三度空间应用安全测试报告

# 短信接口可被恶意利用发送垃圾短信

## 漏洞描述：

短信验证码作为重要的身份验证工具，因其操作简便、安全性高、时效性强等优点已被开发人员广泛使用。但因其获取便利、限制较少容易被不法分子利用进行短信轰炸，恶意刷掉大量短信费用，给公司或个人造成大量的金钱损失，造成这种情况原因主要是在产品实际设计过程中，有些产品人员因为对技术实现不太了解，防范意识薄弱，简单或直接忽略对短信验证码进行限制，这才造成短信接口恶意被不法分子利用。

## 漏洞演示：

* 打开“三度商家管理系统”，点击忘记密码选项；
* 输入手机号，我的手机号没有注册也可以获取到验证码！在获取验证码的时候抓包，获取到的数据包如下：

GET /v1/center/getSms?phoneNumber=15577429534 HTTP/1.1

Host: uc.sanduspace.com

User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: application/json, text/plain, \*/\*

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Referer: https://admin.sanduspace.com/

Content-Type: application/json

Platform-Code: merchantManage

Authorization: undefined

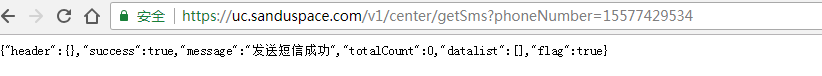
Origin: https://admin.sanduspace.com

Connection: close

* 可以看到是一个跨域请求发送验证码，拼接发送验证码的链接（Payload）：

<https://uc.sanduspace.com/v1/center/getSms?phoneNumber=15577429534>

* 经过测试，phoneNumber参数可任意修改，攻击者可利用其做短信轰炸



## 修复方案：

1. 前端增加图文验证码，防止攻击者低成本的采用自动化攻击；
2. 限制单个手机号每日接收短信次数和时间间隔；
3. 对单IP做请求次数限制，用户请求次数超出预定的阈值，暂停对该IP对服务器请求。

