

Contents

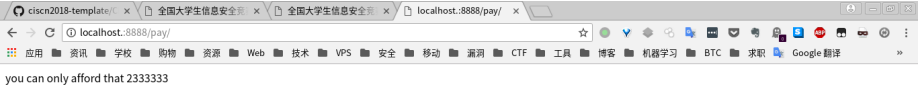
1	解题过程	1
2	设计思路	6
3	考察技能	6

1 解题过程

注册帐号后在个人页面可见，初始积分为 1000。



在一件商品购买 10 件以上后将得到如下提示，说明普通用户同一间商品无法持有 10 件以上。

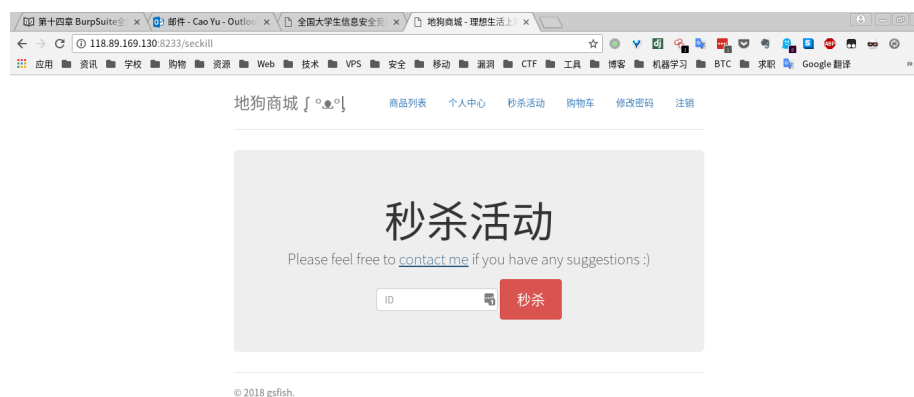


则此刻可考虑使用管理员帐号进行购买，管理员帐号一般为：admin（本题）、root 等常见帐号。

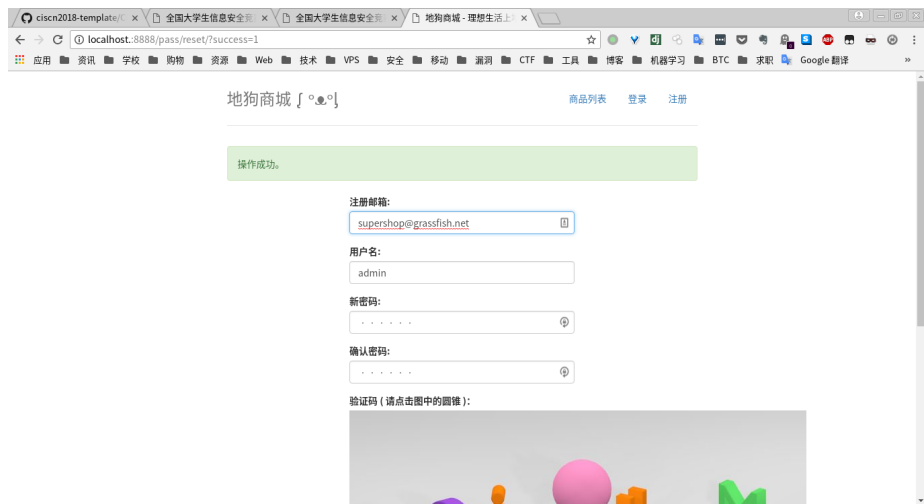
在密码找回页面可以发现，该密码找回逻辑具有严重的缺陷，仅需提供注册时所使用的邮箱及用户名，即可实现密码重置。则此刻可在网站收集与管理员相关的信息。



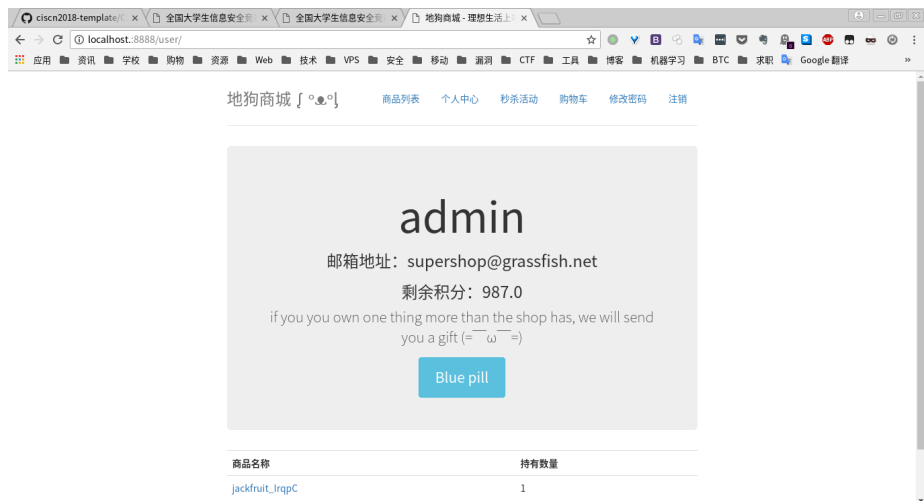
继续浏览站点，在秒杀活动页面，点击“秒杀”按钮后显示如下提示。由于网站还未建好，管理员留下了自己的联系方式（help 处为 mailto 形式的链接）。



