

# 防爬功能调研

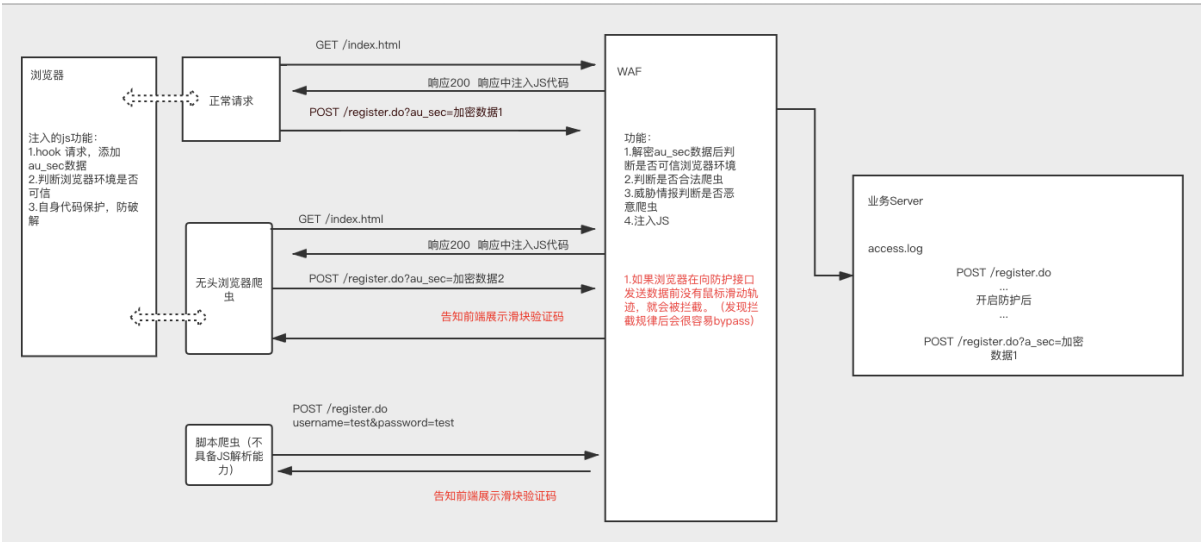
- 阿里云WAF
  - 1.Bot防护重要功能
  - 2.数据流
  - 3.威胁情报测试
- 华为云WAF
  - 防爬原理
- ShareWAF
  - 防爬原理
  - 行为分析的原理
- 可以模拟浏览器的爬虫技术
  - 已有技术
  - 代表工具
  - 无头浏览器攻防
- 浏览器指纹技术
- 总结

## 阿里云WAF

### 1.Bot防护重要功能

- app防护
- 合法爬虫
- 爬虫威胁情报规则
- 数据风控

### 2.数据流



- 浏览器爬虫带上au\_sec值, 但是会被拦截下来?

### 3.威胁情报测试

结论:

- 情报库可能恶意ip库不全,可以使用被微步在线、360识别恶意的机器做代理访问防护接口。

# 华为云WAF

## 防爬原理

### 通过配置反爬虫防护策略阻止爬虫攻击

更新时间：2020/11/20 GMT+08:00

查看PDF 分享

#### 本文导读

- 前提条件
  - 开启Robot检测（识别User-Agent）
  - 开启网站反爬虫（检查浏览器合法性）
  - 配置CC攻击防护（限制访问频率）

网络爬虫为网络信息收集与查询提供了极大的便利，但同时也对网络安全产生以下负面影响：

- 网络爬虫会根据特定策略尽可能多的“爬过”网站中的高价值信息，占用服务器带宽，增加服务器的负载
- 恶意用户利用网络爬虫对Web服务发动DoS攻击，可能使Web服务资源耗尽而不能提供正常服务
- 恶意用户利用网络爬虫抓取各种敏感信息，造成网站的核心数据被窃取，损害企业经济利益

Web应用防火墙可以通过Robot检测（识别User-Agent）、网站反爬虫（检查浏览器合法性）和CC攻击防护（限制访问频率）三个反爬虫策略，全方位帮您解决业务网站遭受的爬虫问题。

和阿里云很不同

- 阿里云不基于user-agent判断
- 没有防护接口的概念，对网站整体做防护

## ShareWAF

## 防爬原理

```
anti_spider.js x 101 sharewaf.js x daemon.js x auto_dynamic_token.js x no_copy.js x anti_xss.js x anti_

1  /*
2  * 反爬虫
3  * 实现方法：
4  * 清空href内容，使链接地址不可见，点击时再还原，可链接生效，且操作起来跟之前无差别
5  */
6
7  /*
8  * 如果页面中原本有window.onload事件，则先进行调用，以免影响原有的代码功能，下同
9  */
10 var pre_window_load=window.onload;
11 if (pre_window_load!=undefined){
12     pre_window_load();
13 }
```

## 行为分析的原理

```
anti_spider.js × action_analyse.js × 101 sharewaf.js × daemon.js × auto_dynamic_token.js × no_copy.js ×
80
81 //向ShareWAF后端提交行为记录
82 function report_to_sharewaf(){
83     //目标URL
84     var sharewaf_url = window.location.protocol + "://" + window.location.host;
85     sharewaf_url = sharewaf_url + ":8080" + "/action/"
86
87     //异步通信
88     $sharewaf.ajax({
89
90         type:'post',
91         url:sharewaf_url,
92         data:{
93             host : window.location.host,
94             url: window.location.href,
95             referrer: document.referrer,
96             keydown: page_keydown,
97             mousedown: page_mousedown,
98             mousemove: page_mousemove,
99             browserid : BrowserWAF_BrowserID,
100             browserid_detail : BrowserWAF_BrowserID_Details
101         },
102
103         //错误
104         error:function(err){
105             console.error("BrowserWAF Error:", err)
106         }
107     });
108 }
```

## 可以模拟浏览器的爬虫技术

### 已有技术

- QtWebkit : PhantomJS ( 废弃 )
- NightmareJS
- Electron
- puppeteer ( CDP协议操作chrome )

### 代表工具

- crawlergo <https://github.com/0Kee-Team/crawlergo/> 漏扫爬虫

### 无头浏览器攻防

- 怎么识别无头浏览器爬 [浏览器爬虫识别demo.html](#)
- crawlergo如何伪装成正常浏览器 [crawlergo爬虫逻辑.js](#)

# 浏览器指纹技术

作用：识别跟踪每一个浏览器

如何在反爬虫场景中应用尚不清楚

## 总结

识别爬虫的方式：

- 判断浏览器环境是否可信
- 威胁情报
- 访问频率
- 是否有爬虫行为，比如
  - 爬取不可见的a标签
  - 触发隐藏元素的事件
  - 没有点击鼠标、键盘、鼠标滑动等行为轨迹