# 运营网站充值支付金额可被篡改

## 漏洞描述：

此次漏洞不太好描述，有点奇怪，不符合逻辑，初步判断是开发未将测试代码删除就直接发布到生产环境，具体请看漏洞演示，危害就是可被羊毛党大量薅羊毛。

## 漏洞演示：

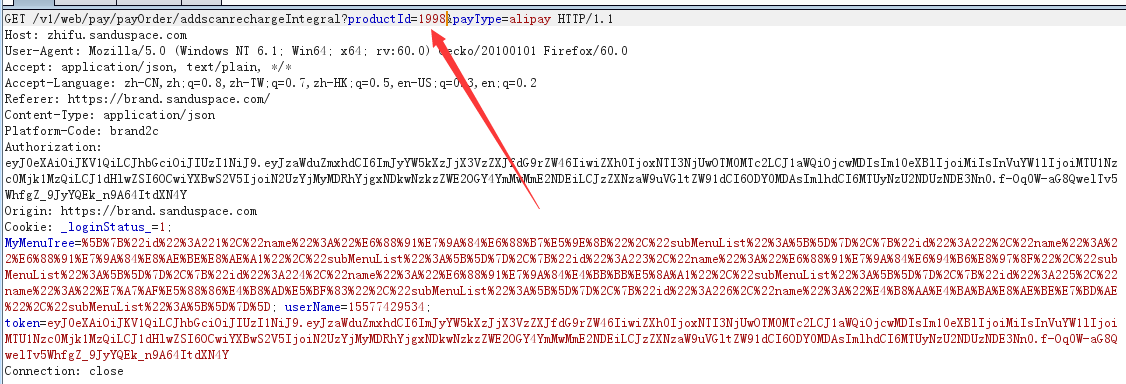
* 在品牌网站上注册一个用户，进去之后来到积分中心页面



* 点击充值，看价格是一块钱10个积分，我选择了5000积分



* 选择支付宝支付并抓包，反复测试之后发现更改productId参数值后会返回一个支付金额和充值积分不匹配的二维码，我改为1998



* 服务器返回一个支付页面



* 使用支付宝扫描，发现只需支付一分钱就行



* 但最终结果并没有获取到5000积分，而只是获取到40积分，虽然说没有获取到5000积分，但也比正常充值的单价1块钱10积分便宜很多，可利用薅羊毛。



## 修复方案：

在接收到客户端充值请求参数时，应严格验证其参数的有效性。但此漏洞不像未对充值参数做校验，应该是历史遗留的（开发测试时的）充值逻辑判断代码未及时删除导致。

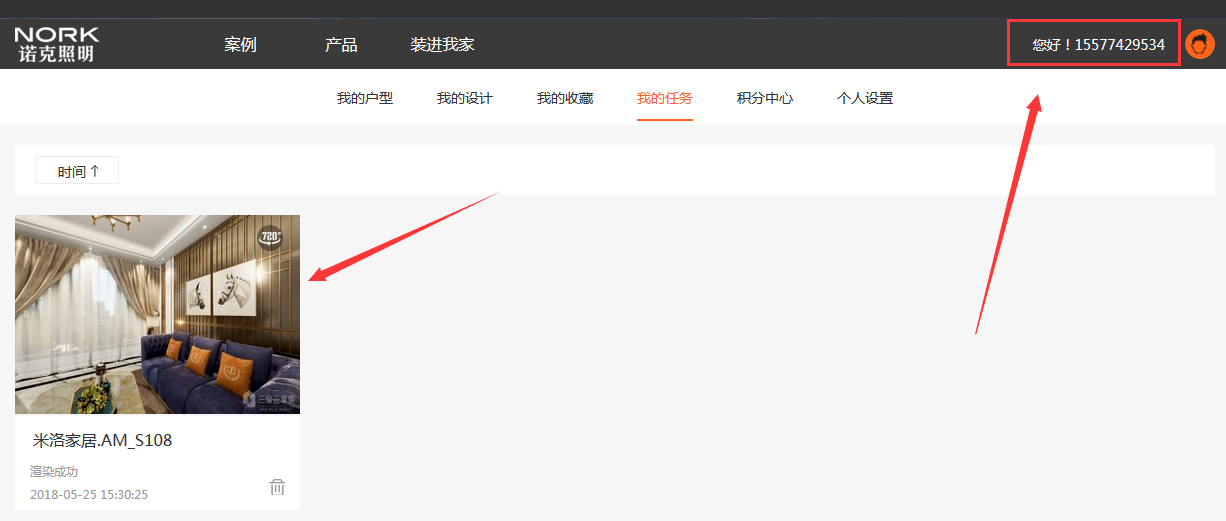
# 3.运营网站可删除其他用户渲染成功的任务

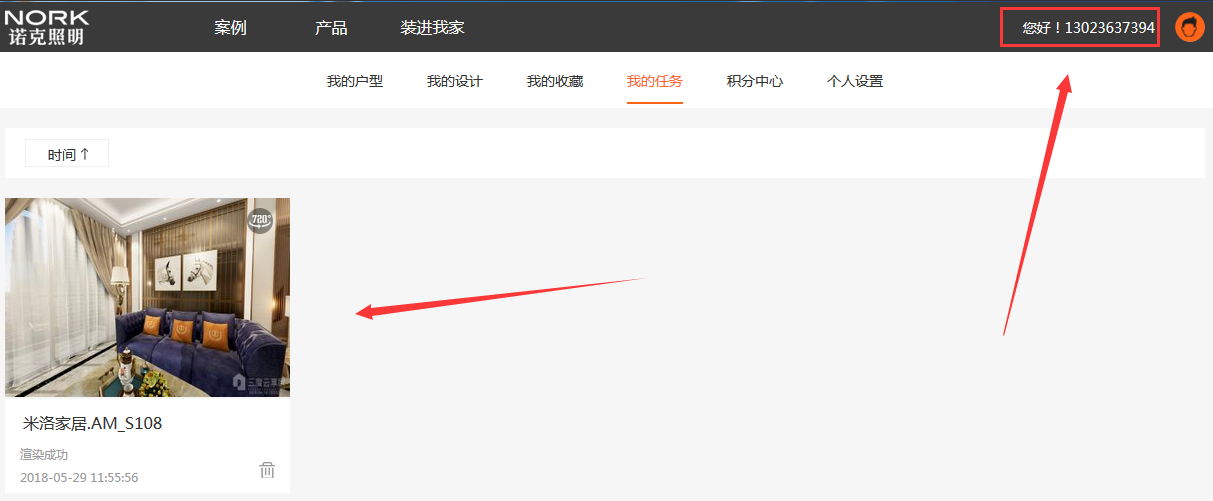
## 漏洞描述：

越权漏洞是比较常见的漏洞类型，越权漏洞可以理解为，一个正常的用户A通常只能够对自己的一些信息进行增删改查，但是由于程序员的一时疏忽未对信息进行增删改查的时候没有进行一个判断，判断所需要操作的信息是否属于对应的用户，可以导致用户A可以操作其他人的信息。