使用 admin 作为帐号 mailto 留下的邮箱即可将管理员的密码进行重置，并登陆其帐号。



使用管理员帐号购买任意商品后得到如下提示。Blue pill 为积分重置，仅管理员可使用该功能。



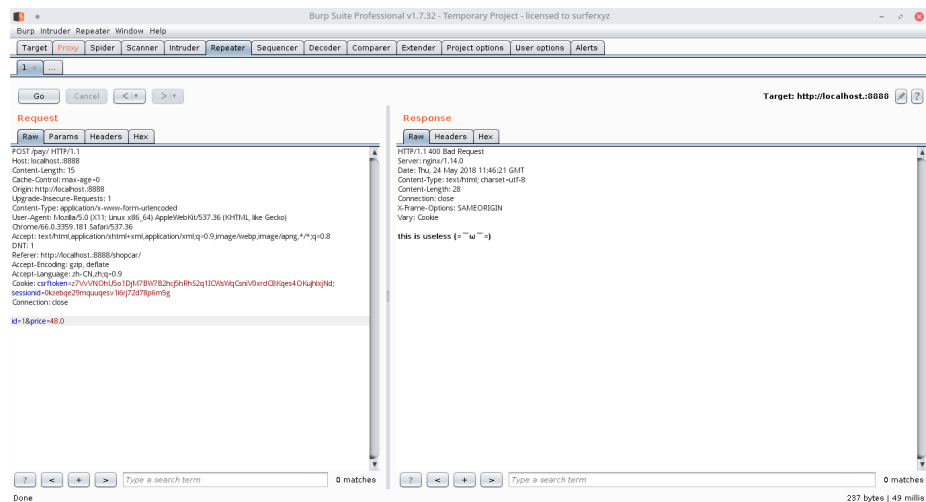
如果管理员能持有额外的商品（多余商城所提供的），那么他将得到商城提供的特别礼物。

由于初始积分只有 1000，由提示可知需要持有 1000 件以上的同一商品。通过尝试可知，在注册时填写邀请人帐号，则邀请人可获得 2 点积分奖励。然而网站对该薅羊毛行为做了防护，一方面注册时需要填写验证码，另一方面每个用户在邀请 10 位新用户注册后将不再获得奖励。

