How SnapManager maintains security

SnapManager Oracle

NetApp
April 01, 2021

Table of Contents

How SnapManager maintains security

You can perform SnapManager operations only if you have the appropriate credentials. Security in SnapManager is governed by user authentication and role-based access control (RBAC). RBAC enables database administrators to restrict the operations that SnapManager can perform against the volumes and LUNs that hold the data files in a database.

Database administrators enable RBAC for SnapManager by using SnapDrive. Database administrators then assign permissions to SnapManager roles and assign these roles to the users in the Operations Manager graphical user interface (GUI) or command-line interface (CLI). RBAC permission checks happen in the DataFabric Manager server.

In addition to role-based access, SnapManager maintains security by requesting user authentication through password prompts or by setting user credentials. An effective user is authenticated and authorized with the SnapManager server.

SnapManager credentials and user authentication differ significantly from SnapManager 3.0:

- In SnapManager versions earlier than 3.0, you would set an arbitrary server password when you install SnapManager. Anyone who wants to use the SnapManager server would need the SnapManager server password. The SnapManager server password would need to be added to the user credentials by using the smo credential set -host command.
- In SnapManager (3.0 and later), the SnapManager server password has been replaced by individual user operating system (OS) authentication. If you are not running the client from the same server as the host, the SnapManager server performs the authentication by using your OS user names and passwords. If you do not want to be prompted for your OS passwords, you can save the data to your SnapManager user credentials cache by using the smo credential set -host command.

The smo credential set -host command remembers your credentials when the host.credentials.persist property in the smo.config file is set to true.

Example

User1 and User2 share a profile called Prof2. User2 cannot perform a backup of Database1 in Host1 without permission to access Host1. User1 cannot clone a database to Host3 without permission to access Host3.

The following table describes different permissions assigned to the users:

Permission type	User1	User2
Host Password	Host1, Host2	Host2, Host3
Repository Password	Repo1	Repo1
Profile Password	Prof1, Prof2	Prof2

In the case where User1 and User2 do not have any shared profiles, assume User1 has permissions for the hosts named Host1 and Host2, and User2 has permissions for the host named Host2. User2 cannot run even the nonprofile commands such as dump and system verify on Host1.