看雪 2017 安全开发者峰会

Kanxue 2017 Security Developer Summit

2000-2017

浅析WEB安全编程

汤青松@中国婚博会

常见漏洞有哪些？

代码注入　CSRF　0元支付

短信轰炸　密码找回　本地限制

验证码爆破　SQL注入　XSS跨站

文件上传　文件包含　命令执行

目录大纲

一、SQL注入

二、XSS跨站

三、请求伪造

四、越权漏洞

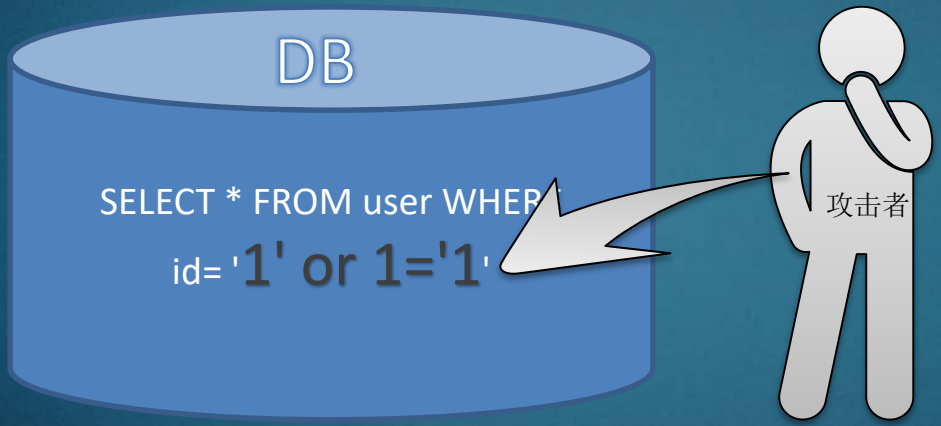
五、支付漏洞

SQL注入

SQL注入

- 漏洞成因
- 攻击方式
- 防御方法

基本原理



1. 使用用户的参数拼接SQL语句

2. 参数改变了原SQL语句的结构

攻击方式

回显注入

报错注入

盲注

回显注入

正常请求返回的数据

URL中加入一些SQL语句后的返回结果

利用注入漏洞可以改变页面返回数据，则称之为回显注入

报错注入

在URL加入了一些错误的SQL语句，被执行后返回了异常信息，这些异常信息当中包含了敏感信息,我们可以通过屏蔽数据库连接错误来防范此问题

盲注

布尔盲注
通过条件是否成立来判断
substr截取第一个字符判断是否大于'a'，成立则页面返回数据

时间盲注
通过返回时间的长短判断
获取第一个字符的ascii码，判断是否大于115，不成立延时5秒返回

