security login commands

ONTAP 9.15.1 commands

NetApp
May 17, 2024

Table of Contents

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 amqp, 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 as the first factor of 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|nsswitch|domain|totp}]` - Second Authentication Method2

This specifies the authentication method for the login. It will be used as the second factor for authentication. This parameter can be specified for *ssh* and *service-processor* applications only. For *ssh* application, possible values include the following:

- password - Password
- publickey - Public-key authentication
- nsswitch - NIS or LDAP authentication
- domain - Active Directory authentication
- none - default value

1. For *service-processor* application, possible values include the following:

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

.

`[-is-ldap-fastbind {yes|no}]` - LDAP Fastbind Authentication

This flag specifies whether the authentication is LDAP fastbind or Not. Default:false

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 *vs1.netapp.com* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *vs1.netapp.com* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *vs1.netapp.com* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *vs1.netapp.com* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *vs1.netapp.com* :

```
cluster1::> security login create -vserver vs1.netapp.com
  -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 *vs1.netapp.com* . Here *is-ns-switch-group* must be set to *yes* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *`vs1.netapp.com`* :

```
cluster1::> security login create -vserver vs1.netapp.com -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 *`vs1.netapp.com`* :

```
cluster1::> security login create -vserver vs1.netapp.com -user-or-group
-name monitor
  -application ssh -authentication-method password
  -second-authentication-method none -role vsadmin
```

The following example illustrates how to create a login that has the user name *`spuser`* , the application *`service-processor`* , the authentication method *`password`* , the second authentication method *`publickey`* and the access-control role *`admin`* for the Administrative Vserver *`cluster1`* :

```
cluster1::> security login create -vserver cluster1 -user-or-group-name
spuser
  -application service-processor -authentication-method password
  -second-authentication-method publickey -role admin
```

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 amqp, 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 *`vs1.netapp.com`* :

```
cluster1::> security login delete -user-or-group-name guest
  -application ssh -authentication-method password -vserver vs1.netapp.com
```

The following example illustrates how to delete a login that has the username *`guest`* , the application *`ontapi`* , and the authentication method *`cert`* for Vserver *`vs1.netapp.com`* :

```
cluster1::> security login delete -user-or-group-name guest
  -application ontapi -authentication-method cert -vserver vs1.netapp.com
```

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 *vs1.netapp.com* :

```
cluster1::> security login delete -user-or-group-name DOMAIN1\adgroup
  -application ssh -authentication-method domain -vserver vs1.netapp.com
```

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 *vs1.netapp.com* :

```
cluster1::> security login delete -user-or-group-name nssgroup
  -application ontapi -authentication-method nsswitch -vserver
vs1.netapp.com
```

The following example illustrates how to delete a login that has the username *spuser* , the application *service-processor* , the authentication method *password* and the second authentication method *publickey* for the Administrative Vserver *cluster1* :

```
cluster1::> security login delete -user-or-group-name spuser
  -application service-processor -authentication-method password -vserver
cluster1
```

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 alogrithm using `-hash -function` parameter.

Examples

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

```
cluster1::> security login expire-password -vserver vs1.netapp.com
-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.netapp.com' Vserver user account named 'jdoe' that is encrypted with the MD5 hash function.

```
cluster1::> security login expire-password -vserver vs1.netapp.com
-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. This command only applies to user accounts configured with the password authentication method where the password is set.

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.netapp.com'.

```
cluster1::> security login lock -vserver vs1.netapp.com -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 amqp, 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|nsswitch|domain|totp}]` **- Second Authentication Method2**

This specifies the authentication method for the login method. It will be used as the second factor for authentication. This parameter can be specified for *ssh* and *service-processor* applications. For *ssh* application, possible values include the following:

- password - Password
- publickey - Public-key authentication
- nsswitch - NIS or LDAP authentication
- domain - Active Directory authentication
- none - default value

  1. For *service-processor* application, possible values include the following:

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

.

`[-is-ldap-fastbind {yes|no}]` - LDAP Fastbind Authentication

This flag specifies whether modify is allowed or not when the authentication is LDAP fastbind.

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 *vs1.netapp.com* :

```
cluster1::> security login modify -user-or-group-name guest
  -application ontapi -authentication-method password -role guest
  -vserver vs1.netapp.com
```

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 *vs1.netapp.com* :

```
cluster1::> security login modify -user-or-group-name guest
  -application ssh -authentication-method publickey -role vsadmin
  -vserver vs1.netapp.com
```

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 *vs1.netapp.com* . 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 vs1.netapp.com -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 *vs1.netapp.com* :

```
cluster1::> security login modify -user-or-group-name guest
  -application ssh -authentication-method publickey
  -second-authentication-method password -vserver vs1.netapp.com
```

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 *vs1.netapp.com* :

```
cluster1::> security login modify -user-or-group-name guest
  -application ssh -authentication-method publickey
  -second-authentication-method none -vserver vs1.netapp.com
```

The following example illustrates how to modify a login method that has the user name *spuser* , the application *service-processor* , and the authentication method *password* to use the second-authentication-method *publickey* for the Administrative Vserver *cluster1* :

```
cluster1::> security login modify -user-or-group-name spuser
  -application service-processor -authentication-method password
  -second-authentication-method publickey -vserver cluster1
```

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 'vs1.netapp.com' Vserver.

```
cluster1::> security login password -username admin -vserver
vs1.netapp.com
```

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 (amqp, 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
- LDAP fastbind authentication

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 amqp, 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|nsswitch|domain|totp}]` - Second Authentication Method2

Displays the login methods that match the specified authentication method to be used as the second factor. Presently, *`ssh`* and *`service-processor`* are the only applications supporting a second factor of authentication. For *`ssh`* application, possible values include the following:

- password - Password
- publickey - Public-key authentication
- nsswitch - NIS or LDAP authentication
- domain - Active Directory authentication
- none - default value

  1. For *`service-processor`* application, possible values include the following:

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

.

`[-is-ldap-fastbind {yes|no}]` - LDAP Fastbind Authentication

Displays the authentication methods that are LDAP fastbind.

Examples

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

```
cluster1::> security login show

Vserver: cluster1
                                                                 Second
User/Group                 Authentication                 Acct
Authentication
Name           Application Method        Role Name        Locked Method
-------------- ----------- ------------- ---------------- ------
--------------
admin          amqp        password      admin            no     none
admin          console     password      admin            no     none
admin          http        password      admin            no     none
admin          ontapi      password      admin            no     none
admin          service-processor
                           password      admin            no     none
admin          ssh         password      admin            no     none
autosupport    console     password      autosupport      no     none
user1          ssh         publickey     admin            -      none
user2          ssh         password      admin            no     publickey
spuser         service-processor
                           password      admin            no     publickey

Vserver: vs1.netapp.com
                                                                 Second
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
12 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. This command only applies to user accounts configured with the password authentication method where the password is set.

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.netapp.com.

