# API接口深度发现的动态爬虫实现5 - API贪心发现

原创 扫到漏洞的 李姐姐的扫描器 2025年05月23日 12:48 北京

API爬虫的复杂度到底在哪里?

某种程度上，笔者认为，是对API接口的贪心，不肯放过任何一个新出现的modal / dialog，不肯放过任何一个可能的交互请求。

笔者尝试了大量方法，试图能够将所有的页面交互顺利走通，尽可能少产生API的遗漏

- 方法1：触发执行页面上已绑定的所有事件函数
- 方法2：**解决多数爬虫发现不了API的关键问题，持续检查页面上是否有新的事件绑定，将剩余的函数也继续触发执行**
- 方法3：自动填充表单并提交，这是基本方法
- 方法4：**通过CSS找出页面上所有的pointer指针的元素，再次对其进行交互点击，避免上述2步产生的少量遗漏。 前2步会漏么？ 答案是会**

  Pointer指针如下图所示，笔者把鼠标悬停在微信公众号网页中的不同位置，就能找到哪些区域是可以点击交互的，这些区域都是pointer指针


- 方法5：监视弹出来的dialog/modal，在该对话框中继续交互尝试发现API
- 方法6：通过正则方式从JS中提取API接口、拼接API接口，使用字典进行接口fuzz

笔者观测到一般工具扫不到接口，常见的原因：

1) API Base URL错误，没有能力正确提取到Base URL

2) Auth Header缺失，没有能力提取到正确的auther header用于接口请求

3）Cookie/ Referer错误等

4）无法进行复杂交互，甚至根本没有headless chrome

5）地狱级难度的参数分析获取，绝大部分工具观察到没有对应的能力

为了解决API参数的这个问题，笔者对Javascript静态分析和分片动态执行做了一些尝试，有一定进展，但也仅限于在1-2层调用下，写法较简单的情形，有概率提取成功。实际上因为打包、封装、继承等原因，函数的调用乱到我无力分析追踪。

肯定有人要扯淡，说大模型在这个分析场景下很牛啊，我想说的是，一个大的JS就是几十万token输入，它最后它能帮你识别出很少几个API接口，还要让你等几十秒，接口数只是个零头，参数也不全部准确。丢整个JS给大模型分析提取API接口的用法，紧急的状况、token能免费消耗的，可以考虑。

第一眼看，真惊艳，丢个几百kb的JS，很快就分析出来了。 仔细一对账，垃圾，又错又少。

## 写本地文件减少HTTP请求

笔者尝试尽可能地去重，减少重复HTTP请求。但因为JS事件的传播机制，同一个事件会在多个父子元素之间重复执行。

笔者的方法，是根据URL+参数进行本地存储，仅第一次请求会产生服务器请求。其余请求则从本地文件读取后返回给浏览器。加速页面上的交互执行。

已存储的JS + API 响应包，则可以用于扫描分析。

## 不放过任何一个对话框

不放过modal框就是不放过可能的API接口。

很多时候我在写代码的时候，纠结的是要不要等（也即sleep/ wait）？

不等，怕错过。

等，扫描太慢了！

等多久？10ms ? 20ms？ 100ms? 甚至1s?

在触发执行的时候可以不等，一股脑全丢过去触发了。

但一旦在这个过程中监视到了dialog，必须有基本的机制能回溯到这个dialog并对其进行深度交互。

## 最后

工具还没开源，扯了半天。 总之，程序的事情，要以效果说话。

No more speech, show me the code