API - Disabling a Power User, its Sub-Users, and its Policies

The following example demonstrates how to disable a Power User's Sub-Users and Policies.

A PHP file called `Disable_Power_User_Sub_Users_And_Policies.php` can be found in `<installdir>/apisamples`. Read more in Accessing Example API Functions.

The PHP script searches and disables objects in the system and prints result messages like the following on the screen:

- "Successfully retrieved all the users" / "No user found with the specified ID" / "Failed to get the specified user"
- "Successfully retrieved all the agents" / "No Agents are owned by the specified user" / "Failed to get the specified agent"
- "Successfully retrieved all the volumes" / "No Volumes are exclusively assigned to the specified user" / "Failed to get all volumes"
- "Successfully retrieved all the diskSafes" / "No Disksafes are associated with the agents owned by the user" / "Failed to get all diskSafes"
- "Successfully retrieved all the policies" / "No Policies are associated with the disksafe assigned to the agents owned by the user" / "Failed to get all the policies"
- "Failed to update all policie(s)" / "All policie(s) update successfully"
- "Failed to update all User(s)" / "All User(s) updated successfully"

Sequence of Automated Actions

The following steps can be accomplished by using this script:

1. Find a User with the specified ID. If a User with the ID does not exist, then exit or save the User ID.
2. Find Sub-Users for the specified Power User. If there are any, save their User IDs.
3. Find Agents which have the specified Users as their owner. Save the Agent IDs.
4. Find Disk Safes associated with the specified Agent. Save the Disk Safe IDs.
5. Find Policies associated with the found Disk Safes. Save the Policy IDs.
6. Disable found Policies.
7. Disable found Users.

How to Fulfill Appropriate Actions in CDP User Interface

Below, you can find steps to take in the program user interface in order to perform the same actions as the script. We also provide you with screen-shots illustrating the scripts for every step.

Defining server configuration variables

```
########====CDP Server Configuration Start====########
#set CDP server host name
$HOST="127.0.0.1";
#set CDP server to access API
$PORT="9443";
#set CDP user
$USER="admin";
#set CDP user password
$PASS="admin";
########====CDP Server Configuration End====########
```

Log in to the CDP Server user interface using your username and password.

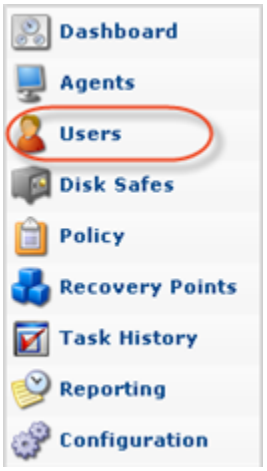
Retrieving Users

```
########====SetUser ID to be deleted Start====########
$ID = "dcad5f7c-27d0-46d0-86b2-24741b6bf25f";
########====SetUser ID to be deleted End====########
########====Get User Start====########

 try{
   $userClient = new soapclient("https://$HOST:$PORT/User?wsdl",
      array('login'=>"$USER",
      'password'=>"$PASS",
      'trace'=>1,
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS)
 );
 # get all the Users
 $allUsers=$userClient->getUsers();
 echo "Successfully retrived all the users \n";
 $selectedUserID = array();
 $selectedSubUserIDs = array();
 $selectedUsersObject = array();
 foreach($allUsers->return as $tmp) {
  // check to see if the specified user id matches the any of the retrived user ids
  if (isset($tmp->id) && $tmp->id == $ID){
   // if it matches store the id
   array_push($selectedUserID, $tmp->id);
   array_push($selectedUsersObject, $tmp);
  }
```

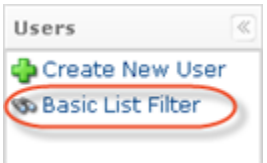
```
  else if (isset($tmp->adminIDs)){
   $tmpAdminIDArray = array($tmp->adminIDs);
   if (in_array($ID , $tmpAdminIDArray)){
    array_push($selectedSubUserIDs, $tmp->id);
    array_push($selectedUsersObject, $tmp);
   }
  }
 }

 // if nothing matches then exit the program
 if (!isset($selectedUserID)){
  echo " No user found with the specified ID $ID \n" ;
  exit(1);
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get the specified user \n";
 echo $exception;
 exit(1);
}

########====Get User End====########
```

To find a User by username, follow the instructions below.

