2018-Xnuca hardphp writeup

wonderkun / 2018-11-28 14:32:00 / 浏览数 3234 安全技术 CTF 顶(0) 踩(0)

Xnuca - hardphp writeup

题目被大佬非预期的注入成功了，膜大佬，tql ..逃 。。。。
非预期解也挺有意思的，详情可以去看https://xz.aliyun.com/t/3428,我这里只说一下预期解法吧。

题目的docker环境已经上传到https://github.com/wonderkun/CTF_web/blob/master/web400-12/README.md?1543386212265了，感兴趣的可以去下载。

下面是题解:

0x1 任意session伪造漏洞

首先看session的处理类:

```
#  file: include/MySessionHandle.php
public function write($session_id,$data){
     $time = time();
     $res = $this->dbsession->query("SELECT * FROM `{$this->dbsession->table_name}` where `sessionid` = '{$session_id}' ");

     if($res){
       $this->dbsession->execute("UPDATE `{$this->dbsession->table_name}` SET `data` = '{$data}',`lastvisit` = '{$time}' where
     }else{

       $res = $this->dbsession->create(
           ["data"=>$data,
           "sessionid"=>$session_id,
           "lastvisit"=>$time]);
     }
     return true;
  }
```

这里的 update 操作没有对 $data 进行 addslashes 操作，所以是有问题的。

虽然前面有简易的waf，对用户的输入进行了过滤：

```
function escape(&$arg) {
   if(is_array($arg)) {
       foreach ($arg as &$value) {
           escape($value);
       }
   } else {
       $arg = str_replace(["'", '\\', '(', ')'], ["’", '\\\\', '■', '■'], $arg);
   }
}
```

执行到这里 $data 已经不可能再单个出现 \ ，' 和(,)了，所以是不存在注入的(不考虑那个非预期解的情况)，但是要注意是 $data
是序列化之后的数据，存在一个长度的问题。

假如说 $data 是 s:6:"test\\"（因为在对 Session 类进行序列化之前，已经对所有成员用 escape 函数处理过一次），那么写入数据库之后就变成了
s:6:"test\"，那么在执行反序列化的时候，就会导致后面一个字符被吞掉。

先来看一下数据库中存储的 session 的格式：

```
data|s:288:"O:7:"Session":4:{s:11:" Session ip";s:10:"172.18.0.1";s:18:" Session userAgent";s:128:"Mozilla/5.0 ?Macintosh; Int
```

利用上面说的那个问题，我们可以在 data 字段注入 \ ，向后面吞字符，吞掉后面的 username字段，然后在 username
值的位置布局我们想要的数据，来伪造session。

php 处理 session 的时候，有一个机制是这样的，后面的值会覆盖掉前面的值，例如：

```
data|s:5:"guest";data|s:5:"admin";
```

那么在获取 $_SESSION['data'] 的时候，后面的值就会覆盖掉前面的值，得到的是 admin。

通过上面的分析，思路就明确了，我们注册的用户名可以是我们需要伪造的 $_SESSION['data']的序列化内容， 然后在真正的 $_SESSION['data']
位置引入 \，让他吞掉后面的字符 ";username|s:9:"，实现伪造 $_SESSION['data'] 的值 ,得到一次反序列化的机会。

0x2 构造执行路径

通过上面的分析，我们可以任意伪造 $_SESSION['data'] 的内容，并且可以获取到一次反序列化的机会。但是题目的代码限制了反序列化：

```
$session = unserialize($_SESSION["data"],["allowed_classes" => ["Session"]]);
```

这里仅仅可以反序列化出来 Session 类，所以这里是没有办法实现任意任意类伪造,所以是没有办法构造 pop chains的。

但是，看代码发现有一个静态的函数调用：

```
$this->now = $session::getTime(time());
```

php是动态语言，是支持动态的函数调用的，比如说：

```
$a = phpinfo; $a();
```

我们可以扩展一下思考，示例代码如下：

```
class A{

   public static  function test(){
       echo "hacked";
   }
}

class B{
   public $name = "i am b";
}
unserialize(serialize("A"),["allowed_classes" => ["B"]])::test();

// hacked
```

也就是说我们其实是可以通过字符串的方式来调用一个类的类方法，等价于下面的方式：

```
$a = "A";  $a::test();
```

通过这种方式，我们就可以在没有反序列出来 A类的情况下，利用字符串"A"调用 A 类的静态方法。

但是我们能调用哪个类的静态方法呢？ 也没有哪个php内置类有 getTime 这个静态方法吧？

0x3 类自动加载器来助攻

```
$__module     = isset($_GET['m']) ? strtolower($_GET['m']) : '';
$__controller = isset($_GET['c']) ? strtolower($_GET['c']) : 'main';
$__action     = isset($_GET['a']) ? strtolower($_GET['a']) : 'index';
$__custom    = isset($_GET['s']) ? strtolower($_GET['s']) : 'custom';

spl_autoload_register('inner_autoload');
function inner_autoload($class){
   // echo $class."\n";
   GLOBAL $__module,$__custom;
   $class = str_replace("\\","/",$class);
   foreach(array('model','include','controller'.(empty($__module)?'':DS.$__module),$__custom) as $dir){
       $file = APP_DIR.DS.$dir.DS.$class.'.php';
       if(file_exists($file)){
           include $file;
           return;
       }
   }
}
```

看到这些代码，那么思路就串起来了，我们可以通过刚才伪造的字符串来调用一个类，然后就会触发一次类加载，
然后类加载器就会在我们制定的目录下寻找这个类的代码，并包含进来。只需要控制 s=img/upload/，就能实现包含我们上传的php文件。

0x4 详细攻击流程

1. 上传一个 php shell 文件，然后记录下其文件名，例如： 28mlzz380bs8e4sr6e98xzqxbdx2lj9m.php
2. 再注册一个账户，账户名为： ;data|s:40:"s:32:"28mlzz380bs8e4sr6e98xzqxbdx2lj9m";
3. 用上面的账号登录，设置 User-Agent为 \\\\\\\\\\\\\\\\ 16个反斜杠，根据自己的情况调整。
4. 然后访问 http://127.0.0.1:8888/main/index?s=img/upload/ 就getshell了。

上一篇：X-NUCA revenge骚思路... 下一篇：PHP利用PCRE回溯次数绕过某些...

1. 3 条回复

pav1 2018-11-30 10:52:14

膜 鲲佬

0 回复Ta

59657****@qq.com 2018-12-17 14:57:50

```
    $session = new Session($res[0]["id"],time(),$ip,$userAgent);
    $_SESSION['data'] = serialize($session);
    $_SESSION['username'] = $username;
```

生成session进行序列化的时候是生成的session['data']，怎么会把sessoin['username']写入数据库呢

0 回复Ta


59657****@qq.com 2018-12-17 15:25:44

@59657**@qq.com 知道了

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