## 0x01   修改支付价格

在支付当中，购买商品一般分为三步骤：订购、确认信息、付款。

那么这个修改价格具体是修改哪一步时的价格呢？在我看来，你可以在这三个步骤当中的随便一个步骤进行修改价格测试，如果前面两步有验证机制，那么你可在最后一步付款时进行抓包尝试修改金额，如果没有在最后一步做好检验，那么问题就会存在，其修改的金额值你可以尝试小数目或者尝试负数。

这里我找到了相关例子：

①：https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=8236

②：https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=21478

③：http://www.anquan.us/static/bugs/wooyun-2016-0174748.html

## 0x02   修改支付状态

这个问题我隐约的遇到过，之前在找相关资料的时候发现了一篇文章讲的是修改支付状态为已支付状态这样的思路，然后勾起了我的回想，这个问题是没有对支付状态的值跟实际订单支付状态进行校验，导致点击支付时抓包修改决定支付或未支付的参数为支付状态的值从而达到支付成功。

这里是一个例子，虽然其文章作者测试失败了，但我觉得思路是非常不错的，例子：

①：https://bbs.ichunqiu.com/forum.php?mod=viewthread&tid=28151

## 0x03   修改购买数量

在支付的过程中，数量也同时决定着价格，比如：1个数量商品对应的是100，2个数据就是200，那么当你修改这个值数量值为负数时，那么其金额也会变为负数，最后就会导致支付问题的产生。

## 0x04   修改附属值

这里是我自己想的一个词，比如在很多购买的时候都可以利用积分或者优惠劵等等进行代替金额付款，那么就容易存在问题。在这里我把附属值分为几类进行讲述。

①：修改优惠劵金额

优惠劵其基本都是优惠一半，一般用优惠劵进行消费一般出现在第二个步骤当中：确认购买信息，在这个步骤页面当中，你可以选择相关优惠劵，然后直接修改金额大于或等于商品的价格就可以，或者直接修改其为负值进行尝试，最后进行支付，如果对这点没有加以验证，那么问题就会产生，直接支付成功。

②：修改优惠劵金额及业务逻辑问题

可能你看到这个标题会想到，你不是上一个讲的就是这个修改优惠劵金额的问题嘛？为什么还要再讲一遍这个？请继续看！

之前遇到过这个漏洞，这个漏洞也是逻辑问题导致了成功利用，同样在是在第二部确认购买信息当中有可选择优惠劵进行支付，但是，当你修改其优惠劵值为任意值或负值想要支付的时候，会回显支付失败，或者金额有误等一些提示，可能这时很多白帽子会很失望然后就会去其它点找问题了，但当你找到个人中心，点击订单详情，如果存在这个逻辑问题，那么此时在你刚刚修改优惠劵金额后点击下一步支付的时候，其实这时候就已经产生了订单了，你在订单详情内就可以看到支付金额为0，因为你刚刚修改了优惠劵金额嘛，然后你点击支付就可以支付成功。

当然，这里还要说下小技巧，有可能会支付失败，但是如果你找到的这个问题是一个一般业务分站点，如果有自带的一个钱包功能，那么你就可以利用这个只带的钱包功能去支付这个订单，而不要利用其它支付类型，那么就可以支付成功！

③：修改积分金额

有些网站有积分，比如你消费多少，评论多少就可以拥有一定的积分数量，这个积分可以在你付款的时候进行折扣其订单金额，如果这个没有做好积分金额的校验，那么当你在支付当中选择用积分为账户减一些金额的时候，可以抓包修改其积分金额为任意数或负金额，然后可0元支付成功。

相关例子：http://www.anquan.us/static/bugs/wooyun-2015-0139556.html

## 0x05   修改支付接口

比如一些网站支持很多种支付，比如自家的支付工具，第三方的支付工具，然后每个支付接口值不一样，如果逻辑设计不当，当我随便选择一个点击支付时进行抓包，然后修改其支付接口为一个不存在的接口，如果没做好不存在接口相关处理，那么此时就会支付成功。

## 0x06   多重替换支付

以前好像也看到过相关的例子，首先去产生两个订单，这两个订单商品是不一样的，其价格不一样，如果服务端没有做好这相关的验证，那么在支付的过程当中抓包，修改其订单值为另一个订单值，最后支付，这时就可以用订单一的支付价格买到订单而的商品。

## 0x07   重复支付

这个其实只是支付当中的一个别类，但是这个思路新颖，所以我就列了出来，比如一些交易市场有一类似于试用牌子或者其它，这个试用牌子可以依靠签到获得，而这个牌子的作用可以去试用一些商品，在你进行试用的时候会扣掉你的试用牌子，当你试用完成或者主动取消试用时，试用牌子会返回到账户当中，你知道，签到得到的牌子肯定很少，且如果想试用好一点的商品那么牌子的数量就尤为重要了。

这里的问题就是如果没有进行对订单多重提交的校验，那么就可导致无限制刷牌子，比如，你试用时抓包，然后你每次试用都会产生一个订单号，然后利用刚抓到的数据包进行批量提交，你就可以看到每次提交的订单号不一样，然后这时你再看订单可以看到同一个商品的无数订单，但试用牌子数只扣了你第一个试验时的牌子数，那么这时你申请批量退出试用，那么这么多订单，每退一个就会退相应的牌子数量到账户当中，这就构成了无限制刷得问题。