1. Click on "Users" in the Main Menu to access the Users page.

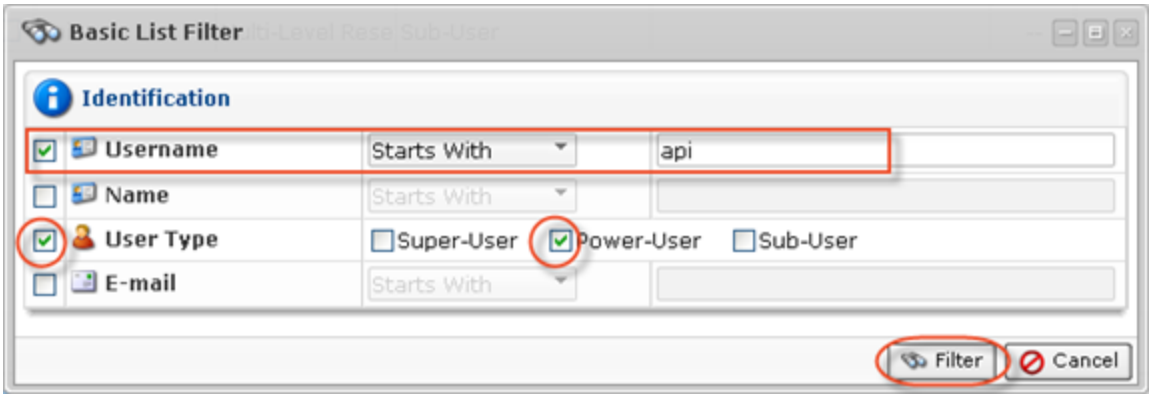


2. Click on "Basic List Filter" located in the Users sub-menu.



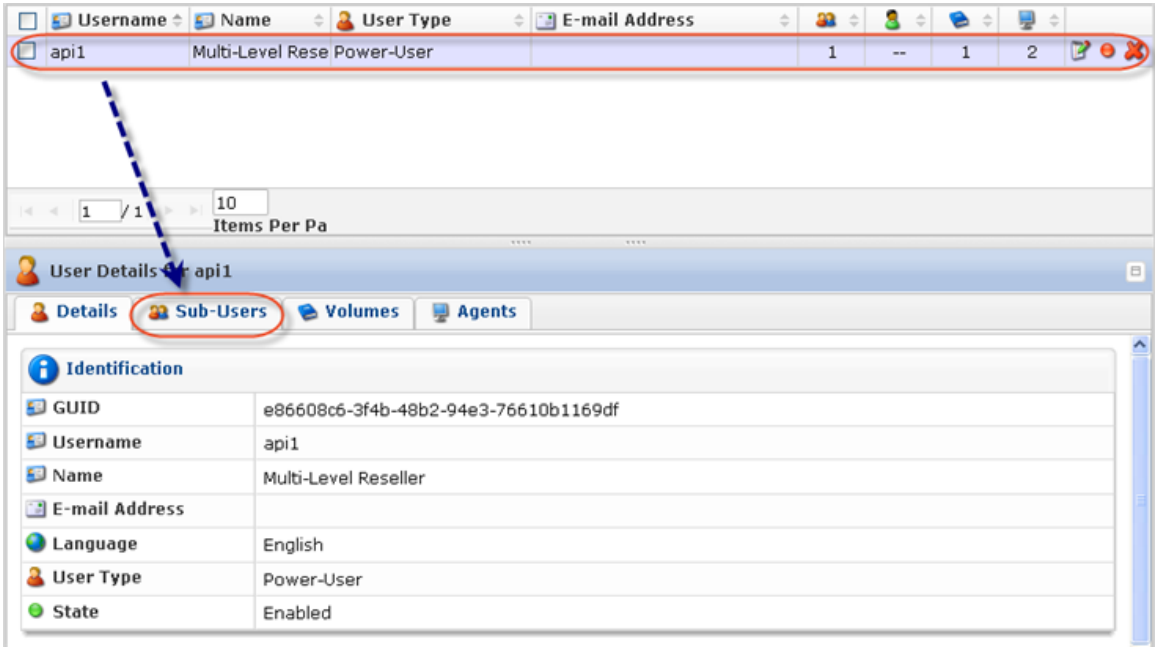
3. Enter a username, select the "Power-User" check-box, and click "Filter."

> **Note**
> While the script searches by User ID, we search by username in the CDP interface.

