

XCTF-CyBRICS2021 几道简单题的WP

原创

是Mumuzi 于 2021-07-25 21:31:54 发布 837 收藏 4

分类专栏: [ctf](#)

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/qq_42880719/article/details/119086019

版权



[ctf 专栏收录该内容](#)

75 篇文章 28 订阅

订阅专栏

Your score: **684**

Your place (overall): **63**

Submits: **48**

Bruteforcer Factor: **85 %**

Last accept: **1h 22m ago**

Solved tasks:

[Profile >](#)

https://blog.csdn.net/qq_42880719

Cyber Bb 50 ⁴⁵³ sol Mic Check ✓	rebyC Bb 50 ²³¹ sol Scanner ✓	Forensic Md 469 ² sol Antarctic	Reverse Md 205 ³⁴ sol Walker	Cyber Md 299 ¹⁵ sol To The Moon	Network Ez 192 ³⁸ sol LX-100	CTB Bb 166 ⁴⁷ sol rm -rf / ✓
Network Md 444 ³ sol Future Tech	Web Md 138 ⁵⁹ sol Multichat	rebyC Ez 50 ¹⁶⁹ sol CAPTCHA The Flag ✓	CTB Md 444 ³ sol rm Escaper	Network Hd 267 ²⁰ sol localhost		
Web Ez 60 ¹¹¹ sol Announcement	Reverse Ez 136 ⁶⁰ sol Kernel Reverse	Forensic Ez 169 ⁴⁶ sol Recording	rebyC Hd 424 ⁴ sol Digital Signature			
CTB Ez 256 ²² sol Little Buggy Editor	Cyber Ez 149 ⁵⁴ sol Signer	CTB Hd 342 ¹⁰ sol GrOSs 1	Network Bb 116 ⁷¹ sol ASCII Terminal ✓	Reverse Bb 72 ¹⁰¹ sol Listing	Reverse Hd 314 ¹³ sol Paired	
Web Bb 50 ²⁰³ sol Ad Network ✓	rebyC Md 500 ¹ sol Pngoshop	Cyber Hd 342 ¹⁰ sol GrOSs 2	Forensic Hd 500 ⁰ sol Smashed Container	Web Hd 500 ⁰ sol Checkin	Forensic Bb 202 ³⁵ sol Namecher ✓	

https://blog.csdn.net/qq_42880719

1.Mic Check (Cyber, Baby, 50 pts)

Mic Check (Cyber, Baby, 50 pts)

Author: Vlad Roskov (@mrvos)

Those organizers are changing [game rules](#) all the time! There's a flag there, and it's not that easy to capture.

Also be sure to join [@cybrics Telegram chat](#) for challenge-related announcements and contacting orgs in case all goes wrong

Added at 10:10 — looks like the little mic check trolling caused massive pain, I've unrolled the rules page :-)

You can now copy-paste freely

https://blog.csdn.net/qq_42880719

签到题，查看rules，那个就是flag

solving the challenge, team gets a flag — some secret string like `cybrics{Th1S_i5_T3h_R34l_m1C_ch3CK_f1A6}`. Team submits it in exchange for points. The team with most points, wins.

To be successful in a CTF, you basically need to know computer systems good and deep.

More info about CTFs on [CTFtime website](#).

CyBRICS 2021

Sat, July 24th, 2021 10:00 UTC — Sun, July 25th, 2021 10:00 UTC (24 hours)

```
<code>cybrics{Th1S_i5_T3h_R34l_m1C_ch3CK_f1A6}</code>
```

```
cybrics{Th1S_i5_T3h_R34l_m1C_ch3CK_f1A6}
```

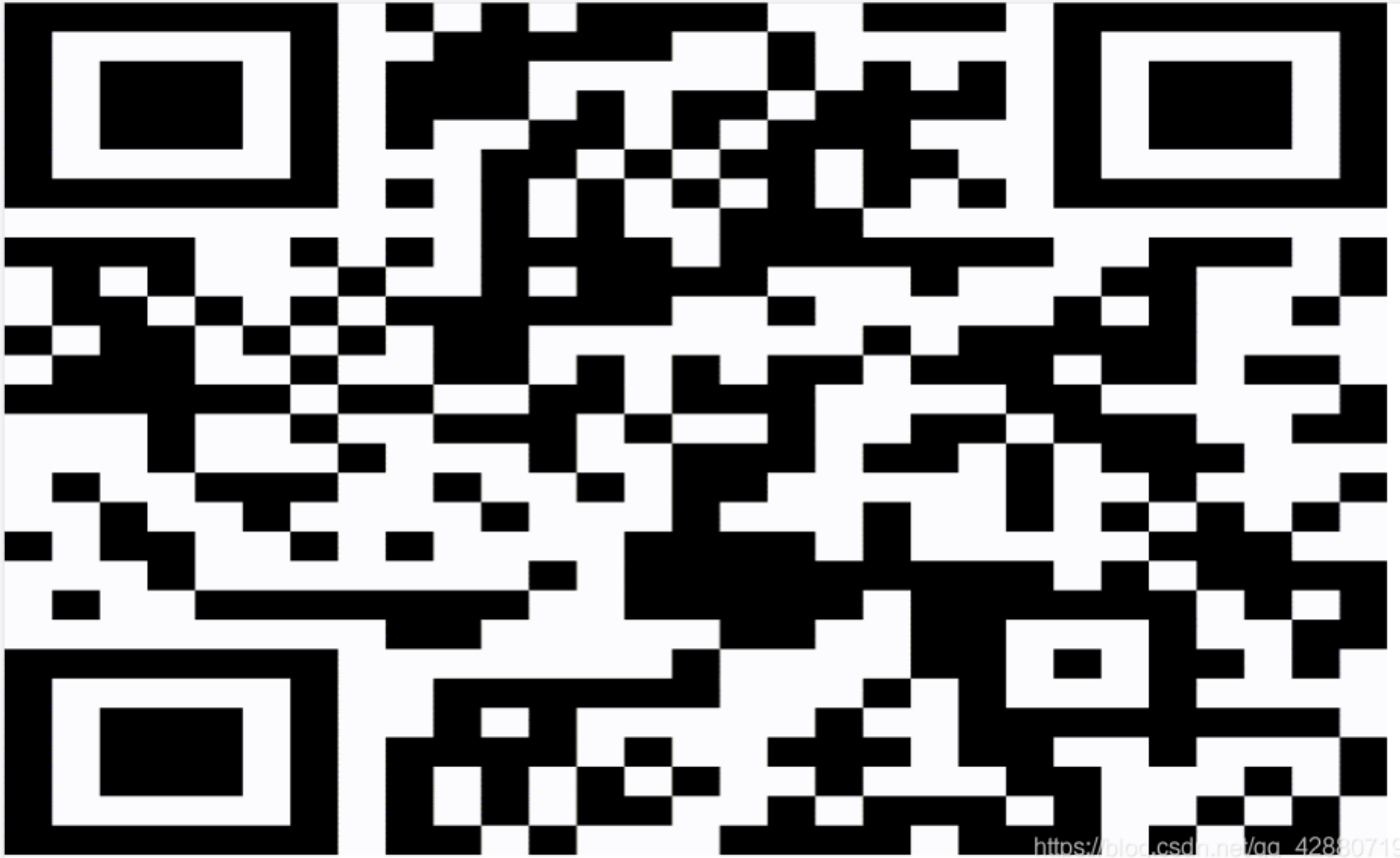
2.Scanner (rebyC, Baby, 50 pts)

