中国婚博会PHP高级工程师、安全顾问汤青松：浅析Web安全编程

原创

csdn业界要闻 于 2017-12-01 15:35:46 发布 597 收藏
文章标签： Web安全 汤青松 看雪安全开发者峰会

本文链接：https://blog.csdn.net/csdn_bang/article/details/80133051

11月18号，2017看雪安全开发者峰会在北京悠唐皇冠假日酒店举行。来自全国各地的开发人员、网络安全爱好者及相应领域顶尖专家，在2017看雪安全开发者峰会汇聚一堂，只为这场“安全与开发”的技术盛宴。

曾问过一位搞 Web 安全的人为什么 PHP 是世界上最好的语言，他的回答是 PHP 网站漏洞多，有饭吃。结合目前黑灰产业中借助 Web 漏洞进行各种薅羊毛的现状，不禁触发了我们的深切反思。问题究竟出在什么地方，为什么网站会存在 SQL 注入、XSS 跨站、CSRF 这些漏洞，我们应该如何避免在代码中产生这些错误？中国婚博会PHP 高级工程师、安全顾问汤青松为我们带来了《浅析 Web 安全编程》主题演讲。作为开发人员你会发现其中有许多点值得我们学习借鉴，安全问题必须引起每一位开发者的重视。

中国婚博会PHP高级工程师 汤青松

> 汤青松，中国婚博会PHP高级工程师、安全顾问。擅长安全测试工具的研发，Web渗透测试，在处理Web漏洞中积累有大量经验，2014年任职于乌云，负责众测开发工作；2015年在网利宝担任研发以及安全建设。目前正专注于PHP的编码安全，PHP安全编码方面的书籍正在撰写中。常常活跃在SegmentFault讲堂以及GitChat，多次分享了Web安全方面的议题。曾在2017 PHP开发者大会上发表演讲《PHP安全开发:从白帽角度做安全》。

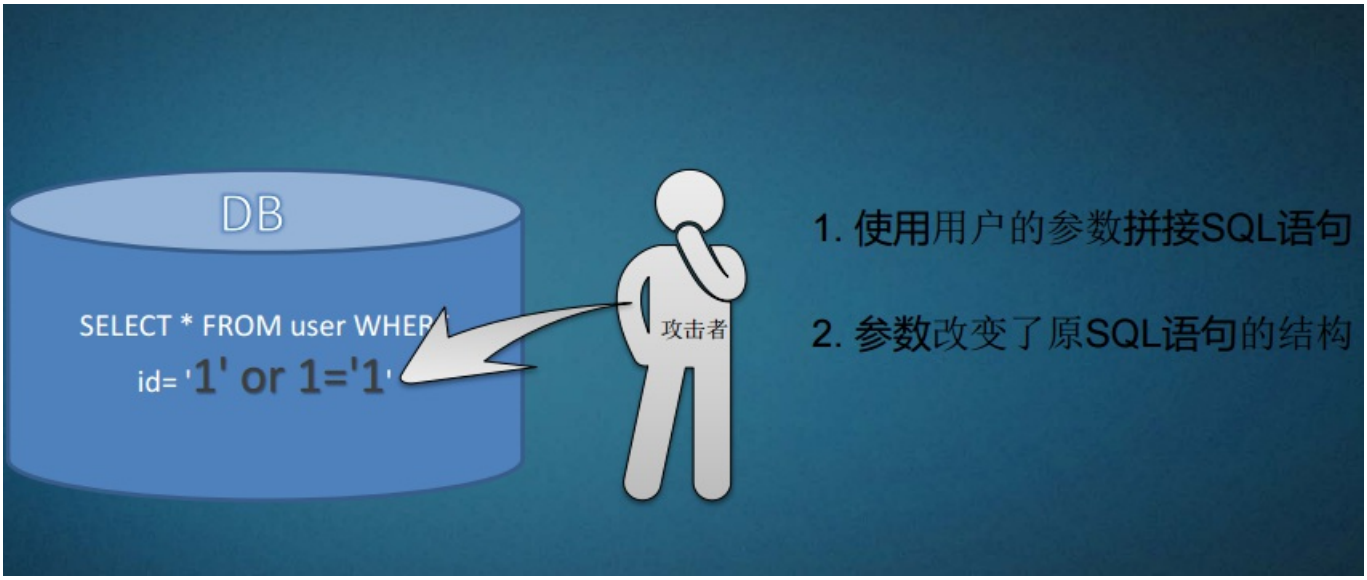
以下为演讲速记：

汤青松：非常高兴跟大家交流这个看雪编码的问题，我是汤青松。说到安全开发的话题，其实我一直是开发者，开发我是专业，安全我是业余，但是安全开发我还是专业的。在场的各位，对于关键词都有清晰的了解，我列举了一些安全的关键词，我们先来看一下。常见代码注入、CSRF、O元支付、短信轰炸这些都是非常常见的漏洞，今天由于时间关系我讲不了那么多，我列举其中几个点跟大家分享一下，希望大家听了以后可以举一反三,在大脑中形成自己的安全意识，从而提高这个安全编码的能力。

