Crawler

2019

第三届爬虫大会报告

爬虫技术的昨天，今天和明天

2019.07.28

背景

Introduction

- 近些年来，爬虫技术的发展突飞猛进
- 爬虫技术逐渐成为一套完整的系统性工程技术
- 涉及的知识面广，平台多，手段越来越多样化
- 对抗性日益显著

提纲

- 回顾一下最近几年爬虫技术的发展线路
- 介绍当今几种主流的爬虫技术
- 前沿的爬虫技术的发展趋势和方向

技术回顾

01 浏览器: 第一阶段

从无到有的过程

- 通过自动化的方法，获取网页信息
- wget, curl

提高抓取效率

- 分布式爬虫框架（scrapy）
- 分布式存储（redis）
- 大量的IP

01 浏览器: 第二阶段

人机验证的出现

- 验证码
- 滑块
- 点选

浏览器自动化工具

- Selenium
- PhantomJS, Chrome Headless, Puppeteer, etc.

01 浏览器: 第三阶段

Js混淆技术（保护请求、加密数据）

- Uglify-js（传统混淆）
- Jscrambler（商用混淆）
- JSF*ck（另类混淆）

破解签名

- PyV8
- NodeJS

01 浏览器: 第三阶段

UglifyJS 3: Online JavaScript minifier

```
/*
 * JavaScript MD5
 * https://github.com/blueimp/JavaScript-MD5
 *
 * Copyright 2011, Sebastian Tschan
 * https://blueimp.net
 *
 * Licensed under the MIT license:
 * https://opensource.org/licenses/MIT
 *
 * Based on
 * A JavaScript implementation of the RSA Data Security, Inc. MD5 Message
 * Digest Algorithm, as defined in RFC 1321.
 * Version 2.2 Copyright (C) Paul Johnston 1999 - 2009
 * Other contributors: Greg Holt, Andrew Kepert, Ydnar, Lostinet
 * Distributed under the BSD License
 * See http://pajhome.org.uk/crypt/md5 for more info.
 */

/* global define */

/* eslint-disable strict */

;(function($) {
  'use strict'

  /**
   * Add integers, wrapping at 2^32.
   * This uses 16-bit operations internally to work around bugs in interpreters.
   *
   * @param {number} x First integer
   * @param {number} y Second integer
   * @returns {number} Sum
   */
  function safeAdd(x, y) {
    var lsw = (x & 0xffff) + (y & 0xffff)
    var msw = (x >> 16) + (y >> 16) + (lsw >> 16)
    return (msw << 16) | (lsw & 0xffff)
  }

  /**
   * Bitwise rotate a 32-bit number to the left.
   *
```

The minified output (3658 bytes, saved 68.35%)

```
!function(n){"use strict";function t(n,t){var r=(65535&n)+(65535&t);return(n>>16)+(t>>16)+(r>>16)
<<16|65535&r}function r(n,r,e,o,u,c){return t((f=t(t(r,n),t(o,c)))<<(i=u)|f>>>32-i,e);var f,i}function
e(n,t,e,o,u,c,f){return r(t&e|~t&o,n,t,u,c,f)}function o(n,t,e,o,u,c,f){return r(t&o|e&~o,n,t,u,c,f)}function
u(n,t,e,o,u,c,f){return r(t^e^o,n,t,u,c,f)}function c(n,t,e,o,u,c,f){return r(e^(t|~o),n,t,u,c,f)}function
f(n,r){var f,i,a,d,h;n[r>>5]|=128<<r%32,n[14+(r+64>>>9<<4)]=r;var
l=1732584193,g=-271733879,v=-1732584194,m=271733878;for(f=0;f<n.length;f+=16)i=l,a=g,d=v,h=m,g=c(g=c(g=c(g=c(
g=u(g=u(g=u(g=u(g=o(g=o(g=o(g=o(g=e(g=e(g=e(g=e(g=e(g,v=e(v,m=e(m,l=e(l,g,v,m,n[f],7,-680876936),g,v,n[f+1],12,-3
89564586),l,g,n[f+2],17,606105819),m,l,n[f+3],22,-1044525330),v=e(v,m=e(m,l=e(l,g,v,m,n[f+4],7,-176418897),g,
v,n[f+5],12,1200080426),l,g,n[f+6],17,-1473231341),m,l,n[f+7],22,-45705983),v=e(v,m=e(m,l=e(l,g,v,m,n[f+8],7,
1770035416),g,v,n[f+9],12,-1958414417),l,g,n[f+10],17,-42063),m,l,n[f+11],22,-1990404162),v=e(v,m=e(m,l=e(l,g
,v,m,n[f+12],7,1804603682),g,v,n[f+13],12,-40341101),l,g,n[f+14],17,-1502002290),m,l,n[f+15],22,1236535329),v
=o(v,m=o(m,l=o(l,g,v,m,n[f+1],5,-165796510),g,v,n[f+6],9,-1069501632),l,g,n[f+11],14,643717713),m,l,n[f],20,-
373897302),v=o(v,m=o(m,l=o(l,g,v,m,n[f+5],5,-701558691),g,v,n[f+10],9,38016083),l,g,n[f+15],14,-660478335),m,
l,n[f+4],20,-405537848),v=o(v,m=o(m,l=o(l,g,v,m,n[f+9],5,568446438),g,v,n[f+14],9,-1019803690),l,g,n[f+3],14,
-187363961),m,l,n[f+8],20,1163531501),v=o(v,m=o(m,l=o(l,g,v,m,n[f+13],5,-1444681467),g,v,n[f+2],9,-51403784),
l,g,n[f+7],14,1735328473),m,l,n[f+12],20,-1926607734),v=u(v,m=u(m,l=u(l,g,v,m,n[f+5],4,-378558),g,v,n[f+8],11
,-2022574463),l,g,n[f+11],16,1839030562),m,l,n[f+14],23,-35309556),v=u(v,m=u(m,l=u(l,g,v,m,n[f+1],4,-15309920
60),g,v,n[f+4],11,1272893353),l,g,n[f+7],16,-155497632),m,l,n[f+10],23,-1094730640),v=u(v,m=u(m,l=u(l,g,v,m,n
[f+13],4,681279174),g,v,n[f],11,-358537222),l,g,n[f+3],16,-722521979),m,l,n[f+6],23,76029189),v=u(v,m=u(m,l=u
(l,g,v,m,n[f+9],4,-640364487),g,v,n[f+12],11,-421815835),l,g,n[f+15],16,530742520),m,l,n[f+2],23,-995338651),
v=c(v,m=c(m,l=c(l,g,v,m,n[f],6,-198630844),g,v,n[f+7],10,1126891415),l,g,n[f+14],15,-1416354905),m,l,n[f+5],2
1,-57434055),v=c(v,m=c(m,l=c(l,g,v,m,n[f+12],6,1700485571),g,v,n[f+3],10,-1894986606),l,g,n[f+10],15,-1051523
),m,l,n[f+1],21,-2054922799),v=c(v,m=c(m,l=c(l,g,v,m,n[f+8],6,1873313359),g,v,n[f+15],10,-30611744),l,g,n[f+6
],15,-1560198380),m,l,n[f+13],21,1309151649),v=c(v,m=c(m,l=c(l,g,v,m,n[f+4],6,-145523070),g,v,n[f+11],10,-112
0210379),l,g,n[f+2],15,718787259),m,l,n[f+9],21,-343485551),l=t(l,i),g=t(g,a),v=t(v,d),m=t(m,h);return[l,g,v,
m]}function i(n){var t,r="",e=32*n.length;for(t=0;t<e;t+=8)r+=String.fromCharCode(n[t>>5]>>>t%32&255);return
r}function a(n){var t,r=[];for(r[(n.length>>2)-1]=void 0,t=0;t<r.length;t+=1)r[t]=0;var
e=8*n.length;for(t=0;t<e;t+=8)r[t>>5]|=(255&n.charCodeAt(t/8))<<t%32;return r}function d(n){var
t,r,e="";for(r=0;r<n.length;r+=1)t=n.charCodeAt(r),e+="0123456789abcdef".charAt(t>>>4&15)+"0123456789abcdef".
charAt(15&t);return e}function h(n){return unescape(encodeURIComponent(n))}function l(n){return function(n)
{return i(f(a(n),8*n.length))}(h(n))}function g(n,t){return function(n,t){var r,e,o=a(n),u=[],c=
[];for(u[15]=c[15]=void 0,o.length>16&&
(o=f(o,8*n.length)),r=0;r<16;r+=1)u[r]=909522486^o[r],c[r]=1549556828^o[r];return
e=f(u.concat(a(t)),512+8*t.length),i(f(c.concat(e),640))}(h(n),h(t))}function v(n,t,r){return t?r?
g(t,n):d(g(t,n)):r?l(n):d(l(n))}"function"==typeof define&&define.amd?define(function(){return
v}):"object"==typeof module&&module.exports?module.exports=v:n.md5=v}(this);
```