第一关每次都是房子，后面三关就看到什么填什么单词就过了，最后是一张二维码。

下载下来用GIFFrame分离出来，然后可以看帧找到哪些图拿来拼起来就是一张完整的二维码

写脚本

```
from PIL import Image
list = [8, 11, 14, 16, 19, 22, 25, 28, 30, 33, 36, 39, 42, 44, 47, 50, 53, 56, 58, 61, 64, 67, 70, 72, 75, 78, 81, 84, 86]
box = [45, 496, 1033, 517]
pic = Image.new('RGB', (1000, 1000), (255, 255, 255))
c = 0
for i in range(len(list)):
    tmp = Image.open("Frame" + str(list[i]) + ".png")
    tmps = tmp.crop(box)
    pic.paste(tmps, (0, c*21))
    c += 1
    print(c)
pic.show()
```



解码即可

```
cybrics{N0w_Y0u_4r3_4_c4sh13r_LOL}
```

3.rm -rf'er (CTB, Baby, 166 pts)

rm -rf'er (CTB, Baby, 166 pts)

Author: Vlad Roskov (@mrvos)

Alarm! We accidentally did `rm -rf /*` on a very important server. Now all that's left is one shell session.

```
ssh rmrfer@178.154.210.26
```

```
Password: sa7Neiyi
```

Rescue the `flag.txt` file from one of the directories by only using your shell

Added at 13:45 — frequent question: yes, if you found `flag.txt`, the flag is right there, in the open, as plain text. Just read it. If you're not seeing the flag, try to find another method that will not hide info from you

https://blog.csdn.net/qq_42880719

很傻的非预期：因为很卡，所以在连接上到执行指令的期间，撤销他的指令，就可以使用cat指令了。大概尝试几次就找到那个点了。我是大概3.5秒的时候疯狂ctrl+c就行

```
rmrfer@178.154.210.26's password:
^C
buildbox:/home/rmrfer#
buildbox:/home/rmrfer#
buildbox:/home/rmrfer# cat
ls
ls
ls
ls
^C
buildbox:/home/rmrfer# cat /etc/ctf/flag.txt
### The flag is: cybrics{TCSHizz13_Ma_Nlzz13} ###

Congrats!
buildbox:/home/rmrfer# https://blog.csdn.net/qq\_42880719
```

预期:

因为指令被删掉,就只能看看剩余的指令,输入字母a-z其中一个然后按tab,就可以查看有哪些指令了

第一步是找flag在哪

这里发现ls-F指令没有被禁用,就利用ls-F找到flag.txt在/etc/ctf/下,cd过去即可

第二步就是输出flag,首先是找到了source,但是只能输出一小点无关内容

```
buildbox:/etc/ctf# source flag.txt
Congrats!: Command not found.
```

结合提示: Added at 13:45 — frequent question: yes, if you found flag.txt, the flag is right there, in the open, as plain text. Just read it. If you're not seeing the flag, try to find another method that will not hide info from you

可以知道flag就在flag.txt中,但是无法cat就很烦。这时第一想法是echo出来,但是echo出来就需要变量,也想到刚刚使用source,可以执行sh脚本。用echo把想执行的输入到sh中,然后再用source打印出来。

首先是试了一下,看了看他用什么执行的。反正其实也就是试然后了解到了用的是tcsh,虽然好像并没有什么用。

然后去百度搜到了有关\$的

```
buildbox:/etc/ctf# echo 'echo $0' > /flag.sh
buildbox:/etc/ctf# source /flag.sh
/usr/bin/tcsh
```

`$#` 是传给脚本的参数个数

`$ 0` 是脚本本身的名字

`$ 1` 是传递给该shell脚本的第一个参数

`$ 2` 是传递给该shell脚本的第二个参数

`$@` 是传给脚本的所有参数的列表，代表目标文件(target)