常见漏洞有哪些？

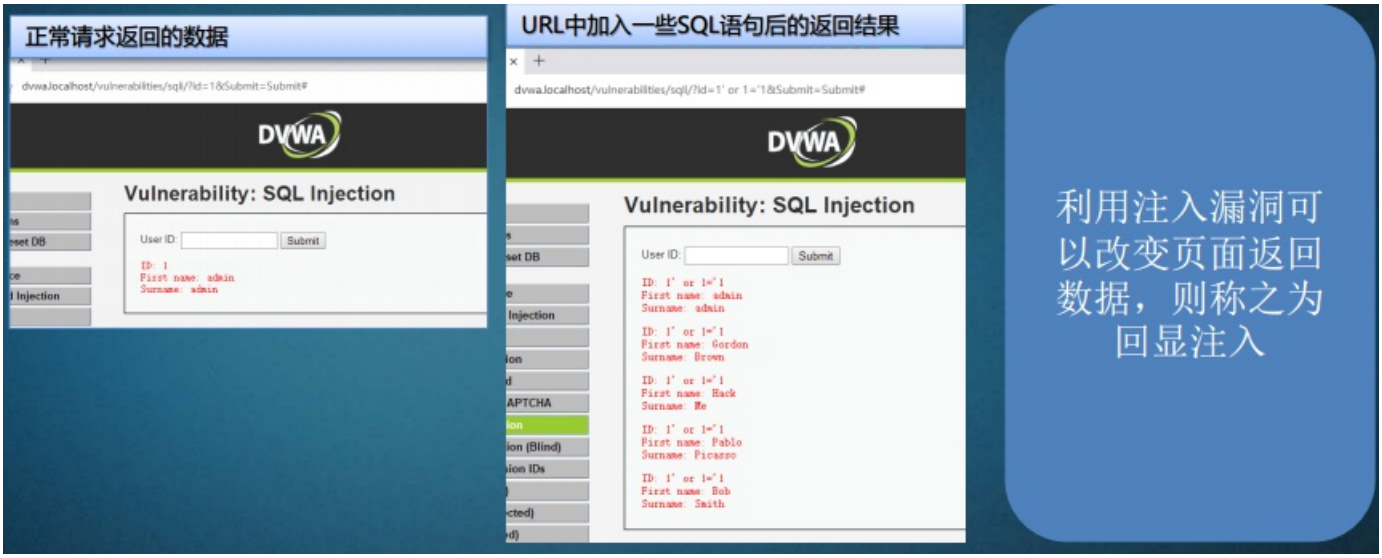
今天要分享的是五个点：SQL注入、XSS跨占，请求伪造，越权漏洞以及支付漏洞。

SQL注入

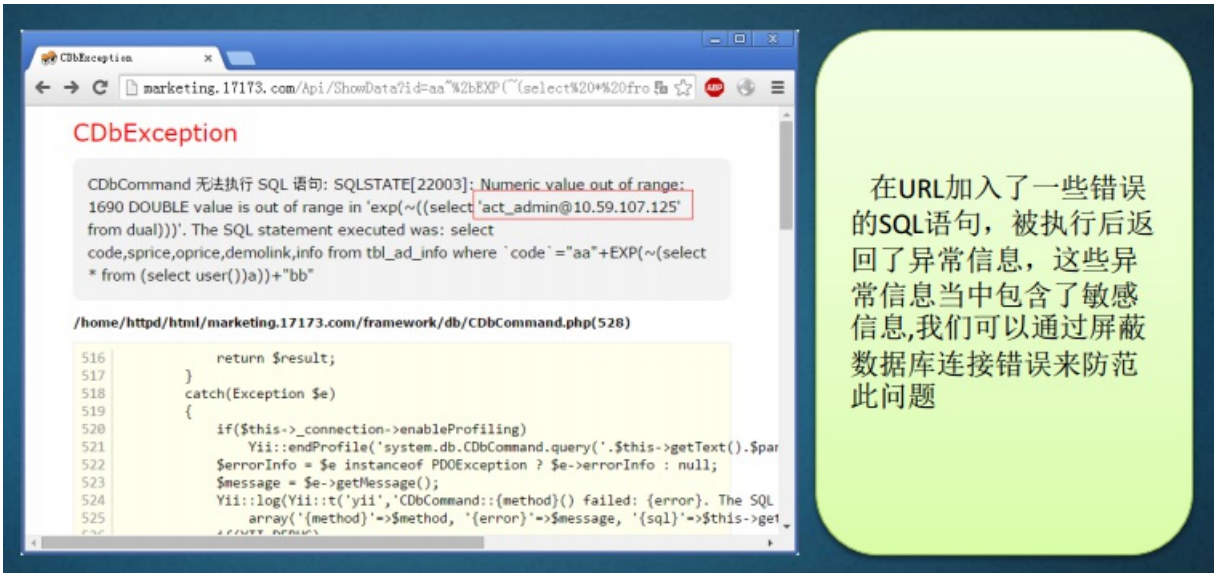


我们看一下SQL注入，首先是漏洞成因，攻击方式以及防御方案。漏洞成因我们可以这两句话，使用用户参数拼接这个SQL语句，这个参数改变了原有的SQL结构，改变了这个SQL的注入。我们看下面一张PPT，左边这是一个数据库，白色部分的字体是我们在代码中写到的SQL结构，黑色部分就是攻击者可能会传入的参数。当我们把这个SQL结构拼接出来之后形成了一个新的结构，这个结构被执行之后把整张表所有的数据传输出来，数据库比较大的访问更多请求，整个可能就挂了，还造成一些数据泄漏的情况，这些就是SQL的注入成因，参数改变了原有的SQL结构。

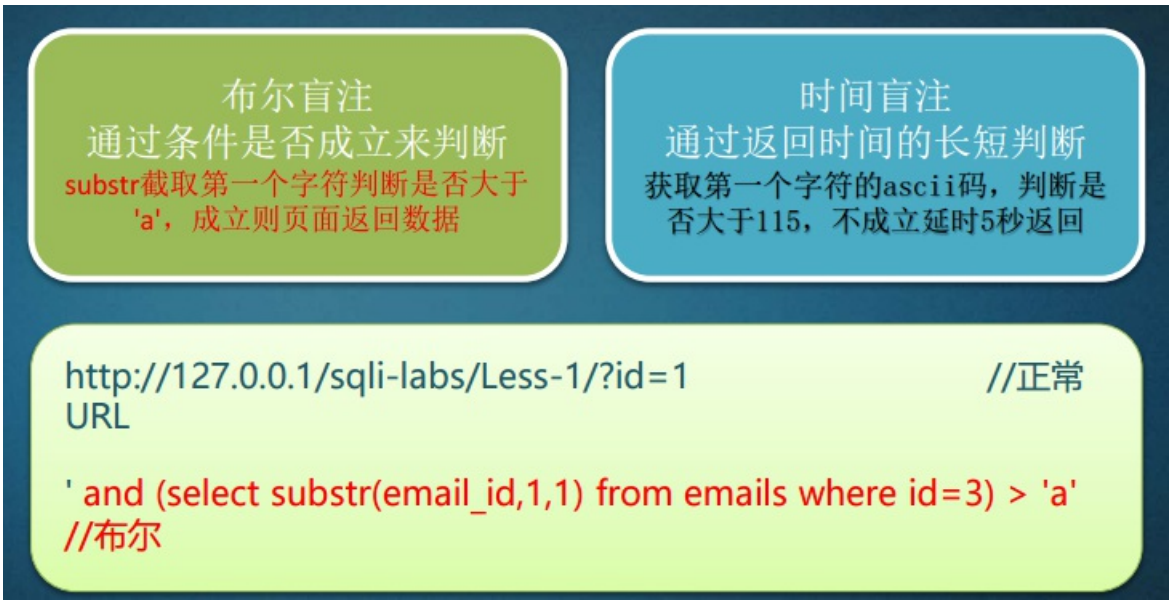
攻击者通常有哪几种攻击方式？我把它分为了三种类型：一种是回显注入，一种是报错注入，一种是盲注。