01 浏览器: 第三阶段

common.js

SOURCE CODE

```
1  //==================================================================
   ========
2  // minimalist DOM helpers
3  //==================================================================
   ========
4  (function () {
5    window.Dom = {
6      get:  function(id)                    { return ((id instanceof
   HTMLElement) || (id === document)) ? id : document.getElementById
   (id); },
7      set:  function(id, html)              { Dom.get(id).innerHTML
   = html;                          },
8      on:   function(ele, type, fn, capture) { Dom.get(ele).addEventL
   istener(type, fn, capture);    },
9      un:   function(ele, type, fn, capture) { Dom.get(ele).removeEve
   ntListener(type, fn, capture); },
10     show: function(ele, type)             { Dom.get(ele).style
   .display = (type || 'block');      },
11     blur: function(ev)                    { ev.target.blur();
                                    },
12
13     addClassName:    function(ele, name)     { Dom.toggleClassName
   (ele, name, true);  },
14     removeClassName: function(ele, name)     { Dom.toggleClassName
   (ele, name, false); },
15     toggleClassName: function(ele, name, on) {
16       ele = Dom.get(ele);
17       var classes = ele.className.split(' ');
18       var n = classes.indexOf(name);
```

PROTECTED CODE

```
1  v411(typeof window === typeof {} ? window : typeof global ===
   typeof {} ? global : this);
2  a1nn(typeof window === typeof {} ? window : typeof global ===
   typeof {} ? global : this);
3  T1II(typeof window === typeof {} ? window : typeof global ===
   typeof {} ? global : this);
4  x7ii.h1WW = h1WW;
5  U0BB(typeof window === typeof {} ? window : typeof global ===
   typeof {} ? global : this);
6  x7ii.W0T = function () {
7    var l8T = 2;
8    for (; l8T !== 1;) {
9      switch (l8T) {
10     case 2:
11       return {
12         n5B: function (I5B) {
13           var u0T = 2;
14           for (; u0T !== 10;) {
15             switch (u0T) {
16             case 8:
17               x5B += p1BB.a1BB(A5B.g0BB(S5B) ^ I5B.g0BB(z5B));
18               u0T = 7;
19               break;
20             case 3:
21               u0T = z5B === I5B.length ? 9 : 8;
22               break;
23             case 4:
24               u0T = S5B < A5B.length ? 3 : 6;
25               break;
```

01 浏览器: 第三阶段

```
> [][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])
  [+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]][([][(![]+[])[+[]]+([!
  []]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+
  []+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+
  []]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+
  [])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!
  []+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+
  ([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+
  [])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+
  (!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])
  [!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])
  [+!+[]+[+[]]]+(!![]+[])[+!+[]]]((![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!![]+
  [])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]+(![]+[][(![]+[])[+[]]+
  ([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])
  [!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[+[]]]+[+!+[]]+(!![]+[][(!
  []+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+
  []]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[+[]]])()

> alert(1)
```

数据的变化

1 数据的载体变化

- PC的流量下降，移动端流量增加
- 有些数据只在手机APP上有

2 数据的类型转变

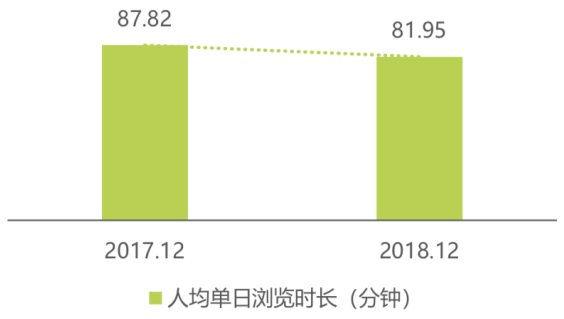
- 图片、短视频、RichText数据

3 数据的通讯协议多样化

- SPDY、Protobuf、私有TCP协议

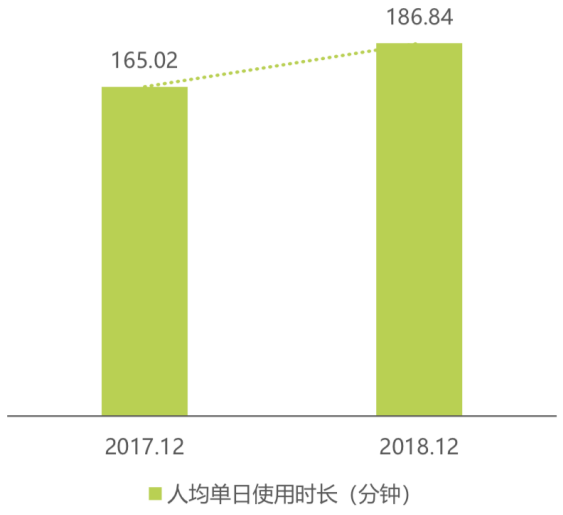
数据的变化

iUserTracker-2017年6月-2018年12月 中国PC网民人均单日上网时长



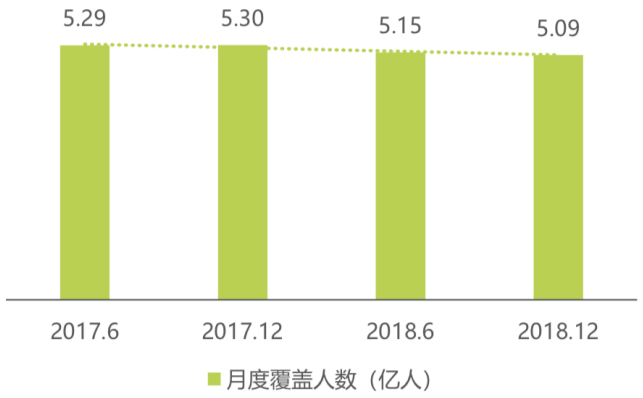
来源：iUserTracker. 家庭办公版 2018.12，基于对40万名家庭及办公（不含公共上网地点）样本网络行为的长期监测数据获得。

mUserTracker-2017年6月-2018年12月 中国移动网民人均单日上网时长



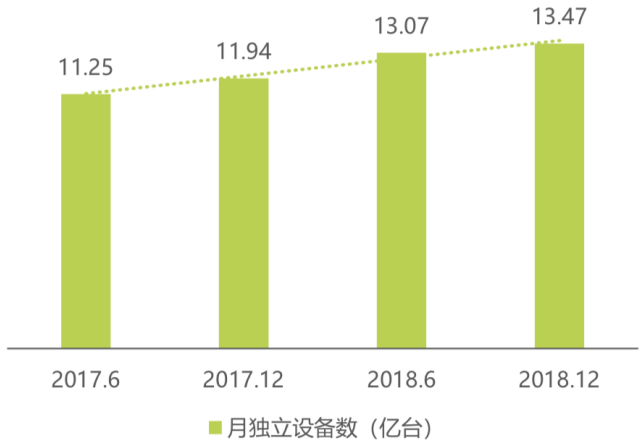
来源： mUserTracker.2018.12，基于日均400万手机、平板移动设备软件监测数据，与超过1亿移动设备的通讯监测数据，联合计算研究获得。

