Nebu1ea 韩磊 [1504895347@qq.com](mailto:1504895347@qq.com)

Web: 代码审计

解题思路:

打开网页直接给源码:

from flask import Flask

from flask import request

import hashlib

import urllib.parse

import os

import json

app = Flask(\_\_name\_\_)

secret\_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):

            os.mkdir(self.sandbox)

    def Exec(self):

        result = {}

        result['code'] = 500

        if self.checkSign():

            if "scan" in self.action:

                resp = scan(self.param)

                if resp == "Connection Timeout":

                    result['data'] = resp

                else:

                    print(resp)

                    self.append\_to\_file(resp)  #追加内容到已存在的文件

                    result['code'] = 200

            if "read" in self.action:

                result['code'] = 200

                result['data'] = self.read\_from\_file()  # 从已存在的文件中读取

            if result['code'] == 500:

                result['data'] = "Action Error"

        else:

            result['code'] = 500

            result['msg'] = "Sign Error"

        return result

    def checkSign(self):

        if get\_sign(self.action, self.param) == self.sign:

            return True

        else:

            return False

@app.route("/geneSign", methods=['GET', 'POST'])

def geneSign():

    param = urllib.parse.unquote(request.args.get("param", ""))

    action = "scan"

    return get\_sign(action, param)

@app.route('/De1ta', methods=['GET', 'POST'])

def challenge():

    action = urllib.parse.unquote(request.cookies.get("action"))

    param = urllib.parse.unquote(request.args.get("param", ""))

    sign = urllib.parse.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):

    try:

        with open(param, 'r') as file:

            content = file.read()

        return content

    except FileNotFoundError:

        return "The file does not exist"

def md5(content):

    return hashlib.md5(content.encode()).hexdigest()

def get\_sign(action, param):

    return hashlib.md5(secret\_key + param.encode('latin1') + action.encode('latin1')).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()

给了task类，

Exec函数逻辑是这样的：假如self.action中包含”scan”，就将self.param中的文件读出来存到类里，假如self.action中包含”read”那么，就把前面存进去的字符读出来显示，这里有一个逻辑漏洞，self.action是可以既包含scan又包含read的，所以可以然self.action=”readscan”或者”scanread”的同时param=想要的文件名，就可以做到读文件完直接回显。

Checksign函数的逻辑，调用get\_sign函数把action，param进行加密，假如和self.sign相等就返回true，因为Exec里调用到了这个函数，所以我们得想办法此函数返回true

接着是/geneSign路由，get输入一个param参数，让action和param进行加密生成sign，但是他把action给锁定成了”scan”。

接着是/De1ta路由，获取cookie里的action，sign以及get传入param，然后用这些参数进行task类初始化并调用Exec函数，最后返回结果

再看getSign函数，加密的逻辑是把param和action连接起来加上密钥进行md5，既然是字符串连接，那就可以令param=”flag.txtread”,这样在连接后就成了”flag.txtreadscan”，正好符合。

接着是步骤，先访问/genesign路由传入param=”flag.txtread”,得到sign=f6427bef944bd611f1ad2690a68b205a

然后访问/De1ta路由传入param=flag.txt cookie: action=readscan; sign=f6427bef944bd611f1ad2690a68b205a即可得到flag