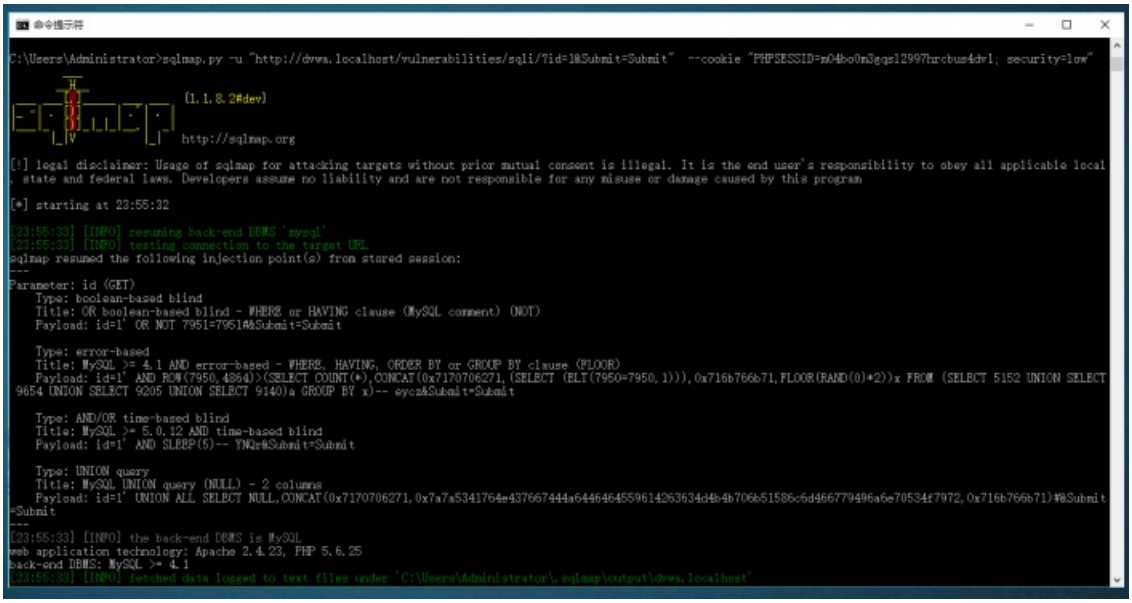
这里面有两张图，第一张图是传入ID是正常的正型数字，返回的结果是用户的一个信息传入ID等于1，上面把这个参数修改了一下，等于1，然后加了 or 1=‘1，当它拼接到之后，跟前面一样把整个表的数据传输出来，这边看到整个用户表的数据都被列举出来了。利用漏洞可以改变这个页面的数据我们叫做回显注入，这个黑客可以直接把这个数据下载下来。



报错注入，这张图非常清楚可以看到URL上面这个部分是正常URL加上攻击者利用的供给代码，其实这上面的供给代码也是执行不了，放到数据库当中，最后会造成数据库变为异常码，然后把异常码抛出来了，把这个用户名展示出来了，这是非常敏感的信息，我们写代码的时候需要把这个数据库抛出来的错误屏蔽掉，不要可以让前台显示出来，通过报错显示了一些敏感信息，我们称之为报错注入。



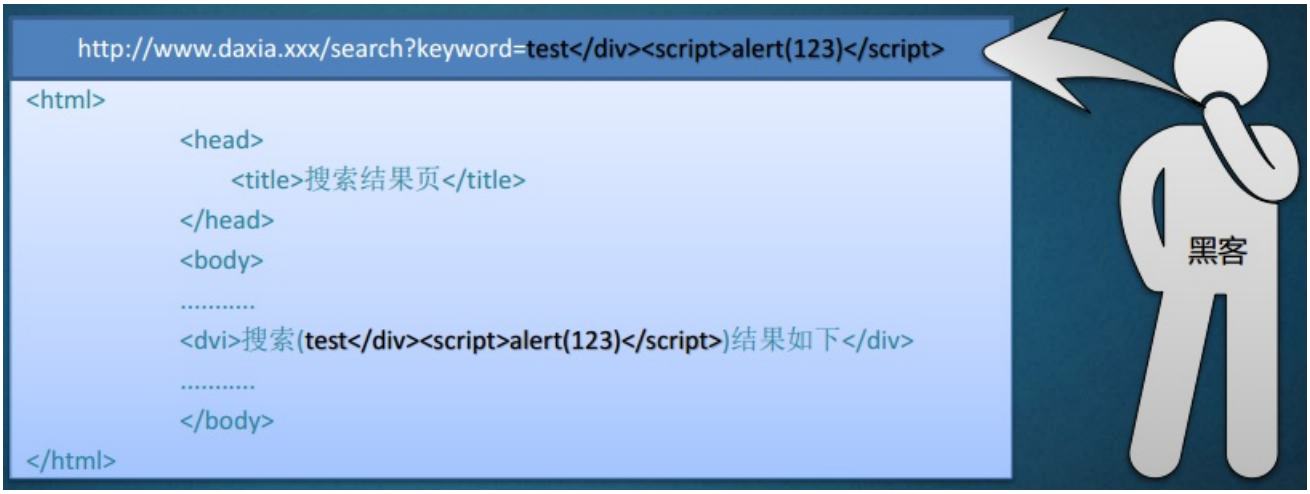
盲注，连接起来稍微可能复杂一点，它和回显注入以及报错注入不一样，我们没有办法通过这个页面数据看到它的区别，可以通过两种方式，比如说步尔盲注和时间盲注，下面的部分是正常URL，红色部分是布尔注入的表示式，前面加一个and截取一个字符，判断一下这个第一个字符是不是大于这个字母A，如果当它成立整个条件都是成立，这个时候页面是有反馈数据的。如果不成立这个页面返回不了数据，这就是布尔数据，我们可以看到有数据和没有数据的情况，当字母A不断变换的时候，也可以把这个数据库里面的数据猜测出来。时间盲注，下面蓝色区域部分，我们知道数据库里面可以用一些IF函数，也是截取第一个字符，如果这个不成立就到五秒钟返回，通过这个页面返回的时间可以判断这个地方是不是有注入的，也可以把这个数据都给下载下来。



刚刚说到攻击者碰到三种攻击方式，下面看一下怎么样检测页面当中有没有什么注入？现在有非常多的工具，如果对注入了解不是太深，也是有快速检测的方法。我们可以看到这是一个CMD窗口，上面是我写到的检测表达式，Splmap.py以及我们需要检测的UI，需要有这个注册点它会告诉你有哪些注入，比如说这个页面是我在本地测试的结果，它就告诉了有回显注入、错误注入以及一些盲注。