iUserTracker-2017年6月-2018年12月 中国PC互联网用户规模



来源：iUserTracker. 家庭办公版 2018.12，基于对40万名家庭及办公（不含公共上网地点）样本网络行为的长期监测数据获得。

mUserTracker-2017年6月-2018年12月 中国移动互联网用户规模



来源： mUserTracker.2018.12，基于日均400万手机、平板移动设备软件监测数据，与超过1亿移动设备的通讯监测数据，联合计算研究获得。

数据的变化

数据的类型转变

- 文本、资讯类数据比重下降
- 多媒体数据井喷
  - 图片、短视频数据
- 生活、娱乐类数据快速增长
  - 电商类数据
  - 社交媒体数据

数据的变化

数据通讯协议变化

- HTTP/HTTPS
- SPDY
- 数据交换的多样性
  - JSON
  - Protobuf
  - 私有TCP协议

新的挑战

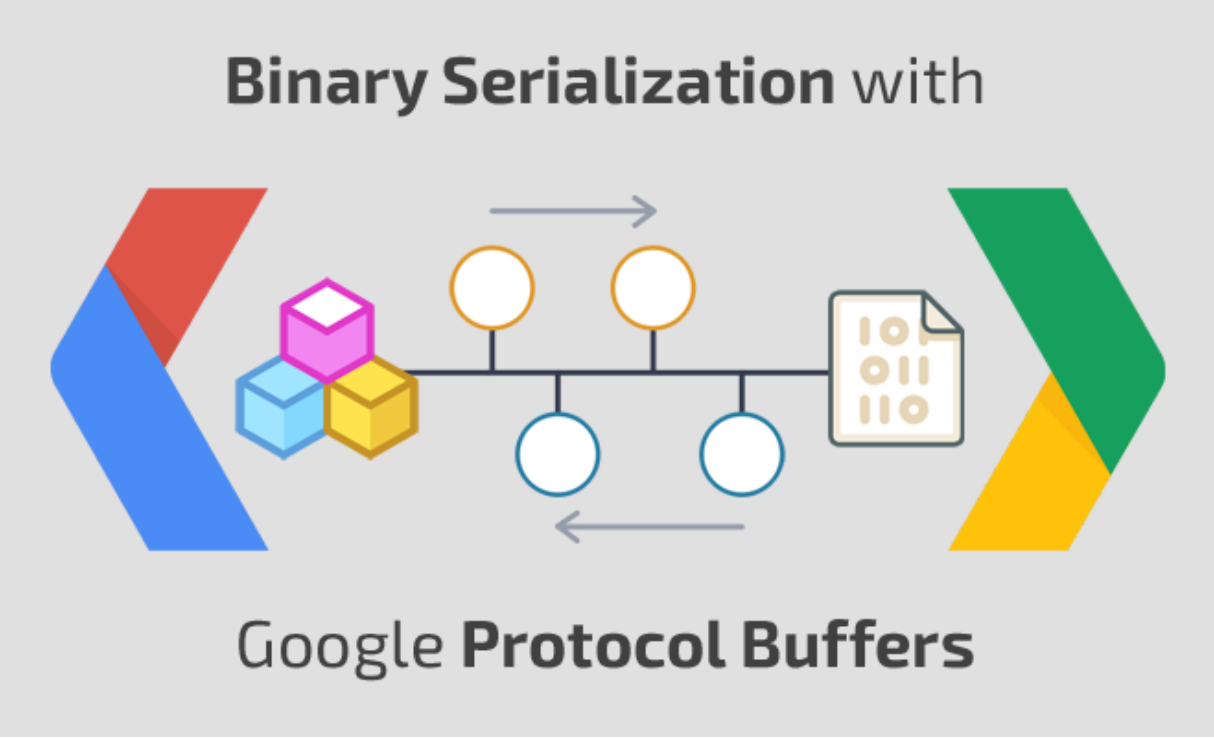
从浏览器到移动端APP

- 浏览器可以看成是一款“APP”
- 各个厂商定制的自己的APP
  - 实现方法差异很大
  - 通信的模式各不相同
  - 反爬策略都不一样
- 系统更复杂，安全指数更高