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

security login whoami

Show the current user, trust score of the 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, role and the trust score 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
     Trust Score: 90
```

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 {scheme://(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
1234567890123456789012345678901234567890123456789012345678901234567890123
4567890
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 companyname.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 duo create

Add a Duo Configuration

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

Description

The `security login duo create` creates the Duo configuration on the Vserver.

Parameters

`-vserver <vserver name>` - Vserver

This parameter specifies the Vserver name of the Duo configurations.

`[-is-enabled {true|false}]` - Duo Enable Status

This parameter enable the Duo authentication.

`-integration-key <text>` - Duo Intgration Key

This parameter specifies Duo integration key.

`-secret-key <text>` - Duo Secret Key

This parameter specifies Duo secret key.

`-apihost <text>` - Duo API Host

This parameter specifies the Duo API host.

`[-autopush {true|false}]` - Duo Auto Push Config

This optionally specifies the autopush configurations. Configuring this to true will allows user to authenticate with single tap on their mobile device.

`[-pushinfo {true|false}]` **- Information with Push**

This parameter specifies the push info configurations. Configuring this to true will provide additional information in the push notification, such as the name of the application. This helps users to verify that they are logging in to the correct service and provides an additional layer of security

`[-fail-mode {safe|secure}]` **- Duo Fail Safe Config**

This parameter specifies the fail mode configurations. Possible values includes safe and secure.

`[-http-proxy <text>]` **- HTTP Proxy URL with Port Number**

This parameter specifies the http proxy configurations.

`[-max-prompts <integer>]` **- Duo Max Attempts**

This parameter specifies the number of retry before Duo authentication return failure.

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

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

Examples

The following command creates a Duo configurations for Vserver vs1.

```
cluster1::> security login duo create -vserver vs1 -integration-key
AA11A1AAAA1AAAA11A1A -secret-key
xxxxxxxxxx11111111117bd5a3b060947b617355ecf353627c50b1xxxxxxxxxx -apihost
api-99X9X9XX.duosecurity.com
-comment "This is a Duo Config"
```

security login duo delete

Delete a Duo Configuration

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

Description

The `security login duo delete` command deletes the Duo configuration.

Parameters

`-vserver <vserver name>` **- Vserver**

This parameter optionally specifies the Vserver name of the Duo configurations.

Examples

The following command deletes the Duo configurations of the Vserver vs1.

```
cluster1::> security login duo delete -vserver vs1
```

security login duo modify

Modify a Duo Configuration

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

Description

The `security login duo modify` modifies the Duo configuration.

Parameters

`-vserver <vserver name>` - Vserver

This parameter specifies the Vserver name of the Duo configurations.

`[-is-enabled {true|false}]` - Duo Enable Status

This parameter enable the Duo authentication.

`[-integration-key <text>]` - Duo Intgration Key

This parameter specifies Duo integration key.

`[-secret-key <text>]` - Duo Secret Key

This parameter specifies Duo secret key.

`[-apihost <text>]` - Duo API Host

This parameter specifies the Duo API host.

`[-autopush {true|false}]` - Duo Auto Push Config

This optionally specifies the autopush configurations. Configuring this to true will allows user to authenticate with single tap on their mobile device.

`[-pushinfo {true|false}]` - Information with Push

This parameter specifies the push info configurations. Configuring this to true will provide additional information in the push notification, such as the name of the application. This helps users to verify that they are logging in to the correct service and provides an additional layer of security

`[-fail-mode {safe|secure}]` - Duo Fail Safe Config

This parameter specifies the fail mode configurations. Possible values includes safe and secure.

`[-http-proxy <text>]` - HTTP Proxy URL with Port Number

This parameter specifies the http proxy configurations.

`[-max-prompts <integer>]` - Duo Max Attempts

This parameter specifies the number of retry before Duo authentication return failure.

`[-comment <text>]` - Comment

This parameter specifies comment text for the Duo. Note that comment text should be enclosed in quotation marks.

Examples

The following command modify the Duo configurations for the Vserver vs1.

```
cluster1::> security login duo modify -vserver vs1 -is-enabled false
```

security login duo show

Display Duo Configurations

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

Description

The `security login duo show` command displays information about the Duo configurations.

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 Duo configuration Vserver that match this parameter value.

`[-is-enabled {true|false}]` - Duo Enable Status

Selects the Duo Configurations that match this parameter value.

`[-integration-key <text>]` - Duo Intgration Key

Selects the Duo Configurations that match this parameter value.

`[-fingerprint <text>]` - Duo Secret Key Fingerprint

Selects the Duo Configurations that match this parameter value.

`[-apihost <text>]` - Duo API Host

Selects the Duo Configurations that match this parameter value.

`[-autopush {true|false}]` - Duo Auto Push Config

Selects the Duo Configurations that match this parameter value.

`[-pushinfo {true|false}]` - Information with Push

Selects the Duo Configurations that match this parameter value.

`[-fail-mode {safe|secure}]` - Duo Fail Safe Config

Selects the Duo Configurations that match this parameter value.

`[-http-proxy <text>]` - HTTP Proxy URL with Port Number

Selects the Duo Configurations that match this parameter value.

`[-max-prompts <integer>]` - Duo Max Attempts

Selects the Duo Configurations that match this parameter value.

`[-status <text>]` - Duo Status

Selects the Duo Configurations that match this parameter value.

`[-comment <text>]` - Comment

Selects the Duo Configurations that match this parameter value.

Examples

The example below displays the Duo information for the Vserver VS1.

```
cluster1::> security login duo show
                    Vserver: VS1
                    Enabled: true
                     Status: OK
            Integration Key: AA11A1AAAA1AAAA11A1A
            SHA Fingerprint:

xxxxxxxxxx11111111117bd5a3b060947b617355ecf353627c50b1xxxxxxxxxx
                   API Host: api-xxxxxxx.duosecurity.com
                   Autopush: true
                  Push info: true
                   Failmode: safe
                 Http-proxy: -
                    Prompts: 1
              Comment: This is a new key
```

security login duo group create

Add a Duo Group Configuration

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

Description

The `security login duo group create` creates the groups that need to included in Duo authentication.

Parameters

`-vserver <vserver name>` - Vserver

This parameter optionally specifies the Vserver of the Duo group that is being created.

`-group-name <text>` - Group Name

This parameter specifies the name of the group that will be part of Duo authentication.

`[-excluded-users <text>,…]` - List of Excluded Users

This parameter optionally specifies the list of users that will be excluded from Duo authentication.

`[-comment <text>]` - Comment

This parameter optionally specifies the comments

Examples

The following command creates a group "test" on vserver "VS1" and exclude the user tsmith from Duo authentication.

```
cluster1::> security login duo group create -vserver vs1 -group-name vs1
-exclude-users tsmith
```

security login duo group delete

Delete a Duo Group Configuration

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

Description

The `security login duo group delete` command deletes the specied group.

Parameters

`-vserver <vserver name>` - Vserver

This parameter optionally specifies the Vserver of the Duo group that is being deleted.

`-group-name <text>` - Group Name

This parameter optionally specifies the group name that is being deleted.

Examples

The following command deletes the group "test" from the Vserver "vs1".