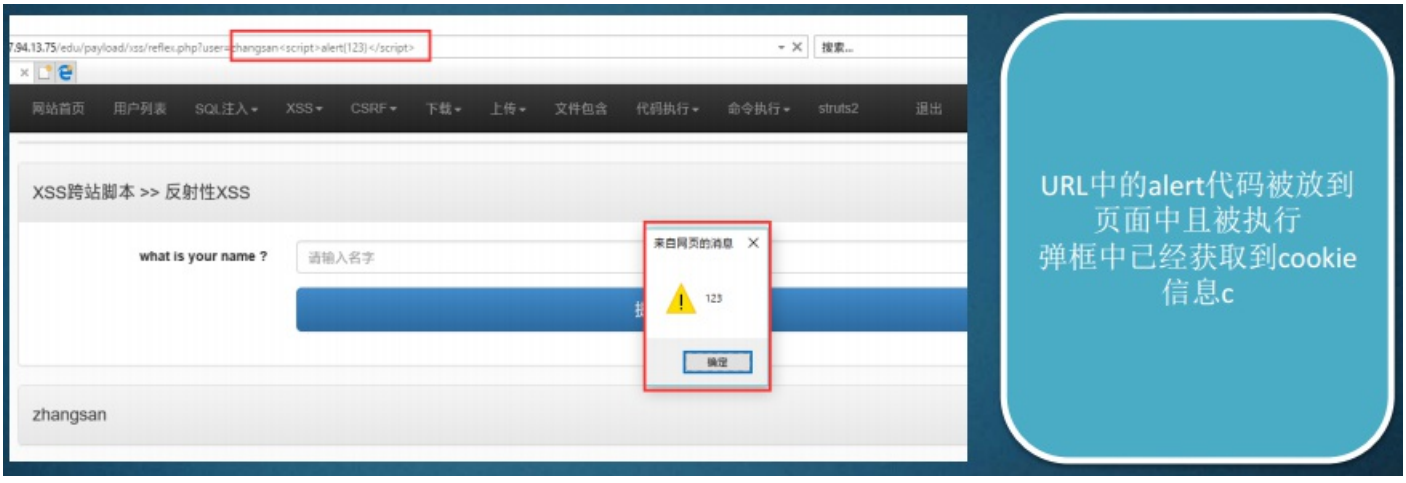
开发者最关心的就是怎么样防范服务器的安全？编码的过程中有三点建议，值得借鉴。1，参数会改变SQL的结构，我们会让这个参数当中不但有这个SQL的结构，当我知道这个参数是整型的时候，我们就把这个参数转型为整型，整型肯定不包括这个SQL的结构，无法改变结构的目的，哪就不存在着SQL注入。有的时候我们无法预测它传什么参数，比如说我们去论坛回复一个帖子，我们肯定没有办法控制的，这个时候我们可以用PDO预处理，这也是最常见的方法，也是一个最好的方法。但是有的时候我们会写一些复杂社会语句，我们会会用第一种方法，我们适前定义好这个SQL的语句结构，再把我们的参数放进去，这个时候是无法达到更改SQL语句处理的目的。当这个业务比较大的时候，这个日志是非常多的，可以找一些SQL的取模软件，可以把这个取模一下，取模之后并不太多的，如果直接看的话是海量的日志，是没法看的。

XSS跨站

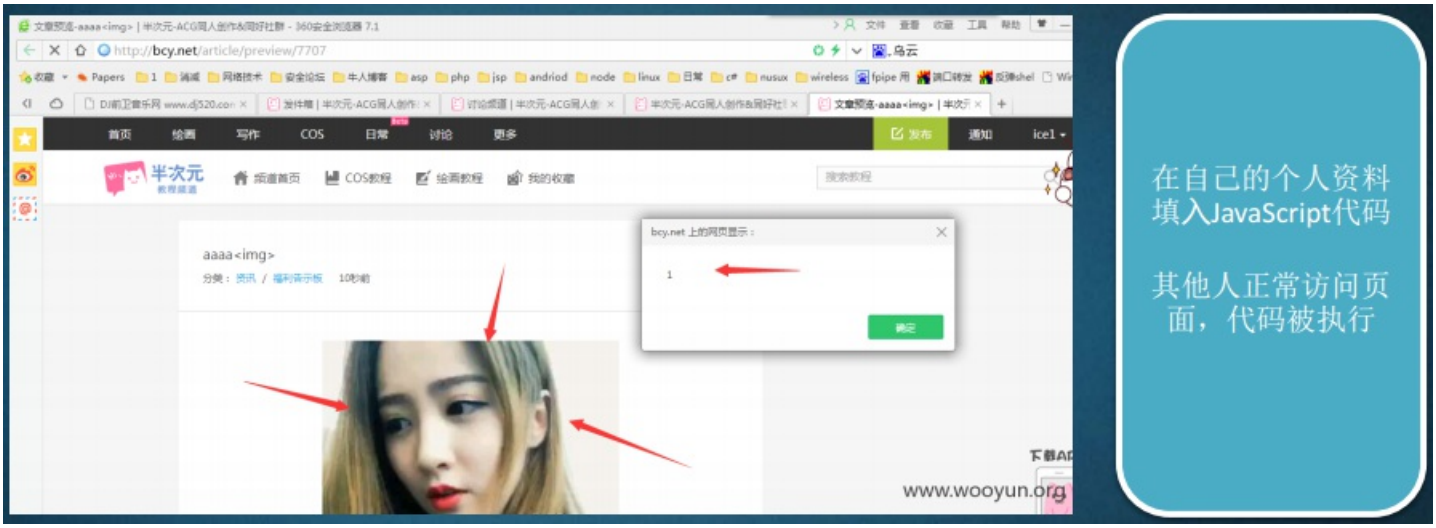


看一下漏洞成因、攻击场景以及防御方法。画了一张图，上面有一个URL，下面是一个页面返回的HTML代码，我们可以看到白色部分HTML是我们事先定义好，黑色部分参数是想用户想搜索的关键词，当我们搜索了test+Div最后等于123，对后反馈页面一般搜索会告诉你用户搜索了什么关键词，结果如何等等。这个地方如果没有做好转移，可能会造成XSS跨3（音），我们可以看到蓝色部分是我们事先定义好的结构，被攻击者利用之后它先把这个DIV结束了，最后加上一个script标签，它也有可能不跟体谈标签，直接发送到它的服务器上。参数未经过安全过滤，然后恶意角本被放到网页当中被执行了，用户浏览的时候执行了这个脚本。