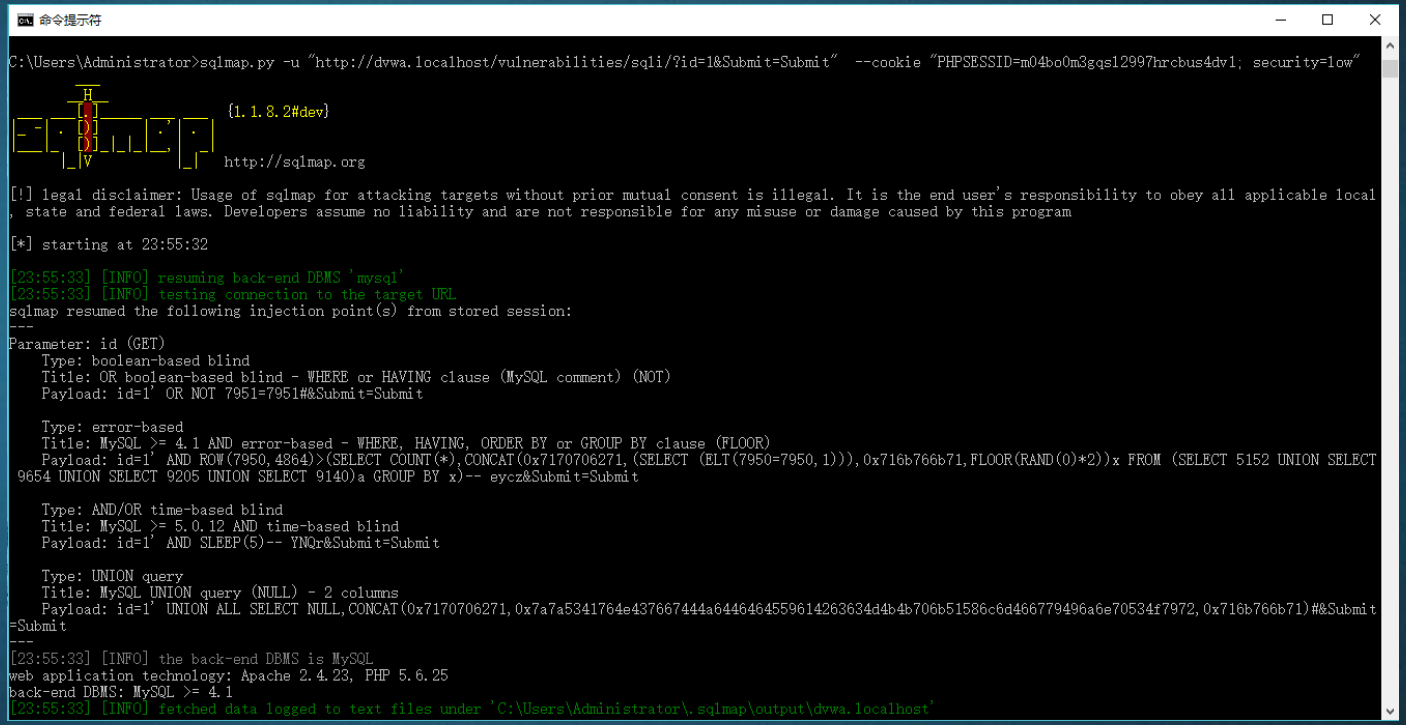
http://127.0.0.1/sqli-labs/Less-1/?id=1 //正常URL

' and (select substr(email_id,1,1) from emails where id=3) > 'a' //布尔

' If(ascii(substr(database(),1,1))>115,0,sleep(5))%23 //时间

检测方法

sqlmap.py –u “http://www.xxxx.com/index.php?id=1”



如何防范?

1. 拦截带有SQL语法的参数的传入

```php
//整型参数过滤处理
$storeId = intval($_GET['store_id']);
//字符串参数过滤处理
$statuArr = ['on','off','del'];
$status = in_array($_GET['status'],$statuArr) ? $_GET['status'] : '';
```

2. 通过预编译处理拼接参数的SQL语句

```php
//不可预测参数处理
$stmt = $pdo->prepare("SELCT * FROM user WHERE name = ?");
$stmt->bindValue(1, "daxia");
$stmt->execute();
```

