Protecting At-Risk Populations from Surveillance, Censorship, and Targeted Attacks: Revisiting BAT Browsers

Esther Rodriguez
Arizona State University

Tenzin Thayai
Tibet Action Institute

Lobsang Gyatso
Tibet Action Institute

Jedidiah Crandall
Arizona State University

Abstract

In this report we examine security and privacy concerns associated with six prominent Chinese web browsers: UC Browser, QQ Browser, Baidu Searchbox, OPPO Browser, Redmi Browser, and VIVO Browser. Our analysis focuses on sensitive data leaks, weak or missing encryption of information during transmission, and third party SDKs that are granted privileges that put users at risk. We found that these browser applications consistently expose sensitive data, including PII, geolocation, device information, and browser activity, often with poor transport-layer security, *e.g.*, purely symmetric cryptography. Some of the browsers transmit this private information even when using incognito mode. We make recommendations for at-risk users and circumvention/privacy tool developers in light of these findings.

1 Introduction

Web browsers are an important part of any at-risk user's set of online tools. Browsers are often the application that the user intends to use with censorship circumvention or privacy tools. Furthermore, as app stores become increasingly restricted (*e.g.*, Apple's App Store, which removes apps for Chinese users at the behest of the Chinese government) the only practical way for users to get access to censorship circumvention and privacy tools is as Progressive Web Applications (PWAs) that run in the browser. With the Tor Browser [1], the model is that the browser (the Tor Browser) and the censorship circumvention and privacy tool (Tor) work together to protect users' privacy and the availability of the tool. However, many users use other tools and other browsers, so a natural question is: how well do the browsers that users are actually using interact with various tools developed by the Internet freedom community? In this report we aim to answer this question for six browsers commonly used in Asian markets. Note that in this market choices of browsers are sometimes limited to various degrees by devices, app stores, or network conditions.

In 2016 Knockel *et al.* [2] reverse engineered what were, at the time, three of the most popular web browsers in the world: Baidu's Baidu Broser, Alibaba's UC Browser, and Tencent's QQ Browser. All three browsers sent private information (such as user IDs, web activity, GPS coordinates, *etc.*) to servers maintained by their respective vendors using poor encryption (*e.g.*, purely symmetric encryption or RSA with a 128-bit modulus that could be factored in under 3 seconds). All three browsers were also vulnerable to machine-in-the-middle attacks in their software update mechanisms. We leave attacking browser update mechanisms for arbitrary code execution as future work, and instead focus on the aforementioned transport-layer privacy issues and another issue that affects at-risk users: the granting of permissions to third-party SDKs.

We analyze six of the most prominent Chinese mobile browsers (including the three from the original BAT browser study): UC Browser, QQ Browser, Baidu Searchbox, OPPO Browser, Redmi Browser, and VIVO Browser. UC Browser is a popular web browser made by UCWeb, a subsidiary of the Alibaba Group. QQ Browser is a popular mobile web browser developed by Tencent. Baidu Searchbox is Baidu's search engine and it is the leading search engine in China, Hong Kong, Taiwan and many other countries [3]. It is the remnants of the formerly popular Baidu Browser. The OPPO, Redmi, and VIVO browsers are the built-in browsers in the OPPO, Xiaomi and VIVO phones respectively. Built-in applications, also known as pre-installed or native apps, are software programs that come preloaded on a phone upon purchase. These applications are designed to work seamlessly with the phone's hardware and operating system to provide core functionality and additional features. Chinese smartphones often come with pre-installed applications, including utility apps, social media, and content platforms popular in China, such as WeChat, QQ,

or Baidu.

We found that all six browsers expose sensitive data by sending it to servers maintained by the vendor, including the user's web activity including full URLs (even for HTTPS), page titles, and search terms. All six do so with no cryptography or poor cryptography. At least three of the browsers do so even when in incognito mode. All six grant potentially dangerous permissions to SDKs.