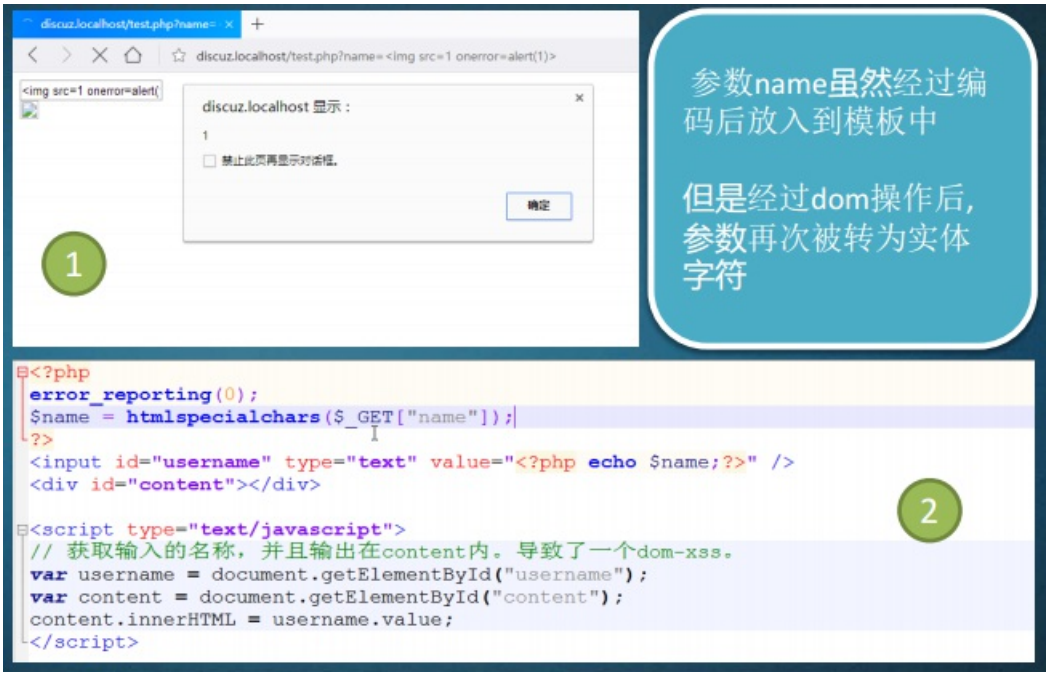
XSS也分好几种类型，比如说这里有三种类型，反射型、存储型以及DOM型。



反射型：这个截图当中是专门训练一些WEB漏洞的练习页面，我们可以输入自己的名字，输入之后会把我们的名字在这里显示了出来，我们输入了一个张三，这个时候弹出来了一个123，在那边显示了一个张三,但是script标签没有出来，因为这个标签被执行了。



接着来说说存储型的漏洞，存储型的XSS，我们可以看一下这个URL上面并没有代码，但是依然弹出了一个“1”其实也是中了XSS代码，这是怎么样做到的？我看了漏洞的帖子。它是发现个人资料页的时候有一个XSS漏洞，它就在个性签名的位置填入了一个XSS标签，弹出了一个1，把这个地址发给了别人，别人看到这个地址并没有什么代码以为这个页面是安全，结果一打开就插入了这个XSS代码。存储型的XSS的攻击危害比较大，因为它的页面当中是看不到这个Script的代码，别人防不胜防。只要管理员没有发现，下一个用户或者下一个用户一直接发它的，反射型需要用户主动点击的。



还有一种是Dom型的XSS，一般是一些有安全意识的开发者弄出来。比如说接受参数会做一些过滤，把一些字符转义一下，但是转义之后依然会存在着XSS的情况，比如说这个图上我们上面输入的可以看到这行代码夺了规律，把这个大括号以及小括号以及双页号进行转移，按理说转移之后它应该不会再作为它的标签存在，不会存在XSS的代码。我们可以看到下面在Script通过ID获得的这个值，复制到了这个DIV上，经过DOM操作之后，之前转义的字符就变为的它的标签，所以经过DOM的操作XSS我们称之为DOMXSS，有可能通过URL传播，也有可能通过服务器的传播。

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

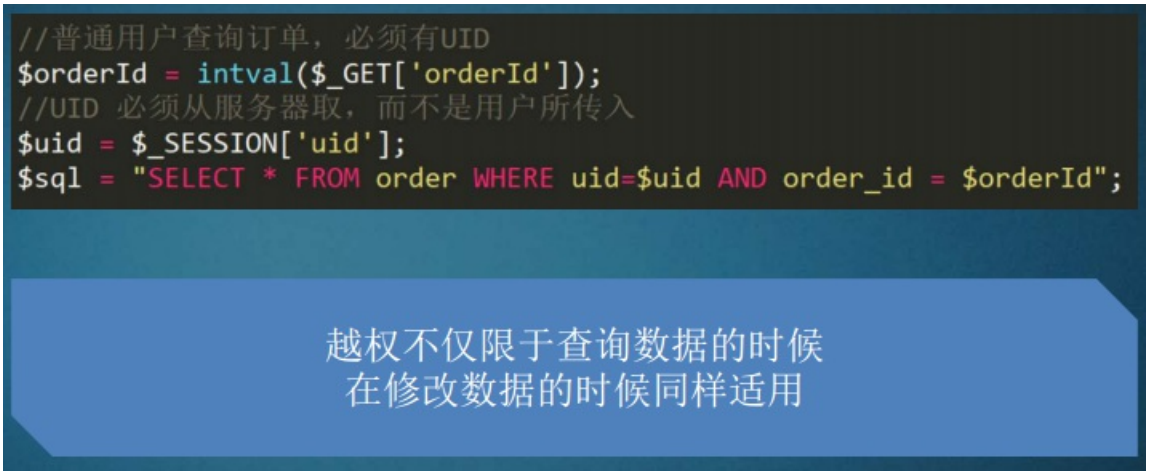
```
//设置HttpOnly
setcookie("user", "daxia", NULL, NULL, NULL, NULL, TRUE);
```

通过XSS有一些建议：有一些时候根本就不需要考虑到它是不是HTML有标签，我们有的时候根本用不到HTML标签，只保留文字部分这是一劳永逸的，但是有一些我们还是需要展示这个标签，比如说程序论坛当中我要贴一个代码不能把这个代码部分干掉，这个时候我们需要用一些转移，它会把这个大括号、小括号以及双引号都可以做一个转意，做为一个字符，就无法执行这个标签型，后面加一个参数，有时候单引号也会造成一席XSS。一个信号当中有那么多的地方存在着这个输入以及检测的地方，可能就有一些地方漏掉了，只要有一个地方漏掉了，用户的cookie信息就被盗取了。服务器发生用户信息的时候我们需要加上一个httponly之后，这个代码无法读取到cookie的信息，那么攻击者也是得不到这个信息的，对于用户来说也是非常好的保护。比如说张三在我们网站上登陆了一下用户明，李四他特意发生了一个攻击请求，他拿不到这个用户ID，就冒充不了这个张三。