3. 定期分析数据库执行日志，是否有异常SQL执行

XSS跨站

漏洞成因

- 漏洞成因
- 攻击场景
- 防御方法

产生原因？

http://www.daxia.xxx/search?keyword=test <script>alert(123)</script>

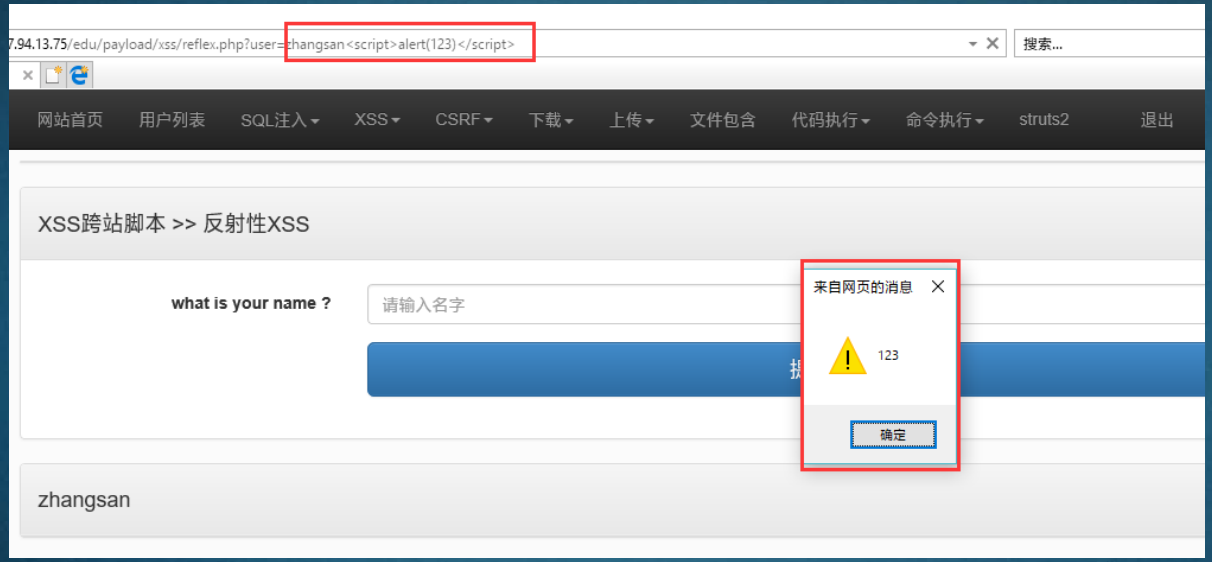
```
<html>
        <head>
            <title>搜索结果页</title>
        </head>
        <body>
        ..........
        <dvi>搜索(test</div><script>alert(123)</script>)结果如下</div>
        ..........
        </body>
</html>
```