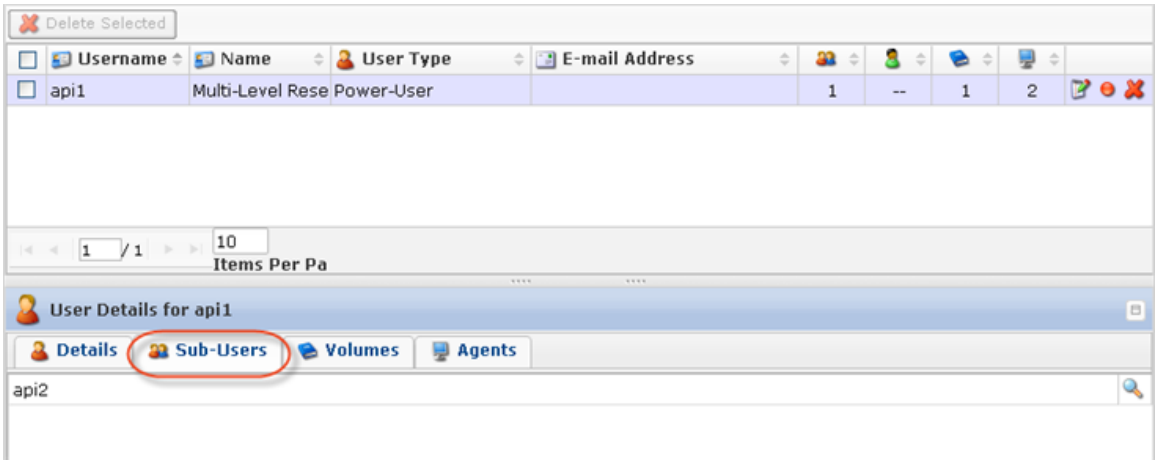


4. The found Users are displayed in the list.

5. Click on the User in the list, and then select the "Sub-Users" tab in the "Details" pane located in the bottom area of the interface.



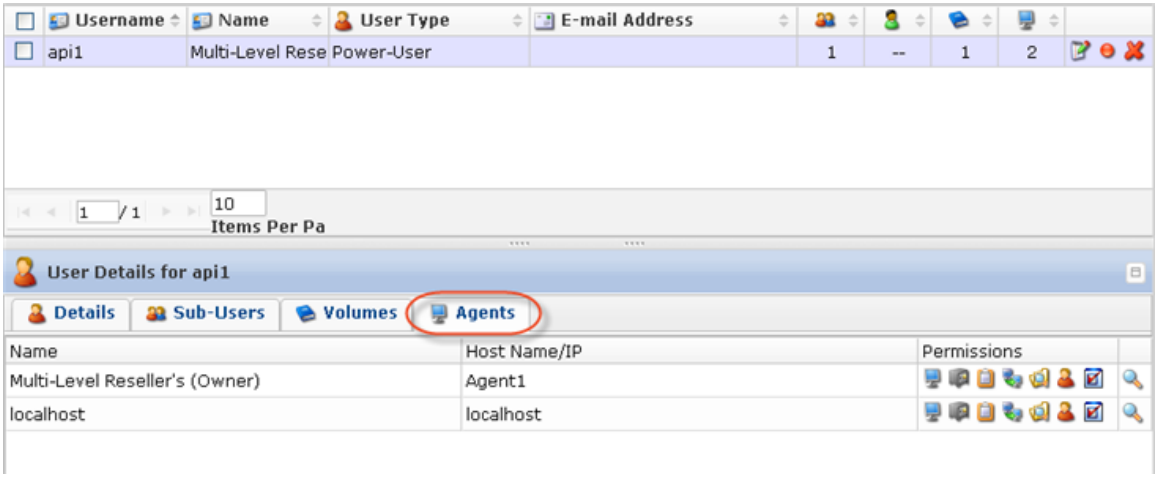
Once the "Sub-Users" tab is selected, all Sub-Users will be listed there.



Retrieving Agents

```
########====Get Agent Start====########

 try{
   $agentClient = new soapclient("https://$HOST:$PORT/Agent?wsdl",
       array('login'=>"$USER",
       'password'=>"$PASS",
       'trace'=>1,
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS)
 );
 # get all the agents
 $allAgents=$agentClient->getAgents();
 echo "Successfully retrived all the agents \n";
 $selectedAgentIDs = array();
 foreach($allAgents->return as $tmp) {
  // check to see if the specified agent has the specified user as owner
  if (isset($tmp->ownerId) && $tmp->ownerId == $ID){
   // if it matches store the id
   array_push($selectedAgentIDs, $tmp->id);
  }
 }

 if (!isset($selectedAgentIDs) || count($selectedAgentIDs) == 0){
  echo " No Agents are owned by the specified user \n" ;
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get the specified agent \n";
 echo $exception;
 exit(1);
}

########====Get Agent End====########
```