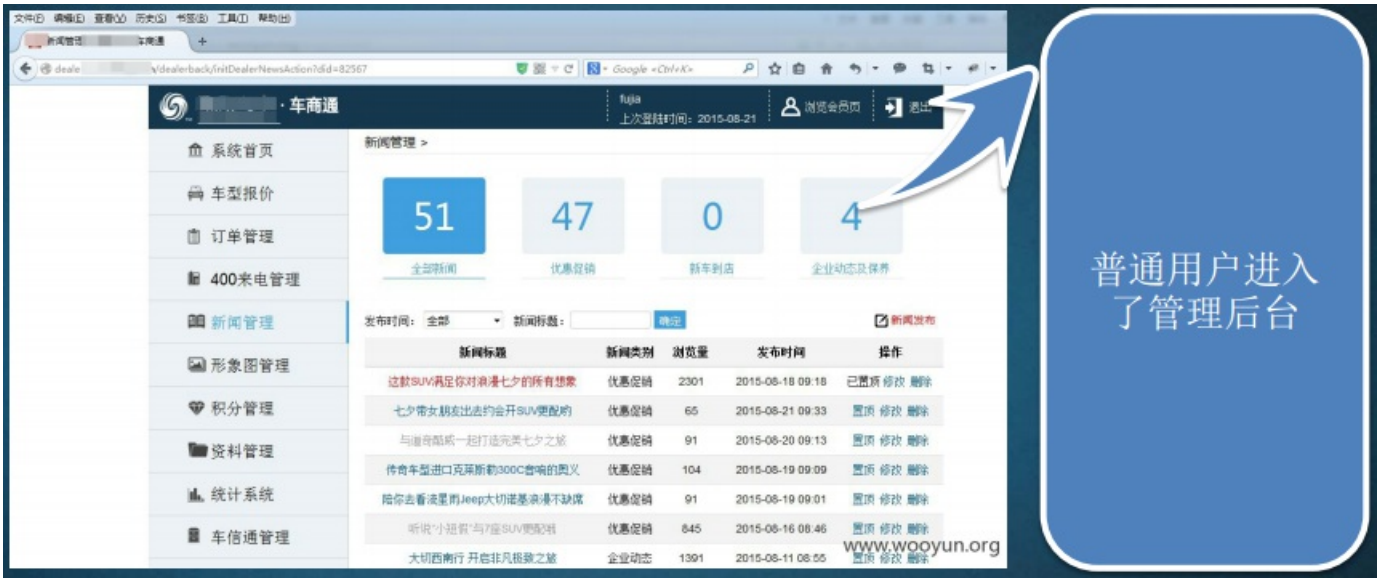
越权漏洞

我们再来看看越权漏洞，在一些系统当中如果存在着多种用户角色，每一种角色有不同的权限。操作业务适合如果权限不严格可能会发生越权漏洞。越权分为垂直越权和平衡越权。

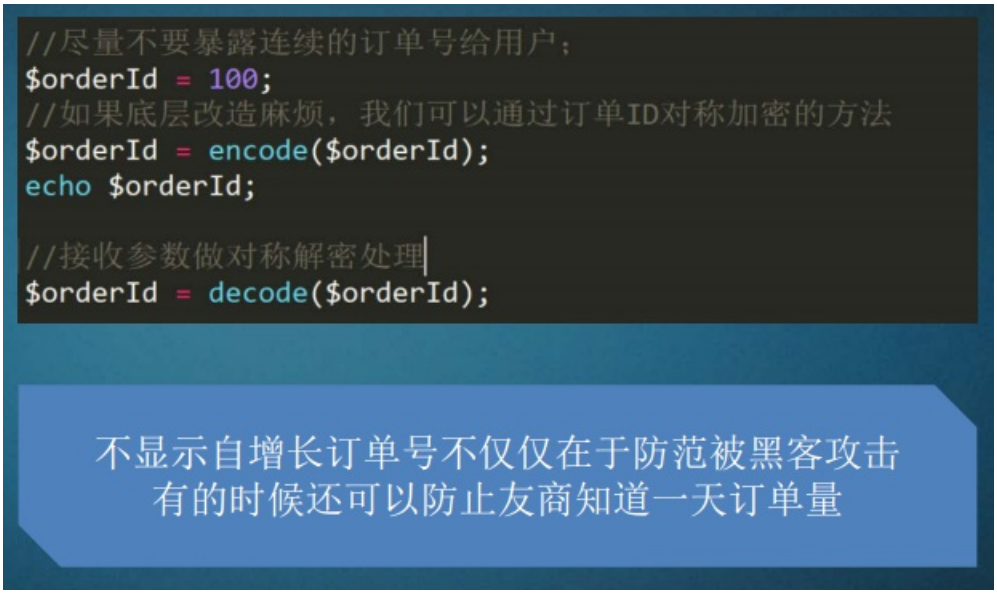
我们先来看一下平行越权。经常在WEB系统当中有商城，这个商城当中必不可少就有订单，订单肯定是有一个店铺ID，我们通常把它设置为一个自增长的ID，这个ID是一个数字型的，在URL上面如果我有一个订单ID就是100，我是一个攻击者我会尝试一下，100+1，当它ID等于101或者99的时候能否访问到，如果能否访问到就看一下这个订单信息不是我的，这个地方就存在着一个漏洞。张三可以看到李四的订单信息，这个时候就存在着越权。张三和李四是平级的用户，我是普通用户，他也是普通用户，他们两个权限是一样，互相可以看平台信息这叫做平级越权。这个有什么危害？比如说这个网站有的漏洞，如果是竞争对手他就可以知道用户在我的平台上下过订单的行为，然后去营销一下，还有一下我们把这个订单ID直接暴露出来，还有一种可能就是竞争对手会根据我们的订单IP的增长量，判断我们的增长量，就知道我们一天到底有多少订单。



