记一次Mysql日志审计

核心内容：如题

起因：昨天晚上做相册模拟网站写到了凌晨三点半。今早上线的时候发现多了一个用户（其实那个用户名从昨晚夜里十点多就一直存在了），甚至还在我的相册下面留言评论了。问了一圈发现不是周边人做的，于是我有一种危机感，开始查看数据库报错日志。乍一看直接惊了，我这个偏僻小站的上线不过一个月的数据库居然已经扛住了二十五万多次的攻击了。我在惊讶之余随便选取了几个IP上网站上查了一下，结果发现这些攻击ip来自天涯海角、四面八方、星辰大海。简单说就说攻击Ip遍布全球，那很明显如果把所有的ip都去进行认真的溯源和分析很明显是不现实的。那么非常重要的一点就是要筛选出有价值的ip了。

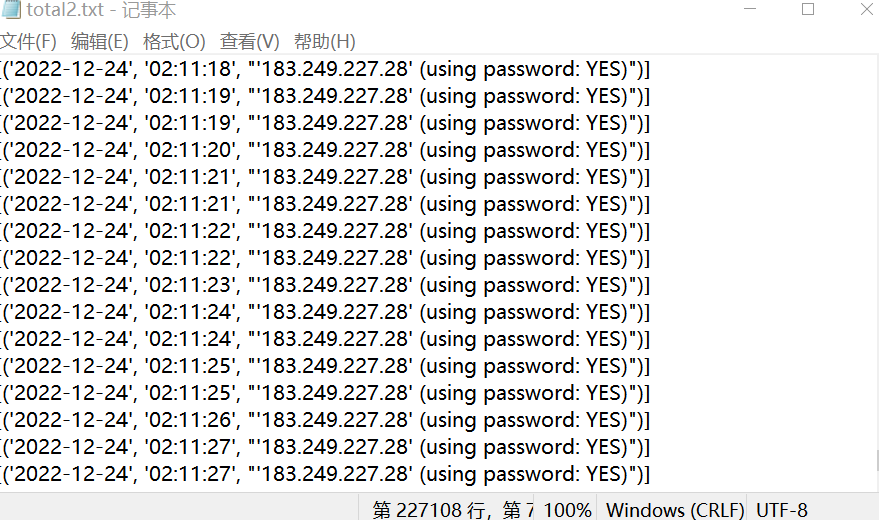
分析：报错日志就26W，要是一条条仔细审计过去，估计不需要用模拟器，我就要重开了。再说了，这其中肯定有大量的爆破记录。爆破的特点是什么？密集，高度密集；还有呢？量大，肯定会比常规的那种一条两条报错长的多。那就简单了。我大致粗略的扫了一下（尤其是近几天），很多记录呈现这样一个规律：“……using password:no…………using password:yes……using password:yes……”稍微数一下就会发现每个ip使用49次以后就会换Ip,然后重复上述过程。我查了几个ip，都是越南的（但不是一个市）的，其实这样的情况很容易想到一个词：代理池。剩下时间差不多的几个ip基本不是越南就是印度，那基本上就是坐实了代理池或者说肉鸡集群的可能性。这样的ip，一个ip就能造成几十甚至几百条报错日志，后续处理的时候很明显要把这种基本一样的给筛选掉。在明确了这个大方向以后就可以开始写处理工具啦

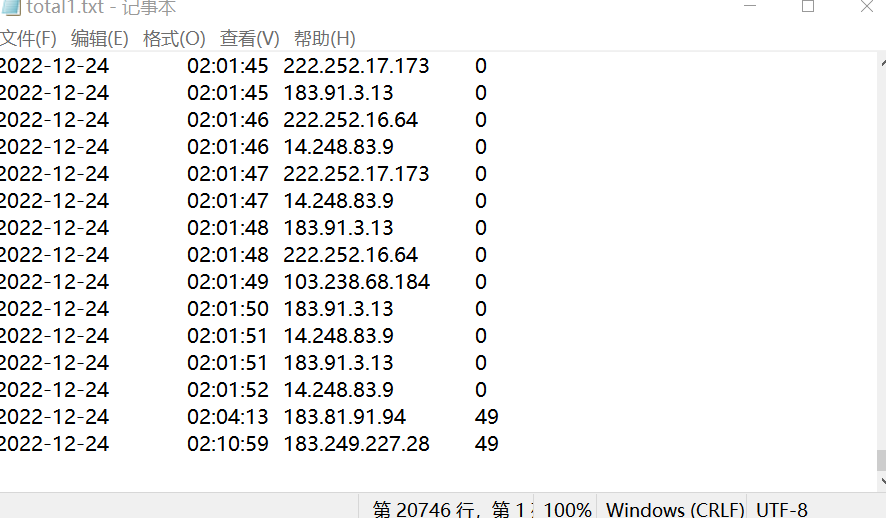
（上个图给大家看看这宏大的日志量，注意Access denied……，这是后续正则的线索）



开发：第一步，从日志中提取出核心信息：日期，时间和Ip。第二步，筛选相同的ip，如果ip相同则把统计次数加一，反之就不理他。我原本非常天真的以为只要这样就行了，结果等拿到效果一看，不是每个ip都能非常顺利的完成49次连续爆破的，会有进来搅局的其他ip。但好在不管多少，5次也好10次也好，好歹已经完成了初步的统计了（此时有两万多条数据），再重复一次类似的统计操作就好了。经过这一次操作，我们把数据压缩到了三千多条，也就是说我的数据库在一个月来经历了三千多人的毒打。

