



Sec Bug #81720 Uninitialized array in pg_query_params() leading to RCE

Submitted: 2022-05-16 14:50 UTC

Modified: 2022-06-06 07:13 UTC

From: c dot fol at ambionics dot io **Assigned:** [stas](#) ([profile](#))

Status: Closed

Package: [PostgreSQL related](#)

PHP Version: 8.1.6

OS:

Private report: No

CVE-ID: [2022-31625](#)

[View](#)

[Add Comment](#)

[Developer](#)

[Edit](#)

[2022-05-16 14:50 UTC] c dot fol at ambionics dot io

Description:

Hello PHP team,

in PHP_FUNCTION(pg_query_params), the array meant to store the char* representation of the query parameters is allocated on the heap, but not cleared:

...

```
params = (char **)safe_emalloc(sizeof(char *), num_params, 0);
```

...

If a conversion error happens (for instance, one of the params is an object), `_php_pgsql_free_params()` gets called *on the whole array*. Since the array is not initialized, a lingering value from a previous request can get freed, leading in the end to remote code execution.

To patch, use calloc or memset-0 it.

There are other functions where you use basically the same code (if cannot convert to string, then free all params) so it might be worth a look.

Patch:

...

```
- _php_pgsql_free_params(params, num_params);
```

```
+ _php_pgsql_free_params(params, i);
```

...

Best regards,

Charles Fol

ambionics.io

Test script:

<?php

```
$strings = [];
```

```
function uenc($v)
```

```
{
```

```
    $out = '';
```

```
    for($i=0; $i<strlen($v);$i++)
```

```
    {
```

```
        $out .= '\u' . '00' . str_pad(dechex(ord($v[$i])), 2, '0', STR_PAD_LEFT);
```

```
    }
```

```
    return '' . $out . '';
```

```
}

$json =
  '{"a": 1, "args":[
    "A","A","A",
    {}
  ]}'
;
$c = pg_connect('host=172.17.0.3 user=postgres password=password');

$data = json_decode($json);

$resultXXX = pg_query_params($c, 'SELECT * FROM test WHERE x NOT IN ($1)', $data->args);
// var_dump(pg_fetch_all($resultXXX));

Expected result:
-----
No crash.

Actual result:
-----
Crash.
```

Patches

[Add a Patch](#)

Pull Requests

[Add a Pull Request](#)

History

| | | | | |
|-----|----------|---------|-----------------|-----------------|
| All | Comments | Changes | Git/SVN commits | Related reports |
|-----|----------|---------|-----------------|-----------------|

[2022-05-17 09:45 UTC] [cmb@php.net](#)

-Status: Open
+Status: Verified
-Assigned To:
+Assigned To: cmb

[2022-05-17 09:45 UTC] [cmb@php.net](#)

Thanks for reporting! I can confirm the issue. I'll have a closer look.

[2022-05-17 11:17 UTC] [c dot fol at ambionics dot io](#)

Cool !

I'm wondering if you also got the previous bug, #81719, related to PDO. I didn't receive a confirmation email.

[2022-05-17 11:35 UTC] [cmb@php.net](#)

-Assigned To: cmb
+Assigned To: stas

[2022-05-17 11:35 UTC] [cmb@php.net](#)

Proposed patch:
<<https://gist.github.com/cmb69/b2b5ab0cb54a5683fe3aff4c7c09f7c2>>.

While fixing this issue, I noticed that pg_send_execute() tries to convert the \$params elements to string, but checks the wrong variable (`tmp` instead of `tmp_str`), what may cause a segfault. The patch also fixes this.

As to whether this is actually a security issue: any potential exploit requires the script to pass values which are not coercible to string to the \$params parameter of `pg_query_params()` or `pg_send_execute()`. That might be regarded as sloppy userland programming, so I'm not sure if we classify this as security issue. On the other hand, the documentation is not explicit about this conversion to string requirement (although the placeholders hint at it).

Stas, what do you think?

> I'm wondering if you also got the previous bug, #81719, related
> to PDO.

I'll have a look at that right away.

[2022-05-17 21:16 UTC] stas@php.net

-CVE-ID:
+CVE-ID: needed

[2022-05-25 21:36 UTC] stas@php.net

-CVE-ID: needed
+CVE-ID: 2022-31625

[2022-06-06 07:13 UTC] git@php.net

Automatic comment on behalf of cmb69 (author) and smalyshev (committer)
Revision: <https://github.com/php/php-src/commit/55f6895f4b4c677272fd4ee1113acdbd99c4b5ab>
Log: Fix #81720: Uninitialized array in pg_query_params() leading to RCE

[2022-06-06 07:13 UTC] git@php.net

-Status: Verified
+Status: Closed

[2022-07-07 13:08 UTC] nutza249943 at gmail dot com

Alcohol and Gaming Commission of Ontario
Contact Us | Français
Home
Gaming-Related Supplier - Manufactures
Type: Gaming-Related Supplier - Manufactures Status: Active
Number: GRSM1236604
Issued to: Relax Gaming Limited
Issue Date: 2022-03-29
Expiry Date: 2024-04-03