```
cluster1::> security login duo group delete -vserver vs1 -group-name test
```

security login duo group modify

Modify a Duo Group Configuration

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

Description

The `security login duo group modify` command modifies the Duo groups.

Parameters

`-vserver <vserver name>` - Vserver

This parameter optionally specifies the Vserver of the group for which the 'exclude-users' or 'comment' is being modified.

`-group-name <text>` - Group Name

This parameter specifies the name of the group for which the 'exclude-users' or 'comment' is being modified.

`[-excluded-users <text>,…]` - List of Excluded Users

This parameter specifies the list of users to be excluded from Duo authentication.

`[-comment <text>]` - Comment

This parameter specifies the comments.

Examples

The following command modifies the excluded users for the group test who is part of Vserver vs1.

```
cluster1::> security login duo group modify -vserver vs1 -group-name test
-exclude-users tsmith, jane
```

security login duo group show

Display Duo Group Configurations

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

Description

The `security login duo group show` command displays information about Duo groups.

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 Vserver that match this parameter value.

[-group-name <text>] - Group Name

Selects the groups that match this parameter value.

[-excluded-users <text>,…] - List of Excluded Users

Selects the excluded users that match this parameter value.

[-comment <text>] - Comment

Selects the comments that match this parameter value.

Examples

The example below displays group information for the Vserver vs1.

```
cluster1::> security login duo group show -vserver vs1
Vserver: vs1
Group Name: NETAPP_ENG
Excluded Users: user1, user2, user12
Comment: This is a new group
```

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 {scheme://(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 and a certificate.

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.

`[-application <text>]` - Application

This parameter optionally specifies the application for which you are adding the public key. Presently, the only supported values are *`ssh`* and *`service-processor`* . The default value for this parameter is *`ssh`* .

`[-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.

`[-x509-certificate <text>]` - Install/Modify/Delete X509 Certificate

If this parameter is used, the specified certificate will be installed. The default when the public key is created is no certificate.

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”. Optionally, we can also specify the certificate.

```
cluster1::> security login publickey create -vserver vs1 -username tsmith
-index 5 -publickey
"ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAspH64CYbUsDQCdW22JnK6J
/vU9upnKzd2zAk9C1f7YaWRUAFNs2Qe5lUmQ3ldi8AD0Vfbr5T6HZPCixNAIza
FciDy7hgnmdj9eNGedGr/JNrftQbLD1hZybX+72DpQB0tYWBhe6eDJ1oPLob
ZBGfMlPXh8VjeU44i7W4+s0hG0E=tsmith@publickey.example.com"
-comment "This is a new key"
-x509-certificate install
Enter certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIIBYTCB56ADAgECAhQybXS4lSL99APGRPqaGTy7CGUICzAKBggqhkjOPQQDAjAS
MRAwDgYDVQQDDAdTU0ggS2V5MB4XDTIyMDkyNjA1MDcwMVoXDTIzMDkyNjAwMDAw
MFowEjEQMA4GA1UEAwwHU1NIIEtleTB2MBAGByqGSM49AgEGBSuBBAAiA2IABEL4
UEJYUBfTO8gGSdRQleLQNvxVFjTiCN9V+8/6qsSTshb+K2zGT7qoF2RNPRMtHvtH
r/EC7Wo+9yP/ovtjFsshC+boJpfe8NN4xpqDzeC0nn1kw1GIavOCGyhGUNauITAK
BggqhkjOPQQDAgNpADBmAjEA4ClCSp3Nb7DlX6Bxvi7utQobj2qQETgNxHpiYzlD
Zr920lNPHDRxJTQ04vIdNeoZAjEAwTfv5jpuowaYxXxPJND2ytpyFcjyl/BUrFpQ
7XyjxyVFzKP3Rfj+uBvhIb8sLbl8
-----END CERTIFICATE-----
```

The following command associates a public key with a 'service-processor' user named 'joed' for the Administrative Vserver. The public key is assigned index number 1 and the comment text is "This is a new publickey". In this case, the certificate has not been specified.

```
cluster1::> security login publickey create -username joed -application
service-processor -index 1 -publickey
"ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFgMRmKHV2+sGDPMD8JdMR
S+vSviIyjAymBZ/vu22Ae1AY51RIOBq5cVKaP49O9A21F1Srkqsd3gHFw/UhYBPy8="
-comment "This is a new publickey"
```

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.

`-application <text>` - Application

This parameter optionally specifies the application for which you are deleting the public key. Presently, the only supported values are *`ssh`* and *`service-processor`* . The default value for this parameter is *`ssh`* .

`-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 along with the certificate if it was installed.

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

The following command deletes the public key at the index number 2 for the 'service-processor' user named 'joed'.

```
cluster1::> security login publickey delete -vserver cluster1 -username
joed -application service-processor -index 2
```

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 {scheme://(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://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 and certificate.

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.

`-application <text>` - Application

Optionally specifies the application for which you are modifying the public key. Presently, the only supported values are *`ssh`* and *`service-processor`* . The default value for this parameter is *`ssh`* .

`-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.

`[-x509-certificate <text>]` - Install/Modify/Delete X509 Certificate

This parameter is used to modify or delete an existing certificate.

Examples

The following command modifies the public key and certificate at index number 10 for the SSH user named tsmith of Vserver vs1.

```
cluster1::> security login publickey modify -vserver vs1 -username tsmith
-index 10 -publickey
"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDDD+pFzFgV/2dlowKRFgym9K910H/u+BVTGitCtHteHy
o8thmaXT
1GLCzaoC/12+XXiYKMRhJ00S9Svo4QQKUXHdCPXFSgR5PnAs39set39ECCLzmduplJnkWtX96p
QH/bg2g3upFcdC6z9
c37uqFtNVPfv8As1Si/9WDQmEJ2mRtJudJeU5GZwZw5ybgTaN1jxDWus9SO2C43F/vmoCKVT52
9UHt4/ePcaaHOGTiQ
O8+Qmm59uTgcfnpg53zYkpeAQV8RdYtMdWlRr44neh1WZrmW7x5N4nXNvtEzr9cvb9sJyqTX1C
kQGfDOdb+7T7y3X7M
if/qKQY6FsovjvfZD" -x509-certificate modify
Enter certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIIBYjCB6aADAgECAhRhhyz6AuV04M5nac2Pq+cdm4IRMzAKBggqhkjOPQQDAjAT
MREwDwYDVQQDDAhhYmhpamVlMjAeFw0yMjEwMTkxNzI4MzZaFw0yMjEwMjAwMDAw
MDBaMBMxETAPBgNVBAMMCGFiaGlqZWUyMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAE
xNEN5jC3IjAvR0kwsAyKXw7aBLwNg5nvDC1D2bRLpbZkoLQ6hzyKZUEqA0ELlQla
u4yX9O1qiRIbMMclqMB1XYJywxOh+3uKOTM6Bz82/IZp4Oaa/4gYVtFRgStHTdPf
MAoGCCqGSM49BAMCA2gAMGUCMQDQIgdSrECuWJ76ZvfEDAvFlHnJHQtNz8zFl0lh
XRNzPlhpltEm6j5V6mPkRmJrmloCMGckXAVkmUCGFxU2e2ZvuKbL5BVCrE5iifet
ly3UApgBg8EgTO+hebMNz3i/Z4p+5w==
-----END CERTIFICATE-----"
```

The following command modifies the public key at index number 3 for the 'service-processor' user named 'joed' belonging to the Administrative Vserver.