漏洞成因

1. 参数输入未经过安全过滤
2. 恶意脚本到被输出到网页
3. 用户的浏览器执行恶意脚本

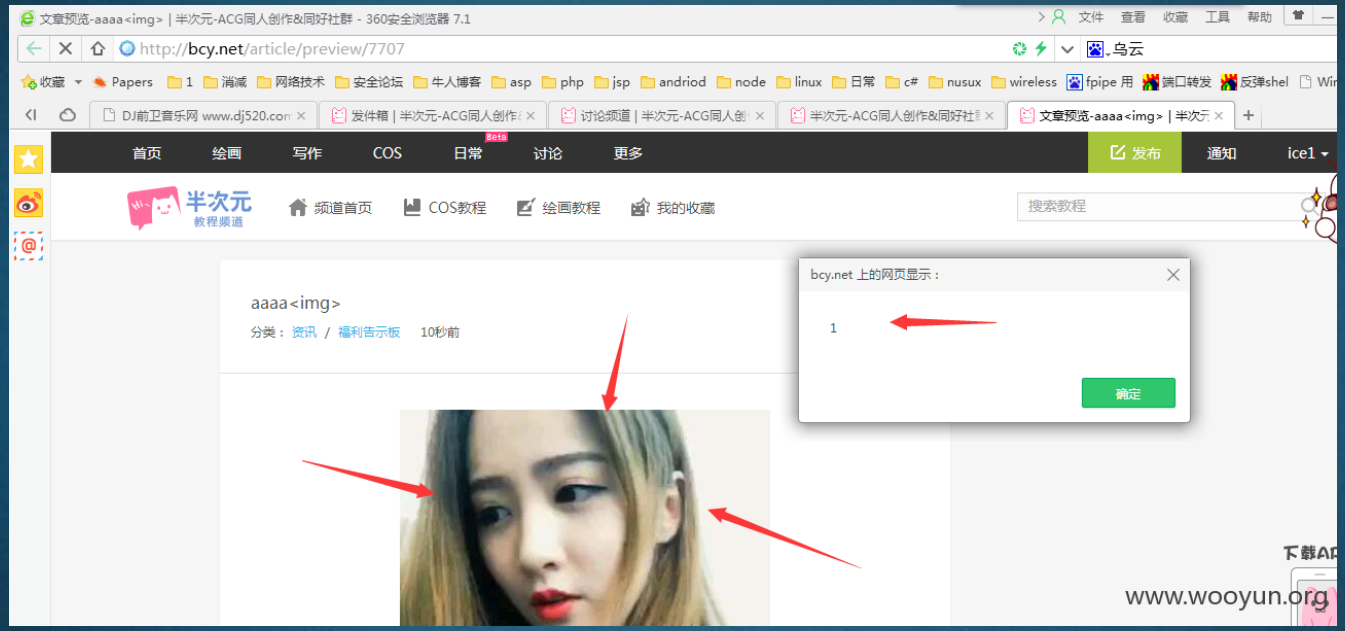
漏洞分类

反射型



URL中的alert代码被放到页面中且被执行
弹框中已经获取到cookie信息c

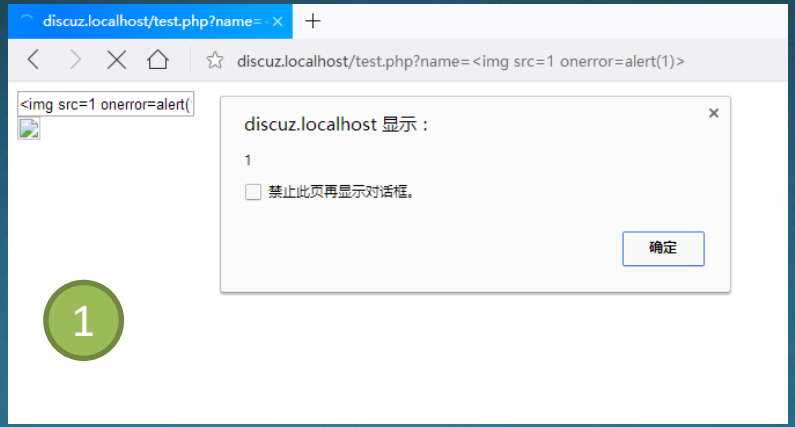
存储型



在自己的个人资料填入JavaScript代码

其他人正常访问页面，代码被执行

DOM型



参数name虽然经过编码后放入到模板中

但是经过dom操作后，参数再次被转为实体字符

```php
<?php
error_reporting(0);
$name = htmlspecialchars($_GET["name"]);
?>
<input id="username" type="text" value="<?php echo $name;?>" />
<div id="content"></div>

<script type="text/javascript">
// 获取输入的名称，并且输出在content内。导致了一个dom-xss。
var username = document.getElementById("username");
var content = document.getElementById("content");
content.innerHTML = username.value;
</script>
```

2

如何防范?

1. 标签黑白名单过滤

```
//过滤HTML标签
$desc = removeXss($_POST['desc']);
```

2. 代码实体转义

```
//把HTML实体标签转为符号，ENT_QUOTES指单引号也需要转义
$desc = htmlspecialchars($_GET['desc'],ENT_QUOTES)
```

3. httponly 防止cookie被盗取

```
//设置HttpOnly
setcookie("user", "daxia", NULL, NULL, NULL, NULL, TRUE);
```

越权漏洞

产生原因？

1. 业务系统存在用户权限验证

2. 对用户的权限验证不严谨

3. 用户能操作不属于自己权限的操作

漏洞分类

平行越权

商城系统中一般会有订单详情

在订单详情URL中，不断的切换订单ID，可以获取到他人订单信息。

可见后端查询数据的条件是并没有加上当前用户的UID。

平行越权防御方法

```php
//普通用户查询订单，必须有UID
$orderId = intval($_GET['orderId']);
//UID 必须从服务器取，而不是用户所传入
$uid = $_SESSION['uid'];
$sql = "SELECT * FROM order WHERE uid=$uid AND order_id = $orderId";
```

越权不仅限于查询数据的时候
在修改数据的时候同样适用

垂直越权

普通用户进入了管理后台

屏蔽自增长主键

```php
//尽量不要暴露连续的订单号给用户;
$orderId = 100;
//如果底层改造麻烦，我们可以通过订单ID对称加密的方法
$orderId = encode($orderId);
echo $orderId;

//接收参数做对称解密处理
$orderId = decode($orderId);
```

不显示自增长订单号不仅仅在于防范被黑客攻击
有的时候还可以防止友商知道一天订单量

越权防范建议

1. 前台和后台的查询尽量不用同一个查询接口
2. 尽量不要暴露出连续ID如订单号
3. 越权不仅限于展示，修改数据也会出现

CSRF跨站请求伪造

产生原因？

1. 服务端错把“浏览器发起的请求”当“用户发起的请求”

