谈谈反垃圾

贴一个两年多以前的文章吧，相信并不完全过时。

《谈谈反垃圾》

由于常年从事用户产品的开发工作，工作中难免遇到过各种各样反垃圾的事，一回生二回熟，在摸爬滚打的对抗中，也摸出了一些门道，此文算是对个人经验的总结，非专业视角的分享。

这里说的垃圾主要针对诸如垃圾评论，机器注册，机器刷接口等等。

反垃圾很重要的两步是：垃圾识别，垃圾处理（包括预防）。

【垃圾识别】

对于判别垃圾，通常有下面一些方法。

1.基于内容的识别

在基于内容的判别上，最直接的是关键词过滤，比如包含“开发票”、“激情视频”这类词的极有可能是垃圾内容，我们通过字符串匹配来判断是否有这类关键词。这里有一个难题，如果是检索一段内容是否包含某一个词还算简单，有很多算法可以实现，比如经典的KMP算法，很多语言内置的字符串查找方法效率也很高。但是，要判断一段内容是否包含一堆关键词中的某一个或某几个，那就有一些难度了，总不能循环一遍所有关键词挨个做匹配吧，所以此法必不可取。

这里推荐两个方法，一个是基于trie树的关键词树，具体有没有开源实现的不清楚，我们使用中是自己基于Memcached改了一个，保留Memcached的简单协议，修改内部逻辑为trie树的查找，简单来说就是将关键词做字节切分，建立一棵trie树，判断一段话中是否包含这些关键词，只需要从根节点向下检索即可。

另外一个方法，是利用贝叶斯算法来进行垃圾概率计算。贝叶斯算法这里就不多展开说了，其原理简单来说就是，收集一组正常内容和一组垃圾内容，用此内容对系统进行训练，让系统能够知道每个词在正常内容中和是在垃圾内容中的概率。做完训练后，再有一段新内容过来，可以直接对其中的词进行综合加权计算，得出整段内容是正常或垃圾的概率。

2.基于特殊内容的识别

上面是纯粹基于随机内容的识别，而实际上我们可能还有一些省力的方法，比如一般的垃圾内容经常会有下面一些特征：带链接（因为要把用户引导到自己的垃圾网站），带图片（为了更醒目），带数字串（比如QQ号，电话号等等），通过这些特征做字符串匹配也是一个好方法，而且就个人经验来看，还比较奏效。其中需要注意的一点就是，上面的链接、数字串这些，通常攻击者都会搞一些变体，不会直接写链接和数字让你判断。比如换成中文数字和字母，你知道，UTF8是很博大精深的。比如：1234567890
这种。所以判断规则上需要多做一些兼容，比如把这种东西先全转成数字来判断。

3.基于请求方式的识别

另外，垃圾毕竟是通过我们暴露给用户的各种接口进来的，而攻击者请求我们接口的方法难免与真实用户有差距。比如说，正常用户会先进入注册页面，再填表单，再提交注册按钮。但是恶意注册程序，很可能是不会先访问你的注册页面的，而是直接请求注册接口（利用这一点我们就可以作文章，比如对用户访问路径进行记录，如果未访问页面就直接请求接口的，判为恶意请求）。另外就是攻击者的http头信息，比如最常见的，UA字段是否是cUrl或者其它非正常浏览器。或者像很多前端团队都有在请求url上添加随机数的习惯，这样本来是为了避免后端缓存，但有些低水平的垃圾请求会原样的每次都用同一个随机数，这就很容易识别他们了。总之，从http请求的层面可识别的东西很多，只要攻击者伪装有一点纰漏，咱们就可以抓到他的尾巴。

4.基于请求主体的识别

如果我们遇到UGC内容的垃圾攻击，那么发起请求的肯定得是一个正常用户（如果是匿名社区请忽略此条）。这时候，内容发送主体的信用级别，就可以转移为对信息质量的判别上来。就像我们都懂的，某些大的平台也会对不同用户执行不同的审核策略（比如都知道的先审后放，还是先放后审），这也需要我们对内容发布主体有充分的信用分级。比如，一个注册24小时内的用户相对一个注册三年发帖无数的用户来说，信用等级就低得多。

5.基于内容载体的识别

目录结构

垃圾内容之所以能形成黑色产业链，通常绝不会是恶作剧玩玩而已，所以跟互联网最传统的广告模式一样，垃圾也希望能够多曝光，多赚点击。那怎么做呢，通常就是选择在用户扎堆的地方去发。比如时下热门的电视剧，热点的新闻事件下面就是垃圾流量的公共厕所了。另外，在一些政治军事内容版块发反动言论，在一些娱乐美女内容版块发成人网站，这些也都是常用的路数。总的来说就是，同样一条内容，在热门版块发布，更有可能会是垃圾内容，需要我们更多的关注。

【垃圾处理】

好吧，上面说了一大堆的方法去给内容和用户评级，以便我们能够对一个用户或者一段发布的内容进行预估，那么，在我们了解了一个用户或者一段内容是否可能是垃圾后，我们脑子里首先蹦出来的可能就是：封杀！但实际处理方法可能不仅封杀一种，下面我们就来探讨一下对垃圾攻击的几种处理方法。

1.制定封杀方法

如果我们已经确切掌握了垃圾流量的规律，比如某一个IP或一组IP，比如同一组参数，比如内容总是包含某网址的变体，那么我们就可以直接大开杀戒，用这些特征直接进行封杀操作。