Based on our findings we make the following recommendations:

- Users of VPNs and other censorship circumvention and privacy tools should be made aware of the private information collected by these six browsers.
  - The information collected by these browsers, particularly web activity and search terms, violates the privacy assumptions users typically make when using tools such as a VPN or the browser's incognito mode.
  - The poor or missing cryptography that is typical of the Internet traffic carrying this information opens threats up beyond the browsers' vendors to any actor that can view Internet traffic between, *e.g.*, the VPN server and the vendor's servers.
  - The inclusion of information such as user IDs, GPS coordinates, and local network information makes it trivial for an attacker who can view and decrypt this information to detect the use of a VPN or other circumvention tool. For example, GPS coordinates in China and a client IP address outside China is a clear indication that the user identified by the user ID is using a VPN.
- Developers and users of PWAs should also be aware of the data collected by these browsers.
  - If the name or any other identifier of the PWA appears in the title bar or URL shown to the PWA user, this information is also being collected by the browser's vendor and is visible to local network actors (*e.g.*, the user's local ISP).
  - For PWAs with a privacy focus (*e.g.*, private encrypted chat), there is a risk that a software vendor could use elevated privileges to monitor the user in ways that would not be possible with other browsers. This extends beyond the vendors of the browsers we looked at to any vendors whose SDKs they include and give dangerous permissions to.
- At-risk users in the diaspora should be made aware of the risks of using these six browsers.
  - Web activity, user ID, GPS coordinates, *etc.* are constantly being send to servers in China while using these browsers.
  - The poor transport-layer security of these browsers means that this information is accessible to actors on local networks, as well.
  - If members of a diaspora use these browsers in combination with certain VPNs, the issues we find in this research could be combined with CVE-2021-3773 to redirect all of the PII and private data leaked to any other part of the world. For example, an attacker in Viet Nam could redirect traffic for a user in Japan using a VPN in the U.S. so that all of the information collected by one of the Chinese browsers in this report could be tracked by the attacker in Viet Nam.
  - Members of the diaspora should understand that if they use one of these browsers and they communicate with individuals in China, their web browsing acitivites can be tied to the individuals they chat with (*e.g.*, in WeChat).

2 Background and Related Work

Our work is a follow-up to the BAT (Baidu, Alibaba, Tencent) browsers work done by Knockel *et al.* [2, 4]. They found that these three browsers sent personal data to their respective vendors servers without any encryption or with encryption that can be easily decrypted. This information includes the user's IMEI, IMSI, search queries, and full URLs (even for HTTPS) and titles of pages visited. Moreover, they found that QQ Browser and Baidu browser also send location information like nearby WiFi access points. Knockel *et al.* Notified the BAT vendors of the security issues in the BAT browsers and some issues were addressed but the BAT browsers still have not adopted cryptography best practices and the APIs that had been found to be vulnerable in the past are still used by hundreds of millions of users, in one way or another.

Pradeep et al. [5] performed a large scale privacy analysis of Android browsers. In this analysis they added some of browsers from Chinese app stores that can be installed on Android devices, including QQ Browser and Baidu Browser. They report that QQ and Baidu leak browsing history data.

Liu et al [6] analyzed personal data being transmitted by the Redmi (Xiaomi), OnePlus (OPPO) and Realme phones. Their analysis focuses on information transmitted by all of the preinstalled applications on the phones. They found that a large amount of device-specific, geolocation, user profile, and social relationship informa-

tion gets transmitted by applications preinstalled on the phone. A specific mention of the use of preinstalled browser for their analysis is not found in their work.

It is well documented [7–9] that UC Browser sends a considerable amount of PII during incognito mode using easily decryptable encryption. It has been found that the URLs of visited pages, the IP of the user and other PII were sent to the vendors servers using purely symmetric cryptography with hardcoded keys and different block cipher modes (with hardcoded initialization vectors when CBC is used).

It was reported [10] that Xiaomi devices recorded all the websites visited by users, even when using incognito mode. Xiaomi phones sending user data to servers hosted by Alibaba. To collect some of the data, Xiaomi was using the services of a behavioral analytics company called Sensors Analytics [10]. Furthermore, browsers shipped by Xiaomi on Google Play were collecting the same data. After the disclosure of the above findings, Xiaomi released changes to its incognito browsing mode [11], these changes are different in the different versions of the Mi browser. An option to turn off aggregated data collection was added to incognito mode in the international versions. However, this option only prevents websites visited from collecting information but does not address what gets sent to Xiaomi's own servers.