```
cluster1::> security login publickey modify -vserver cluster1 -username
joed -index 3 -publickey
"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDYaIWUyYFtW+8xeCFgVg5OEM2P4m6mZvOBj7ES6TTUSI
cYk4gaOyLcJNwm05Q+Op3tidvakx16ALCtkP9TBBWgHPwwHma0+z689ekK88myabMst12iwMR3
9OGmen2KKW7NJQwwWmed47tTkEi5VU1BfX3CAsXiw4jp1HuFYi2iuHUQDtN6MlL4ON51X2IYUv
StH4N1UNn89bn3Q0l4UYFpFwF2ixIuR8Pm11bZvDy1yP4hpxmoisUpkwsmr/SJiBsX381Ogg3C
+Gnthtqe5/xoYx2CCb93Ff2UqKQyf41MgK8P1gNpOt5Vns1LQ+K1S+fh4ZmtkmBwHvQEXwBKQg
mb"
```

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.

[-application <text>] - Application

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.

[-certificate <certificate>] - Certificate Associated with Public Key

Selects the public keys that match this parameter value.

[-certificate-details <text>] - Details about the Certificate

Selects the public keys that match this parameter value.

`[-certificate-expired <text>]` - Expiry Status of Certificate

Selects the public keys that match this parameter value.

`[-certificate-revoked <text>]` - Revocation Status of Certificate

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
Vserver: vs1
UserName: tsmith Index: 5
Public Key:
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAspH64CYbUsDQCdW22JnK6J
/vU9upnKzd2zAk9C1f7YaWRUAFNs2Qe5lUmQ3ldi8AD0Vfbr5T6HZPCixNAIza
FciDy7hgnmdj9eNGedGr/JNrftQbLD1hZybX+72DpQB0tYWBhe6eDJ1oPLob
ZBGfMlPXh8VjeU44i7W4+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
Certificate:
-----BEGIN CERTIFICATE-----
MIICHTCCAaKgAwIBAgIUU+YJjeaOzvs+w+56JSm8Amow0nAwCgYIKoZIzj0EAwIw
RTELMAkGA1UEBhMCSU4xCzAJBgNVBAgMAktBMQwwCgYDVQQHDANCTFIxDTALBgNV
BAoMBE5UQVAxDDAKBgNVBAsMA1ImRDAeFw0yMjA5MjAwNjAyNDRaFw0yMzA5MjAw
NjAyNDRaMEUxCzAJBgNVBAYTAklOMQswCQYDVQQIDAJLQTEMMAoGA1UEBwwDQkxS
MQ0wCwYDVQQKDAROVEFQMQwwCgYDVQQLDANSJkQwdjAQBgcqhkjOPQIBBgUrgQQA
IgNiAASbaxCYB6XRDOFgdHBghvMoUpideGnd2jNrQJANeSaWVMnPUpXzg2tcPnsu
c87AR75BcfwhSurrFGLIw7TLcR22IFTggcrKmhjI8QwvomMZWfIoeHZlwsI+msm4
PLyaCqmjUzBRMB0GA1UdDgQWBBREeGdLZ3YTEL4CXLvTa8XQRAhRqDAfBgNVHSME
GDAWgBREeGdLZ3YTEL4CXLvTa8XQRAhRqDAPBgNVHRMBAf8EBTADAQH/MAoGCCqG
SM49BAMCA2kAMGYCMQDv9ZEselgteBlbqOYScKCyVcq3d89zz8Y9GBBB4FXJ3J+q
/h4zDk2Y2IJG63d7Kf0CMQDJl7v9I/NRNtS09qkavJh6snjJvUe3C5RhAkMPDhBO
2sfbUx1UQSo/md6U1CQBewM=
-----END CERTIFICATE-----
Certificate Details:
Subject:
C=US, ST=NC, L=RTP, O=NETAPP, OU=NTAP,
CN=scspr2692789021.gdl.englab.netapp.com
Issuer:
C=US, ST=NC, L=RTP, O=NETAPP, OU=NTAP, CN=NTAP-INTERCA2
Expiration: Jan 29 04:46:20 2024 GMT
Certificate Expiration Status: Not Expired
Certificate Revocation Status: good
```

The example below displays public key information for all 'service-processor' users.

```
cluster1::> security login publickey show -application service-processor
Vserver: cluster1
UserName: joed Index: 1
Public Key:
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFgMRmKHV2+sGDPMD8JdMR
S+vSviIyjAymBZ/vu22Ae1AY51RIOBq5cVKaP49O9A21F1Srkqsd3gHFw/UhYBPy8=
Fingerprint:
SHA256:ETxoqJyjq2tXjMKiJlCSTOT5vj+s+h6OxtXUIl28PcI
Bubblebabble fingerprint:
xiges-husyn-fyzim-sanok-bihos-sizuv-ribyt-cyryz-lelel-sekan-poxyx
Comment:
This is a new publickey
Certificate:
-
Certificate Details:
-
Certificate Expiration Status: -
Certificate Revocation Status: -
```

security login rest-role create

Add a REST access control role

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

Description

The `security login rest-role create` command creates a Representational State Transfer (REST) access-control role. A REST access-control role consists of a role name and an Application Programming Interface (API) to which the role has access. It optionally includes an access level (*`none`* , *`readonly`* , *`read_create`* , *`read_modify`* , *`read_create_modify`* or *`all`*) for the API. After you create a REST 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 REST role.

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

This specifies the REST role that is to be created.

`-api <text>` **- API Path**

This specifies the API to which the REST role has access. This API can be a private CLI API or a resource-qualified endpoint. Currently, the only supported resource-qualified endpoints are the following:

- Snapshots APIs
- */api/storage/volumes/{volume.uuid}/snapshots*
- File System Analytics APIs
- */api/storage/volumes/{volume.uuid}/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/clients*
- */api/storage/volumes/{volume.uuid}/top-metrics/directories*
- */api/storage/volumes/{volume.uuid}/top-metrics/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/users*
- */api/svm/svms/{svm.uuid}/top-metrics/clients*
- */api/svm/svms/{svm.uuid}/top-metrics/directories*
- */api/svm/svms/{svm.uuid}/top-metrics/files*
- */api/svm/svms/{svm.uuid}/top-metrics/users*
- Ontap S3 APIs
- */api/protocols/s3/services/{svm.uuid}/users*

In the above APIs, wildcard character * could be used in place of *{volume.uuid}* or *{svm.uuid}* to denote *all* volumes or *all* SVMs, depending upon whether the REST endpoint references volumes or SVMs.