`$*` 是以一个单字符串显示所有向脚本传递的参数，与位置变量不同，参数可超过 9 个

`$$` 是脚本运行的当前进程ID号

`$?` 是显示最后命令的退出状态， 0 表示没有错误，其他表示有错误

@这个符号通常用在“规则”行中，表示不显示命令本身，而只显示它的结果

`$^` 代表所有的依赖文件(components)

`$<` 代表第一个依赖文件(components中最左边的那个)。

https://blog.csdn.net/qq_42880719

之后发现`$<`可行

```
buildbox:/etc/ctf# echo 'echo $<' >/flag.sh
buildbox:/etc/ctf# source /flag.sh <flag.txt
### The flag is: cybricsTCSHizzl3_Ma_Nlzzl3 ###
buildbox:/etc/ctf#
```

}掉了诶(我记得当时做题的时候没掉emmmm)

```
cybrics{TCSHizzl3_Ma_Nlzzl3}
```

4.CAPTCHA The Flag (rebyC, Easy, 50 pts)

CAPTCHA The Flag (rebyC, Easy, 50 pts)

Author: Vlad Roskov (@[mrvos](#))

Guessing challenges? On *my* CyBRICS? It's more likely than you think.

Prove you're a true CTFer!

captf-cybrics2021.ctf.su/

https://blog.csdn.net/qq_42880719

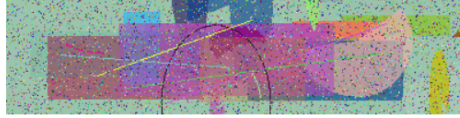
没啥意思，就浪费点时间，就验证码，要验证25次。下载下来stegsolve看R/G/B的0或者1通道即可

The Real CTF CAPTCHA

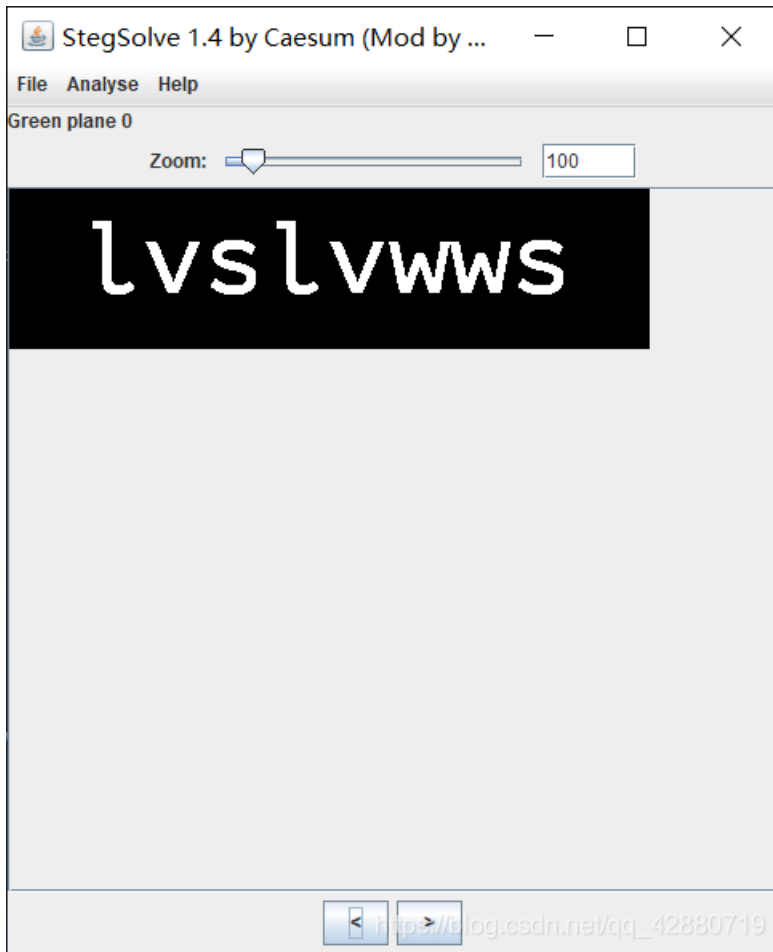
If you're a CTFer, you should be able to pass this CAPTCHA test.

Enter the letters you see in this picture:

Enter the letters you see in this picture:



https://blog.csdn.net/qq_42880719

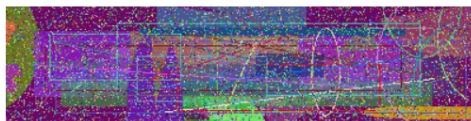


The Real CTF CAPTCHA

If you're a CTFer, you should be able to pass this CAPTCHA test.

