安全小课堂第120期【国内SRC漏洞挖掘经验和自用技巧】

京东安全应急响应中心 11月26日

国内企业近年来越来越注意企业自身安全，也都开始逐步建立自己的安全应急响应中心平台。由于各家厂商情况不一和白帽子水准层次不齐，因此目前国内厂商和白帽子都还有一定的进步空间。

2015年-2017年三年整，先后主要给阿里，网易，唯品会提交过一些高质量漏洞。因为工作，2018年一整年我都没有给任何平台提交过漏洞，以后我也不准备在SRC这块投入过多精力了，因此希望能多分享一些自用经验技巧以及个人想法给大家。

JSRC **安全小课堂第120期**，邀请到**硬糖**作为讲师就**国内SRC漏洞挖掘经验和自用技巧**为大家进行分享。同时感谢小伙伴们的精彩讨论。

国内SRC漏洞挖掘的需要注意什么？

京安小妹

硬糖：

之前想过，这一部分主要还是针对白帽子的。

在参与国内厂商漏洞挖掘时，作为白帽子我觉得应该注意以下两点：

1. 遵从厂商规则。厂商规则不一定是完整的，厂商规则也肯定不能让所有人满意，但厂商规则的制定一定符合厂商自己的利益。漏洞挖掘测试过程说到底也是未授权的非法行为，如果因此超越了厂商规定的范畴，触及了厂商的底线，例如私自下载公开相关数据等，自己可能面对的就是一个厂商的法务人员而不是审核人员。这类事情，可能公开的并不多，但是真实发生的事情还是挺多的。
2. 第二点其实和技术没太大关系了,尊重工作人员。其实我是一个很较真，又暴脾气的人。我清楚记得我曾因为漏洞定级问题曾经对网易的审核人员出言不逊（虽然至今我依然认为定级有问题），对审核人员造成一定困扰，幸亏对方对我一直很忍让，否则我现在可能失去一个很好的朋友。在此还是向@网易大枫哥 表示抱歉。而且你也并不知道你提交的漏洞会具体到个人身上造成什么样的后果，可能会让人离职也可能让人……提交漏洞获取奖金的目的无非是提高技术和生活质量，其实蛮希望大家可以对技术较真而不是对人。

国内SRC平台的两种路线选择模式

京安小妹

硬糖：

刚刚说的有点煽情了，来说说国内SRC平台两种路线选择模式。

因为其实我接触的厂商并不是很多，所以只摸索出了两条路线选择模式，如果有补充欢迎提出。

我的路线区分的依据是厂商的业务线长短,其实也就是厂商规则中接受的业务范围大小。

1. 业务线较短的，即业务范围较窄的。这种业务线较短的厂商其实不用每天每周都去投入很大精力，因为业务线只有那么长，很多人都在同一时期挖掘业务线上的漏洞，由于观察力，技术甚至手速等原因很可能会一无所获。在挖掘此类SRC漏洞时，我的周期都是半年看一次，虽然可能会错过一些低危漏洞，由于半年周期内厂商自身业务线的延长会增加自己的业务系统，因此你就可以通过新增的业务系统发现一些高危或者严重漏洞从而可能获得高额的奖金，我在某SRC获取几万元奖励的严重漏洞就是通过此方法。之前发现在一个已有系统中发现一个安全隐患但无法扩展利用，在等待半年后发现一个新增系统存在同样隐患并可以扩展利用。因此面对这类厂商需要做的是，紧盯新增业务，新增业务出现时，要有足够的行动力尽快将业务过一遍，避免手速过慢被撞洞。标记是我觉得比较重要的，虽然和技术没太大关系。
2. 第二种，也就是业务线较长的，即业务范围较广的。这类厂商接可以投入大量精力去挖洞，去提炼，去刷洞。大概是这样一个过程。当开始面对一家新的厂商时，可以投入大量精力挖洞。积累到一定程度后，可以发现其实每一家厂商都有自己的“通病”，从开发角度看这些业务线较长大厂商都有一套自己的“行为准则”，虽然可能不是写入公司的明文规定，但一些自带BUG的习惯和框架可能因为用顺手了，所以就应用到各个业务系统中了。因此面对这类厂商需要做的是，挖洞积累到一定量，可以去尝试总结提炼自己挖到的漏洞存在哪些共同点，这些共同点往往可能都会存在于其他系统包括重要系统中，利用共同点去刷洞，达到快速获取相应赏金的目的。

讲师

我的漏洞挖掘信仰——“爆破”