3 Methodology

We analyzed three Chinese Android browsers (UC Browser, QQ Browser, Baidu Searchbox) available on all Chinese app stores and three Chinese browsers that come pre-installed (OPPO, Redmi, VIVO) on the OPPO, Xiaomi and VIVO phones, respectively. The details of the versions analysed are in table 1.

Browser	Version
UC Browser	13.9.4.1175
QQ Browser	12.2.3.7052
Baidu Searchbox	13.27.0.12
OPPO Browser	40.7.9.9
Redmi Browser	15.5.8
VIVO Browser	9.3.27.2

Figure 1: Apps analyzed and their version

Our analysis focused on:

1. Private data that is sent out by the app and is easily decrypted or not encrypted at all. We categorize this data as PII, geolocation and browsing activity data.

2. Private data being collected and transmitted during incognito mode, including browsing activity.

3. Permissions granted by the browser applications to third party SDKs, which can then potentially access private data protected by these permissions.

3.1 Environment Setup

We set up our environment for reverse engineering mobile applications by performing static and dynamic analysis. For static analysis we use JADX [12]. For dynamic analysis, we set up a virtual environment with Genymotion [13], Frida [14] and Mobile Security Framework (MobSF) [15]. MobSF has an httptool integrated [16] that aids with the capture, repeat and live intercept of HTTP requests with scripting capabilities, and is built on top of mitmproxy [17].

We installed all of the official APKS on our Genymotion instance and allowed any permissions for which the apps asked. We did not sign in into specific accounts like QQ or Weibo acounts, however, for most of the apps, except for QQ apps, most of the content and behavior of the app does not change when not logged in. For the browsers supporting incognito mode, we used the default settings for incognito browsing.

It is challenging to run built-in browsers (Xiaomi, VIVO, OPPO) in a research environment, since installable APKs are not available online and they are inextricably tied to specific devices for specific markets. For Vivo and Redmi we combined static analysis of available APKs combined with live packet captures taken from the relevant devices. In total there were 12 packet captures, for different versions, actions, and modes.

Using both static and dynamic analysis, we made a packet capture while using the browser, and then used a combination of tcpflow and tshark Linux command one liners to find and count certian byte patterns. The mitmproxy tool is set up to intercept and log HTTPS flows, which allows us to examine all the transmitted fields in plaintext for these flows. With our setup we were able to capture and decrypt most of the connections and trace the encrypted or unencrypted data back to the code in the APK. Furthermore, with Frida and MobSF we were able to collect the plaintext corresponding to the encrypted body of traffic (gziped). We were able to find hardcoded keys and easily decryptable or non encrypted data transmitted. We have set up existing genymotion scrips to dump uses of the crypto library for AES.

4 Results

We discovered the same privacy issues identified by Knockel *et al.* in all six browsers that we analyzed. Furthermore, this behavior occurs in incognito mode as well for most of the browsers we analyzed. We found that the information collected and transmitted from some of the applications is different depending on the location of the phone and gets sent to servers located in the given country where the user is located. Some data is even transmitted using plain HTTP.

Some browsers opt for sending sensitive data encrypted while in incognito mode. This data can get sent unencrypted when the default browsing method is used.

We found that although all of the browsers grant dangerous permissions to multiple third party SDKs, the built-in browsers grant a lot more permissions to third party SDKs than the browsers from the app store.

4.1 Transmission of Sensitive Information

By analyzing network traffic, we found sensitive data being sent to the browser's own server or to third party servers. By using static analysis we found that the data was being collected and transmitted by the browser application and not by the websites visited. We found data being transmitted in different ways (1) unencrypted as part of the header and/or body of HTTP(S) requests (2) poorly encrypted as part of the header and/or body of HTTP(S) requests (3) encrypted as part of the header and/or body of HTTP(S) requests.

The data we observed being collected and sent over the network divided in the following categories [2].

