

First request

URL	https://www.nulled.to/authkeys.php
Method	POST
Data	register=1&key={authkey}&hwid={hwid}&program_id={programid}

A successful answer of the above will look like the following:

```
{
  "status": true,
  "data": {
    "message": "Succesfully registered"
  }
}
```

This means that the auth key is succesfully tied to the HWID provided. Further requests to the validation endpoint will have to provide the same HWID.

Second request

URL	https://www.nulled.to/authkeys.php
Method	POST
Data	validate=1&key={authkey}&hwid={hwid}&program_id={programid}

Data is the same as the first request, only the first parameter is changed, from “register” to “validate”.

This is a successful answer:

```
{
  "status": true,
  "data": {
    "hash": "C5D39A73C63F6E1CB9DB0C30B39B0A17E58D901C8C0A0CE20629E43EEAD02AB3",
    "mid": "1468487",
    "name": "Username",
    "Likes": "1337",
    "groups": [
      "99"
    ],
    "extra": "1338",
    "message": "Challenge accepted"
  }
}
```

Request parameters

- **Auth Key**

Gotten through user input.

- **HWID**

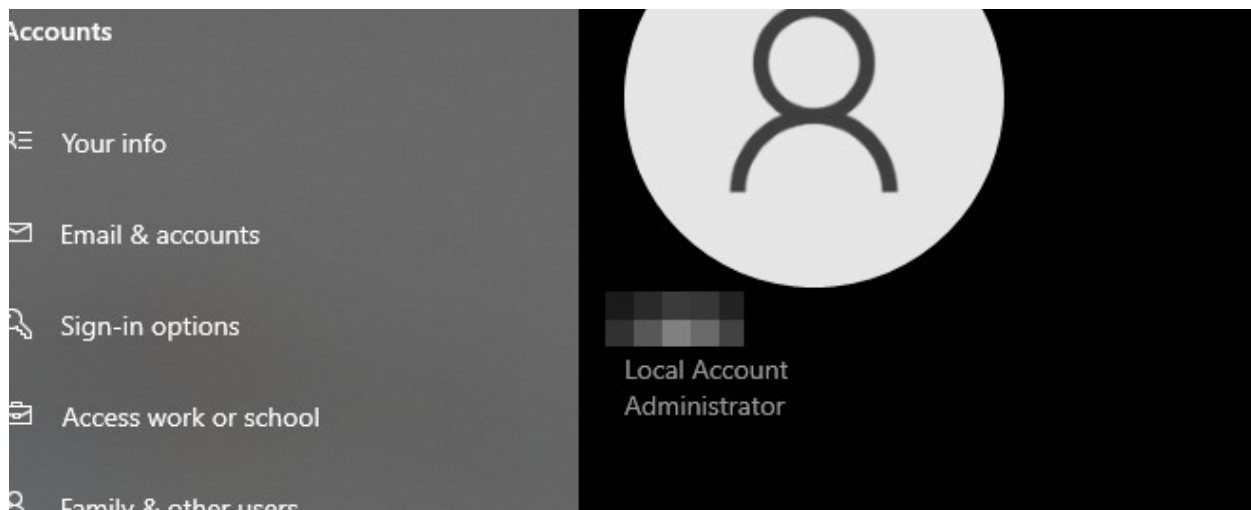
SHA256 String. Original string has the following format:

```
{c_name}{u_name}{p_rev}{disk}{uuid}{guid}
```

{c_name} is the computer name. You can find an example of this in “System Information”



{u_name} is the logged in username.



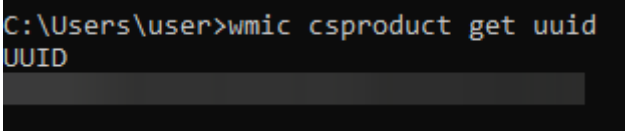
{p_rev} is the processor revision number.

All of the above can be found as an environment variable.

Further information: <https://gist.github.com/lsauer/2834199>

{disk} is the total space of the [C:\](#) drive, in bytes.

{uuid} is one of the variables than can be used to identify a computer. You can get it with “wmic csproduct get uuid” in the console.



```
C:\Users\user>wmic csproduct get uuid
UUID
```

{guid} is another variable that can be used to identify a computer. Can be found in the registry. (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MachineGuid)

You will have to find a way to get these variables programmatically.

Python example: <https://pastebin.com/9sDU3CcN>

Java example: <https://pastebin.com/5a30cZeN>

- **Program ID**

This one is provided by staff.

Validate Request Response

Getting back to the example response:

```
{
  "status": true,
  "data": {
    "hash": "C5D39A73C63F6E1CB9DB0C30B39B0A17E58D901C8C0A0CE20629E43EEAD02AB3",
    "mid": "1468487",
    "name": "Username",
    "Likes": "1337",
    "groups": [
      "99"
    ],
    "extra": "1338",
    "message": "Challenge accepted"
  }
}
```

I won't talk about the self explaining values (mid, name, likes...), I will focus on the important ones.

I will explain the data.hash string a bit later, but for now keep in mind that it's a SHA256 hash.

The data.groups array only returns the display usergroup ID.

data.extra field will tell us the permission level of the user (VIP, Aqua or Nova).
If the value is:

- **1338**

The user is *Nova*. He should have access to any software.

- **1337**

The user is *Aqua*. He will not have access to Nova only tools.

- **An epoch time in the future**

The user is *VIP*. He will only have access to VIP only tools.

Needless to say, any user that has access to auth will be able to use tools that doesn't have any usergroup requirement.

Failure conditions

- **Hash retrieved from the server doesn't match the one generated in the client side.**

The hash, as I said before, is a SHA256 string. The original string has the following format:

```
{program_secret}{auth_key}{hwid}{u_timestamp}
```

I have already explained {auth_key} and {hwid} before, so I will skip them.

{program_secret} will be provided by Staff.

{u_timestamp} is an epoch time, divided by 200, rounded and then multiplied back by 200.

Python example: <https://pastebin.com/JJSLzZ0j>

Java example: <https://pastebin.com/TM3vCqV7>

- **Epoch time from the extra field is in the past.**

Self explaining.

- **Status is set to false**

Also self explaining.

Extra Content: Group Ids

ID	Name
110	Refund God
109	Chief
108	Titan
104	Heavenly
103	Godly Refunder
102	Coder
100	Disinfector
99	Godly
98	Legendary Refunder
97	SE God
92	Nova
91	Aqua
90	Developer
80	Trial Moderator
78	Insane
73	Retired Staff
72	Middleman
38	Legendary
12	Royal
10	Contributor
8	Reverser
7	VIP
6	Moderator
5	Banned
4	Administrator
3	Members
2	Guests
1	Validating