ClearPass Guest Endpoints Cleanup

Bruce Osborne - Liberty University

This is how we chose to clean up Guest Endpoints. We chose the particular API account setup because we migrated from the eTIPS API that used a username & password. We chose PHP because, up until this time, all out API work was web based and we already had some of the parts built & working. Since we use our operator account for other automations too, the profile access setup listed here is "best guess"

1. Basically, we use the REST API to get all Known Endpoints. Filtering choices are limited so this is the best filter we found. We get the Endpoints up to 1000 at a time because that is the API limit. We process them in REVERSE order (high to low) because deleting an endpoint shifts the other records on the server.
2. We then are only interests in the Endpoints that have a "Guest Role ID" attribute defined.
3. We use the "Username" attribute to look for the Guest account.
4. If the Guest Account does not exist, we delete the endpoint. Otherwise we leave it.

CPPM API Setup

1. Set up a user account with a new Guest Operator Profile.
2. Set up the guest operator profile
    a. API Services
        i. Allow API Access
    b. Guest Manager
        i. Manage Guest Accounts **Read Only**
    c. Policy Manager
        i. Identity – Endpoints **Read, Write, Delete**
    d. Operator Filter: **No operator filter**
3. Set up the API Client
    a. Client ID (we matched the Profile name)
    b. Enable API client
    c. Operator Profile **name used in step 2**
    d. Grant Type: **Username and password credentials (grant_type=password]**
    e. Public Client: **This client is a public (trusted) client**
    f. Refresh Token: **Unchecked**
    g. Access Token Lifetime: **your choice**

Script Customization

1. Define **$apiUser** with the API account username
2. Define **$apiPassword** with the API account password
3. Define **$apiClient** with the Operator Profile name

4. Define **$serverIp** with the ip address of your Publisher
5. Be sure to ssh to the Publisher by IP address to store in **known_hosts**.

Test Run

1. Set up the **.forward** file for the script user so you get the script output from cron.
2. There are some commented out print statements that can be used to help troubleshoot.

Set up Cron

We run this script automatically once a day. Here is our crontab entry.

```
5 3 * * *       /home/nowires/scripts/guest-cleanup/cron-cleanup-guests.sh
```

Script

```
#!/bin/php
<?php


// Production server constants
$apiUser = 'define';
$apiPassword = 'define';
$apiClient = 'define';
$serverIp = 'nnn.nnn.nnn.nnn';

$numKeepEvent = 0;
$numKeep = 0;
$numDelete = 0;

function processEndpoints ($offset, $num)
{
// global variables

  global $apiUser;
  global $apiPassword;
  global $apiClient;
  global $serverIp;
  global $authToken;
  global $numKeepEvent;
  global $numKeep;
  global $numDelete;
```

```
// loop through (up to 1000) entries)
   $cmd = "/usr/bin/curl -k -s -m 30 -X GET
\"https://$serverIp/api/endpoint?filter=%7B%22status%22%3A%20%22Known%22%7D&sort=%2Bid&of
fset=$offset&limi
t=$num&calculate_count=true\" -H \"Authorization: $authToken\"";
//   print "$cmd\n";


  $curlJSON = array(json_decode(exec($cmd), TRUE));
//   print_r($curlJSON);


  for ($i = 0; $i < $num; $i++) {


    $guestFlag = 0;
    $userName = "";


    $macAddress = $curlJSON[0]['_embedded']['items'][$i]['mac_address'];
//     print("Mac: $macAddress\n");


    if (isset($curlJSON[0]['_embedded']['items'][$i]['attributes']['Guest Role ID'])) {
      $guestFlag = 1;
      $userName = $curlJSON[0]['_embedded']['items'][$i]['attributes']['Username'];
    }


       if ($guestFlag == 1) {
//                print ("Endpoint Mac: $macAddress\n");
//                print ("UserName: $userName\n");


//           Look up if Guest Account exists


//           Build the initial query
           $guestQuery = "curl -X GET
\"https://$serverIp:443/api/guest?filter=%7B%22Email%22%3A%22".urlencode($userName)."%22%7D
&calculate_count
=true\" -H \"Authorization: $authToken\"  -m 30 -k -s";
//           print ("Query: $guestQuery\n");
```

```
            $curlJSON2 = array(json_decode(exec($guestQuery), TRUE));
//            print_r ($curlJSON2);

            $numRecords = $curlJSON2[0]['count'];
//            print "Records: $numRecords Mac: $macAddress\n";

//         Only proceed if NumRecords = 1
          if ($numRecords == 1) {;
            $numKeep += 1;
//               print "Keep\n";
                print ".";
                if (substr_count($userName, "@event")) {
                    $numKeepEvent += 1;
                    print "e";
                }
          } else {
                $numDelete += 1;
                print "\n$macAddress  ";

//        Now lets delete it!
                // Build the command
             $cmd = "/usr/bin/curl -X DELETE 'https://$serverIp/api/endpoint/mac-address/$macAddress' -H
\"Authorization: $authToken\" -m 30 -k -s";
//                print "$cmd\n";

                $curlJSON3 = array(json_decode(exec($cmd), TRUE));
//                if (count($curlJSON3[0]) == 0) {
                if ($curlJSON3[0] == "") {
                  print "Successfully deleted\n";
                } else {
                     print "Error deleting\n";
                    print_r($curlJSON3[0]);
                }
          }
     }
  }
```

```
}



// Get OAuth token
  // Build the command
$cmd = "/usr/bin/curl -X POST 'https://$serverIp/api/oauth' -H 'Content-Type: application/json' -d
'{\"grant_type\": \"password\", \"username\": \"$ap
iUser\", \"password\": \"$apiPassword\", \"client_id\": \"$apiClient\"}' -m 30 -k -s";


//   print "$cmd\n";


 $curlJSON = array(json_decode(exec($cmd), TRUE));


 $authToken = $curlJSON[0]['token_type']." ".$curlJSON[0]['access_token'];
// print "Token: $authToken\n";


# Get all Known Endpoints
// Seed the offset and count
$offset = 0;
unset($count);
$mod = 0;



$cmd = "/usr/bin/curl -k -s -m 30 -X GET
\"https://$serverIp/api/endpoint?filter=%7B%22status%22%3A%20%22Known%22%7D&sort=%2Bid&of
fset=$offset&limit=1
&calculate_count=true\" -H \"Authorization: $authToken\"";
//   print "$cmd\n";


$curlJSON = array(json_decode(exec($cmd), TRUE));
// print_r($curlJSON);


$count = $curlJSON[0]['count'];
print("Known Count: $count\n");


if ($count > 0) {
```

```
  $mod = $count % 1000;
  $offset = $count / 1000;
  $offset = (int) $offset;
}

if ($mod > 0) {
  processEndpoints($offset * 1000, $mod);
  --$offset;
}

if ($offset > 0) {
  for ($i = $offset; $i >= 0; $i--) {
    processEndpoints($i * 1000, 1000);
  }
}

print "\nGuest Kept: $numKeep\n";
print "Event Guest Kept: $numKeepEvent\n";
print "Guest Deleted: $numDelete\n";

?>
```