因此此时需对支付逻辑进行漏洞挖掘。通过抓包可发现，在购物车结算时会发出包含以下参数的 POST 请求

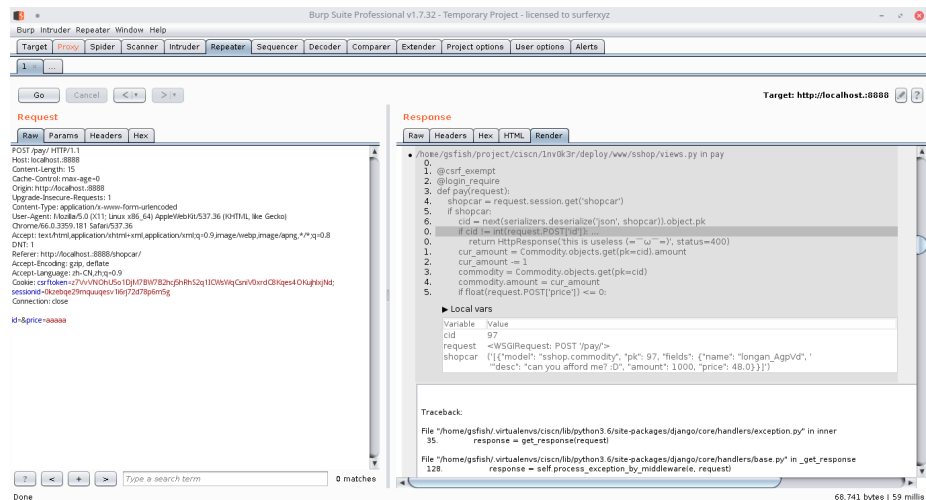
`id=97&price=48.0`

测试 id 可以发现，该值必须与所选商品的 id 相同。



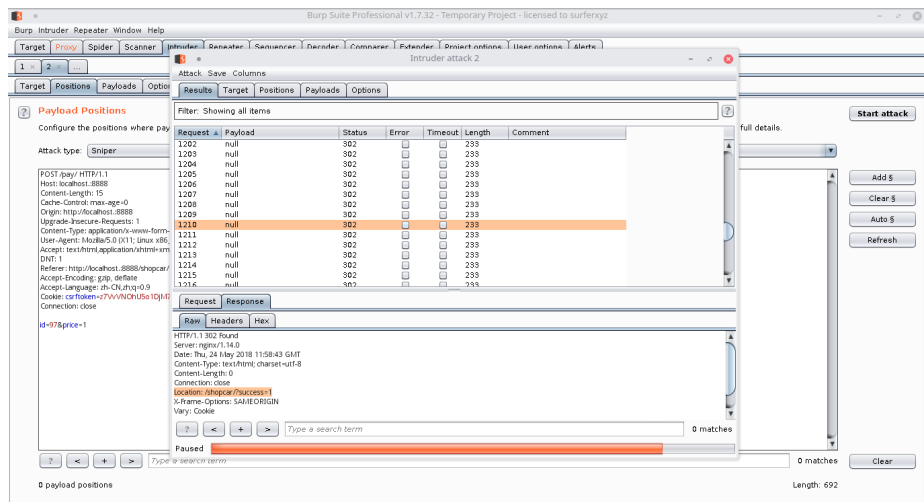
测试 price 可以发现，在 price 大于 0 的情况下是可以支付成功的，因此可以将 price 改为 1。然而即使售价为 1，最多也只能持有 1000 个，因此继续测试。

经过测试发现，若将 id 或 price 改为空，或者字符时，服务端会报错。由于网站未将调试模式关闭，因此可以获得详细的信息。

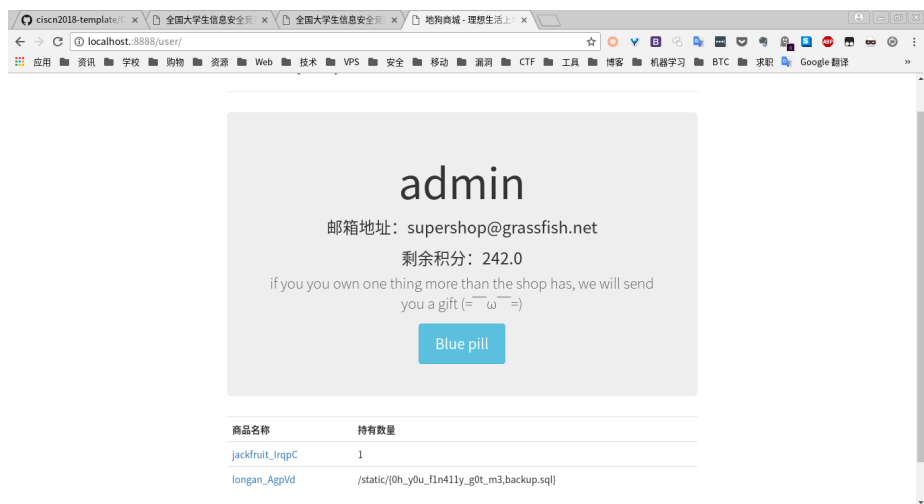


通过分别尝试，可以从报错信息中获得处理 id 和 price 具体代码。从中可以发现，在处理商品数量和用户积分的代码逻辑中，存在竞争条件漏洞。

因此，此时将 price 改为 1，并通过多线程发包即可利用竞争条件漏洞购买 1000 件以上的商品。



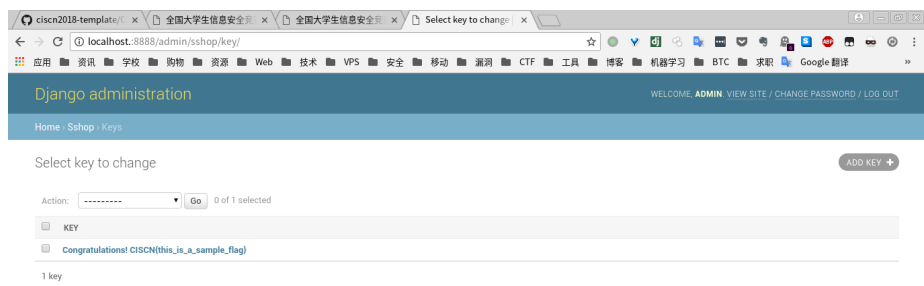
攻击完成之后,重新查看 个人中心页面,得到两个关键文件的路径。其中 0h_y0u_f1n411y_g0t_m3 为一个密码字典, backup.sql 为 Django 导出的数据库备份。



从得到的数据库备份中可以得到 Django superuser 的密码哈希。格式如下:

md5\$<salt>\$<hash>

可使用 Django 自带的哈希 API `django.contrib.auth.hashers.check_password`, 用 0h_y0u_f1n411y_g0t_m3 作为字典, 对数据库备份中的哈希进行破解。得到明文后, 用其登陆 Django 自带的管理页面, 从 Key 表中拿到最终 Flag。



2 设计思路

1. 找回密码功能的逻辑缺陷，并使用在网站搜集的信息重置管理员帐号；
2. 通过测试请求参数获取调试信息，并通过部分代码审计发现竞争条件漏洞；
3. 通过利用条件竞争漏洞实现商品超售；
4. 利用 Django API 对数据库备份中的哈希进行破解，登陆 Django 管理界面；

3 考察技能

1. Web 逻辑漏洞挖掘；
2. 敏感信息利用；
3. 代码审计；
4. 竞争条件漏洞利用；
5. 脚本编写；