口说无凭，上图为证

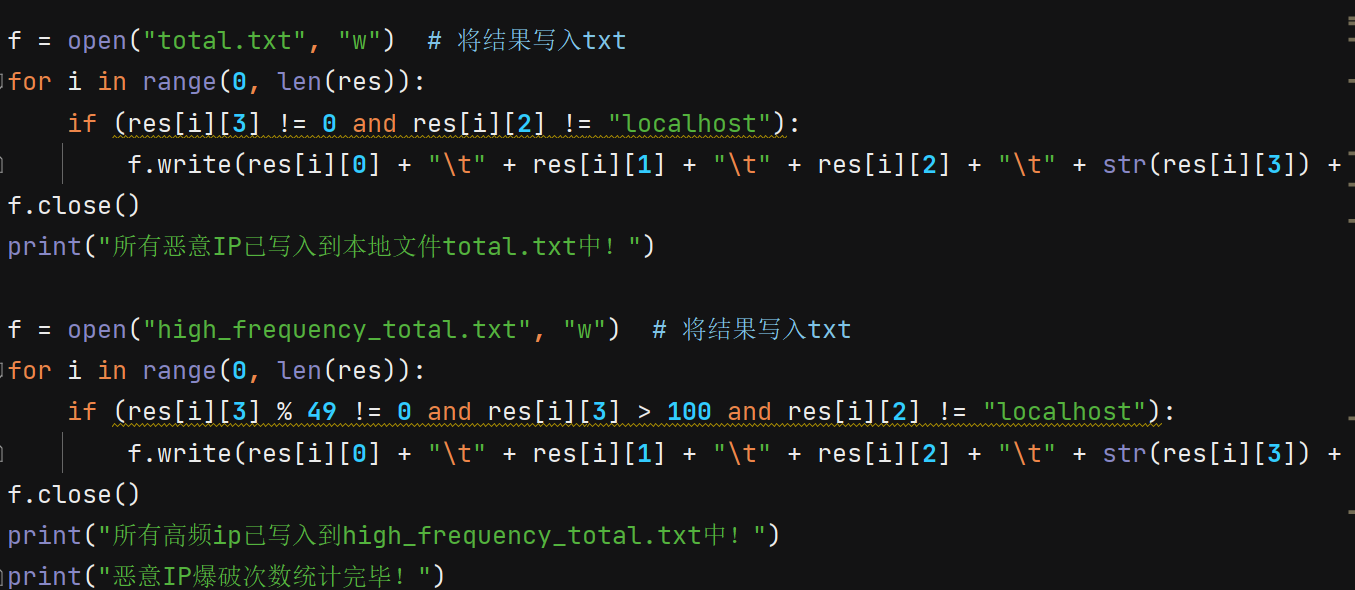






（注意，日志所选取的时间是服务器的实际时间，并非作者所在的中国大陆北京时间，所以即使是最新的日志，时间也会显得滞后一些。但可以很明显看到数据总量从逼近26W减少到了22W减少到了2W减少到了3000，相比于一开始已经算是一大进步了。但作为一个有经验的研判人员怎么可能满足于此。而且3000这个数字虽然相比于二十多万是小的，但是在绝对意义上还是很大的。试想一下：如果国护期间让你重点封禁或者溯源几千个ip现实么？不现实的呀。最好能控制在百来个左右。所以我们还要更进一步）

但是注意一点：好多的49啊。有没有感觉这个数字好熟悉，哎，就是上面分析过程中我说的经典规律。但不仅如此，还有很多的98,147,196……对没错，都是49的倍数。这种很明显就是代理池做的爆破ip自然还是直接扔了算了。还有嘛就是这种出现次数太少的也扔了算了。十以下的直接淘汰，十以上的……我觉得有的Ip能持续猜超过98次应该还是说明需要一定恒心和毅力的。所以综合考虑最后筛选高频率ip的标准为：非49的倍数且超过100。



那么这样筛选出来的有多少个Ip呢？107个。这个数说实话就比较能接受了，就算一个个去ip138上查都来得及。当然了，我们必不可能这么做。

查位置：众所周知，ip138是个用来查看Ip的好东西，不仅可以看自己目前的ip情况更可以说查看目标ip所属的国家省份城市等等。于是乎，我就有了一个想法，即将这样一个需要走网页一个个输入的操作改为自动化。说白了就是爬虫、正则表达式，最后用一个excel输出结果即可

在具体使用过程中有个新问题：就是这个解码和编码问题。众所周知我们的网站的编码常见的utf8但也有例外。比如我们的Ip138，他居然是用gb2312做的。但你以为你只要爬下来一份用gb2312解码的content就万事大吉了？不，并不是这样的。我一开始也是这么以为的直到触发了报错（尤其是那种快成功了来个报错是真的遭不住啊）。在百度后决心使用最强的gb18030（还有补充条款会选择ignore，这里个人建议还是尽量不要用，毕竟用的感觉怪怪的）。

不仅如此，有的IP在我们的正则匹配的过程中可能会涉及到一些特殊情况，需要对数据集进行单独的极个别处理，不过那都是细节的微调了。

在一通折腾后我们终于看到了我们梦寐以求的结果。

