河北师范大学 软件学院
Software College of Hebei Normal University

Linux平台PHP服务端开发——

第十九讲 XSS和CSRF

XSS简介

- 跨站脚本攻击（Cross Site Script），为避免和CSS样式表冲突，简称为XSS。
- 前端页面可以提交JS代码，HTML标签等数据，后台不做处理，或处理不当。导致恶意代码嵌入到前端页面中。
- XSS攻击方式有多种：获取用户信息，账户权限，控制受害者机器发起网络攻击…

XSS示例

- 编写一个简单的测试网站，文章内容页面允许评论，后台处理评论内容的代码如下：

```
public function comment()
{
    $id = input('?post.id')?input('post.id'):'';
    if (empty($id)) {exit('Deny!');}
    $comment = input('?post.comment')?input('post.comment'):'';
    if (empty($comment)) {exit('Deny');}
    $cid = Db::name('content_comment')
            ->insertGetId([
                'user_id' => Session::get('user_id'),
                'content_comment' => $comment,
                'content_id' => $id
            ]);
    return $cid?$cid:'failed';
}
```

- 这段代码没有对提交的数据做处理，如果用户提交的数据含有JS，就会在前端页面执行。

XSS示例

- 前端页面评论区域内容:

- 再次访问前端页面:

XSS示例

- 如果填写<img src="http://www.atest.com/u/126" alt="">，而存在XSS漏洞的网站用户量巨大。此内容发布出去，点击量非常大，而每个用户点击页面都会尝试加载此img链接。

- 这等于是利用网站的XSS漏洞以及网站用户对一个其他站点发起DDOS攻击。如果http://www.atest.com站点没有应对高并发的措施，尤其是单机系统，很可能会因此而崩溃。

CSRF简介

- 跨站请求伪造（Cross-site request forgery），简称CSRF。
- CSRF要伪装来自受信任的用户，在用户登录过要攻击的网站，并且cookie或SESSIONID未过期才有效。
- CSRF无法获取用户的cookie等登录凭证，而是通过其他手段直接利用。
- CSRF并不一定会攻击成功，往往要经过一段时间甚至很长时间的等待才会有用户受害。

CSRF触发条件

- 攻击者要非常了解网站。
- 被攻击者已登录过被攻击的网站，并且会话有效。
- 引诱被攻击者点击含有恶意脚本或者是恶意请求的链接。
- 被攻击网站没有对请求进行token验证或其他验证的处理；或者即使有处理但是网站存在XSS漏洞。

- 注意：访问其他链接指向的页面，如果页面中存在指向被攻击网站的请求，而此时被攻击者已经登录了被攻击网站，那么在其他链接页面中的请求也会带上被攻击者的cookie或是SESSIONID，如果此时用户会话有效则攻击成功。这是CSRF攻击成功的因素。

CSRF示例

- 对后台管理员的攻击：

  这种情况是在了解后台路径的基础上进行的攻击，比如使用wordpress，drupal，joomla这些比较流行的CMS；ThinkPHP，Laravel等流行的框架。如果这些工具发现了问题，则基于这些工具开发的系统就会存在问题，后台路径是固定的或是类似的。

  假设后台有后台接口：

  ```
  /index.php/admin/index/delcontent?id=5
  ```

  可以删除id为5的文章。此时如果后台管理员在登录以后，打开了另一个链接，而此链接中包含以下内容：

  ```
  <img
  src=http://192.168.56.103:6789/index.php/admin/index/delcontent?
  id=5 alt="">
  ```

  此时img标签的请求由于获取不到正常的图像数据，就显示空白，但是由于管理员登录了，请求会被正确执行。因为这段代码虽然在其他网站页面，但是由于管理员的登录会话有效，而在其他页面发起的请求仍然会带有会话ID。

XSS与CSRF配合攻击

- 如果网站同时存在XSS漏洞和CSRF漏洞。则利用XSS和CSRF会导致更严重的问题。

- 如果网站同时存在XSS漏洞和CSRF漏洞，按照示例网站，直接在评论区输入以下内容：

```
<img
src="http://192.168.56.103:6789/index.php/index/user/setnickname?nickname=phpbest"
alt="">
```

- 示例网站提交评论后，这段代码会在用户查看内容时，在评论中加载，而图片无法正常显示，会根据alt属性显示为空。但是请求执行成功，如果用户登录，则会直接把用户的昵称设置为nickname。

如何防止XSS

- 不要引入不可信的第三方JS库。
- 采用HTML标签过滤：htmlentities，此函数会把HTML的标签进行转义。

- 但是注意转义的数据并不是完全失效，如果是放在<script>标签中或者某些标签属性里，有可能仍然会被运行。所以对用户的输入仅仅放在指定的标签内，并做一些强过滤处理，移除某些标签的内容，移除事件绑定。

- 不要把不可信的输入插入到标签的属性里。

防止CSRF

- HTTP请求头referer字段指明了请求来源地址，通过验证此字段可以屏蔽除本站以外的其他链接，但是此字段是浏览器提供的，可以伪造。

- 所有需要更新数据库操作的请求都要加上token验证，每次请求随机生成token并在前端页面保存token到cookie或是页面的隐藏标签里。每次请求要验证token是不是正确。这种方式要求系统不能有XSS漏洞，否则还是会被攻击。