



越权漏洞分享

白色键盘`

目录

CONTENTS



- 01. 越权漏洞概述
- 02. 越权漏洞案例
- 03. 越权演示即分析
- 04. 如何防范越权漏洞



PART 1

越权漏洞概述

越权风险问题即产生

越权风险问题

越权访问是Web应用程序中一种漏洞

越权访问漏洞的产生

比如，某个订单系统，用户可以查询自己的订单信息。A用户查询订单时，发送的HTTP请求中包含参数`orderid=A`，订单系统取得orderid后最终会查询数据库，查询语句类似于`select * from tablename where orderid = A`。B用户查询订单时，发送的HTTP请求中包含参数`orderid=B`，系统查询数据库语句类似于`select * from tablename where orderid = B`。正常情况下，每个用户只会查询到自己的订单。但是，当B用户将自己的HTTP请求参数修改为`orderid=A`，那么最终B用户执行的数据库语句变成了`select * from tablename where orderid = A`，导致A的订单信息被B用户获取到了。

越权漏洞简单概述

如果使用A用户的权限去操作B用户的数据，A的权限小于B的权限，如果能够成功操作，则称之为越权操作。越权漏洞形成的原因是后台使用了不合理的权限校验规则导致的。一般越权漏洞容易出现在权限页面（需要登录的页面）增、删、改、查的地方，当用户对权限页面内的信息进行这些操作时，后台需要对当前用户的权限进行校验，看其是否具备操作的权限，从而给出响应，而如果校验的规则过于简单则容易出现越权漏洞。

因此，在在权限管理中应该遵守：

- 1.使用最小权限原则对用户进行赋权;
- 2.使用合理（严格）的权限校验规则;
- 3.使用后台登录态作为条件进行权限判断,别动不动就瞎用前端传进来的条件;



漏洞产生条件

越权漏洞高于逻辑漏洞（逻辑漏洞就是指攻击者利用业务的设计缺陷，获取敏感信息或破坏业务的完整性。一般出现在密码修改、越权访问、密码找回、交易支付金额用户登录等功能），是由于权限校验的逻辑不够严谨导致的。每个应用系统其用户对应的权限是根据其用户功能划分的，而每个企业的业务又都是不一样的。因此越权漏洞很难通过扫描工具发现出来，往往需要手动进行测试。



PART 2

越权漏洞案例

越权漏洞案例

以下是几个案例：

1. 通过修改GET传参进行越权：<http://cn-sec.com/archives/2572.html>
2. 通过修改POST传参进行越权：<http://cn-sec.com/archives/1682.html>
3. 修改cookie传参进行越权：<http://cn-sec.com/archives/6421.html>
4. 业务逻辑绕过：<http://cn-sec.com/archives/17524.html>



PART 3

越权演示即分析



越权演示

Pikachu靶场项目: <https://github.com/zhuifengshaonianhanlu/pikachu>

1. 水平越权（平行越权）：A用户和B用户属于同一级别用户，但各自不能操作对方个人信息，A用户如果越权操作B用户的个人信息的情况称为平行越权操作。
2. 垂直越权：权限较低的用户去执行高权限用户的操作。

以下是几张分析附图

```
File Edit Selection View Go ... op2_admin.php - overpermission - Visual Studio Code [Administrator]
op1_mem2.php op2_admin.php x
op2 > op2_admin.php
5 /*
6
7
8 $SELF_PAGE = substr($_SERVER['PHP_SELF'],strrpos($_SERVER['PHP_SELF'],'/')+1);
9
10 if ($SELF_PAGE == "op2_admin.php"){
11     $ACTIVE = array('','','','','','','','','','','','','','','','','','','','','','','','','','');
12 }
13 $PIKA_ROOT_DIR = "../..../";
14 include_once $PIKA_ROOT_DIR . 'header.php';
15 include_once $PIKA_ROOT_DIR . 'inc/mysql.inc.php';
16 include_once $PIKA_ROOT_DIR . 'inc/function.php';
17 include_once $PIKA_ROOT_DIR . 'inc/config.inc.php';
18
19
20 $link=connect();
21 // 判断是否登录，没有登录不能访问
22 //如果没登录 或者level不等于1，都干掉
23 if(!check_op2_login($link) || $_SESSION['op2']['level']=1){
24     header("location:op2_login.php");
25     exit();
26 }
27
28 //删除
29 if(isset($_GET['id'])){
30     $id=escape($link,$_GET['id']); //转义
31     $query="delete from member where id={$id}";
32     execute($link,$query);
33 }
34
```

The screenshot shows a Visual Studio Code editor window titled "op2_admin_edit.php - overpermission - Visual Studio Code [Administrator]". The active tab is "op2_admin_edit.php". The code is as follows:

```
1 <?php
2 /**
3  * Created by runner.han
4  * There is nothing new under the sun
5  */
6
7
8 $SELF_PAGE = substr($_SERVER['PHP_SELF'],strrpos($_SERVER['PHP_SELF'],'/')+1);
9
10 if ($SELF_PAGE == "op2_admin_edit.php"){
11     $_ACTIVE = array('','','','','','','','','','','','','','','','','','');
12 }
13 $PIKA_ROOT_DIR = "../..../";
14 include_once $PIKA_ROOT_DIR . 'header.php';
15 include_once $PIKA_ROOT_DIR.'inc/mysql.inc.php';
16 include_once $PIKA_ROOT_DIR.'inc/function.php';
17 include_once $PIKA_ROOT_DIR.'inc/config.inc.php';
18
19 $link=connect();
20 // 判断是否登录，没有登录不能访问
21 //这里只是验证了登录状态，并没有验证级别，所以存在越权问题。
22 if(!check_op2_login($link)){
23     header("location:op2_login.php");
24     exit();
25 }
26 if(isset($_POST['submit'])){
27     if($_POST['username']==null && $_POST['password']!=null){//用户名密码必填
28         $getdata=escape($link, $_POST);//转义
29         $query="insert into member(username,pw,sex,phonenum,email,address) values('{$_GETdata['username']}'"
30         $result=execute($link,$query);
```

A red arrow points from the Chinese text annotation to the `!check_op2_login($link)` condition in line 22.

只验证了用户是否登录，如果没登录就跳转到登录页面，没有验证用户权限等级，但前端显示添加用户是权限等级为1的用户才能执行的操作，因此这里存在垂直越权漏洞



PART 4

如何防范越权漏洞

防范越权漏洞方式

1. 前后端同时对用户输入信息进行校验，双重验证机制
2. 执行关键操作前必须验证用户身份，验证用户是否具备操作数据的权限
3. 特别敏感操作可以让用户再次输入密码或其他验证信息。
4. 可以从用户的加密认证cookie中获取当前用户id，防止攻击者对其修改。或在session、cookie中加入不可预测、不可猜解的用户信息。
5. 直接对象引用的加密资源ID，防止攻击者枚举ID，敏感数据特殊化处理
6. 永远不要相信来自用户的输入，对于可控参数进行严格的检查与过滤



Thanks