Correct! Done 1 / 25

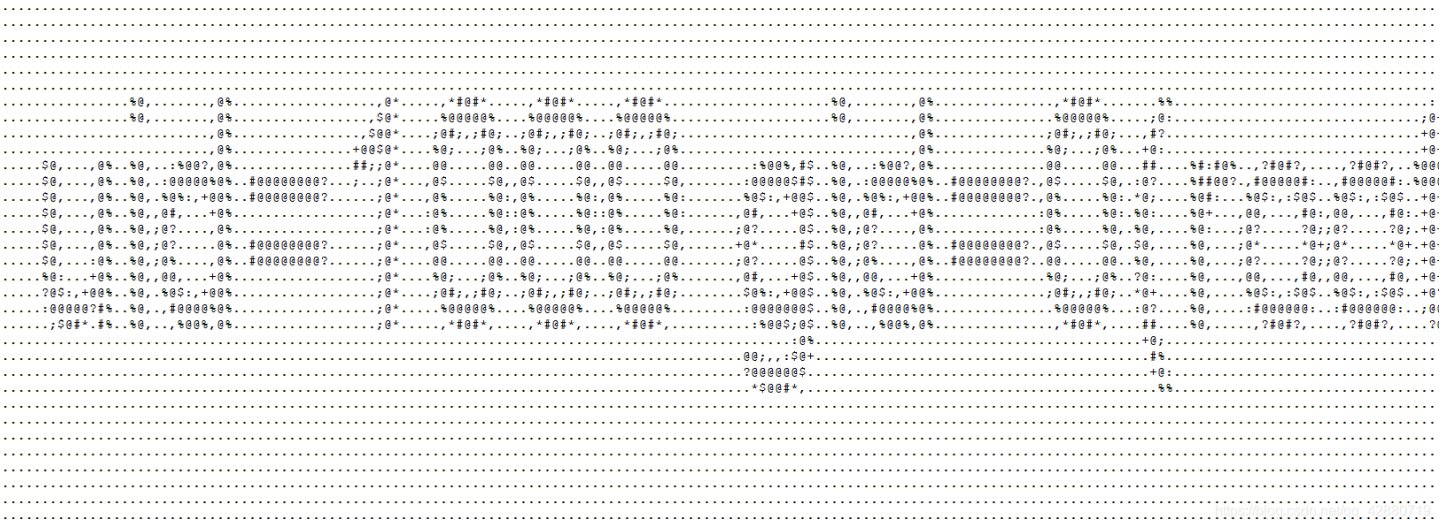
Enter the letters you see in this picture:



https://blog.csdn.net/qq_42880719



然后他说执行了id，并且后面打印了一堆东西



所以这道题的意思是输入的ascii art会被识别然后转成指令。

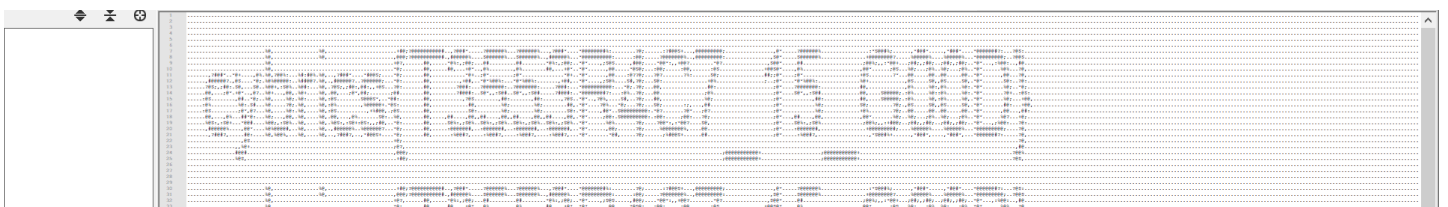
我当时是ls之后发现flag.txt，然后cat的flag.txt，用的图片转换成ascii

<https://www.qtool.net/imgascii>

关于ls指令被识别成大写的i和s

这里换一个字，换成像我们那种手写体的就行，或者用别的指令吧。

不去操作了，当时粘贴到了记事本，直接放那个图了。





```
cybrics{T3553R4C7_15_G00D}
```

6. Ad Network (Web, Baby, 50 pts)

Ad Network (Web, Baby, 50 pts)

Author: Alexander Menshchikov (@n0str)

We are so tired of advertising on the internet. It feels like it breaks the internet. Try to follow the ad, try to follow its rules.

[Adnetwork website](#)

There is a flag 1337 redirects deep into the network...

https://blog.csdn.net/qq_42880719

写的很明确了，重定向1337次即可。

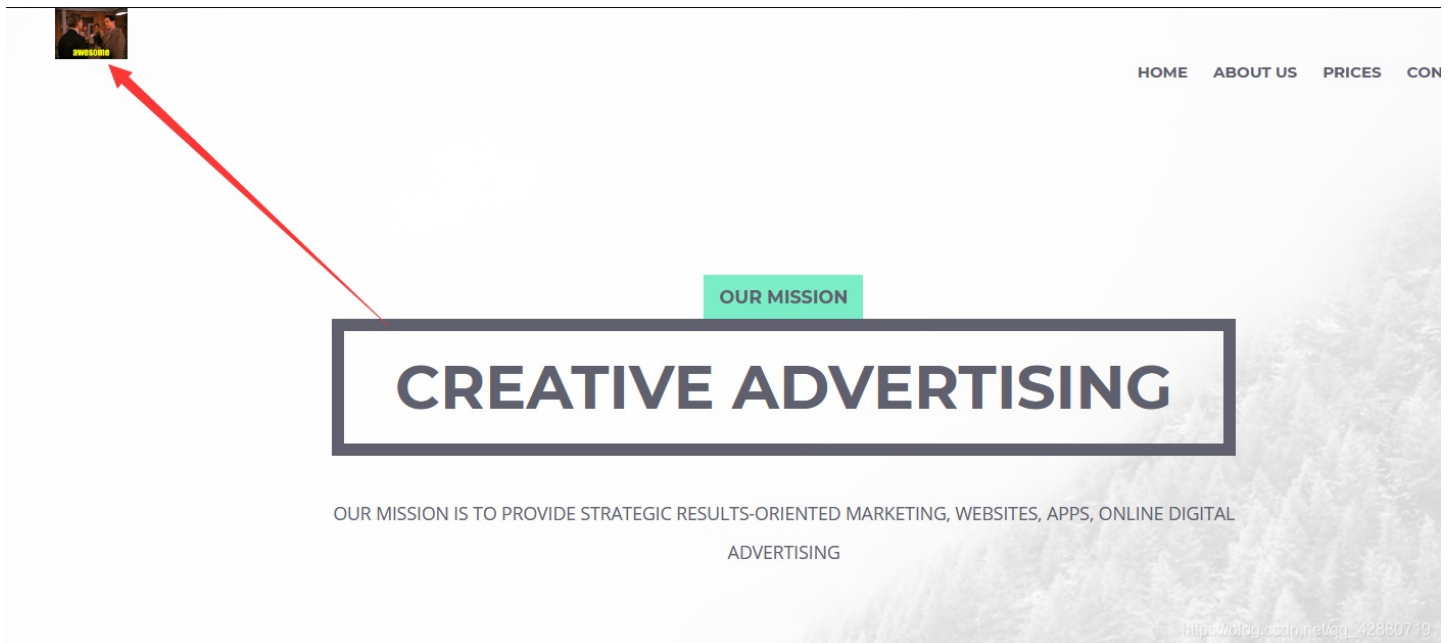
方法一：火狐修改限制，因为火狐默认貌似是20次(我反正每20次就自动停止重定向)，所以要去about:config里面改然后搜redirect，就找到了