移动端: 第一阶段

UI层与数据层的分离

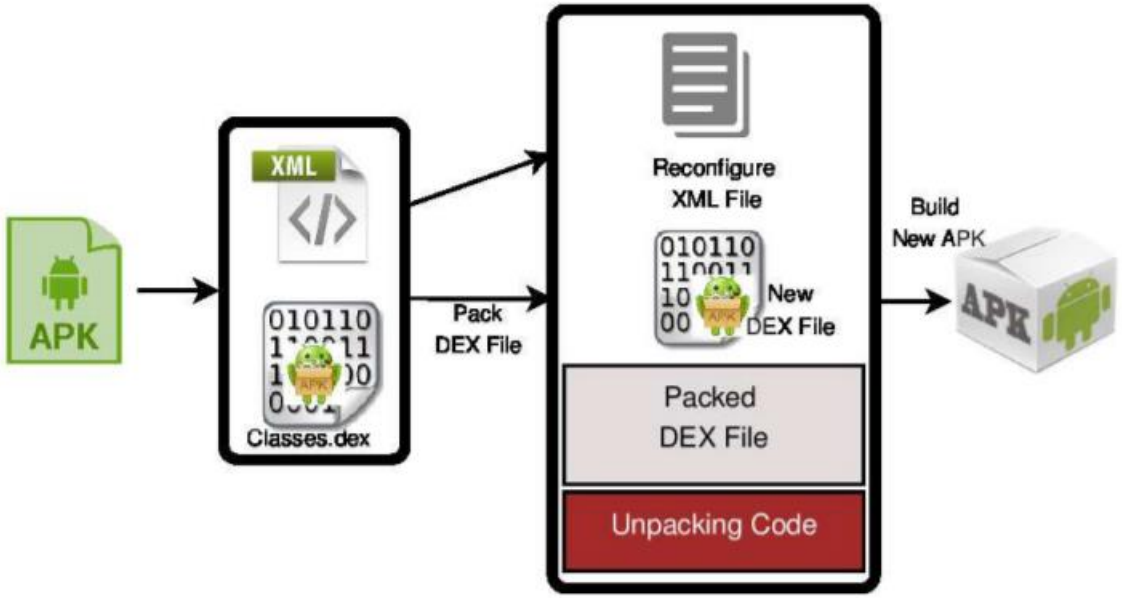
- 序列化协议Protobuf



移动端: 第二阶段

安卓Java层混淆与加壳

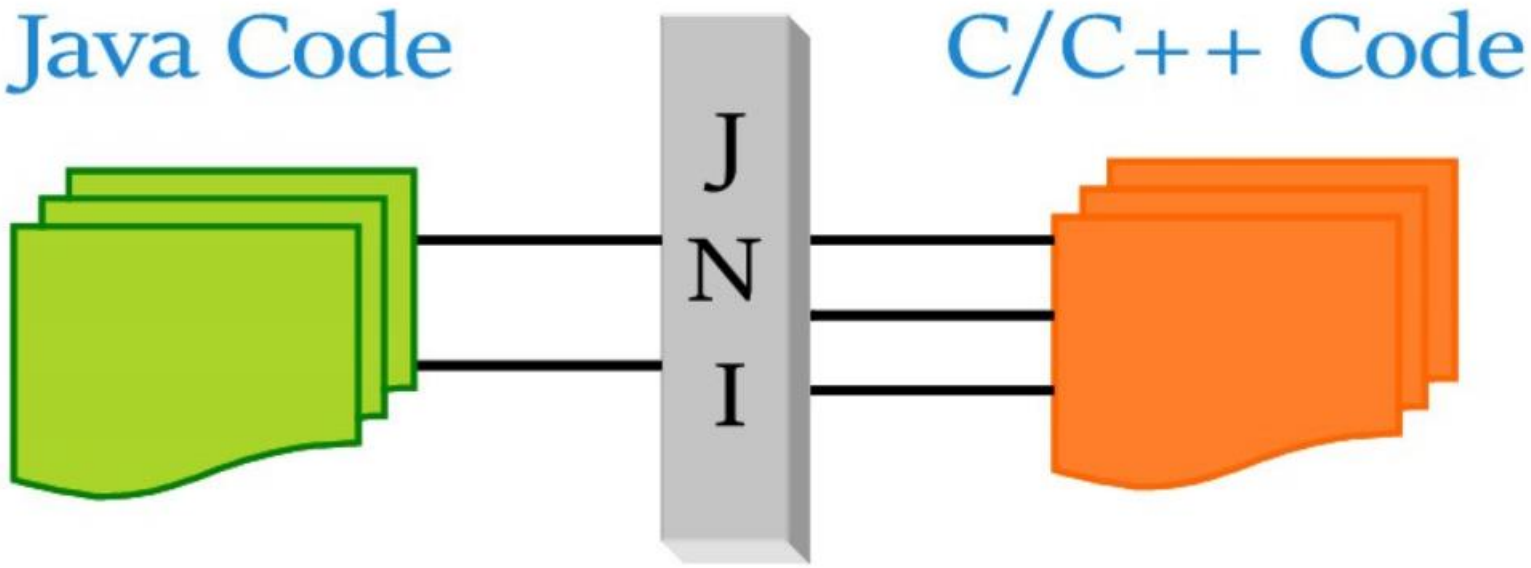
- 代码反混淆
- 动态脱壳机



02 移动端: 第三阶段

关键代码逐渐下沉到Native层

- 逆向代码



03 小结

1. PC浏览器上的数据保护越来越强
2. 数据从文本到图片、视频的转变
3. 数据从PC浏览器，到移动端迁移的趋势
4. 移动端的关键代码往native层下沉

爬虫技术分享

三个主题

CONTENT

01

JS签名破解

02

Frida

03

手机机房

JS签名破解

JS签名破解

处理复杂JS签名的一种思路

- 直接分析js代码还原程序逻辑的难度太大
- 可以使用Nodejs或者Chrome加载原生js代码
- 调用js中的关键函数获得计算结果
- 以我司的企业查询网站为例
  - http://www.qymss.com
  - 站点请求参数中使用了复杂js签名函数
  - 如果静态分析还原签名代码，难度很大。

JS签名破解

请求中的签名参数

Group by frame | Preserve log | Dis

Hide data URLs

Font Doc WS Manifest Other

600000 ms 800000 ms 1000000 ms

× Headers Preview Response Timing

Status Code: 200 OK

Remote Address: 43.241.216.100:8080

Referrer Policy: no-referrer-when-downgrade

▾ **Response Headers**

Access-Control-Allow-Origin: *

Content-Encoding: none

Content-Length: 0

Content-Type: text/html;charset=UTF-8

Server: Bayou Tech Web Srv 1.0

X-Frame-Options: SAMEORIGIN

▾ **Request Headers**

⚠ **Provisional headers are shown**

Accept: application/json, text/javascript,
0.01

Origin: http://qymss.pullwave.com

Referer: http://qymss.pullwave.com/search.h
y=boy

User-Agent: Mozilla/5.0 (Macintosh; Intel M
10_14_5) AppleWebKit/537.36 (KHTML, like
hrome/75.0.3770.142 Safari/537.36

▾ **Query String Parameters** view source view UR

page: 0

name: boy

sign: d446588017422e95c81b20a40bb4371d

ts: 1564151713458

boy 开始搜索

商务合作请联系 cl_workmail@163.com 或微博私信 @梁斌penny

网站备案号：苏ICP备19027551号-1

Icons made by surang from www.flaticon.com is licensed by CC 3.0 BY

Photo by Nick Chung from Burst

JS签名破解

被混淆的关键js代码

Filter | Hide data URLs | All | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

100000 ms | 200000 ms | 300000 ms | 400000 ms | 500000 ms | 600000 ms | 700000 ms | 800000 ms | 900000 ms | 1000000 ms

Name | × Headers | Preview | Response | Timing

- search.html?key=boy
- material-icons.min.css
- bootstrap-material-design....
- index.css
- jquery.min.js
- popper.min.js
- bootstrap-material-design....
- mithril.min.js
- 1561782120976.js
- hm.js?983387e8d4dcd2a85...
- MaterialIcons-Regular.woff2
- blob:http://qymss.pullwave....
- search.js?t=1564151712646
- hm.gif?cc=1&ck=1&cl=24-b...
- hm.gif?cc=1&ck=1&cl=24-b...
- blob:http://qymss.pullwave....
- index.php?page=0&name=...
- blob:http://qymss.pullwave....
- MaterialIcons-Regular.woff2
- data:image/svg+xml;...
- MaterialIcons-Regular.woff2
- MaterialIcons-Regular.woff2
- MaterialIcons-Regular.woff2

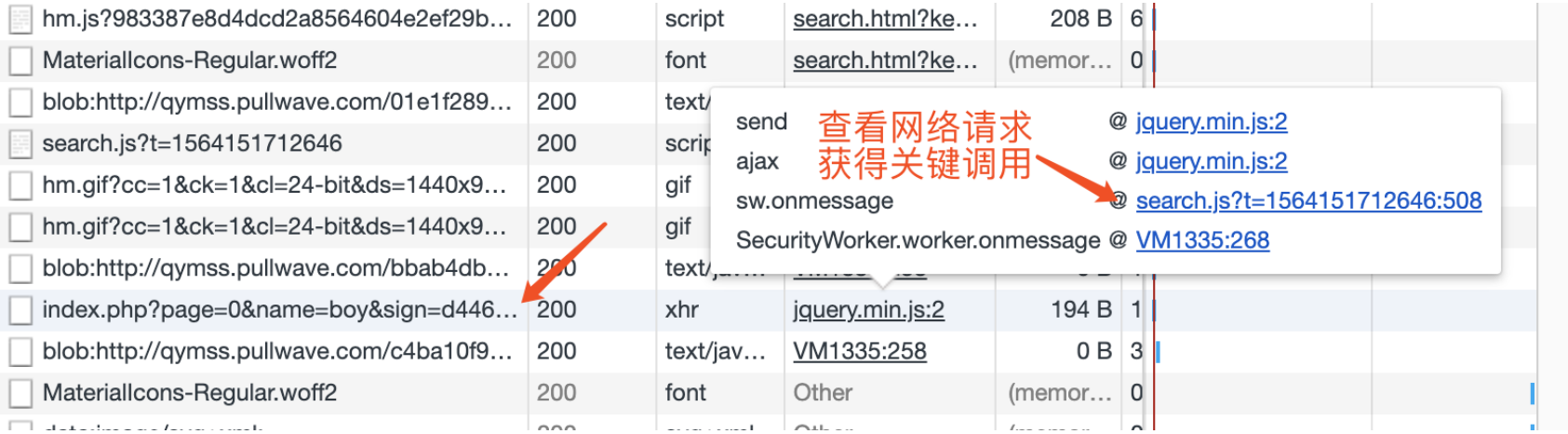
混淆后的js

```
                e = "";
            return e
        }
        function C(e) {
            for (var i = 0, n = 0; n < e.length; ++n) {
                var r = e.charCodeAt(n);
                55296 <= r && r <= 57343 && (r = 65536 + ((1023 & r) << 10) |
                r <= 127 ? ++i : i = r <= 2047 ? i + 2 : r <= 65535 ? i + 3 :
            }
            var A = i + 1;
            if (n = i = D(A),
            r = E,
            0 < A) {
                A = n + A - 1;
                for (var a = 0; a < e.length; ++a) {
                    var s = e.charCodeAt(a);
                    if (55296 <= s && s <= 57343)
                        s = 65536 + ((1023 & s) << 10) | 1023 & e.charCodeAt(
                    if (s <= 127) {
                        if (A <= n)
                            break;
                        r[n++] = s
                    } else {
                        if (s <= 2047) {
                            if (A <= n + 1)
                                break;
                            r[n++] = 192 | s >> 6
                        } else {
                            if (s <= 65535) {
                                if (A <= n + 2)
                                    break;
                                r[n++] = 224 | s >> 12
                            } else {
                                if (A <= n + 3)
                                    break;
                                r[n++] = 240 | s >> 18,
                                r[n++] = 128 | s >> 12 & 63
                            }
                            r[n++] = 128 | s >> 6 & 63
                        }
```