2.制定审核级别

顺着上面的思路，我们可以对不同的用户和内容施加不同的审核策略，比如是直接放行、先审后放、先放后审还是直接毙掉。我们还可以对用户施加不同的限制策略，比如新注册用户每天只能发3条内容（在审核通过一条后又可以再发）。

3.工作量证明

工作量证明是一个在反垃圾邮件中的方法，最近火得不得了的比特币，工作量证明也是其核心理论支柱之一。通过引入工作量证明方法，我们甚至可以不用对垃圾流量进行判别。只要加一道隐形的门槛，就足以让很多攻击者却步。

举个例子，如果攻击者原来只需要请求一次接口就能够发布一条信息，现在我们需要他在接口请求之前先填一个验证码，他就没那么容易自动狂发内容了。上面这个逻辑大家都能理解，也确实能奏效，但是很抱歉，这样做很伤用户体验，产品经理说不行。

那我们换一种做法，我们让用户在请求前先做大约10w次的md5运算，普通用户的机器偶尔进行一次这样的计算不算什么，但是对攻击者来说，它需要单机发布大量内容，如果我们要求每条内容都需要做10w次md5的话，对的硬件资源是很大的挑战，也是让他放弃对你网站进行攻击的一个方法。

当然，如果我们直接用上面的10w次md5的方法，我们在服务端也需要做同样多的工作才能对传入的接口进行验证，对我们服务器本身也是很大的挑战。所以上面只是一个为了让我们理解的例子，通常的做法是，服务端给定一个随机字符串 s1，客户端需要找到一个数
d，这个数要满足下面条件：这个数追加在这个随机串后同组成一个新串
s2，这个新串进行md5后，前5位都要是0。大家可以想一下，要达到这样的标准，客户端需要不断循环来寻找这个合适的d，而服务端验证却是只需要进行一次md5就可以了。这就是所谓的工作量证明。

4.请求签名

请求签名也是一个省时省力的好方法，前后端约定一种hash算法（最好是自创的），前端对请求内容进行签名，后端验证签名。通过对前端代码进行混淆，让攻击者很难实现你的hash算法。增加他的攻击成本。

5.查出源头

发垃圾内容的攻击者通常都不会用自己机器或服务器IP（要不你就赚到了，直接封IP就行了），而是用手里控制的肉鸡或者扫描来的http代理来做，其实识别肉鸡和代理也比较简单，最直接的方法就是看看开没开着80、8080、3128等端口。这是一般代理的常用接口，另外一般情况下被拿下的肉鸡也都是web接口防范不严造成的。如果是普通http代理，很可能会很有良心的通过x-forward-for，或者x-real-ip等http头信息把源ip传给你，而对于肉鸡找到肉鸡，如果你的黑客水平够，你可以直接也黑上去，看看是哪个IP在控制它，从而查到真实IP。查到攻击者的真实IP后如何处理就看你的了，是联系攻击方和平解决，直接报案还是把攻击者给黑了。那就看个人想法和水平了。

【策略与战略】

上面说了一堆战术层面的东西，下面聊一点战略上的原则。

1.反垃圾是一场成本的较量

反垃圾，其实不是一项技术竞赛，更不像是个人恩怨，更多的是成本较量。
如果你的网站流量大，但防护措施做得不够，那垃圾流量过来是必然的。我们所有的反垃圾策略只

有一个目的，就是增加攻击者的成本，当成本上升到某一阈值时，攻击者会发现在你的网站玩太费劲，投入产出比太低，于是会去找同类型的其它网站。所以就像狮子和羊群一样，只要不是跑得最慢的那一只，就能逃过狮子的爪牙。

2.多数攻击者痛点在IP

无论是用代理，还是肉鸡，攻击者的IP资源总是比较有限的，所以收集到足够多的IP进行封杀，通常能够解决大问题。

3.实而示之虚

上面说反垃圾是一场成本较量，但在我们实际操作中，却要尽量避免真正的较上劲。比如当你发现了恶意请求的规律，如果你选择直接对此规则的请求返回404，那么攻击者也会马上知道它的攻击特征被你发现了，从而迅速进行升级对抗。但是如果你只是让他的操作无实际效果，但还照样返回“注册成功”、“发布成功”，那么攻击者可能会麻痹大意很长时间才会发现。正如《孙子兵法》中说的：“实而示之虚”。实际上在垃圾与反垃圾的较量中，最忌讳的就是无止境的军备竞赛。

4.发现特征之钓鱼策略

有的攻击者很高明，能够将自己的请求伪装得得正常用户一模一样，所有的http头信息，请求参数，都完全仿真。对于这样的攻击者，我们有什么办法抓到他的尾巴呢。这里给大家介绍一种钓鱼策略。首先你修改一下你的网站的前后端逻辑，比如前端增加某一个参数，后端判断没有这个参数请求就会失败，这时候攻击者马上就会发现自己请求失败了，通过对正常请求的抓包，他很快发现你增加了一个参数，那他会跟着进行修改。这时我们让他爽几天。然后偷偷地把这个无关紧要的参数撤掉。这时候，所有正常用户请求中都不会有这个参数了，但是，攻击者不会时时关注我们的请求参数，所以还会在一段时间内，继续加上这个参数请求。这时钓鱼成功，正是我们的好机会，在这段时间内，我们可以尽量收集垃圾的IP，发布账号等信息。等收集到一定程度一起封掉（当然，这里的封掉也不要暴力封掉，而是让看起来没有被封掉）。

总的来说，反

转载地址

作者：iammutex
链接：https://www.zhihu.com/question/20103086/answer/105715337
来源：知乎