京安小妹

硬糖：

其实这个我之前在山东济南的DC0531的分享上分享过了，不过由于“爆破”是我唯一会的东西了，自己也投入了一些精力，所以就拿出来多说说。

粗略统计了下，在我的严重漏洞中用“爆破”发现的可能占到了75%。开始爱上“爆破”的启蒙案例是这样的：

在一次做安全测试时，因为测试对象刚做完漏洞的修补和一些软硬件的加固，因此并没有测试出质量较高的漏洞。因为此类网站经常被一些黑阔入侵，我想到应该会有一些残存的东西在站点内，例如webshell，因此我拿出了burp开始对网站的根目录和图片目录进行爆破，希望能爆破到webshell，如愿以偿，我最终在网站根目录下发现一个webshell并成功爆破出了密码。网站虽然做过修补加固，但是不知道为啥留了shell在里面，也是有点尴尬，从那时起，我开始关注“爆破”，希望能够挖掘出爆破的其他技巧。 后来，我通过爆破先后挖到了Uber以及国内几个大厂的严重漏洞。

“爆破”最需要的当然是字典。

之后，我做了这样一个事情，我下载了1000多个网络上开源的web源码，通过正则提取这些源码的目录结构，可执行脚本文件名，参数名， JS脚本名。收集目录结构是为了爆破目录，收集可执行脚本文件名是为了爆破一些爬虫获取不到的可执行脚本文件，参数名当然也是为了爆破可执行脚本的参数，收集JS脚本名是因为其实我碰到很多站都会把一些重要的API存放在JS文件中，这种针对性的收集，肯定会减少爆破时遗漏的情况。（字典将整理后上传 Blog ）

这里说了字典的获取，但问题也来了:字典当然是越全越好，越多越好，但是也要考虑效率问题,如此庞大的字典怎么才能更高效的使用，这是一个不太好处理的问题。

我们可以将字典入库，增加keywordcount字段用于计数，当我们的关键词命中时，对应的keywordcount值加1。每次使用字典时，从数据库中order by keyword_count desc提取关键词，这样会生成一个根据关键词命中次数降序的字典，这样经常命中的关键词就会靠前，我们使用的字典的效率也会提高，循环往复我们的字典将会越加成熟。

一些其他漏洞挖掘小技巧

京安小妹

硬糖：

关于SRC的其他小技巧，其实上面也提过，在各个其他的会议分享上也有其他师傅提过，所以内容可能重复，见谅。

1. 多关注新增业务。

可以关注每个业务对应的微信公众号，因为一般有新业务新活动都会通过微信公众号通知；也可以关注招聘广告，一般朋友圈刷屏的招人广告代表该业务方向是开始着重发展的，可以重点关注下。我之前的一个最高奖励的严重漏洞，就是因为关注到了一个业务在招安全人员，因此赶紧去测试该业务了。

1. 尝试申请非普通用户拥有的权限。

如：商家，合作方。这些用户权限因为能申请到的不多，因此被测试的次数也较少，更容易发现漏洞。

例如一个漫画业务，漫画既有读者也有作者，这里不仅可以测试读者权限的点，也可以申请作者权限去测试。

1. 留意标记暂未发现漏洞的站点。

一般站点多多少少都会有一点问题，没有问题的站点大概率是因为一些知识点暂时没有关注到，在以后学习到新的知识点时，可能就会猛然回想起当时没测试的站点。

不管在安全测试还是攻防实战中经常回头看看都会有意料之外的惊喜。

这些技巧主要都是技术之外的技巧，在我看来，技术上的东西只要多学习多交流，应该都会有很好的技术，在针对漏洞挖掘这块更多考验的是思路和耐心。

讲师

可以分享一个实际案例吗？

京安小妹

硬糖：

这里案例还是分享通过“爆破”方式形成的案例。

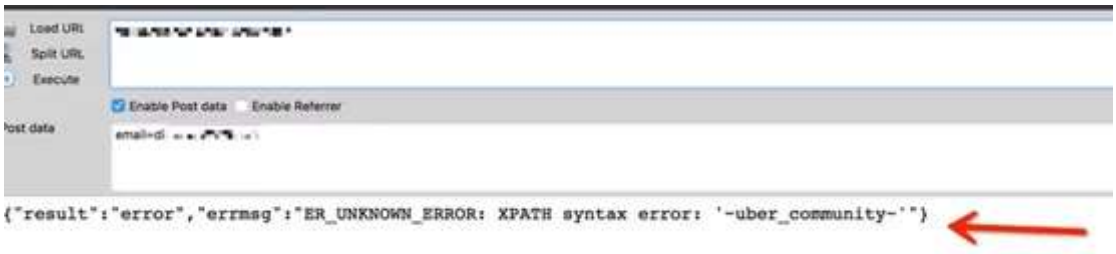
案例在曾经在其他地方分享过了，因为一些其他漏洞的保密原因，实在只能找到分享过的漏洞案例了。