redirect	
browser.urlbar.tipShownCount.searchTip_redirect	1
extensions.webextensions.identity.redirectDomain	extensions.allizom.org
network.http.prompt-temp-redirect	false
network.http.redirection-limit	20
network.websocket.auto-follow-http-redirects	false

https://blog.csdn.net/qq_42880719

改成1500吧，等他跑完就完事

啊还忘了说在哪：



方法二：写脚本

```
import requests
import re
url = 'http://adnetwork-cybrics2021.ctf.su/adnetwork'
for i in range(1500):
    r = requests.get(url,allow_redirects=False)
    url = re.findall('<a href="(.*?)>',r.text)[0]
    print("[*]重定向{}次, ".format(i) + "url为" + str(url))
    if 'cybrics{' in r.text:
        print(r.text)
        break
```

最后的url

<http://tend.adnetwork-cybrics2021.ctf.su/military-front-low/learn-fill-though-factor-line/hear-hundred-subject-wind/enough-lot-tree-will-color>

cybrics{f0lL0w_RU3Z_F0ll0W_r3d1r3C7z}

Namecheck (Forensic, Baby, 202 pts)

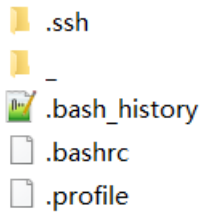
Author: Alexander Menshchikov (@n0str)

We have got the home folder from a criminal's computer. Try to find his/her real name.

eyebulling.tar.gz

Flag format in uppercase: LASTNAME FIRSTNAME (ex: IVANOV IVAN)

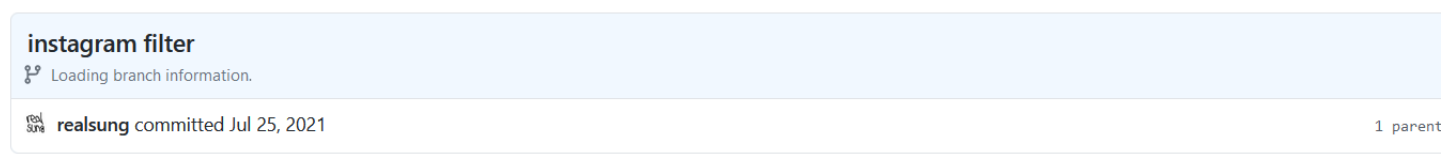
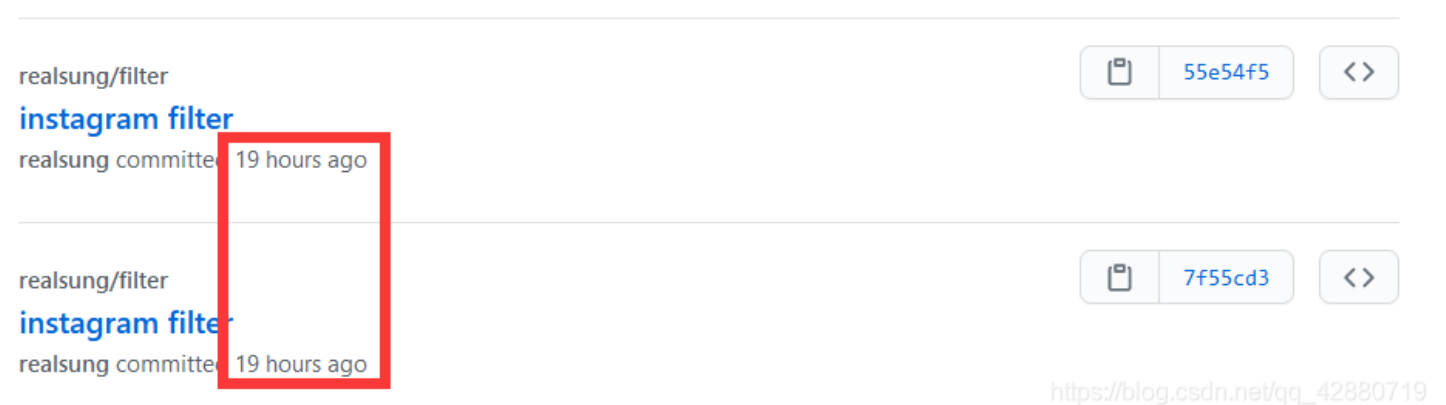
首先解压得到这些



bash_history发现添加了一条commit

```
git add *
git commit -m "instagram filter"
git push origin main
rm *
ls -la
rm -rf .git
```

然后就是去github搜



```
... @@ -1 +1 @@
1 - FLAG{FAKE_ZZLOL}
1 + cybrics{criminal name is Ugonfor Lee}
```

