Guide to API Authentication - Access Hash Authentication

Guide to API Authentication

- Access Hash Authentication
- API Tokens in WHM
- Browser-Based Authentication
- Secure Remote Logins
- Single Sign On
- Two-Factor Authentication
- Username and Password Authentication

Introduction

Access hashes allow you to authenticate with the server as the `root` user. To view or retrieve an access hash, use WHM's *Remote Access Key* interface (*WHM >> Home >> Clusters >> Remote Access Key*).

Important:

- We deprecated WHM's *Remote Access Key* feature in cPanel & WHM version 64. We **strongly** recommend that you use API tokens instead.
- This method is **only** available for WHM authentication.
- API calls that use a method that includes a URL **must** use the correct port:
  - `2082` — Unsecure calls to cPanel's APIs.
  - `2083` — Secure calls to cPanel's APIs.
  - `2086` — Unsecure calls to WHM's APIs, or to cPanel's APIs via the WHM API.
  - `2087` — Secure calls to WHM's APIs, or to cPanel's APIs via the WHM API.
  - `2095` — Unsecure calls to cPanel's APIs via a Webmail session.
  - `2096` — Secure calls to cPanel's APIs via a Webmail session.

  Otherwise-correct calls will return `Permission denied` or `Function not found` errors if they use an incorrect port number.
- This document **only** includes cPanel & WHM authentication methods. For Manage2 authentication information, read our Guide to the Manage2 API documentation.

Access hashes

Scripts can include an access hash in the HTTP header that they send to the server during API functions. The system stores access hashes in the `.accesshash` file in each user's home directory.

Example Perl script

Click to view...

Notes:

- Replace `accesshashhere` in line 7 with the contents of the `/root/.accesshash` file. You **must** supply the access hash as a single line with no breaks.
- This script requires the `LWP::Protocol:https` module. If you attempt to run this script, you **must** first run the `/scripts/perlinstaller LWP::Protocol::https` command to install the module.
- This script calls WHM API 1's `listaccts` function. Make **certain** that you update this code for the correct API version, port, and other function-specific call information.

Related documentation

```
#!/usr/bin/perl
use strict;
use LWP::UserAgent;
use LWP::Protocol::https;
use MIME::Base64;

my $hash = "accesshashhere";

$hash =~ s/\n//g;

my $auth = "WHM root:" . $hash;

my $ua = LWP::UserAgent->new(
ssl_opts   => { verify_hostname => 0,
SSL_verify_mode => 'SSL_VERIFY_NONE',
SSL_use_cert => 0 },
 );
my $request = HTTP::Request->new(GET
=>
"https://127.0.0.1:2087/json-api/list
accts?api.version=1");
$request->header( Authorization =>
$auth );
my $response =
$ua->request($request);
print $response->content;
```

- In line 7, the script declares the `$hash` variable and assigns the access hash to it as a value.
- In line 11, the script declares the `$auth` variable, and assigns it a value of `WHM root: $hash`.
- Line 15 declares the `$request` variable, which stores information about the call. To set its value, the `HTTP::Request` module's `new()` method creates a function to the WHM API 1 `listaccts` function.
    - This call uses the GET method.
    - When you construct URLs to use this method, use the same methods as for a browser-based call.
- Line 16 uses the `header()` method to use the `$auth` value as the call's authentication information.
- Line 17 uses the `LWP::UserAgent` module to run the function.
- Line 18 prints the function's output.

Example PHP script

Click to view...

Notes:

- Replace `accesshashhere` with the contents of the `/root/.accesshash` file. You **must** supply the access hash as a single line with no breaks.
- This script calls WHM API 1's `listaccts` function. Make **certain** that you update this code for the correct API version, port, and other function-specific call information.

Content by label

There is no content with the specified labels

```
<?
$whmusername = "root";

# The contents of /root/.accesshash
$hash = "accesshashhere";

$query =
"https://127.0.0.1:2087/json-api/list
accts?api.version=1";

$curl = curl_init();
curl_setopt($curl,
CURLOPT_SSL_VERIFYHOST,0);
curl_setopt($curl,
CURLOPT_SSL_VERIFYPEER,0);
curl_setopt($curl,
CURLOPT_RETURNTRANSFER,1);

$header[0] = "Authorization: WHM
$whmusername:" .
preg_replace("'(\r|\n)'","",$hash);
curl_setopt($curl,CURLOPT_HTTPHEADER,
$header);
curl_setopt($curl, CURLOPT_URL,
$query);

$result = curl_exec($curl);
if ($result == false) {
    error_log("curl_exec threw error
\"" . curl_error($curl) . "\" for
$query");
}
curl_close($curl);

print $result;
?>
```

- Line 2 sets the `$whmusername` value as the `root` user.
- Line 5 sets the `$hash` value as the contents of the appropriate access hash.
- Line 7 assigns a WHM API 1 `listaccts` function to the `$query` value .
  - This call uses the GET method.
  - When you construct URLs to use this method, use the same methods as for a browser-based call.
- Line 14 assigns the `$header[0]` variable a value of `WHM $whmusername: $hash`.
  - The `$whmusername` variable contains the account's username, which **must** be `root` for this authentication method.
  - The `$hash` variable contains the account's access hash.
- Line 15 uses the `$header` hash to properly configure the HTTP header for the function.
- Line 16 uses the `$query` variable to pass in the function itself.
- Lines 18 through 22 execute the function.
- Line 24 prints the function's output.