23 requests | 25.2 KB transferre | {} Line 1, Column 1

观察网络请求调用，获得关键指向函数

JS签名破解



JS签名破解

分析关键函数的调用代码

```
486     }());
487     var Main = (function () {
488         function Main() {
489             var _this = this;
490             this.url = "http://pibao.pullwave.com:8080/index.php";
491             this.resultPage = new ResultPage(this);
492             var res = Util.hrefToParams();
493             var page = res['page'] || 0;
494             if (res['key']) {
495                 SecurityWorker.ready(function () {
496                     _this.sw = new SecurityWorker();
497                     _this.sw.oncreate = function () {
498                         _this.onSearch(res['key'], page);
499                     };
500                 });
501             }
502         }
503         Main.prototype.onSearch = function (key, page) {
504             var _this = this;
505             if (page === void 0) { page = 0; }
506             key = key.trim();
507             this.sw.onmessage = function (msg) {
508                 $.ajax({
509                     type: 'GET',
510                     url: _this.url,
511                     data: { page: page, name: key, sign: msg.data.sig
512                     dataType: "json",
513                     success: function (data) {
514                         if (data == null || data == '') {
515                             data = [];
516                         }
517                         _this.result = new ResultData(decodeURI(key ,
518                         m.redraw();
519                     },
520                     error: function (e) {
521                         console.log(e);
522                         _this.result = new ResultData(decodeURI(key ,
523                         m.redraw();
524                     }
525                 });
526             };
527             this.sw.postMessage({
528                 type: "sign", data: {
529                     name: key,
530                     page: page.toString(),
531                 }
532             });
533         };
```

关键函数1

关键函数2

控制台中验证关键函数的调用逻辑

JS签名破解

```
> sw = new SecurityWorker();
<· ▶SecurityWorker {blob: Blob, worker: Worker, onmessage: ƒ, oncreate: ƒ, onterminate: ƒ}
> sw.onmessage = function(msg){console.log(msg)}
<· ƒ (msg){console.log(msg)}
> sw.postMessage({type:"sign",data:{name:"女装", page:1}})
<· undefined
  ▼{type: "sign", data: {…}}                                          VM1421:1
    ▶data: {sign: "08d87e4372031667cd541409b8d0ad1d", ts: "1564143775819"}
     type: "sign"
    ▶__proto__: Object
>
```

填入参数

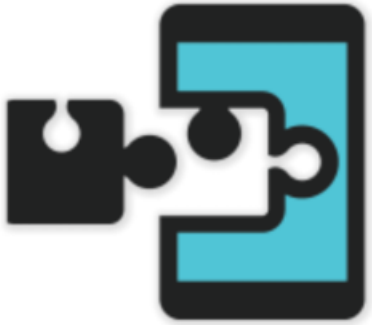
拿到计算结果

Frida

Frida

为什么要用Frida

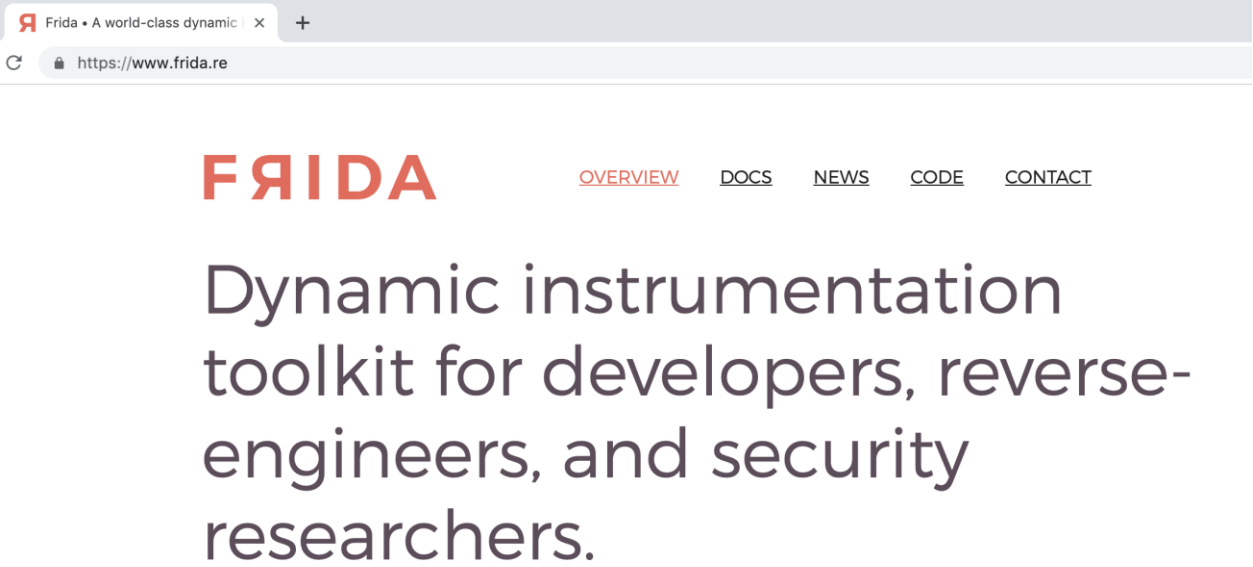
- 移动端逆向调试和Hook是刚需
- Xposed
  - 调试代码过于臃肿，效率低
  - Native调试非常不方便
  - 支持平台单一



Frida

Frida优点

- 入门简单，只需熟悉基本js语法即可上手
- 社区活跃，文档资料丰富
- 安卓、苹果、桌面全平台支持
- 即改即生效，无需编译打包



Frida

跟踪应用对文件的操作

```
'use strict';
Interceptor.attach(Module.findExportByName(null, 'open'), {
  onEnter: function (args) {
    console.log('[*] open(\"' + Memory.readUtf8String(args[0]) + '\")');
  }
});
Interceptor.attach(Module.findExportByName(null, 'close'), {
  onEnter: function (args) {
    console.log('[*] close(' + args[0].toInt32() + ')');
  }
});
Interceptor.attach(Module.findExportByName(null, 'read'), {
  onEnter: function (args) {
    console.log('[*] read(' + args[0].toInt32() + ')');
  }
});
Interceptor.attach(Module.findExportByName(null, 'write'), {
  onEnter: function (args) {
    console.log('[*] write(' + args[0].toInt32() + ')');
  }
});
```

Frida

绕过安卓应用请求的双向证书检测

```
setTimeout(function(){
  Java.perform(function (){
    console.log("");
    console.log("[.] Cert Pinning Bypass/Re-Pinning");

    var CertificateFactory = Java.use("java.security.cert.CertificateFactory");
    var FileInputStream = Java.use("java.io.FileInputStream");
    var BufferedInputStream = Java.use("java.io.BufferedInputStream");
    var X509Certificate = Java.use("java.security.cert.X509Certificate");
    var KeyStore = Java.use("java.security.KeyStore");
    var TrustManagerFactory = Java.use("javax.net.ssl.TrustManagerFactory");
    var SSLContext = Java.use("javax.net.ssl.SSLContext");

    // Load CAs from an InputStream
    console.log("[+] Loading our CA...")
    cf = CertificateFactory.getInstance("X.509");

    try {
      var fileInputStream = FileInputStream.$new("/data/local/tmp/cert-der.crt");
    }
    catch(err) {
      console.log("[o] " + err);
    }

    var bufferedInputStream = BufferedInputStream.$new(fileInputStream);
    var ca = cf.generateCertificate(bufferedInputStream);
    bufferedInputStream.close();

    var certInfo = Java.cast(ca, X509Certificate);
    console.log("[o] Our CA Info: " + certInfo.getSubjectDN());

    // Create a KeyStore containing our trusted CAs
    console.log("[+] Creating a KeyStore for our CA...");
    var keyStoreType = KeyStore.getDefaultType();
    var keyStore = KeyStore.getInstance(keyStoreType);
    keyStore.load(null, null);
    keyStore.setCertificateEntry("ca", ca);

    // Create a TrustManager that trusts the CAs in our KeyStore
```

Frida

获得动态注册的JNI函数

```
var hook_registNatives = function() {
  var env = Java.vm.getEnv();
  var handlePointer = Memory.readPointer(env.handle);
  console.log("handle: " + handlePointer);
  var nativePointer = Memory.readPointer(handlePointer.add(215 * Process.pointerSize));
  console.log("register: " + nativePointer);
      Interceptor.attach(nativePointer, {
       onEnter: function(args) {
         var methods = args[2];
         var methodcount = args[3];
         var name = env.getClassName(args[1]);
         console.log("=== class: " + name + " ====");
         console.log("==== methods: " + methods + " nMethods: " + methodcount + " ====");
         for (var i = 0; i < methodcount; i ++ ) {
           var idx = i * 12;
           console.log("name: " + Memory.readCString(Memory.readPointer(methods.add(idx)))
             + " signature: " + Memory.readCString(Memory.readPointer(methods.add(idx + 4)))
             + " fnPtr: " + Memory.readPointer(methods.add(idx + 8))
           );
         }
       }
  });
}

Java.perform(function () {
  hook_registNatives();
});
```