-access {none|readonly|read_create|read_modify|read_create_modify|all} - Access Level

This optionally specifies an access level for the REST role. Possible access level settings are *none* , *readonly* , *read_create* , *read_modify* , *read_create_modify* and *all* .

Examples

The following command creates a REST access-control role named *admin* for the *vs1.example.com* Vserver. This REST role has an access-level of *all* for the */api/storage/volumes* API.

```
cluster1::> security login rest-role create -role admin -api
"/api/storage/volumes" -access all -vserver vs1.example.com
cluster1::>
```

The following command creates a REST access-control role named *rest_role1* for the *cluster1.example.com* administrative Vserver. This REST role has an access-level of *read_create_modify* for the */api/snapmirror/policies* API.

```
cluster1::> security login rest-role create -role rest_role1 -api
"/api/snapmirror/policies" -access read_create_modify -vserver
cluster1.example.com
cluster1::>
```

The following command creates a REST access-control role named *`vs1_role`* for the *`vs1.example.com`* Vserver. This REST role has an access level of *`readonly`* for all snapshots on the volume with UUID *`f8a541b5-b68c-11ea-9581-005056bbabe6`* . The volume UUID refers to the `-instance-uuid` field value in the volume show command output at diagnostic privilege level.

```
cluster1::> security login rest-role create -role vs1_role -api
"/api/storage/volumes/f8a541b5-b68c-11ea-9581-005056bbabe6/snapshots"
-access readonly -vserver vs1.example.com
Warning: Operating on an alias operates on the target of the specified
alias:
           "volume snapshot"
cluster1::>
```

The following command creates a REST access-control role named *`vs2_role`* for the *`vs2.example.com`* Vserver. This REST role has an access level of *`readonly`* for all files on the volume with UUID *`15d489b5-1d40-11ec-992e-005056bba268`* .The volume UUID refers to the `-instance-uuid` field value in the volume show command output at diagnostic privilege level.

```
cluster1::> security login rest-role create -role vs2_role -api
"/api/storage/volumes/15d489b5-1d40-11ec-992e-005056bba268/files" -access
readonly -vserver vs2.example.com
cluster1::>
```

The following command creates a REST access-control role named *`vs3_role`* for the *`vs3.example.com`* Vserver. This REST role has an access level of *`read_create_modify`* for all top-metrics directories on the SVM with UUID *`881764b5-9ea1-11ec-8771-005056bb1a7c`* .

```
cluster1::> security login rest-role create -role vs3_role -api
"/api/svm/svms/881764b5-9ea1-11ec-8771-005056bb1a7c/top-
metrics/directories" -access read_create_modify -vserver vs3.example.com
cluster1::>
```

The following command creates a REST access-control role named *`vs4_role`* for the *`vs4.example.com`* Vserver. This REST role has an access level of *`all`* for command directory *`cluster`* .

```
cluster1::> security login rest-role create -role vs4_role -api
"/api/private/cli/cluster" -access all -vserver vs4.example.com
cluster1::>
```

Related Links

- security login modify
- security login create

- volume show

security login rest-role delete

Delete a REST access control role

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

Description

The `security login rest-role delete` command deletes a Representational State Transfer (REST) access-control role.

Parameters

`-vserver <vserver name>` - Vserver

This optionally specifies the Vserver name associated with the REST role.

`-role <text>` - Role Name

This specifies the REST role that is to be deleted.

`-api <text>` - API Path

This specifies the Application Programming Interface (API) to which the REST role has access. This API can be a private CLI API or a resource-qualified endpoint. Currently, the only supported resource-qualified endpoints are the following:

- Snapshots APIs
- */api/storage/volumes/{volume.uuid}/snapshots*
- File System Analytics APIs
- */api/storage/volumes/{volume.uuid}/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/clients*
- */api/storage/volumes/{volume.uuid}/top-metrics/directories*
- */api/storage/volumes/{volume.uuid}/top-metrics/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/users*
- */api/svm/svms/{svm.uuid}/top-metrics/clients*
- */api/svm/svms/{svm.uuid}/top-metrics/directories*
- */api/svm/svms/{svm.uuid}/top-metrics/files*
- */api/svm/svms/{svm.uuid}/top-metrics/users*
- Ontap S3 APIs
- */api/protocols/s3/services/{svm.uuid}/users*

In the above APIs, wildcard character *** could be used in place of *{volume.uuid}* or *{svm.uuid}* to denote *all* volumes or _all _ SVMs, depending upon whether the REST endpoint references volumes or SVMs.

Examples

The following command deletes a REST access-control role entry with the role name *readonly* and the API */api/storage/volumes* from Vserver *vs.example.com* .

```
cluster1::> security login rest-role delete -role readonly -api
"/api/storage/volumes" -vserver vs.example.com
cluster1::>
```

The following command deletes a REST access-control role entry with the role name *vs1_role* and the resource-qualified endpoint corresponding to all snapshots on the volume with UUID *0aa39ec1-b68d-11ea-9581-005056bbabe6* from Vserver *vs1.example.com* .The volume UUID refers to the `-instance-uuid` field value in the volume show command output at diagnostic privilege level.

```
cluster1::> security login rest-role delete -role vs1_role -api
"/api/storage/volumes/0aa39ec1-b68d-11ea-9581-005056bbabe6/snapshots"
-vserver vs1.example.com
cluster1::>
```

The following command deletes a REST access-control role entry with the role name *vs2_role* and the resource-qualified endpoint corresponding to all top-metrics clients on the volume with UUID *373eb9ef-1d40-11ec-992e-005056bba268* from Vserver *vs2.example.com* . The volume UUID refers to the `-instance-uuid` field value in the volume show command output at diagnostic privilege level.

```
cluster1::> security login rest-role delete -role vs2_role -api
"/api/storage/volumes/373eb9ef-1d40-11ec-992e-005056bba268/top-
metrics/clients" -vserver vs2.example.com
cluster1::>
```

The following command deletes a REST access-control role entry with the role name *vs3_role* and the resource-qualified endpoint corresponding to all top-metrics directories for the Vserver *vs3.example.com* with UUID *6dfeb2a7-9a16-11ec-819e-005056bb1a7c* .

```
cluster1::> security login rest-role delete -role vs3_role -api
"/api/svm/svms/6dfeb2a7-9a16-11ec-819e-005056bb1a7c/top-
metrics/directories" -vserver vs3.example.com
cluster1::>
```

The following command deletes a REST access-control role entry with the role name *vs4_role* and the API */api/private/cli/cluster* for the Vserver *vs4.example.com* .

```
cluster1::> security login rest-role delete -role vs4_role -api
"/api/private/cli/cluster" -vserver vs4.example.com
cluster1::>
```

Related Links

- volume show

security login rest-role modify

Modify a REST access control role

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

Description

The `security login rest-role modify` command modifies a Representational State Transfer (REST) access-control role.

Parameters

`-vserver <vserver name>` - Vserver

This optionally specifies the Vserver name associated with the REST role.

`-role <text>` - Role Name

This specifies the REST role that is to be modified.