2. 已登录的浏览器，打开恶意网址后，被执行了相应操作

POST型CSRF

```
<html>
  <body>
    <form action="http://i.emao.com/homecp/user/dosetting" method="POST">
      <input type="hidden" name="nickname" value="eeeeeeeeq" />
      <input type="hidden" name="sex" value="1" />
      <input type="hidden" name="year" value="" />
      <input type="hidden" name="month" value="" />
      <input type="hidden" name="day" value="" />
      <input type="hidden" name="provinceid" value="" />
      <input type="hidden" name="cityid" value="" />
      <input type="hidden" name="areaid" value="" />
      <input type="hidden" name="major" value="" />

          <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

后端没有限定数据的来源，导致用户在不知情的情况下修改了用户信息

检测方法?

1. 去掉token参数尝试能否正常请求

2. 去掉referer是否可以提交成功

3. 是否能用GET提交替代POST提交

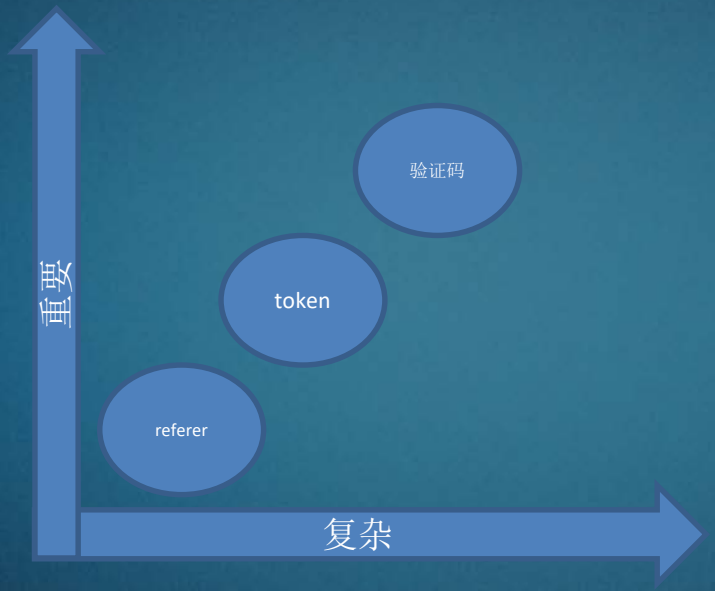
编码方法