We observed the data we observed being collected and sent over the network is the following:

- PII: MEI, AD ID, Android ID, MAC, WifiMac, IMSI, ClientIP, OS Version, OS, Phone Model, Manufacturer, Screen Size.
- Location: MCC+MNC, GPS coordinates, LAC.
- Browser activity: terms searched and/or URLs visited.

For PII we found both persistent (IMEI) and resetable (Ad ID) identifiers being exposed by all of the browsers. Persistent identifiers are typically randomly generated to identify users and devices. They're better for identifying users than, *e.g.*, IP addresses and MAC addresses that can change. One important finding is that the full URLs of websites visited are collected and sent to the respective vendor of each browser, even for HTTPS websites. Some browsers also send the page title.

UC Browser

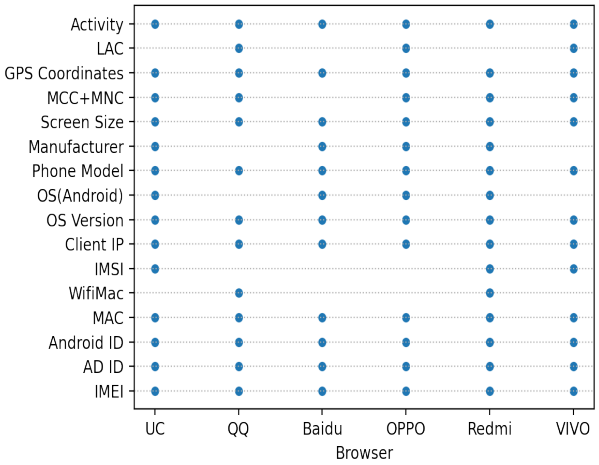

Figure 2

We found that UC Browser transmits sensitive data as detailed in 2. The browser checks for updates by making an HTTPS POST request to "https://puds.ucweb.com/upgrade". This request contains details of the phone like the OS version, phone model, and screen dimensions. For every website visited, basic information such as the domain name for the website and the title of the web page is sent in JSON format over unencrypted HTTP to 'logsug.ucweb.com'. This behavior is only observed when the browser is not in Incognito mode. This behavior occurs whether or not the web page being visited uses HTTPS.

We confirmed that, in both regular and incognito modes, the app still sends encrypted (with hardcoded, purely symmetric keys as explained in the next section) sensitive information to various ucweb domains that include the website visited, IP, *etc.*

QQ Browser

We found that QQ Browser leaks sensitive data over the network as shown in 2. We found that the local IP of the phone on the WiFi network, the GPS coordinates, and the MAC address goes to the browser's own servers unencrypted. All of the other sensitive information information is contained in WUP requests that are sent to Tencent's servers.

QQ browser sends to Tencent's servers HTTP POSTs request called WUP requests. The body of each request can be encrypted, partially encrypted or unencrypted. Different WUP requests send different information. Everything else is sent on WUP requests which are encrypted using AES and textbook RSA, which is known to be vulnerable [4]. The client uses the AES session key to encrypt the WUP request, in ECB mode.

Additional identifiers not present in 2 are also sent in WUP requests, these identifiers are the GUID, QUA,

LC, Cellphone, Uin, Cellid, ServerVer, Save Channel, UA, LanguageType, APN, CellNumber, LBSKeyData VenderID, and FirstChannel. The Q-GUID is a unique string used by QQ Browser to identify a particular user. Q-UA is a value used by QQ Browser that identifies the version of the application used and the type of hardware on which it is installed. The Q-GUID and Q-UA appear in the headers of WUP requests unencrypted and in the payloads of WUP requests encrypted.

QQ Browser and WUP requests are of broad interest, especially considering that we have evidence to suggest that a substantial proportion of applications in Chinese app stores utilize WUP requests. Notable examples include WeChat, Tencent's app store, and QQ's chat application. In the Anzhi app store, for instance, 14% of the apps are likely to issue WUP requests. As a result of prior ethical disclosures made by Citizen Lab, Tencent has made considerable strides in enhancing the encryption of WUP requests. However, the current encryption scheme still falls short of adhering to numerous best practices in cryptography.