`-api <text>` - API Path

This specifies the Application Programming Interface (API) to which the REST role has access. This API can be a private CLI API or a resource-qualified endpoint. Currently, the only supported resource-qualified endpoints are the following:

- Snapshots APIs
- *`/api/storage/volumes/{volume.uuid}/snapshots`*
- File System Analytics APIs
- *`/api/storage/volumes/{volume.uuid}/files`*
- *`/api/storage/volumes/{volume.uuid}/top-metrics/clients`*
- *`/api/storage/volumes/{volume.uuid}/top-metrics/directories`*
- *`/api/storage/volumes/{volume.uuid}/top-metrics/files`*
- *`/api/storage/volumes/{volume.uuid}/top-metrics/users`*
- *`/api/svm/svms/{svm.uuid}/top-metrics/clients`*
- *`/api/svm/svms/{svm.uuid}/top-metrics/directories`*
- *`/api/svm/svms/{svm.uuid}/top-metrics/files`*

- `/api/svm/svms/{svm.uuid}/top-metrics/users`
- Ontap S3 APIs
- `/api/protocols/s3/services/{svm.uuid}/users`

In the above APIs, wildcard character `*` could be used in place of `{volume.uuid}` or `{svm.uuid}` to denote `all` volumes or `all` SVMs, depending upon whether the REST endpoint references volumes or SVMs.

`[-access {none|readonly|read_create|read_modify|read_create_modify|all}]` - Access Level

This specifies a new access level for the REST role. Possible access level settings are `none` , `readonly` , `read_create` , `read_modify` , `read_create_modify` and `all` .

Examples

The following command modifies a REST access-control role with the role name `readonly` and the API `/api/storage/volumes` to have the access level `readonly` for Vserver `vs.example.com` :

```
cluster1::> security login rest-role modify -role readonly -api
"/api/storage/volumes" -access readonly -vserver vs.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name `rest_role1` and the API `/api/snapmirror/policies` to have the access level `read_create` for Vserver `cluster1.example.com` :

```
cluster1::> security login rest-role modify -role rest_role1 -api
"/api/snapmirror/policies" -access read_create -vserver
cluster1.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name `vs1_role` and the resource-qualified endpoint `/api/storage/volumes/*/snapshots` to have the access level `readonly` for Vserver `vs1.example.com` :

```
cluster1::> security login rest-role modify -role vs1_role -api
"/api/storage/volumes/*/snapshots" -access readonly -vserver
vs1.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name `vs2_role` and the resource-qualified endpoint `/api/storage/volumes/4d383f47-1d40-11ec-81af-005056bb3eae/top-metrics/users` to have the access level `none` for Vserver `vs2.example.com` :

```
cluster1::> security login rest-role modify -role vs2_role -api
"/api/storage/volumes/4d383f47-1d40-11ec-81af-005056bb3eae/top-
metrics/users" -access none -vserver vs2.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name *`vs3_role`* and the resource-qualified endpoint *`/api/svm/svms/6dfeb406-9a16-11ec-819e-005056bb1a7c/top-metrics/files`* to have the access level *`read_modify`* for Vserver *`vs3.example.com`* :

```
cluster1::> security login rest-role modify -role vs3_role -api
"/api/svm/svms/6dfeb406-9a16-11ec-819e-005056bb1a7c/top-metrics/files"
-access read_modify -vserver vs3.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name *`vs3_role2`* and the wildcard resource-qualified endpoint *`/api/svm/svms/*/top-metrics/clients`* to have the access level *`readonly`* for Vserver *`vs3.example.com`* :

```
cluster1::> security login rest-role modify -role vs3_role2 -api
"/api/svm/svms/*/top-metrics/clients" -access readonly -vserver
vs3.example.com
cluster1::>
```

The following command modifies a REST access-control role with the role name *`vs4_role`* and api *`/api/private/cli/cluster`* to have the access level *`readonly`* for Vserver *`vs4.example.com`* :

```
cluster1::> security login rest-role modify -role vs4_role -api
"/api/private/cli/cluster" -access readonly -vserver vs4.example.com
cluster1::>
```

security login rest-role show

Show REST access control roles

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

Description

The `security login rest-role show` command displays the following information about Representational State Transfer (REST) access-control roles:

- Vserver
- Role name

- Application Programming Interface (API) to which the REST role has access
- Access Level (*none* , *readonly* , *read_create* , *read_modify* , *read_create_modify* , or *all*)

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 REST roles that match this parameter value.

[-role <text>] - Role Name

Selects the REST roles that match this parameter value. If this parameter and the `-api` parameter are both used, the command displays detailed information about the specified REST access-control role.

[-api <text>] - API Path

Selects the REST roles that match this parameter value. If this parameter and the `-role` parameter are both used, the command displays detailed information about the specified REST access-control role. This API can be a private CLI API or a resource-qualified endpoint. Currently, the only supported resource-qualified endpoints are the following:

- Snapshots APIs
- */api/storage/volumes/{volume.uuid}/snapshots*
- File System Analytics APIs
- */api/storage/volumes/{volume.uuid}/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/clients*
- */api/storage/volumes/{volume.uuid}/top-metrics/directories*
- */api/storage/volumes/{volume.uuid}/top-metrics/files*
- */api/storage/volumes/{volume.uuid}/top-metrics/users*
- */api/svm/svms/{svm.uuid}/top-metrics/clients*
- */api/svm/svms/{svm.uuid}/top-metrics/directories*
- */api/svm/svms/{svm.uuid}/top-metrics/files*
- */api/svm/svms/{svm.uuid}/top-metrics/users*
- Ontap S3 APIs
- */api/protocols/s3/services/{svm.uuid}/users*
- Private-cli APIs
- */api/private/cli/cluster*

In the above APIs, wildcard character `*` could be used in place of *{volume.uuid}* or *{svm.uuid}* to denote *all* volumes or *all* SVMs, depending upon whether the REST endpoint references volumes or SVMs.

[-access {none|readonly|read_create|read_modify|read_create_modify|all}] - Access Level

Selects the roles that match this parameter value.

Examples

The example below displays information about all REST access-control roles:

```
cluster1::> security login rest-role show
Role                            Access
      Vserver           Name            API             Level
      ----------        -------------   -----------     -----------
      vs                vsrole1         /api            none
      vs                vsrole1         /api/storage/volumes/f8a541b5-
b68c-11ea-9581-005056bbabe6/files
                                                        all
      vs                vsrole1         /api/storage/volumes/f8a541b5-
b68c-11ea-9581-005056bbabe6/snapshots
                                                        readonly
      vs                vsrole1         /api/storage/volumes/843b87f9-
2f5e-11ec-9524-005056bb0bee/snapshots
                                                        read_create
      vs                vsrole1         /api/svm/svms/843b87f9-2f5e-11ec-
9524-005056bb0bee/top-metrics/clients
                                                        read_create
      cluster1          readonly        /api/storage    none
      cluster1          custom          /api/cluster    read_modify
      cluster1          custom          /api/security/accounts
                                                        read_create_modify
      cluster1          custom          /api/storage/volumes/*/top-
metrics/users
                                                        readonly
      cluster1          custom          /api/storage/volumes/*/snapshots
                                                        all
cluster1::>
```

security login rest-role expanded-rest-roles modify

Modify the status of Expanded REST roles for granular resource control feature

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

Description

The `security login rest-role expanded-rest-roles modify` command enables or disables *`Expanded REST roles for granular resource control`* feature.

Parameters

`[-is-enabled {true|false}]` - Is Enabled? (privilege: advanced)

This parameter specifies whether the *`Expanded REST roles for granular resource control`* feature is enabled or disabled. The default value is *`true`* i.e. the feature is enabled by default.

Examples

The following command disables the *`Expanded REST roles for granular resource control`* feature.