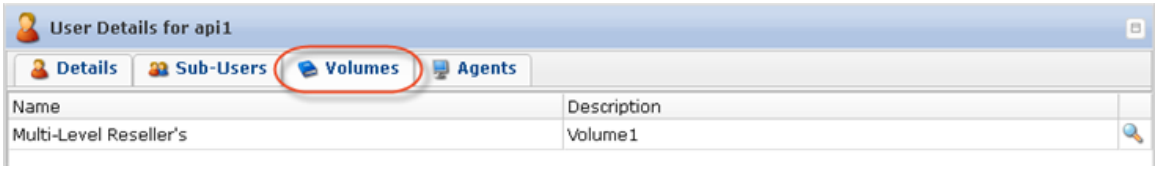
Select the "Agents" tab in the "Details" pane in the bottom area of the interface to list all associated Agents.



Retrieving Volumes

```
########====Get Volumes Start====########

try {
 $volumeClient = new soapclient("https://$HOST:$PORT/Volume?wsdl",
  array('login'=>"$USER",
  'password'=>"$PASS",
  'cache_wsdl' => WSDL_CACHE_NONE,
  'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
  'trace'=>1)
 );

 $allVolumes=$volumeClient->getVolumes();
 echo "Successfully retrived all the volumes \n";
 $selectedVolumeIDs = array();
 foreach($allVolumes->return as $tmp) {
  // check if the user is assigned the volume and the user is the only user in the assigned list
  if (isset($tmp->userIDs) && count($tmp->userIDs) == 1 && in_array($selectedUserID, $tmp->userIDs)) {
   // if the above condition is true put the volume in the volume id in the selectedVolume list
   array_push($selectedVolumeIDs, $tmp->id);
  }
 }
 if (!isset($selectedVolumeIDs) || count($selectedVolumeIDs) == 0){
  echo " No Volumes are exclusively assigned to the specified user \n" ;
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get all volumes \n";
 echo $exception;
 exit(1);
}

########====Get Volumes Start====########
```

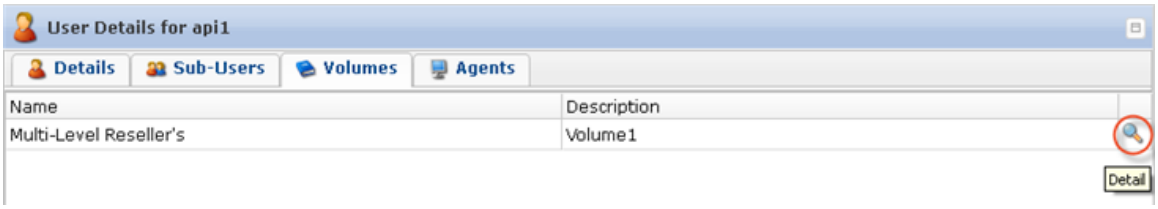
In the bottom "Details" pane, select the "Volumes" tab to list the Volumes assigned to the selected User.



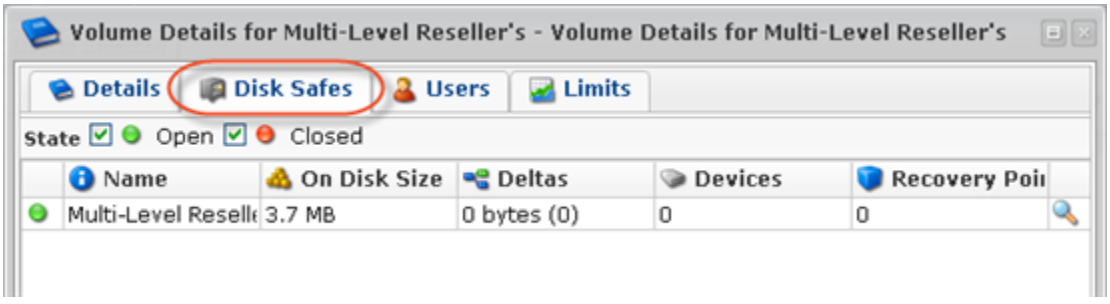
Retrieving Disk Safes

```
########====Get DiskSafes Start====########

try{

 $diskSafeClient = new soapclient("https://$HOST:$PORT/DiskSafe?wsdl",
  array('login'=>"$USER",
  'password'=>"$PASS",
  'trace'=>1,
  'cache_wsdl' => WSDL_CACHE_NONE,
  'features' => SOAP_SINGLE_ELEMENT_ARRAYS)
 );

 // retrive all the ids
 $allDiskSafes = $diskSafeClient->getDiskSafes();
 echo "Successfully retrived all the diskSafes \n";
 $selectedDiskSafeIDs = array();
 foreach($allDiskSafes->return as $tmp) {
  // check if the disksafe has a valid agent set and it's id == the agent to be deleted
  if (isset($tmp->agentID) && in_array($tmp->agentID, $selectedAgentIDs)){
   // if the condition is true store then store the ids
   array_push($selectedDiskSafeIDs, $tmp->id);
  }
 }
 if (!isset($selectedDiskSafeIDs) || count($selectedDiskSafeIDs) == 0){
  echo " No Disksafes are associated with the agents owned by the user \n" ;
 }

}
catch (SoapFault $exception)
{
 echo "Failed to get all diskSafes \n";
 echo $exception;
 exit(1);
}

########====Get DiskSafes End====########
```

