders. Specify zero for the minimum possible timeout. The default value is 10 seconds.

## **[-ocsp-max-response-age <unsigned32\_or\_unlimited>] - Maximum Allowable Age for OCSP Responses (secs)**

This parameter sets the maximum allowable age (freshness) in seconds for the OCSP responses. The default value for this parameter is unlimited, which does not enforce a maximum age and the OCSP responses are considered valid as long as their expiration date field is in the future.

## **[-ocsp-max-response-time-skew <[<integer>h] [<integer>m] [<integer>s]>] - Maximum Allowable Time Skew for OCSP Response Validation**

This parameter sets the maximum allowable time difference for OCSP responses (when validating their ThisUpdate and NextUpdate fields).

## **[-ocsp-use-request-nonce {true|false}] - Use a NONCE within OCSP Queries**

This parameter determines whether the queries to the OCSP responders should contain a NONCE or not. By default, a query NONCE is always used and checked against the OCSP response. When the responder does not use NONCEs, this parameter should be disabled.



A NONCE is a unique identifier included in each OCSP request or OCSP response to prevent a replay attack.

## **Examples**

The following example enables SSL server authentication for a Vserver named vs0 with a certificate that has ca as www.example.com and serial as 4F4EB629.

```
cluster1::> security ssl modify -vserver vs0 -ca www.example.com -serial 4F4EB629 -server-enabled true
```

The following example disables SSL server authentication for a Vserver name vs0.

```
cluster1::> security ssl modify -vserver vs0 -server-enabled false
```

The following example enables SSL client authentication for a Vserver named vs0.

```
cluster1::> security ssl modify -vserver vs0 -client-enabled true
```

The following example disables SSL client authentication for a Vserver named vs0.

```
cluster1::> security ssl modify -vserver vs0 -client-enabled false
```

## Related Links

- [vserver services web show](#)
- [security certificate create](#)
- [security certificate install](#)

## security ssl show

Display the SSL configuration for HTTP servers

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

This command displays the configuration of encrypted HTTP (SSL) for Vservers in the cluster. Depending on the requirements of the individual node's or cluster's web services (displayed by the [vserver services web show](#) command), this encryption might or might not be used. If the Vserver does not have a certificate associated with it, SSL will not be available.

### 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.

**| [-ocsp ]**

If you specify the `-ocsp` parameter, the command displays the Online Certificate Status Protocol configuration.

**| [-instance ] }**

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

**[-vserver <Vserver Name>] - Vserver**

Identifies a Vserver for hosting SSL-encrypted web services.

**[-ca <text>] - Server Certificate Issuing CA**

Filters the display of SSL configuration by specifying the Certificate Authority (CA) that issued the server certificate.

**[-serial <text>] - Server Certificate Serial Number**

Filters the display of SSL configuration by specifying the serial number of a server certificate.

**[-common-name <FQDN or Custom Common Name>] - Server Certificate Common Name**

Filters the display of SSL configuration by specifying the common name for the server certificate.

**[-server-enabled {true|false}] - SSL Server Authentication Enabled**

Filters the display of SSL configuration according to whether the SSL server authentication is enabled or disabled. Vservers have self-signed certificates automatically generated during their creation. These Vserver self-signed certificates are server-enabled by default.

**[-client-enabled {true|false}] - SSL Client Authentication Enabled**

Filters the display of SSL configuration according to whether the SSL client authentication is enabled or disabled. You can enable client authentication only when server authentication is enabled.

**[-ocsp-enabled {true|false}] - Online Certificate Status Protocol Validation Enabled**

Filters the display of SSL configuration when the Online Certificate Status Protocol validation is enabled.

**[-ocsp-default-responder <text>] - URI of the Default Responder for OCSP Validation**

Filters the display of SSL configuration according to the URI of the default responder for OCSP validation.

**[-ocsp-override-responder {true|false}] - Force the Use of the Default Responder URI for OCSP Validation**

Filters the display of SSL configuration, which forces the use of the default responder URI for OCSP validation.

**[-ocsp-responder-timeout <[<integer>h] [<integer>m] [<integer>s]>] - Timeout for OCSP Queries**

Filters the display of SSL configuration according to the timeout for queries to OCSP responders.

**[-ocsp-max-response-age <unsigned32 or unlimited>] - Maximum Allowable Age for OCSP Responses (secs)**

Filters the display of SSL configuration according to the maximum allowable age (freshness) in seconds for the OCSP responses.

**[-ocsp-max-response-time-skew <[<integer>h] [<integer>m] [<integer>s]>] - Maximum Allowable Time Skew for OCSP Response Validation**

Filters the display of SSL configuration according to the maximum allowable time difference for OCSP responses (when validating their ThisUpdate and NextUpdate fields).

**[-ocsp-use-request-nonce {true|false}] - Use a NONCE within OCSP Queries**

Filters the display of SSL configuration by specifying whether the queries to the OCSP responders should contain a NONCE or not.



A NONCE is a unique identifier included in each OCSP request or OCSP response to prevent a replay attack.

## Examples

The following example displays the configured certificates for Vservers.

```

cluster1::security ssl> show
      Serial
Vserver  Number Common Name          Server  Client
          Enabled Enabled
-----
cluster1  516C3CB3
          cluster1.company.com      true   true
vs0      516816D4
          vs0.company.com        true   false
2 entries were displayed.

```

## Related Links

- [vserver services web show](#)

## Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

**LIMITED RIGHTS LEGEND:** Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.