平行越权防御方法，我们查询的时候必须加上当前用户的ID，就是orderID加上UID，这样不会出现张三可以看到李四的订单了。



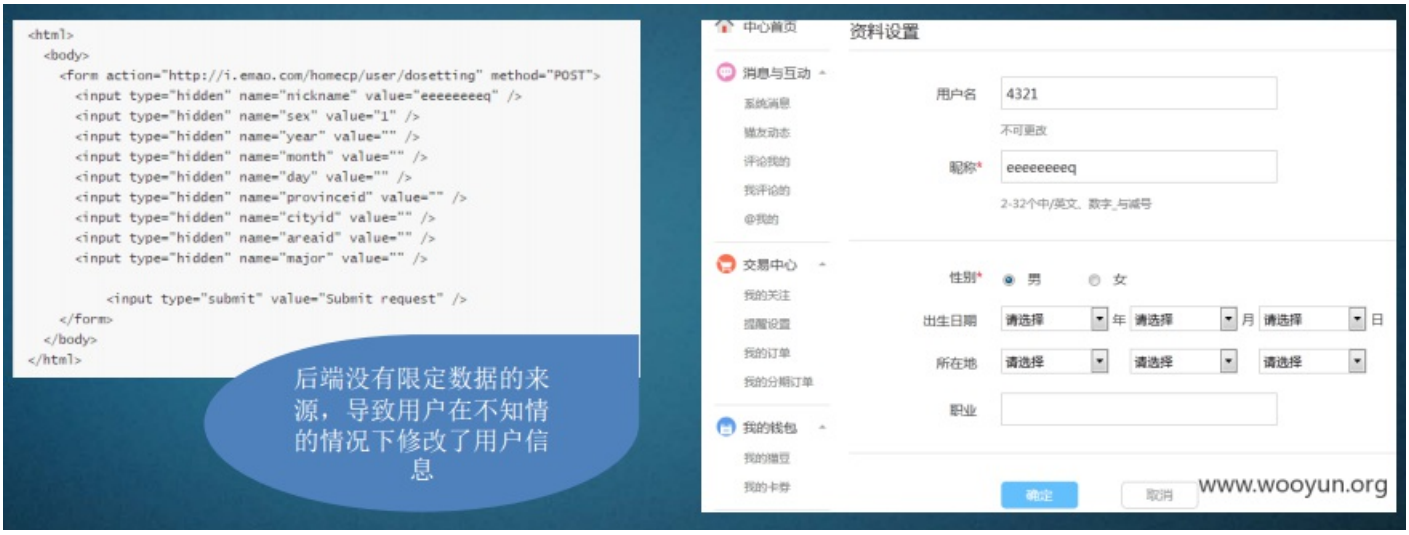
接下来我们再看一下垂直越权，这是一个普通用户进入到后台管理系统当中，他的权限就扩大化了，这个时候可以一些其他的操作，他有更多的权限了，通常发现这种情况原因，通常后台会集成到更多的控制器来统一管理。依然有一些邮件就漏掉了，没有集成到，就会出发这种情况。黑客不会一个一个找，会通过一些扫描器发现了，他就可以进去了。不要把自增长ID暴露出来，可以做对称加密或者非对称加密都可以，先转换为一个字母型的，让别人看不到你的数字型的ID是多少。别人就没有办法去通过这个加一减一的方式越权，也看不到你的一天业务增长量。



我看到一些代码在前台和后台会供应到长接口，前台和后台是有一些区别，前台有一个订单列表不会+UID，造成代码一直写写，前台也不会加入这个UID，我建议尽量把这个前台的方法和后台的方法区分开来。越权，其实不仅仅限于展示，我们刚刚看到了这个订单，张三可以看到李四的订单信息是查看，但是有的时候我们修改订单的时候也会主线这个问题。所以在读写的时候我们都需要注意一下这个越权的问题。

CSRF跨站请求伪造

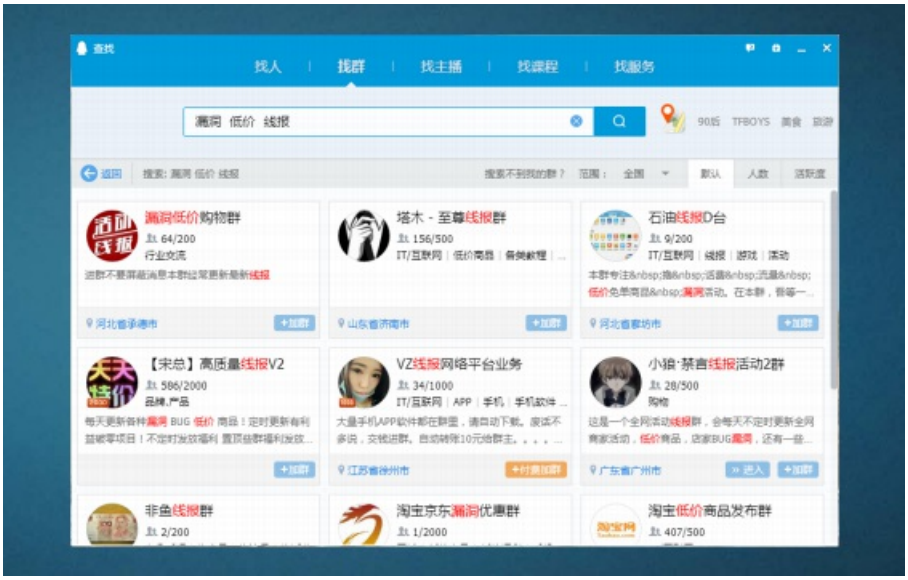
CSRF，这个通常会配合XSS使用。服务端错把浏览器发起的应请求当用户发起的请求，会造成XSS的问题。比如说我打开了张三的网站，登陆了一个用户信息，李四网站上有一个攻击代码，向张三这个网站发起请求，张三的网站会以为你本人发起的请求，他实际上是浏览器发出的请求。



比如PPT中有一张图片，图片有一个表单，左边是它的源码，我们可以看到表单每一项都在，但是从安全的角度上考虑它是少了一样东西，没有一些验证码或者TOKEN等等一些相关信息。服务端如果没有验证这个问题，就会造成这个CSRF的攻击。如何检测我们的系统当中是否存在这个CSRF？

可以去掉token信息看一下能否提交，把Referer能否提交成功，以及替代POST请求，如果都存在它就存在着CSRF。我们有一些建议，一定要验证Reeferer信息，Token的验证，图片验证码等等。根据我们的业务安全等级越高，最基础的我们可以用这个refefe需要验证，再高一级就是token，再高一级就是图片验证。