我的朋友@cy(5up3rc)之前在 Hackerone 上提交了1个Uber的SQL注入漏洞，了解了下大概内容，由于该站点把一些API及其参数方法写入了JS中，可以通过分析JS构造数据包进行请求，其中1处API存在SQL注入，提交给Uber 后获得相应奖金。

这里其实也是我上面提到的，一些站点会把接口写在js中。前台接口会写入，后台接口也会写入，我朋友发现的就是前台的接口，当时存在漏洞的API早已修复，但是我凭一种直觉，当时我觉得应该还是有漏网之鱼的。通过爆破API路径，成功获得2个未写入前台JS的的未授权后台API，然后根据JS里已有的参数名，最终构造爆破出了8个参数。最终成功获取到一个未授权的数据写入API和一个未授权的数据查询API。

这是写入数据的api，在数据写入的API中我在openid这个参数里面添加了注入语句，可以看到返回正常页面，并未形成SQL注入。

接着我再通过查询API，查询刚刚添加的用户，形成二次注入。
这样数据库名就出来了，一个完整的二次注入。

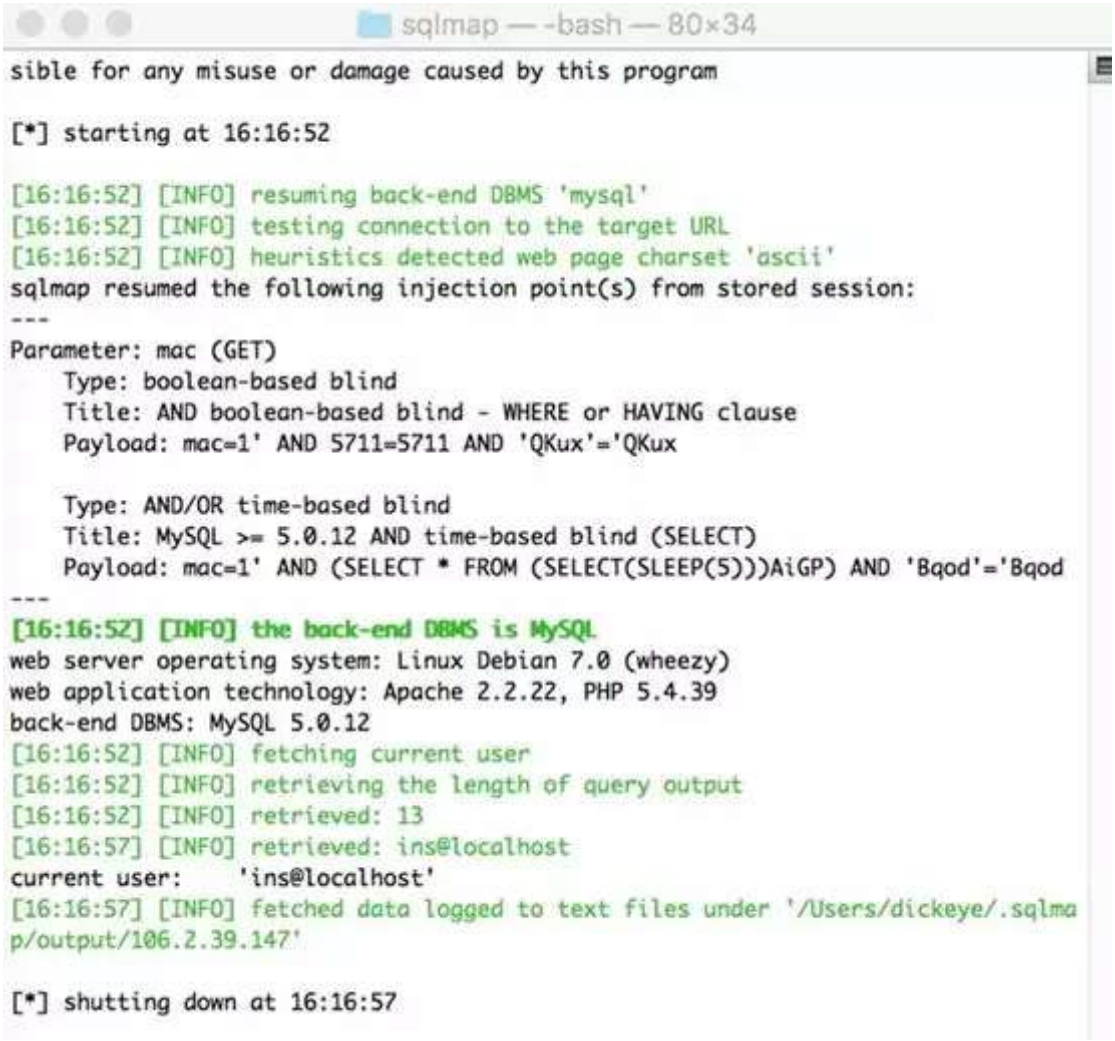


这个漏洞是两年前的了，Uber这个案例虽然路由参数并不复杂，但的确让我开始注重字典的精准收集与合理利用，这也是我分类收集字典的初始原因。
这里还有另外一个案例，当我们碰到站点首页403或者404时，一般都会放弃，但其实这些站点一定程度上往往更容易出现问题，这里是个在我看来有相当一部分概率是此地无银三百两的操作。
因为既然开了web，为何没有内容?
之前我遇到一个IP 106.**.**.147 它的首页是403页面。

这里我通过目录爆破和可执行脚本文件爆破，获取到
http://106.**.**.147/adver/landing.php

可以看到提示参数错误，那么很明显我需要一个参数，通过爆破参数，成功枚举到参数。
完整URL： http://106.**.**.147/adver/landing.php?mac=1
其他案例签了保密协议，所以只能说这些案例的，不好意思啊，各位。
完整URL： http://106.**.**.147/adver/landing.php?mac=1 ，添加单引号，成功发现SQL注入，获得数据库权限，而且这是个游戏业务

成功发现SQL注入，获得数据库权限，而且还是个游戏业务。