```php
//验证Referer
if ($_SERVICE['HTTP_REFERER'] != $host) {
    return false;
}
//验证token
if ($_POST['token'] != $token) {
    return false;
}
//图片验证码
if ($_POST['verify_code'] != $verify_code) {
    return false;
}
```

REFERER 信息不符合
TOKEN 验证不通过
图片验证码验证失败

则不接受请求

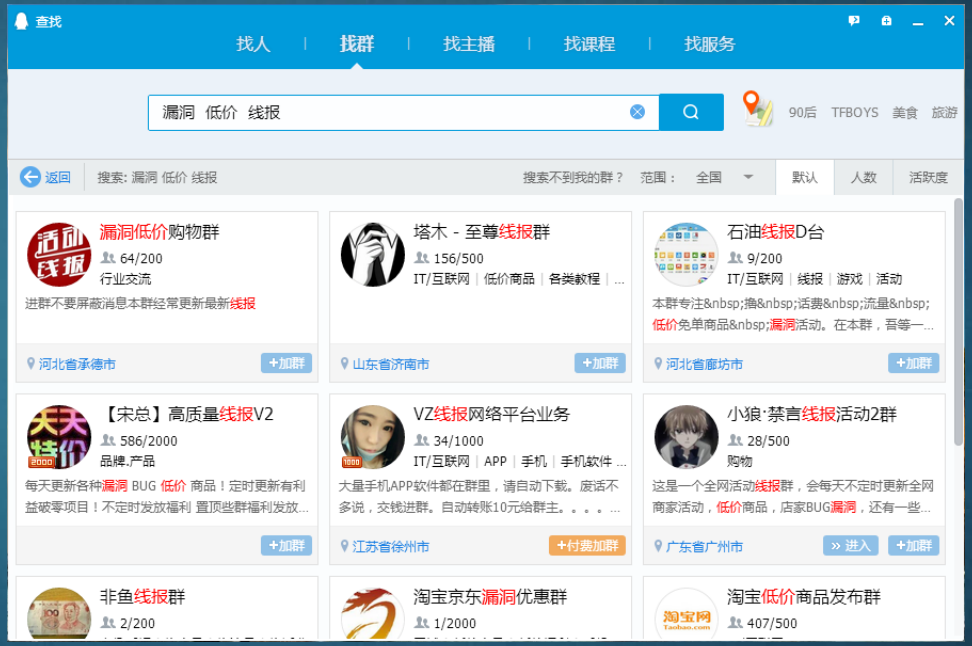
防范等级



业务类型越重要的位置
使用相对复杂的验证机制

支付漏洞

QQ群黑产线报群


查找
找人
找群
找主播
找课程
找服务
漏洞 低价 线报
90后 TFBOYS 美食 旅游
返回
搜索: 漏洞 低价 线报
搜索不到我的群？
范围： 全国
默认
人数
活跃度
漏洞低价购物群
64/200
行业交流
进群不要屏蔽消息本群经常更新最新线报
河北省承德市
+加群
塔木 - 至尊线报群
156/500
IT/互联网 | 低价商品 | 各类教程 | ...
山东省济南市
+加群
石油线报D台
9/200
IT/互联网 | 线报 | 游戏 | 活动
本群专注 撸 话费 流量
低价免单商品 漏洞活动。在本群，吾等一...
河北省廊坊市
+加群
【宋总】高质量线报V2
586/2000
品牌.产品
每天更新各种漏洞 BUG 低价 商品！定时更新有利益破零项目！不定时发放福利 置顶些群福利发放...
+加群
VZ线报网络平台业务
34/1000
IT/互联网 | APP | 手机 | 手机软件 ...
大量手机APP软件都在群里，请自动下载。废话不多说，交钱进群。自动转账10元给群主。。。。...