支付漏洞

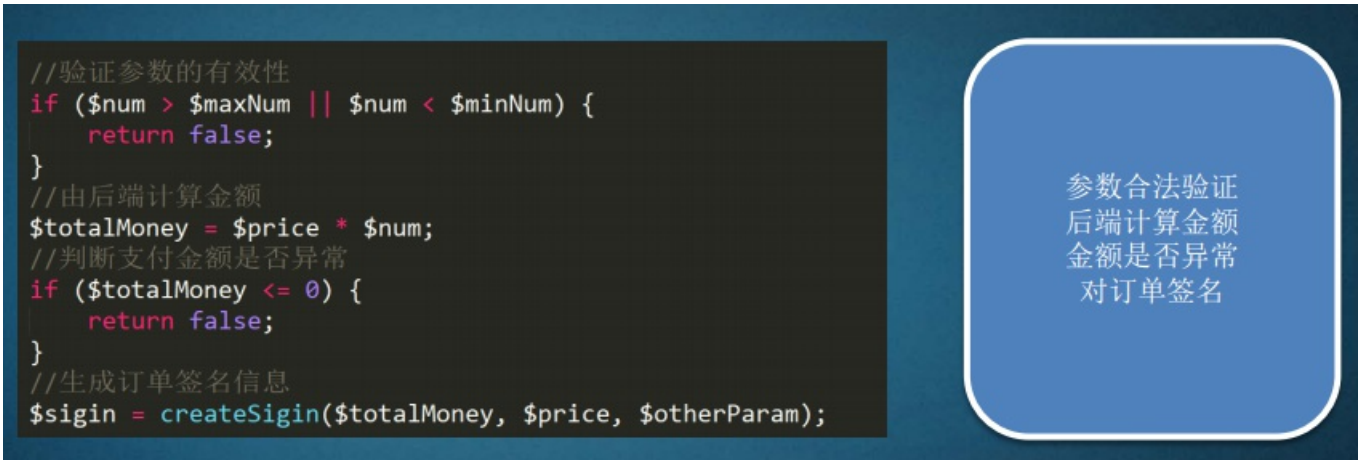


接着我们来看看支付漏洞,在这张图中,我搜索了"漏洞 低价 线报",搜索到一批QQ群; 我尝试加入过这些QQ群中观察，在群中看到有人在卖低价支付购买商品的漏洞，在跟其中的一个群友哪里了解到有很多做黑产的会把一些漏洞的信息卖到群里面去，其他的人就会去把这个漏洞给利用起来。

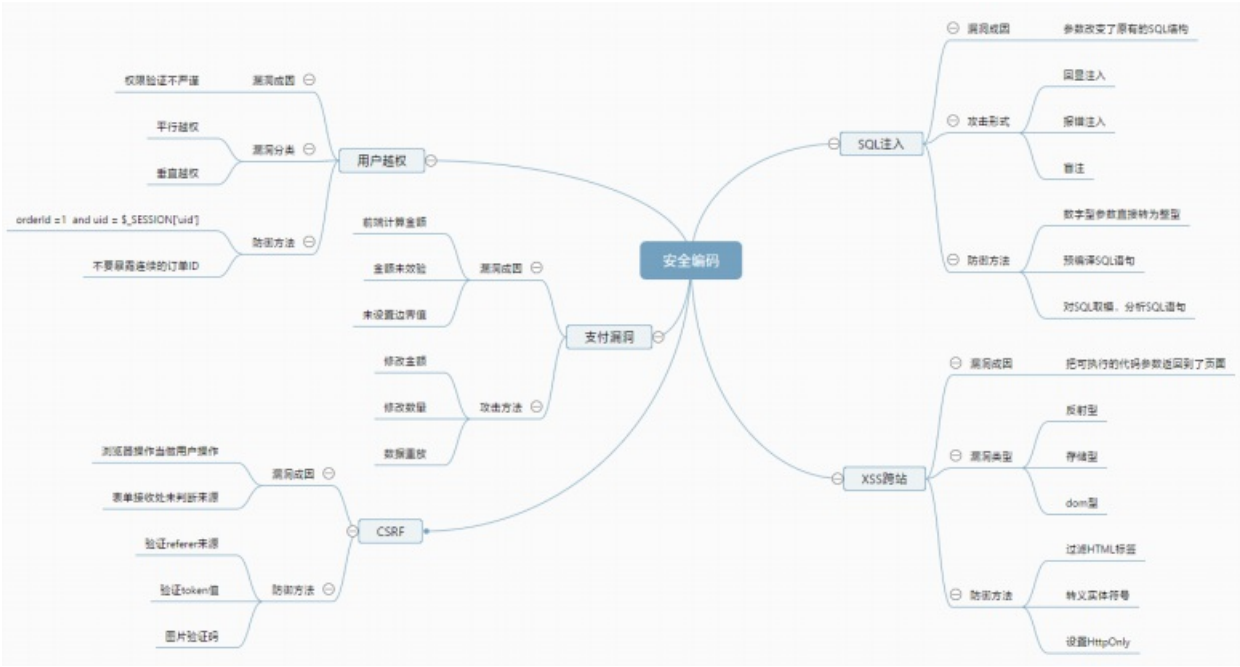
之前看到一个新闻，有一个浙江的老板，他想做线上找人做了一个网站，这个网站存在着一些支付漏洞，一周之后他发现这个订单量极速上升，卖了70多万，结果看了一下帐户余额只有几千块钱，报警之后才查到原因，但是货物已经发出去了。造成这些漏洞原因有很多，比如说支付金额是由前端提交的数据，不是后端计算的，而且没有对这个金额做校验，直接信任的前端提交的金额，导致这个攻击者可以随即修改这个金额，比如说修改为一分钱，这是非常典型的可以随意更改这个金额。

上面的金额是94元，这个表单里面改为一分钱，最后提交的时候是一分钱，这是非常好的漏洞，也是非常典型的。

还有一个问题是数量的限制，一个价格是26元，一个是27元，把这个数量变为负一之后，一提交变为一块钱了。这是之前数据包的漏洞，他充值了一块钱，他发现有一个数据包向网站发送，他就把这个数据包反复重放，就加了好几次，实际上只充值了一块钱。



如何防范？我们限制这个低价购买产品，比如说负数的时候肯定不行，等于零的商品的根据业务情况也是需要多注意的。限制免费商品获得金钱和积分的情况。有一些商品免费了，但是它可以获得一些积分，那就存在着刷积分的情况。



最后这个脑图这是我画的这次讲解的内容，大家可以下载看一下。我的演讲到此结束。

> 注：本文根据大会主办方提供的速记整理而成，不代表CSDN观点。

2017看雪安全开发者峰会更多精彩内容：

- 2017看雪安全开发者峰会在京召开 共商网络安全保障之策
- 中国信息安全测评中心总工程师王军：用技术实现国家的网络强国梦
- 兴华永恒公司CSO仙果：Flash之殇—漏洞之王Flash Player的末路
- 威胁猎人产品总监彭巍：业务安全发展趋势及对安全研发的挑战
- 启明星辰ADLab西南团队负责人王东：智能化的安全——设备&应用&ICS
- 自由Android安全研究员陈愉鑫：移动App灰色产业案例分析与防范
- 腾讯反病毒实验室安全研究员杨经宇：开启IoT设备的上帝模式
- 绿盟科技应急响应中心安全研究员邓永凯：那些年，你怎么写总会出现的漏洞
- 腾讯游戏安全高级工程师胡和君：定制化对抗——游戏反外挂的安全实践
- 绿盟科技网络安全攻防实验室安全研究员廖新喜：Java JSON 反序列化之殇
- 阿里安全IoT安全研究团队Leader谢君：如何黑掉无人机