Frida

Frida还能做

- 快速dump应用动态加载的dex
- 动态打印函数调用栈
- 实现okhttp3的Interceptor
- 对native层进行inline hook
- ……

手机机房

手机机房

为什么要搭建手机机房

- 逆向APP的难度逐渐加大
- APP内藏有陷阱，纯协议可能导致封号
- 手机机房的优势：
  - 真实App运行环境，应对复杂的检测手段
  - 减少或降低的逆向分析的难度
  - 受应用版本升级影响较小，鲁棒性高

手机机房

构建手机机房的挑战 I

- 基础环境设施的搭建
  - 租用场地
  - 网络设备配置
  - 人工运营（管理、日常维护）
- 配套的软件产品
  - 批量安装应用、刷机
  - 远程脚本调度控制
  - 单机排查问题（远程控制VNC）
  - 定时计划任务

手机机房

构建手机机房的挑战 II

- 可靠性与分组管理
  - 构建多个机房，保障业务可靠性
  - 不同机房构造不同的子网络拓扑
  - 不同任务多机房调度派发
- 异常情况的处理
  - 停电、断网
  - 电池鼓包（预防燃火）
  - 硬件坏损，长期持久性维护

手机机房

手机机房竣工图

手机机房

机房日常维护

手机机房

机房扩容

手机机房

手机机房的缺点

- 抓取效率低、成本高
- 维护成本高
  - 存在硬件故障维修维护问题（电池鼓包、屏幕坏损）
  - 较多的人工参与运维
- 可扩展性差
  - 扩容比较麻烦，部署需要时间。

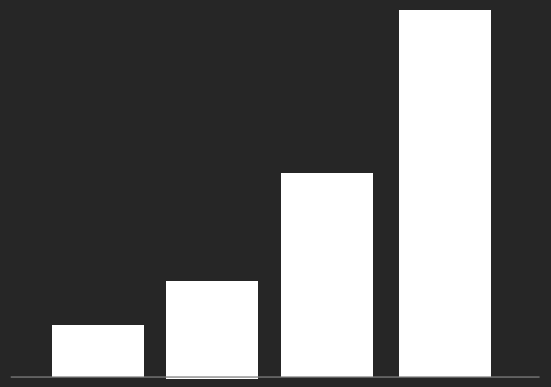
手机机房

纯逆向协议 v.s. 手机机房

	优点	缺点
逆向协议 难!	执行效率极高 可扩展性极好	前期消耗大量时间精力 受版本升级影响略大
手机机房 (模拟器) 苦!	鲁棒性好 躲避很多检测	可扩展性很差 执行效率很低 维护成本高

前沿技术展望

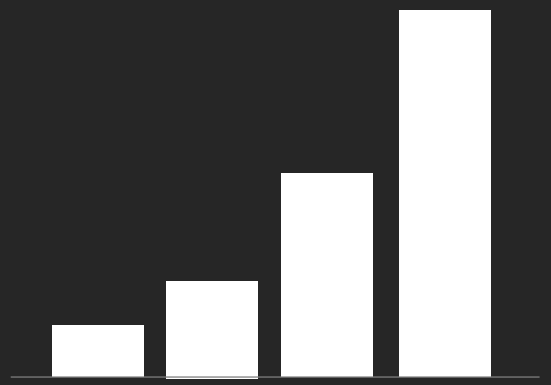
浏览器指纹



浏览器指纹的作用

- 浏览器指纹是指通过采集浏览器的信息，进而分析得到该浏览器（甚至是机器）唯一标识的技术
- 对爬虫的影响
  - 使用selenium开多个浏览器，可能是同一个指纹id
  - Headless浏览器可能会出现指纹获取失败