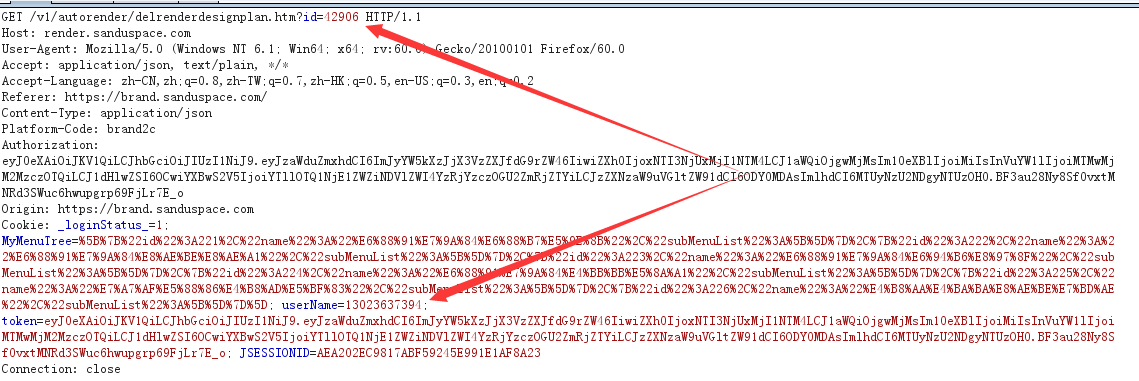
## 漏洞演示：

1. 注册两个账号进行测试
2. 每个账号都渲染一个720任务



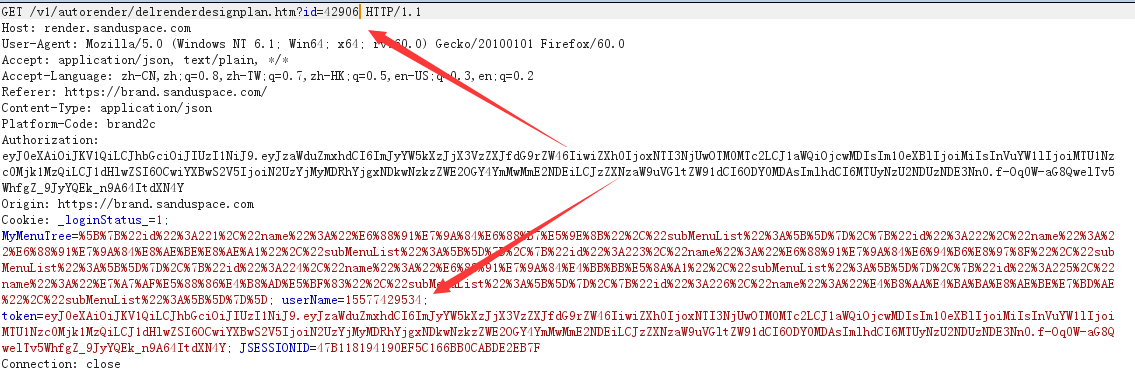


1. 查看130手机号渲染成功的任务删除id为42906



1. 来到155手机号渲染成功的任务，点击删除，抓包修改，将删除id修改为130手机号渲染成功任务的删除id42906





1. 服务器返回删除方案成功



1. 来到130手机号的渲染任务里，发现渲染成功的任务已被删除



## 修复方案：

用户对某一些数据进行增删改查时，需要去校验该用户所操作的数据是否属于该用户。客户端登录成功后，应将客户端身份标识UserID保存在服务器端的会话中，在后续的交互中，依据客户端cookie 中的会话 ID(即JSESSIONID)在服务器会话中查找身份标识 UserID，进行数据库查询后返回 UserID对应数据。