```
cluster1::*> security login rest-role expanded-rest-roles modify -is
-enabled false
cluster1::*>
```

security login rest-role expanded-rest-roles show

Show the status of Expanded REST roles for granular resource control feature

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

Description

The `security login rest-role expanded-rest-roles show` command specifies whether the *`Expanded REST roles for granular resource control`* feature is enabled (*`true`*) or disabled (*`false`*).

Examples

The command below specfies that the *`Expanded REST roles for granular resource control`* feature is enabled.

```
cluster1::> security login rest-role expanded-rest-roles show

Is Enabled? true
```

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. The command or command directory must be specified either within double quotes or inside curly brackets. To specify the default setting, use the special value "`DEFAULT` ".

`[-access {none|readonly|read_create|read_modify|read_create_modify|all}]` - 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 {none|readonly|read_create|read_modify|read_create_modify|all}]` - 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-rest

Display the mapping between ONTAP REST APIs and CLI commands

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

Description

The `security login role show-rest` command displays ONTAP REST APIs 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.

[`-endpoint <text>`] - URI of the REST endpoint

This parameter will filter the output to show CLI commands that match the provided REST endpoint.

[`-commands <text>`,…] - CLI Command

This parameter will filter the output to show the REST API or APIs that match the provided CLI command.

[`-supported-operations <text>`,…] - Supported REST operations <GET,POST,PATCH,DELETE>

This parameter filters the output to show REST APIs or CLI commands that match the provided REST operations <GET,POST,PATCH,DELETE>

Examples

The following command displays all the ONTAP REST APIs and their mapped CLI commands:

```
cluster1::> security login role show-rest
REST Endpoint               CLI Commands                   Supported
Operations
--------------------------- ------------------------------
--------------------
/application/applications   application                    GET, PATCH,
POST, DELETE
/application/applications/$id/components/$id/metadata
                            -                              GET, PATCH,
POST, DELETE
/application/applications/$id/metadata
                            application metadata           GET, PATCH,
POST, DELETE
/application/applications/{application.uuid}/components
                            -                              GET
/application/applications/{application.uuid}/components/{component.uuid}/s
napshots
                            -                              GET, POST,
DELETE
/application/applications/{application.uuid}/components/{component.uuid}/s
napshots/{uuid}/restore
                            -                              POST, DELETE
/application/applications/{application.uuid}/snapshots
                            -                              GET, POST,
DELETE
/application/applications/{application.uuid}/snapshots/{uuid}/restore
                            -                              POST, DELETE
/application/consistency-groups
Press <space> to page down, <return> for next line, or 'q' to quit...
```

The following examples display all ONTAP REST APIs that are mapped to a specified CLI command:

```
cluster1::> security login role show-rest -commands statistics
REST Endpoint               CLI Commands                   Supported
Operations
--------------------------- ------------------------------
--------------------
/cluster/counter/tables     statistics                     GET
/cluster/counter/tables/{counter_table.name}/rows
                            statistics                     GET
2 entries were displayed.
```

```
cluster1::> security login role show-rest -commands "network interface"*
REST Endpoint               CLI Commands                   Supported
Operations
--------------------------- ------------------------------
--------------------
/cluster/nodes              cluster add-node, network interface create,
storage aggregate auto-provision, system node, system node modify, system
node show, system service-processor network modify
                                                           GET, PATCH,
POST, DELETE
/network/fc/interfaces      network interface create, network interface
show, vserver fcp interface show
                                                           GET, PATCH,
POST, DELETE
/network/fc/interfaces/{uuid}/metrics
                            network interface show         GET
/network/ip/interfaces      network interface, network interface create
                                                           GET, PATCH,
POST, DELETE
/network/ip/interfaces/{uuid}/metrics
                            network interface show         GET
/network/ip/service-policies
                            network interface service-policy, network
interface service-policy show
                                                           GET, PATCH,
POST, DELETE
/svm/svms                   network interface create, network route
create, vserver, vserver add-aggregates, vserver add-protocols, vserver
cifs create, vserver create, vserver fcp create, vserver iscsi create,
vserver nfs create, vserver nvme create, vserver object-store-server
create, vserver remove-protocols, vserver services name-service dns
create, vserver services name-service ldap client create, vserver services
name-service nis-domain create, vserver services name-service ns-switch
create, vserver show
                                                           GET, PATCH,
POST, DELETE
7 entries were displayed.
```

```
cluster1::> security login role show-rest -commands *"fcp"*"create"
REST Endpoint               CLI Commands                       Supported
Operations
--------------------------- ------------------------------
--------------------
/protocols/san/fcp/services vserver fcp create, vserver fcp show
                                                           GET, PATCH,
POST, DELETE
/svm/svms                   network interface create, network route
create, vserver, vserver add-aggregates, vserver add-protocols, vserver
cifs create, vserver create, vserver fcp create, vserver iscsi create,
vserver nfs create, vserver nvme create, vserver object-store-server
create, vserver remove-protocols, vserver services name-service dns
create, vserver services name-service ldap client create, vserver services
name-service nis-domain create, vserver services name-service ns-switch
create, vserver show
                                                           GET, PATCH,
POST, DELETE
2 entries were displayed.
```

The following examples display the CLI command that is mapped to a specified ONTAP REST API:

```
cluster1::> security login role show-rest -endpoint /cluster/metrocluster*
REST Endpoint               CLI Commands                   Supported
Operations
--------------------------- ------------------------------
--------------------
/cluster/metrocluster       metrocluster configuration-settings connection
connect, metrocluster configuration-settings dr-group create, metrocluster
configuration-settings interface create, metrocluster configuration-
settings mediator add, metrocluster configure, metrocluster show, storage
aggregate create, storage aggregate mirror
                                                           GET, PATCH,
POST, DELETE
/cluster/metrocluster/diagnostics
                            metrocluster check show        GET, POST,
DELETE
/cluster/metrocluster/dr-groups
                            metrocluster configuration-settings dr-group
create, metrocluster configuration-settings dr-group show
                                                           GET, PATCH,
POST, DELETE
/cluster/metrocluster/interconnects
                            metrocluster interconnect mirror show
                                                           GET, PATCH
/cluster/metrocluster/nodes metrocluster node show         GET
/cluster/metrocluster/operations
                            metrocluster operation show    GET
/cluster/metrocluster/svms  metrocluster vserver show      GET
7 entries were displayed.
```