0 comments on commit 55e54f5

https://blog.csdn.net/qq_42880719

然后就是，你被骗了。

当然没那么简单，首先可以知道instagram是Facebook的一款应用，拿来放图片的。

然后再看刚刚文件夹里面，ssh里有个key，拿去直接base解码在最后就可以发现他的邮箱

Base64 编码或解码的结果:

```
MDI0p08w9vmbQ7mZLnRZ07mCqai0WixFtVKNKXtPFPQbvgsadyA9vognmZKtmM7
MN2lO/joZnKLkUYK5YPuA2rcSzURf835w+OMqelsor3Yg/hPxWA22q/UKjvub5SGEG0x02
1gch9WxCCgdzk2dgbMhX9b0NevCT2gehQihHtqSz1oUD0U4hGPKfaadoYttiG66Beah3gn
lfXJOsU5zW8G/DvEvBBznAJ3k+8nB5AAAawQD25PIGa2EaeCVct/jaM2JBRJCn4EymdNf9
F5uQGmPeTkBaxGvYo8Y7r5QLJ5TJYFvJ1MGdVMyp9H1jG6leEAags2XWIEtkjH6YQ/+58
i9OVkujMYCYkcv+y63eah5QKlK7+ZXjhiaHYG2s7BWI0gKlw813Xr081JVqvYMFildNmD
Er2HFGcOvURpww49jLfz1z9ZL6FbltgaKa+A02Ci8kVJR8M8INDRfupaPPdbAeGymmS5EM
0i+6RftAV/kw8AAAAUdml2aWRjb2FsYUBsb2NhbGhvc3QBAgMEBQYH
```

编码 (Encode) 解码 (Decode) ↑ 交换 (编码快捷键: **Ctrl** + **Enter**)

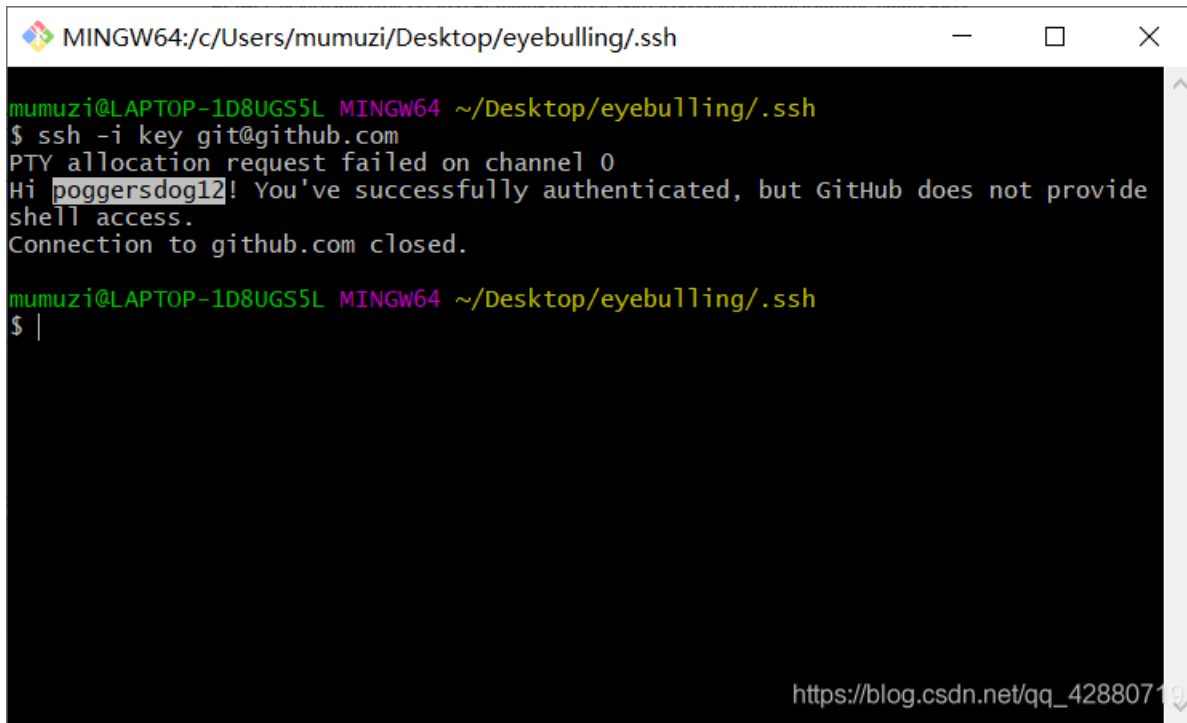
Base64 编码或解码的结果:

□ 编

```
zun86^LcSh_m_0~li2#@&fe1B(GZ)vCq|f_纒+fpxt7bK-)5搵dYK阕Tz6
mE_lqWo3金C_t宏C_Sw>eDC?rIK'qx,wx55_Q07@!q|¥@_OR%SC<[pm]_g=
(pA(Ⓢ)+Xu3
قم(X>زijQ8<v|y|z?V_mB凜a|;|6L!IB
sg`lū
zB!O'o!io!ihb!ywr!\SoüKG9'y>py_@鯨ka_x%_bADt_c_s@Zi2!g!U'S_uS2vnx@X1a爽9R5c)Z梛
Z](<IV>Xt!gDip=Y/[!])3EIGüB~!d_ividcoala@localhost
```