浏览器指纹

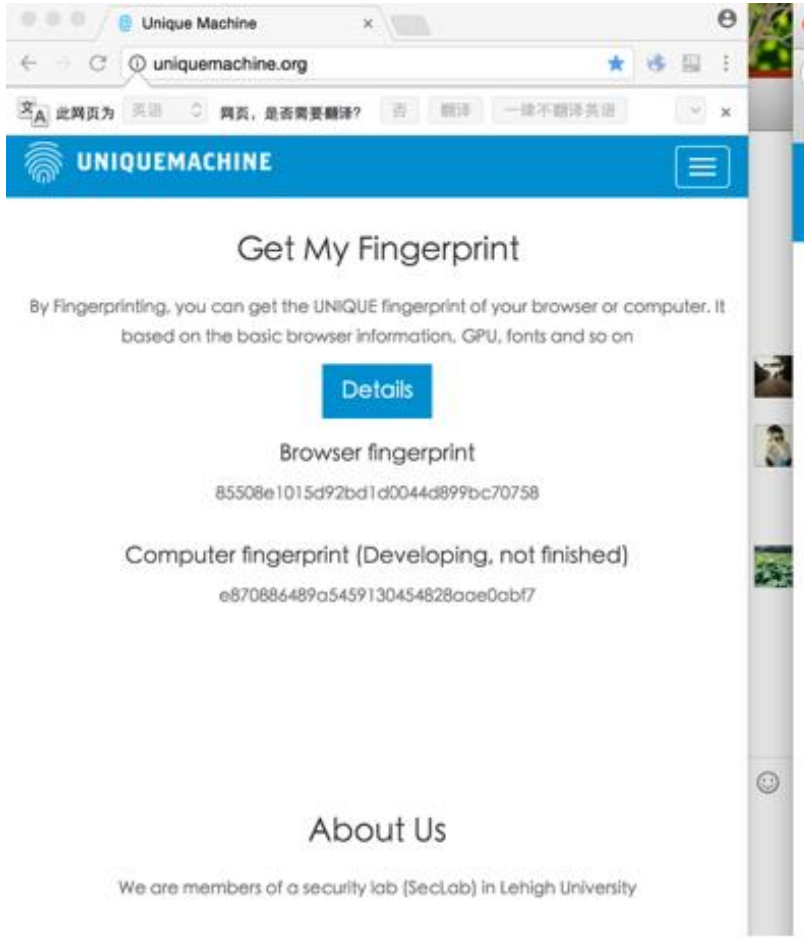


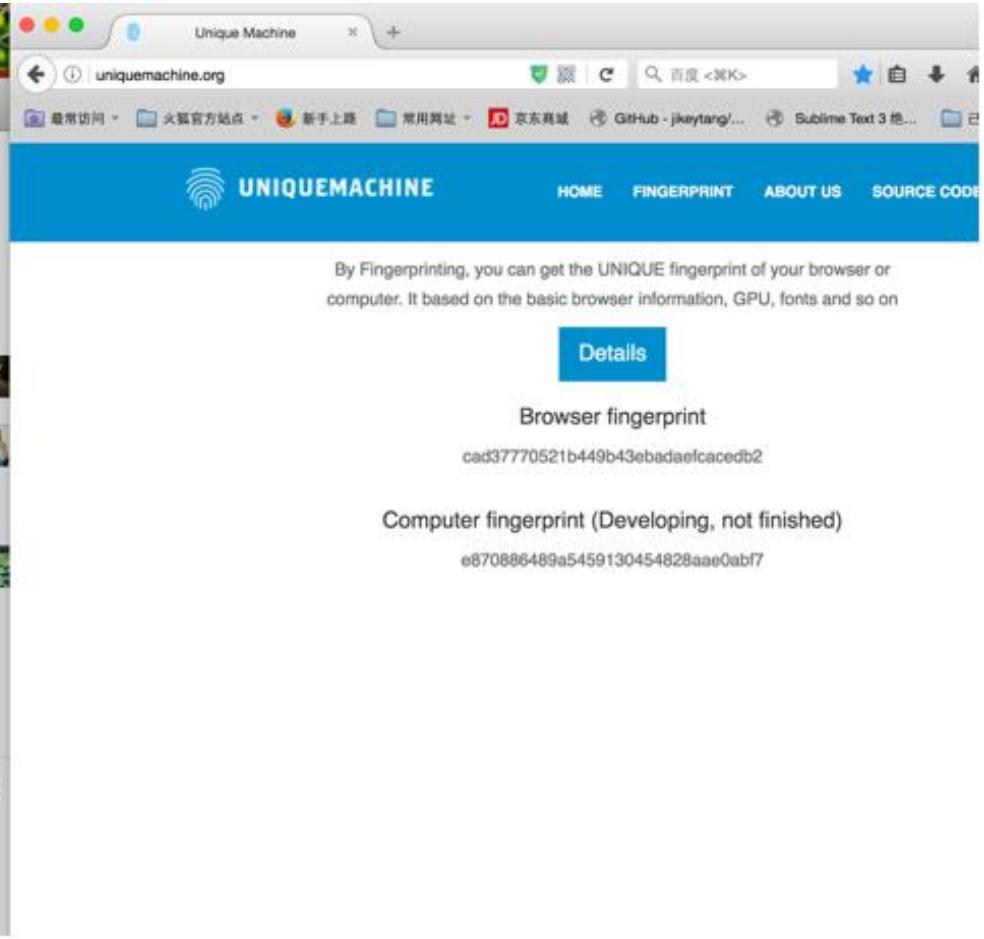
浏览器指纹的特征采集

- 常用的浏览器指纹检测手段
  - Canvas指纹
  - 插件指纹
  - 字体指纹
  - 音频指纹
  - 显卡绘图指纹
- NDSS17(http://yinzhicao.org/TrackingFree/crossbrowsertracking_NDSS17.pdf)

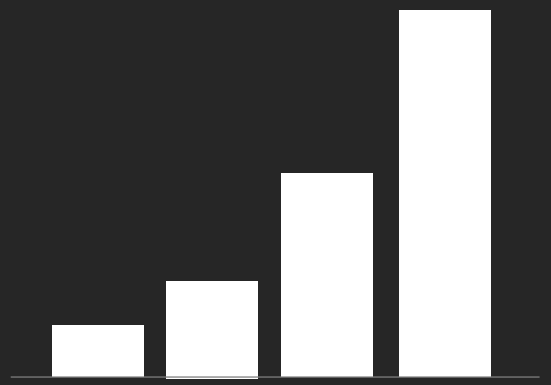
浏览器指纹

机器指纹


Unique Machine
uniquemachine.org
UNIQUEMACHINE
Get My Fingerprint
By Fingerprinting, you can get the UNIQUE fingerprint of your browser or computer. It based on the basic browser information, GPU, fonts and so on
Details
Browser fingerprint
85508e1015d92bd1d0044d899bc70758
Computer fingerprint (Developing, not finished)
e870886489a5459130454828aae0abf7
About Us
We are members of a security lab (SecLab) in Lehigh University


Unique Machine
uniquemachine.org
UNIQUEMACHINE
HOME
FINGERPRINT
ABOUT US
By Fingerprinting, you can get the UNIQUE fingerprint of your browser or computer. It based on the basic browser information, GPU, fonts and so on
Details
Browser fingerprint
cad37770521b449b43ebadaefcacedb2
Computer fingerprint (Developing, not finished)
e870886489a5459130454828aae0abf7

浏览器指纹



反-浏览器指纹

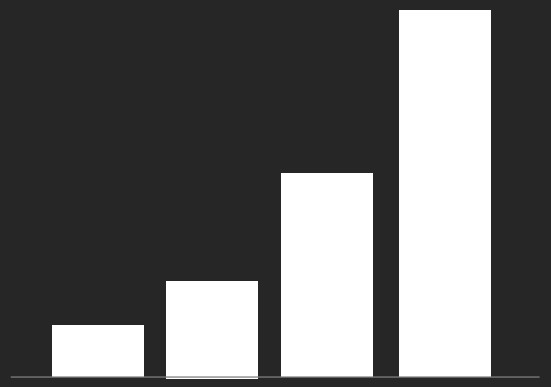
- 熟悉前沿浏览器内核采集的数据特征
- 改造浏览器内核
  - 通过源码改造
  - Hook改造
- 市面已经有类似的商业产品

WASM

WebAssembly

- WebAssembly可以简单理解成是在浏览器上执行的“汇编代码”
- 可以把C/C++代码编译成WASM，由浏览器导入使用

WASM



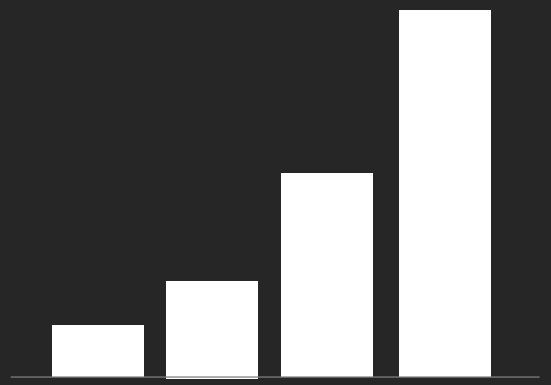
WebAssembly的样例

main.wasm.x86

```
wasm-function[0]:
  sub rsp, 8                          ; 0x000000 48 83 ec 08
  nop                                 ; 0x000004 66 90
  add rsp, 8                          ; 0x000006 48 83 c4 08
  ret                                 ; 0x00000a c3

wasm-function[1]:
  sub rsp, 8                          ; 0x000000 48 83 ec 08
  mov eax, 0x2a                       ; 0x000004 b8 2a 00 00 00
  nop                                 ; 0x000009 66 90
  add rsp, 8                          ; 0x00000b 48 83 c4 08
  ret                                 ; 0x00000f c3
```

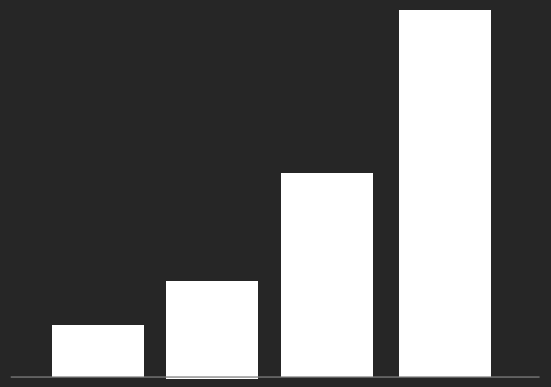
WASM



WebAssembly的现状

- 目前，WASM的已经开始“悄悄的”应用在很多站点上了
- WASM仍然处在发展期，还是有很多缺点
  - Emscripten打包文件太大
  - 文档并不那么丰富
- 相比js，这个的破解难度更大
- 逆向工具很少（Decompiler）

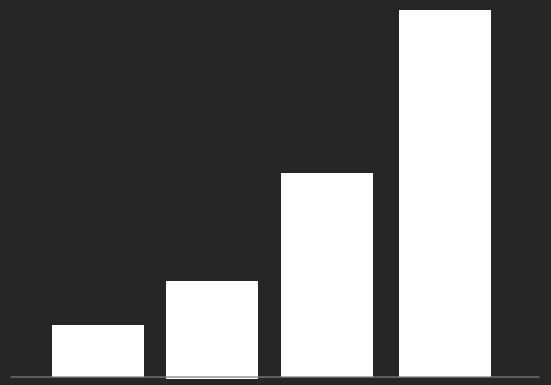
WASM



反-WebAssembly

- 掌握传统汇编的知识
- 了解linux的一些系统原理知识
- 开发WASM逆向工具（或者未来等大神开发出来再用）😂

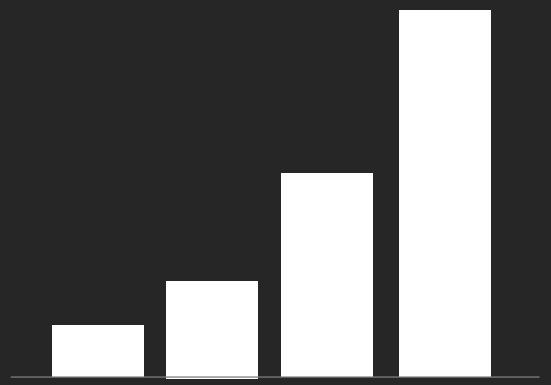
Native混淆



Native混淆的现状

- 各大厂商逐渐把安全模块下沉
- 红蓝双方几十年的攻防碰撞，沉淀了无数的安全领域的知识
- 从2016年开始，大量PC(x86)上的安全技术，逐渐被移植到手机(arm)上