江苏省徐州市
+付费加群
小狼·禁言线报活动2群
28/500
购物
这是一个全网活动线报群，会每天不定时更新全网商家活动，低价商品，店家BUG漏洞，还有一些...
广东省广州市
» 进入
+加群
非鱼线报群
2/200
淘宝京东漏洞优惠群
1/2000
淘宝低价商品发布群
407/500

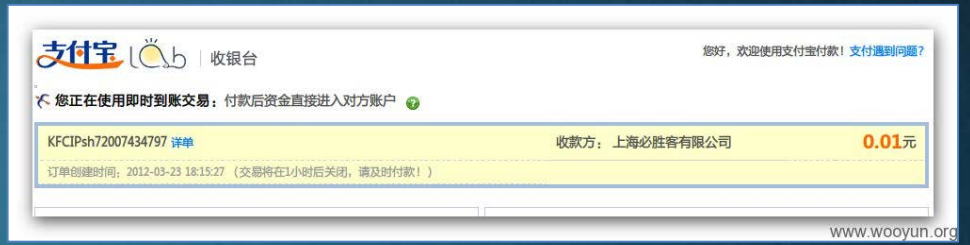
产生原因？

1. 开发者在数据包中传递支付的金额

2. 后端并没有对金额做校验或者签名

3. 导致攻击者可以随意篡改金额提交

修改支付金额



后端在确认订单信息的时候，完全依靠前端传过来的信息，导致金额可以被用户自己所修改

修改数量

商品数量可以为负数，导致最终的支付金额减少

数据包重放问题

在充值过程中，把数据包多次提交，发现多了一笔订单，并且增加了余额

如何防范？

1. 限制超量购买限量商品

2. 限制低价、免费购买付费商品

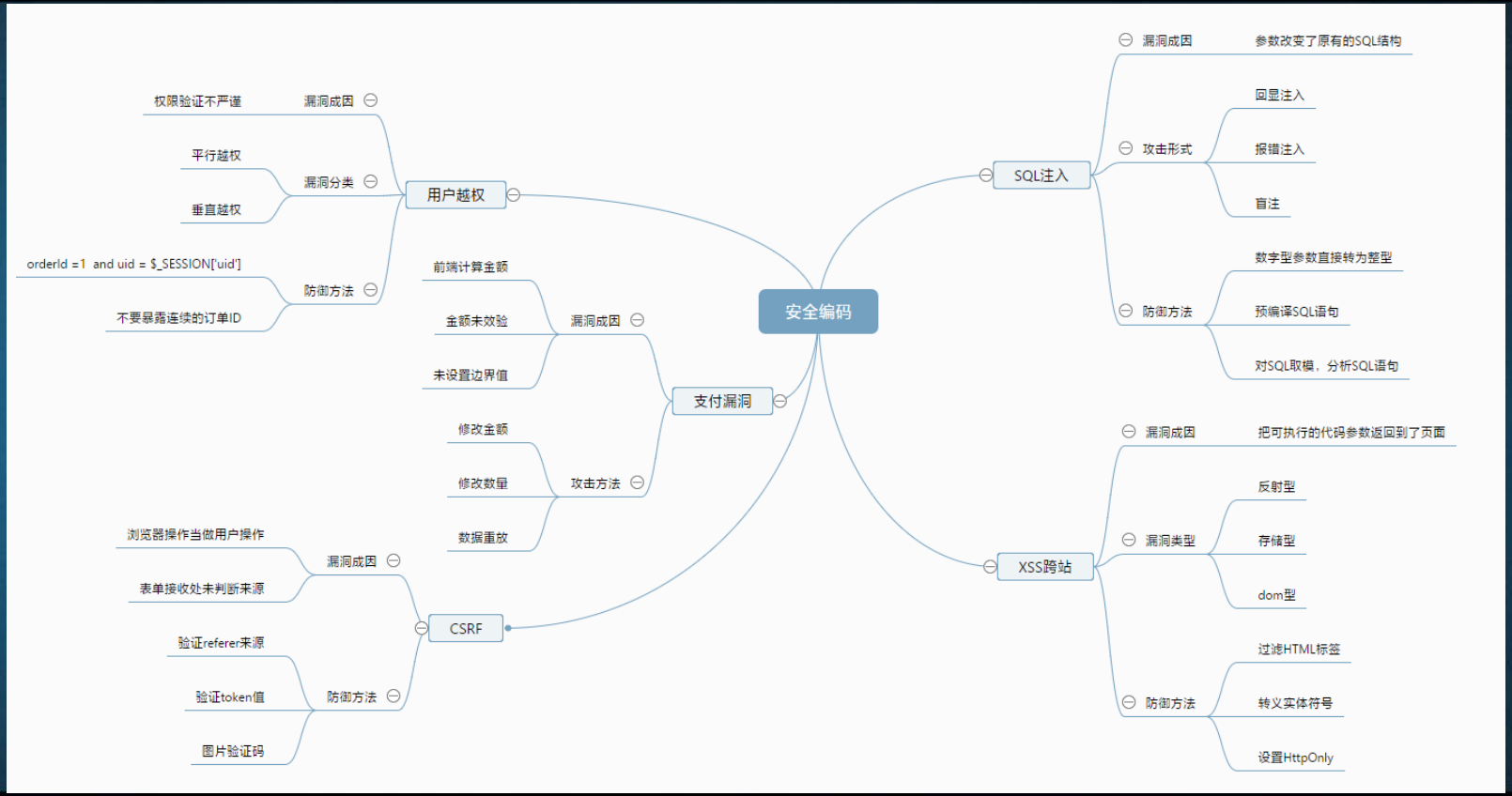
3. 限制免费商品获得金钱、积分等利益

如何防范?

```php
//验证参数的有效性
if ($num > $maxNum || $num < $minNum) {
    return false;
}
//由后端计算金额
$totalMoney = $price * $num;
//判断支付金额是否异常
if ($totalMoney <= 0) {
    return false;
}
//生成订单签名信息
$sigin = createSigin($totalMoney, $price, $otherParam);
```

参数合法验证
后端计算金额
金额是否异常
对订单签名

回顾总结



WeChat: songboy8888

Email:soupqingsong@foxmail.com