https://blog.csdn.net/qq_42880719

并且在git里输入ssh -i key git@github.com还能解析出他github用户名



```
MINGW64:/c/Users/mumuzi/Desktop/eyebulling/.ssh
mumuzi@LAPTOP-1D8UGS5L MINGW64 ~/Desktop/eyebulling/.ssh
$ ssh -i key git@github.com
PTY allocation request failed on channel 0
Hi poggersdog12! You've successfully authenticated, but GitHub does not provide
shell access.
Connection to github.com closed.

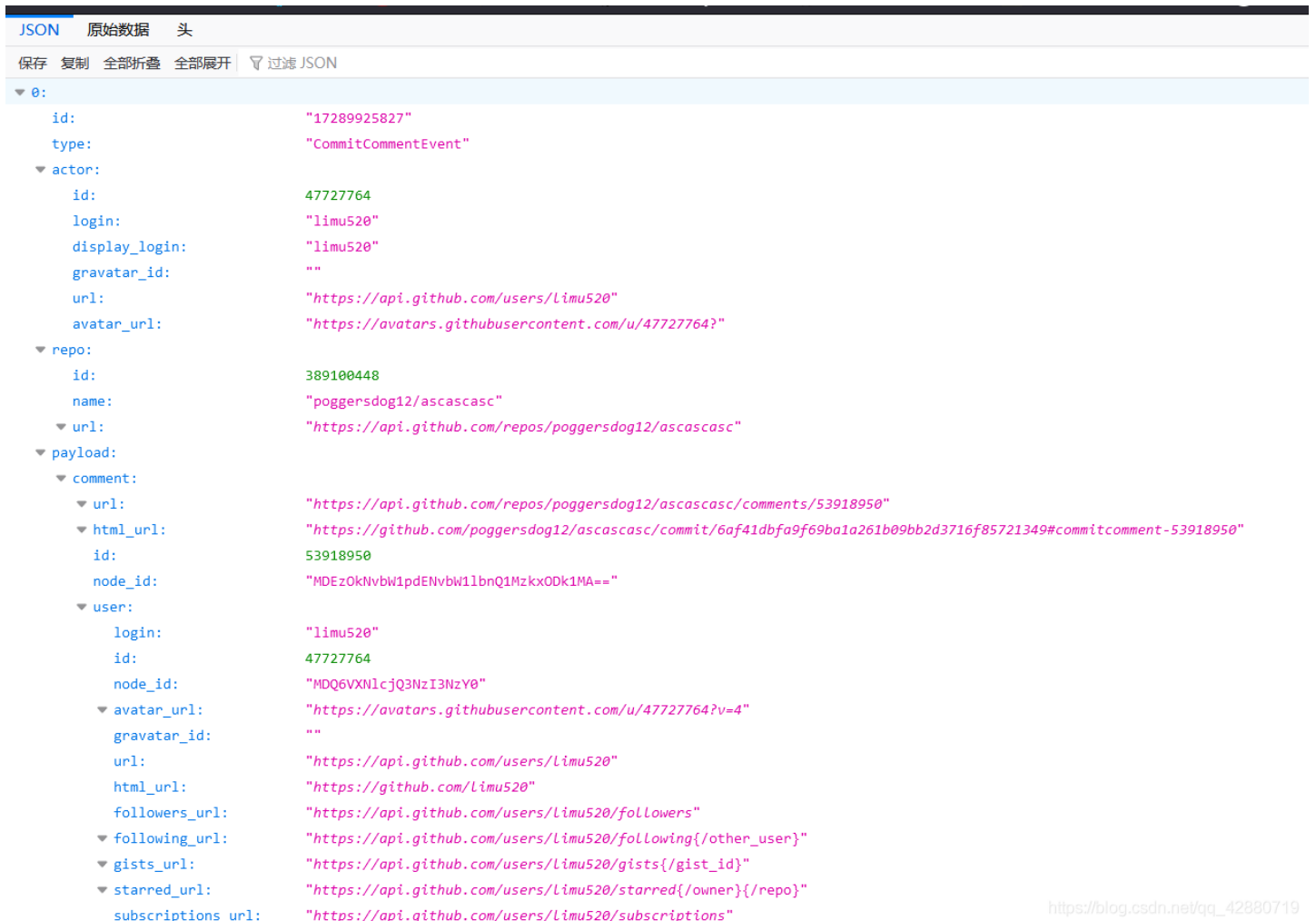
mumuzi@LAPTOP-1D8UGS5L MINGW64 ~/Desktop/eyebulling/.ssh
$ |
```

嗯！查有此人。

但是他没啥记录，然后用github的api

<https://api.github.com/repos/poggersdog12/ascascasc/events?page=1>

(?为什么看到了limu大佬



```
JSON 原始数据 头
保存 复制 全部折叠 全部展开 | 过滤 JSON
0:
  id: "17289925827"
  type: "CommitCommentEvent"
  actor:
    id: 47727764
    login: "limu520"
    display_login: "limu520"
    gravatar_id: ""
    url: "https://api.github.com/users/Limu520"
    avatar_url: "https://avatars.githubusercontent.com/u/47727764?"
  repo:
    id: 389100448
    name: "poggersdog12/ascascasc"
    url: "https://api.github.com/repos/poggersdog12/ascascasc"
  payload:
    comment:
      url: "https://api.github.com/repos/poggersdog12/ascascasc/comments/53918950"
      html_url: "https://github.com/poggersdog12/ascascasc/commit/6af41dbfa9f69ba1a261b09bb2d3716f85721349#commitcomment-53918950"
      id: 53918950
      node_id: "MDEzOkNvbW1pdENvbW11bnQ1MzkxODk1MA=="
      user:
        login: "limu520"
        id: 47727764
        node_id: "MDQ6VXNlcjQ3NzI3NzY0"
        avatar_url: "https://avatars.githubusercontent.com/u/47727764?v=4"
        gravatar_id: ""
        url: "https://api.github.com/users/Limu520"
        html_url: "https://github.com/Limu520"
        followers_url: "https://api.github.com/users/Limu520/followers"
        following_url: "https://api.github.com/users/Limu520/following{/other_user}"
        gists_url: "https://api.github.com/users/Limu520/gists{/gist_id}"
        starred_url: "https://api.github.com/users/Limu520/starred{/owner}/repo"
        subscriptions_url: "https://api.github.com/users/Limu520/subscriptions"
```