Native混淆



Decompiler降低了难度

File Analyse View Help

Instructions

```
8048094:  push   ebp
8048095:  mov    ebp, esp
8048097:  sub    esp, 0x18
804809a:  cmp    [ebp + 0xc]:32, 0x0
804809e:  jnz    0x80480a5
80480a0:  mov    eax, [ebp + 0x8]:32
80480a3:  jmp    0x80480c1
80480a5:  mov    eax, [ebp + 0x8]:32
80480a8:  mov    edx, eax
80480aa:  sar    edx, 0x1f
80480ad:  idiv   [ebp + 0xc]:32
80480b0:  mov    eax, edx
80480b2:  mov    [esp + 0x4]:32, eax
80480b6:  mov    eax, [ebp + 0xc]:32
80480b9:  mov    [esp]:32, eax
80480bc:  call   0x8048094
80480c1:  leave
80480c2:  ret
```

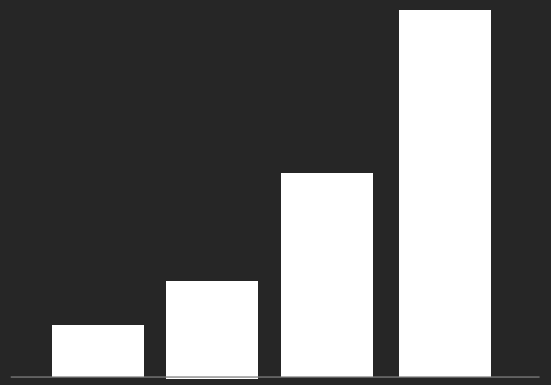
C++

```
int32_t gcd(int32_t arg1, int32_t arg2) {
  int32_t eax1;

  if (arg2 != 0) {
    eax1 = gcd(arg2, arg1 % arg2);
  } else {
    eax1 = arg1;
  }
  return eax1;
}
```

Line 6, Column 27

Native混淆

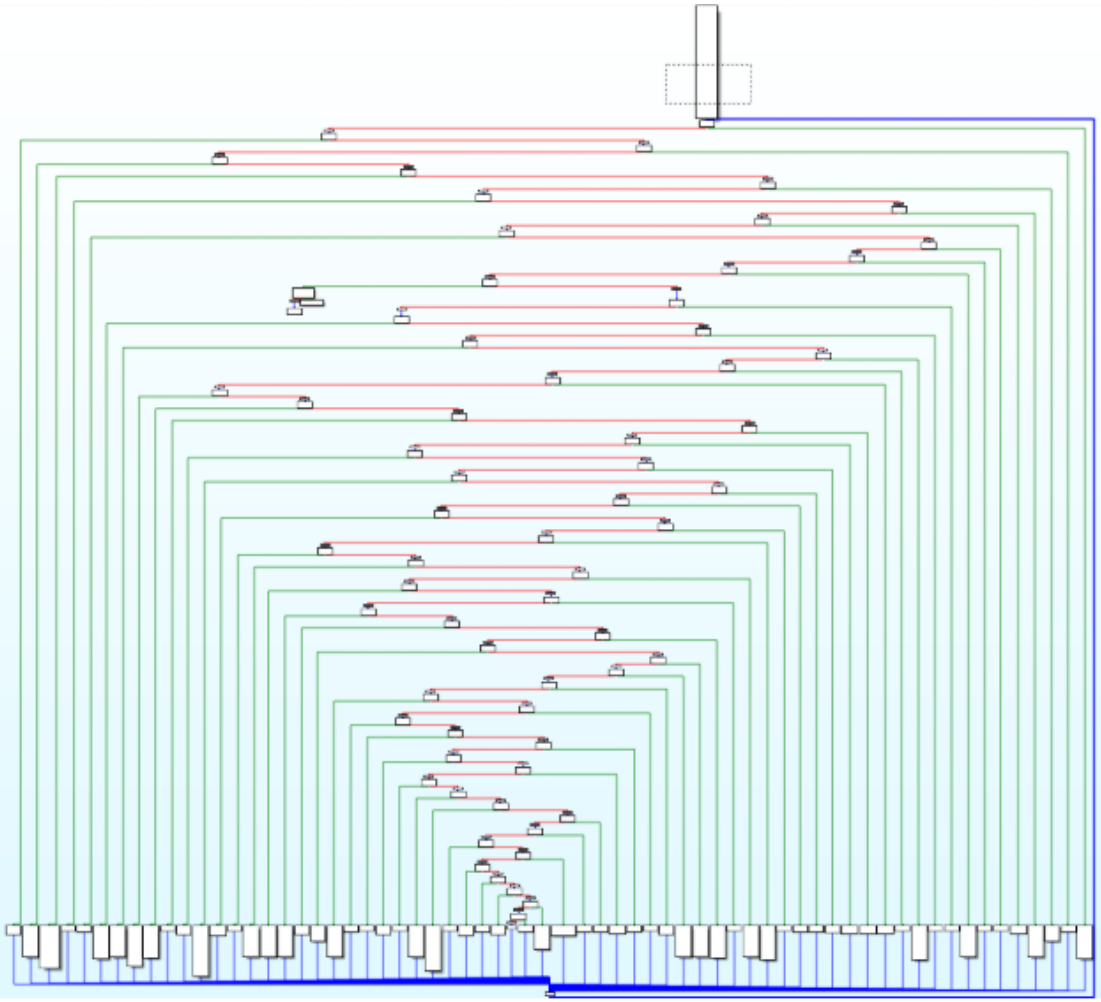


Native上混淆技术

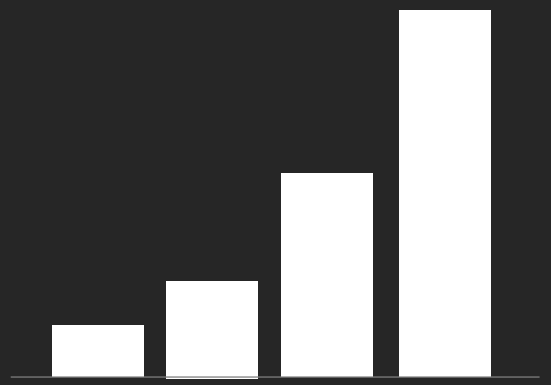
- 加壳（Packer）
- 混杂代码（Opaque Predicates）
- 花指令（Junk Code）
- OLLVM（Obfuscator-LLVM）
  - 控制流扁平化

Native混淆

OLLVM控制流扁平化



Native混淆

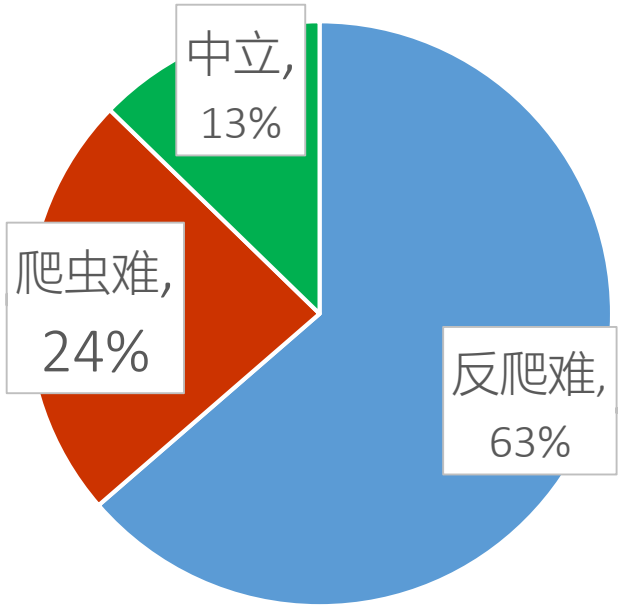


反-Native混淆

- 这个更多是经验性的方法
- 需要长期实践总结其中的模式和套路
- 对于固定的混淆，有一些工具
- 多关注安全社区的讨论

爬虫 / 反爬

正反观点



爬虫难

- 知识面要求广
- 每个站点策略都不同
- 未知法律风险
- 被动

反爬难

- 容易误杀真实用户
- 防守面大，进攻点小
- 数据有价值就有爬虫
- 被动

猫鼠游戏？

- 爬虫和反爬，日渐成为对抗性的行业
- 技术都是有生命周期的，不存在永远的真理
- 知己知彼，百战不殆

Crawler

2019

THANKS!