1. Click on the "Detail" icon in front of the Volume to drill down to the Disk Safes.



2. In the displayed window, select the "Disk Safes" tab to list the Disk Safes assigned to the selected Volume.



Retrieving Policies

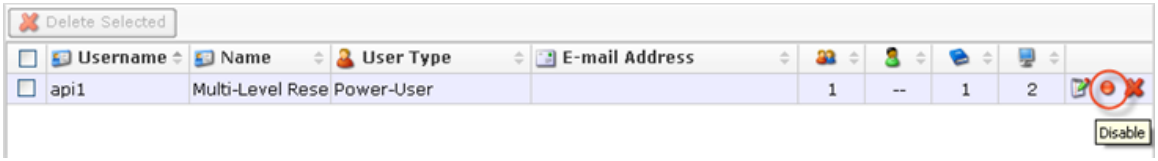
```
########====Get policies Start====########

if (count($selectedDiskSafeIDs) > 0){
 try{
  $policyClient = new soapclient("https://$HOST:$PORT/Policy2?wsdl",
   array(
    'login'=>"$USER",
    'password'=>"$PASS",
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
    'trace'=>1)
  );
  // get all the policies
  $allPolicies = $policyClient->getPolicies();

  echo "Successfully retrived all the policies \n";
  $selectedPolicies = array();
  foreach($allPolicies->return as $tmp) {
   // check to see if disksafe id of the policy belongs in the list selectedDiskSafeIDs
   if (in_array($tmp->diskSafeID, $selectedDiskSafeIDs)){
    // if the condition is true save the policy id
    array_push($selectedPolicies, $tmp);
   }
  }
  if (!isset($selectedPolicies) || count($selectedPolicies) == 0){
  echo " No Policies are associated with the disksafe assigned to the agents owned by the user \n" ;
  }

 }
 catch (SoapFault $exception)
 {
  echo "Failed to get all the policies \n";
  echo $exception;
  exit(1);
 }
}

########====Get Policies End====########
```

Disabling Policies and Users

```
 ########====Update Policies Start====########
// check to see if there are any policies to be deleted
 if (isset($selectedPolicies) && count($selectedPolicies) > 0){
  try {
   // iterate over the list of policy IDs and delete them
   foreach($selectedPolicies as $tmp) {
    $tmp->enabled = false;
    $policyClient->updatePolicy(array('policy'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to update all policie(s) \n";
   echo $exception;
   exit(1);
  }
  echo "All policie(s) update successfully \n";
 }

########====Update Policies End====########
########====Update User Start====########

 if (isset($selectedUsersObject) && count($selectedUsersObject) > 0){
  try {
   // iterate over the list of sub user IDs and delete them
   foreach($selectedUsersObject as $tmp) {
```

```
  // disable the user
  $tmp->enabled = false;

  if ($tmp->userType == "SUB_USER" || $tmp->userType == "SUPER_USER"){
   $tmp->userAttributes = array();
  }
  $userClient->updateUser(array('user'=>$tmp));
  }
 }
 catch (SoapFault $exception)
 {
  echo "Failed to update all User(s) \n";
  echo $exception;
  exit(1);
 }
 echo "All User(s) updated successfully \n";
}

########====Update User End====########
```

In the "Users" list, find the appropriate User and click on the "Disable" icon in the "Actions" column for this User.

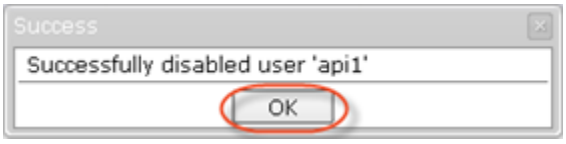


2. To confirm the deletion of the associated Policies and Sub-Users, select the following check-boxes:

- Disable Policies
- Disable Sub-Users

Click "OK."

3. Click "OK" in the confirmation dialog window.



4. After clicking on "OK," you will be returned to the Users List.