## 0x08   最小额支付

在很多白帽子测试支付的漏洞时候，修改的金额往往都是0.01等或者负数，我想说这很容易错失掉一些潜在的支付问题，我就深有体会，在挖掘支付漏洞的过程当中，就遇到过，直到第三次再一次检测时才发现，比如一些网站有金币或者积分什么就相当于支付可以用这些支付，那么在充值的时候，比如：10元对应的积分值为100、50对应的是5000、100对应的是10000。

这个问题如果你在充值时进行修改其支付金额为负数或者0.01等是会显示支付失败的，但是如果你修改其金额为1.00，那么支付就会成功，也就用1元购买到任意值得积分数量了，这是为什么呢？

其实你在测试过程当中细心点就可以很好发现的，这里最低就是1元，1元对应100积分，而你如果修改为0.01，那么对应的积分就是空值了，所以会显示失败，而当你修改为1元，那么1元这个支付接口是存在的，其后面积分数为其它金额的积分数，然后跳转过去支付就会以1元购买到比它多得多的积分数量，也可以是任意积分值。

## 0x09   值为最大值支付问题

以前也是看到过相关的例子，一些网站比如你购买商品，这里有2个思路修改值，1是直接修改支付金额值为最大值，比如999999999，或者修改附属值，如优惠卷，积分等为999999999，如果这里逻辑设计有问题，那么其支付金额会变为0。

## 0x10   越权支付

这个问题很早之前有过，现在可能很少存在这类问题，在支付当中会出现当前用户的ID，比如：username=XXXXX，如果没有加以验证，其支付也是一次性支付没有要求输入密码什么的机制，那么就可以修改这个用户ID为其它用户ID，达到用其他用户的账号进行支付你的商品。

## 0x11   无限制试用

一些网站的一些商品，比如云系列产品支持试用，试用时期一般为7天或者30天，一个账户只能试用一次，试用期间不能再试用，但如果这个试用接口会做好分配那么很容易导致问题的发生。

这也是我遇到过的例子，比如：在支付的时候它URL后面的支付接口是3，而试用接口是4，那么此时你已经使用过了，复制下确认试用时的URL，修改后面的支付接口为3，那么此时就会调用购买支付接口，但是由于你本身这个产品就是试用的，其相应值绑定了这个试用商品，那么金额就肯定是0，那么最后点击支付，你就可以看到支付成功，试用成功，又重复试用了一次，然后他们的试用时间会累加在一起，这就导致了可无限制购买任何产品了。

## 0x12   修改优惠价

比如一些商品有优惠价，优惠多少多少，那么在支付时抓包，修改这个优惠价就可造成支付问题的产生。

支付问题的相关分析文章：

①：http://wooyun.jozxing.cc/static/drops/papers-345.html

②：http://xdxd.love/2015/12/02/%E6%94%AF%E4%BB%98%E6%BC%8F%E6%B4%9E%E6%80%BB%E7%BB%93/

# 以下是不常见思路

## 0x01   多线程并发问题

可能很多白帽子知道，也有可能不知道，或者听说过，但是没有实际挖掘过，那么我相信，这个思路会让你们有新的挖掘方向了。

现在可能还有一些大厂商存在该问题，多线程并发问题就是没有实时的处理各种状态所导致的问题，之前挖掘过刷钱问题，就是利用该思路，比如很多平台有自家的钱包，而这个钱包是一个迷你钱包，这个钱包作用也仅是用于这当前一个业务平台网站，在提现时，没有任何验证码或者校验机制，只要输入体现金额就可以提现，并且是秒到账，如果什么负数，修改金额都测试过了都不行，那么你就可以试试多线程并发问题，提现时抓包，比如我现在钱包内有0.1元，那么按理说每提0.01可以体现10次，也就是发送10次进程，但是利用这个问题可以达到多发现几次成功的进程，提现时抓包，然后把数据包发送到BurpSuite工具的Intruder当中，进行批量发送18次，然后可以看到成功的提现到了12次。

这里我贴出相关证明图片：

这里是从0开始到11截止，我账户内只有0.1 而这里体现了0.12 也就是提现的进程为12次，369为提现成功，349为提现失败的长度值，从这里就可以看出这个问题的危害了，当然此时账户的金额肯定是为负的了，如果把这个提现金额变大，那么这多提现的金额可不是闹着玩的。

当然，多线程也可以在其它功能处进行测试，比如我之前讲到的试用商品问题，就可以通过多线程进行多几次的使用，比如利用积分总换礼品，一个账户只能进行总换一次，利用这个问题，可以多几次总换，一些转账功能，提现功能，购买功能等等很多。

多线程并发的相关分析文章：http://wooyun.jozxing.cc/static/drops/papers-831.html

## 0x02   支付问题挖掘技巧

如果你习惯用BurpSuite工具，那么在你测试抓包的时候通常请求包都有很多，比如有3个请求包，第一个请求包是一个干扰数据包，第二个是一个图片加载的数据包，第三个可能才是支付相关的数据包，所以有时候要细心，不要认为抓不到，如果你用的是其它工具，那么可以查看整个提交过程，所以很容易看到提交的各种数据包，在BurpSuite当中你可以通过Target这模块进行分析，这个模块会有你测试时相关的数据包。