然后可以查到他的repo

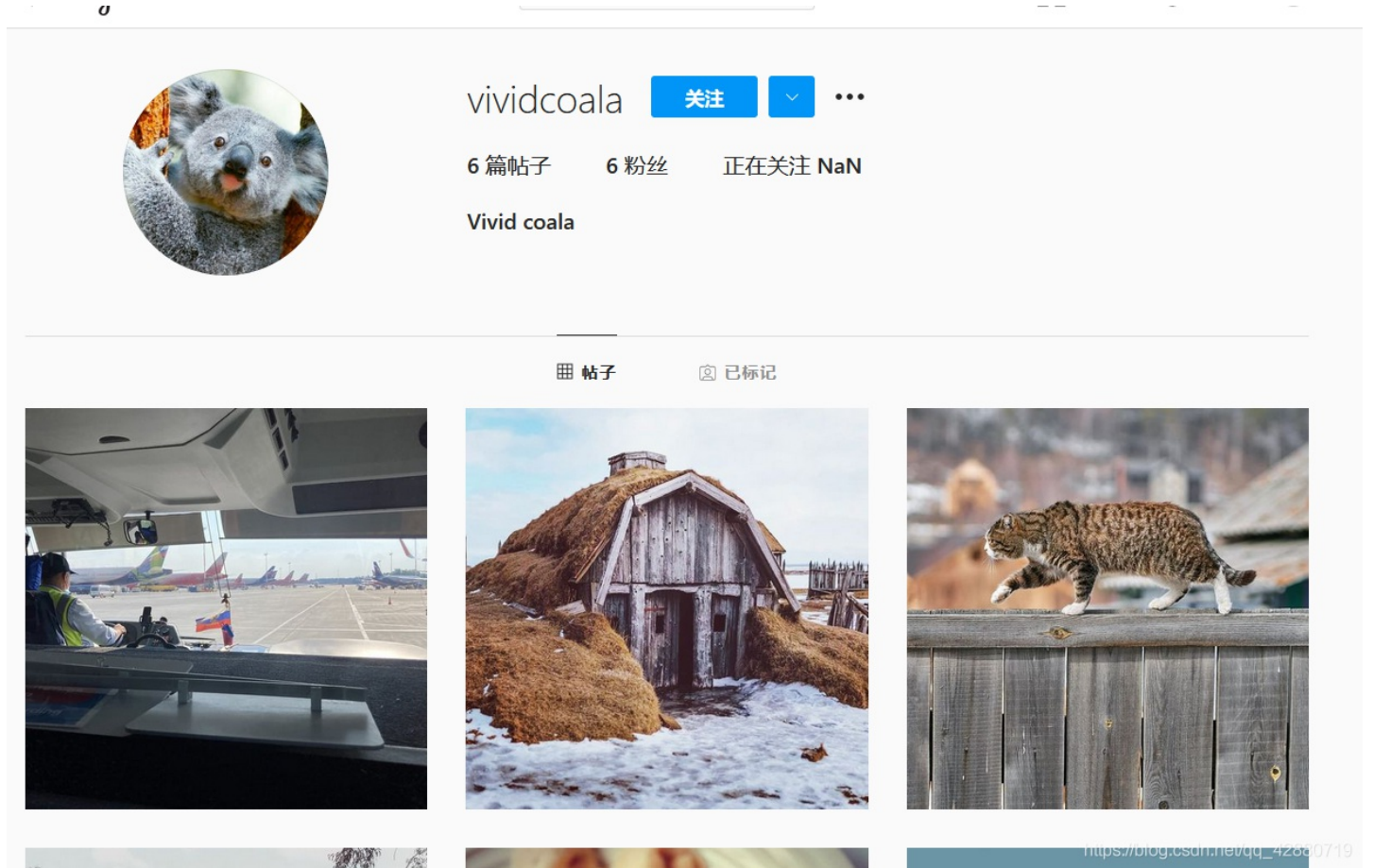
<https://api.github.com/repos/poggersdog12/ascascasc/commits/6af41dbfa9f69ba1a261b09bb2d3716f85721349>

只说了，反正没用，也是只能查到名字vividcoala@localhost.com

反正最后都要跑到instagram去

在instagram搜vividcoala(不要走错了

<https://www.instagram.com/>



点他头像，就可以看到他机票的一个图。

(但是那张图貌似只能保存24小时，而且并不能下载，要去下载插件才能下载下来)





放大那个条码，扫一扫<https://online-barcode-reader.inlitteresearch.com/>
记得勾选第六个 Driver License, ID cards






Pages: 1 **Barcodes:** 1

Barcode: 1 of 1 **Type:** Pdf417 **Page 1 of 1**
Length: 157 **Rotation:** right
Module: 3.1pix **Rectangle:** {X=25,Y=16,Width=162,Height=552}

Barcode Text processing:
Signature: IATA-BCBP

M1DIVOV/NIKOLAI MR EQCMYKK SVOLEDSU 0024 197Y020D0053 162<532
1MR1197BSU 2A555604939055
9 1 N



https://blog.csdn.net/qq_42880719

所以他的名字(flag)就是

DIVOV NIKOLAI