大概就是通过不断枚举爆破获得路径，可执行脚本，参数。完成一次漏洞挖掘，通过这种暴力测试我在这几年获得了相当可观的奖金。

因为这是SRC的安全测试，因此这种动静较大留存日志较多的测试我也不需要担心什么问题，但是在真正的攻防对抗时还是不建议大动作。

可以分享的案例就只有这些了，一些其他案例可能得等我保密协议过了，才能脱敏分享。

平顶山：

另外一种就是：IP+端口访问 403 或者 404 既然开了端口，它为什么开呢？多数都是后面有东西的，只是你访问姿势不对。

讲师

对于SRC厂商的一些建议？

京安小妹

硬糖：
规则虽然主要是针对白帽子来设定的，但其实作为规则制定者厂商更应该注意遵守规则。
经常白帽子会碰到以下问题，恰恰是规则制定者违反了自己制定的规则。
1. 当一种类型漏洞被发现存在于一个系统或多个系统中，可能会被综合为一个漏洞或者漏洞等级前后不一致。
2. 同一种漏洞，危害影响相同，不同审核人员认定等级不一致。
3. 有些漏洞已标记为修复，再次测试发现依然有问题，提交后被告知还未被修复，漏洞被取消。
4. 还有一个建议是：厂商不要被SRC漏洞数量和质量迷惑，安全测试和渗透攻防还是有一定区别的，就连参与SRC漏洞挖掘的白帽子都不能完全保证遵守规则不要说真正居心叵测的人了，曾经发现一个白帽子为了提交漏洞换取奖金的构造钓鱼邮件发送木马给厂商员工。一些真正的渗透攻防对抗，红蓝演练也是必须要有的。

互动问答环节：

1. 硬糖师傅在挖洞的时候有没有碰到过一段时间(比如说半个月或者一个月)都没有产出的情况啊，如果有师傅当时是怎么解决的啊？

讲师：

一般碰到瓶颈期的时候，就是什么洞都挖不出来的时候我就会出去玩，然后或者是干别的事情。

2. 硬糖师傅，这么多站点，爆破怎么提高效率呢？

讲师：

还是要针对性的去爆破，不能什么都去爆破。
我主要爆破的对象都是那些新业务，不太成熟的业务或者完全不是业务系统的那些站点，其实也就是经验吧，直觉他有洞，他就必须得有。

本期JSRC 安全小课堂到此结束。更多内容请期待下期安全小课堂。如果还有你希望出现在安全小课堂内容暂时未出现，也欢迎留言告诉我们。
安全小课堂的往期内容开通了自助查询，点击菜单栏进入“安全小课堂”即可浏览。

简历请发送：cv-security@jd.com

微信公众号：jsrc_team

新浪官方微博：京东安全应急响应中心