爬虫与反爬虫的对抗

大家好，之前我们所讲的内容都是爬虫相关知识，但是要立志成为一名爬虫工程师，仅仅掌握爬虫是不够的，重要的是对抗反爬，今天我们就来讨论下爬虫与反爬虫的对抗。

在讨论之前，有几个概念需要我们先明白：

****爬虫****：自动获取网站数据的程序，关键是****批量的获取****。（仅获取几条数据严格来说并不是爬虫）

****反爬虫****：使用技术手段防止爬虫程序的方法

****误伤****：反爬技术将普通用户识别为爬虫，从而限制其访问，如果误伤过高，反爬效果再好也不能使用（例如封ip，只会限制ip在某段时间内不能访问）

****成本****：反爬虫需要的人力和机器成本

****拦截****：成功拦截爬虫，一般拦截率越高，误伤率越高

现在，我们来了解下爬虫与反爬虫之间的斗争，这场斗争中有两个角色，一个是爬虫，一个是被爬取的网站，现在游戏开始：

第一场

爬虫：对网站的数据感兴趣，着手分析网络请求，用Scrapy写爬虫爬取网站数据

网站：后台监控发现请求中的User-Agent都是python，直接限制访问（不能封ip）

第二场

爬虫：通过在请求头中传递User-Agent参数模拟浏览器请求

网站：后台监控发现同一ip在某时间段内请求过于频繁，直接限制访问（不能封ip）

第三场

爬虫：在原来的基础上，再通过ip代理向网站发起请求

网站：后台监控发现ip发生变化，但某一时间段内的请求量过大，对服务器造成过大压力，网站中某些数据直接要求登录才能访问（例如：新浪微博）

第四场

爬虫：注册账号，每次请求携带cookie或者token

网站：健全账号体系，用户只能访问好友信息，非好友信息不能访问，或只能访问部分信息（例如：Facebook）

第五场

爬虫：注册多个账号，多个账号联合爬取，设置程序，每个账号每天固定添加好友，但是养号是个相对耗时的过程

网站：后台监控发现请求过于频繁，为避免服务器压力过大，进一步加剧ip访问频率限制

第六场

爬虫：模拟人去请求，限制请求速度

网站：后台监控到ip访问时间段过长，例如一天24小时都在有规律地请求，弹出验证码

第七场

爬虫：通过各种手段识别验证码（建议使用云打码，自己去研究识别验证码太费时，时效性也太差）

网站：增加动态网站，数据通过js动态加载，增加网络分析复杂度，或者发现大量请求时只请求html，而不请求image和css以及js，亦或者直接返回假数据给爬虫

第八场

爬虫：通过selenium和phantomjs（无界面浏览器）完全模拟浏览器操作

网站：还有完没完了，成本太高了，我先暂时放弃吧，等我有钱了，或者等我找到了新技术再来解决你

但是，游戏并没有结束，爬虫与反爬虫一直在对抗，一直在互相进步。

正因为有对抗，技术才会飞跃发展。

有些网站不是做不出反爬技术，而是他们需要衡量所付出的成本与获得的利益是否匹配，没有两全其美的方案，只是在中间寻求一个平衡而已。

最后，祝大家在这场游戏中玩得愉快，切莫中途掉线，大家加油。

三十一  阅