```
cluster1::> security login role show-rest -endpoint *cifs/session*
REST Endpoint               CLI Commands                   Supported
Operations
--------------------------- ------------------------------
--------------------
/protocols/cifs/session/files
                            vserver cifs session file      GET
/protocols/cifs/session/files/$id/$id
                            vserver cifs session file      GET, DELETE
/protocols/cifs/sessions    vserver cifs session           GET, POST,
DELETE
/protocols/cifs/sessions/$id/$id
                            vserver cifs session           GET, POST,
DELETE
4 entries were displayed.
```

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 {none|readonly|read_create|read_modify|read_create_modify|all}] - 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 <integer_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 *`5`* , which means *`5`* failed login attempts will cause an account to be locked.

`[-lockout-duration <integer>]` - (DEPRECATED)-Maximum Lockout Period (Days)

(DEPRECATED)-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`* hour. For roles which were created in a release before ONTAP 9.15.0 with the default value of *`0`* , this value will be automatically changed to *`1`* during upgrade to ONTAP 9.15.0. In other words, the value of this field for roles created before ONTAP 9.15.0 is defaulted to 24 hrs. For the roles which are created in ONTAP 9.15.0

or later, the value of this field defaults to 1 hour. This parameter is deprecated in ONTAP 9.15.0 and later. It may be removed from a future release of ONTAP.

`[-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 charaters. The default setting is *0* , which requires no digits.

`[-passwd-expiry-warn-time <integer_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 <integer_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 <integer_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.

`[-account-lockout-duration {P[<integer>D]T[<integer>H][<integer>M][<integer>S] | P<integer>W | disabled}]` - Account Lockout Duration (ISO 8601 Duration Format)

This specifies the duration in ISO 8601 format for which an account is locked if the failed login attempts reach the allowed maximum. The default is 1 hour. Specifying this field will set the field `lockout-duration` as *0* and the field `account-lockout-duration` will be used for further operations.

Examples

The following command modifies the user-account restrictions for an account with the role name admin for a Vserver named vs1. The minimum size of the password is set to 12 characters.

```
cluster1::> security login role config modify -role admin -vserver vs1
    -passwd-minlength 12
```

The following command sets the maximum allowed number of consecutive invalid login attempts to 3 and the maximum account lockout duration to 1 minute 30 seconds after 3 failed login attempts for role admin for vserver vs1:

```
cluster1::> security login role config modify -role admin -vserver vs1
    -max-failed-login-attempts 3 -account-lockout-duration PT1M30S
```

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")
- Minimum number of uppercase characters required in password ("0")
- Minimum number of lowercase characters required in password ("0")
- Minimum number of digits 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 ("5")
- Maximum time period for which the user account is locked out after the maximum number of failed login attempts is reached ("1 Hour")

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`
- (DEPRECATED)-Number of days for which the user account is locked after the maximum number of failed login attempts is reached. For roles which were created in a release before ONTAP 9.15.0 with the default value of *0* , this value will be automatically changed to *1* during upgrade to ONTAP 9.15.0. In other words, the value of this field for roles created before ONTAP 9.15.0 is defaulted to 24 hrs. For the roles which are created in ONTAP 9.15.0 or later, the value of this field defaults to 1 hour. This parameter is deprecated in ONTAP 9.15.0 and later. It may be removed from a future release of ONTAP `-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`
- Duration for which the user account is locked after the maximum number of failed login attempts is reached `-account-lockout-duration`

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 <integer_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>]` - (DEPRECATED)-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 <integer_or_unlimited>]` - Display Warning Message Days Prior to Password Expiry (Days)

Selects the profile configurations that match this parameter value.

`[-account-expiry-time <integer_or_unlimited>]` - Account Expires in (Days)

Selects the profile configurations that match this parameter value.

`[-account-inactive-limit <integer_or_unlimited>]` - Maximum Duration of Inactivity before Account Expiration (Days)

Selects the profile configurations that match this parameter value.

`[-account-lockout-duration {P[<integer>D]T[<integer>H][<integer>M][<integer>S] | P<integer>W | disabled}]` - Account Lockout Duration (ISO 8601 Duration Format)

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 login totp create

Add a TOTP secret

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

Description

The `security login totp create` associates an new secret key with a user account. This command requires that you enter a user name.

Parameters

`-vserver <vserver name>` - Vserver

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

`-username <text>` - Username

This parameter specifies the name of the user for whom you are adding the secret key. If you do not specify a user in case of Cserver, the user named admin is specified by default. If it is a data vserver, it defaults to logged-in user

`[-comment <text>]` - Comment

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

Examples

The following command creates a secret key with a user named tsmith for Vserver vs1. The secret key has a comment text is “This is a new key”.

```
cluster1::> security login totp create -vserver vs1 -username tsmith
-comment "This is a new key"
```

security login totp delete

Delete a TOTP secret

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

Description

The `security login totp delete` command deletes secret key for a specific user. To delete a secret key, you must specify a user name.

Parameters

`-vserver <vserver name>` - Vserver

This parameter optionally specifies the Vserver of the user for whom you are deleting the secret key.

`-username <text>` - Username

This parameter specifies the name of the user for whom you are deleting a secret key. If you do not specify a user in case of cserver, the user named admin is specified by default. In case of data vserver, it defaults to logged-in user

Examples

The following command deletes the secret key for the user named tsmith.

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

security login totp modify

Modify a TOTP status

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

Description

The `security login totp modify` command modifies the TOTP status. Only admin can modify the TOTP status and it requires that you enter a user name and enabled

Parameters

`-vserver <vserver name>` - Vserver

This parameter optionally specifies the Vserver of the user for whom you are modifying the TOTP status or comment.

`-username <text>` - Username

This parameter specifies the name of the user for whom you are modifying the TOTP status.

[-enabled {true|false}] - TOTP Status

This parameter specifies the TOTP status of the user for whom you are modifying.

[-comment <text>] - Comment

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

Examples

The following command disable the TOTP for the user named tsmith for Vserver vs1.

```
cluster1::> security login totp modify -vserver vs1 -username tsmith
-enabled false
```

security login totp show

Display TOTP secret

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

Description

The `security login totp show` command displays information about secret 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 secret keys that match this parameter value.

[-username <text>] - Username

Selects the secret keys that match this parameter value.

[-fingerprint <text>] - TOTP Secret key fingerprint

Select the secret keys that match this parameter value.

[-enabled {true|false}] - TOTP Status

Selects the secret keys that match this parameter value.

[-comment <text>] - Comment

Selects the secret keys that match this parameter value.

Examples

The example below displays secret key information for the user named tsmith.

```
cluster1::> security login totp show -username tsmith
UserName: tsmith
TOTP SHA Fingerprint:
7038eb494f8b86726bdfae9da1fbadb348f8fe26116f49e09f718a7b8bdd73b8
TOTP status: true
Comment: This is a new key
```

Copyright information

Trademark information