Baidu Searchbox

We found that Baidu transmits PII and device-related data to multiple Baidu domains. This includes client IP, GAID, IMEI, OS version, phone model, and app-specific parameters such as CUID and BAIDUID. The information is sent either encrypted or unencrypted. The client IP and CUID are transmitted in plaintext to various domains, including: https://wappass.baidu.com/v8/sdkconfig, https://passport.baidu.com/v3/api/login/sharev3app, and https://nsclick.baidu.com/v.gif. Note that HTTPS adds adequate encryption to the flow, so that the vendor still has the PII but actors on the network between the client and vendor's server do not in this case.

As detailed by Knockel et al. in their report "Privacy Security Issues in Baidu Browser," the collection and leakage of sensitive data are attributable to Baidu Mobile Tongji (Analytics) SDK, one of Baidu's software development kits (SDKs). Furthermore, Baidu Push SDK, another of Baidu's SDKs, AES encrypts the CUID and sends it as the "devinfo" field. According to a report by Palo Alto Networks, their analysis of Android malware indicates that SDKs like Baidu Push SDK or ShareSDK are often used by malicious applications to extract and transmit device data.

Baidu Mobile Tongji (Analytics) SDK collects and sends information such as OS version, phone model, manufacturer, OS (Android), Baidu Browser version number, screen dimensions in pixels (width and height), IMEI number, UUID, CUID, GAID, device MAC ID, device Bluetooth MAC, and package name. Some fields are encrypted using AES/ECB with the hardcoded key "h9YLQoINGWyOBYYk" before being transmitted to https://hmma.baidu.com/app.gif via TLS (HTTPS).

Baidu sends the BAIDUID parameter as a tracking cookie that gets stored for some visited domains. The BAIDUID is stored and transmitted to https://passport.baidu.com/v3/api/login/sharev3app, along with the CUID. Knockel et al.'s report on Baidu Browser revealed that the CUID parameter was a concatenated string of an MD5 hash of Android version information and the phone's IMEI number written backward, which was then encrypted with an easily decryptable algorithm. The CUID that we identified in Baidu Searchbox appears to be different from the one previously reported.

OPPO Browser

Our research discovered that OPPO transmits PII and device-related data to their own domains, which raises concerns about user privacy. This information includes data such as client IP, URL visited, and an MD5 signature, which are sent unencrypted via HTTP to support.browser.heytapmobi.com. The lack of encryption exposes users' sensitive information to potential interception and misuse by malicious actors.

Furthermore, OPPO Browser was found to leak encrypted IMEI information in the header of GET requests sent to "api-cn.cdo.heytapmobi.com/usertrace/log/...". Static analysis and Frida scripts revealed that the IMEI and OpenID are AES encrypted using the hardcoded key "puwQbwBb9CMen91BMLD+UA==". In addition to this, the browser also sends location information to https://i6.weather.oppomobile.com/weather/. Users should be aware of these privacy risks when using the OPPO Browser and consider opting for alternative browsers that prioritize user privacy and security.

Additionally, OPPO Browser uses Baidu as its search engine, which involves sending and receiving data from Baidu servers. It has been found that the browser contains code (libcuid.so) to generate a CUID, which is an MD5 hash of the Android version information and the phone's IMEI number written backward. The CUID is sent to Baidu domains in plaintext *via* HTTPS requests to https://api.map.baidu.com/sdkcs/verify.

Mi Browser

We found that Xiaomi browser sends data to tracking.intl.miui.com, sdkconfig.ad.xiaomi.com, and staging.tracking.miui.com. We found that Xiaomi doesn't send data to sa.api.intl.xiaomi.com, according to [6] they didn't find this to be the case for all the apps on Xiami's phones. We looked into browsers shipped by Xiaomi on Google Play and they do not send data to that domain, either.

Using static analysis of this browser, we found that Xiaomi sends encrypted data to their own domains: tracking.intl.miui.com, sdkconfig.ad.xiaomi.com, and staging.tracking.miui.com. The data is being Gziped and

encrypted using AES/ECB/PKCS5Padding before being sent to Xiaomi's servers.

We found that Mi Browser use Baidu as the search engine by default, for this reason they share all of the search terms information to Baidu servers, even when using incognito mode on Mi Browser. If you input a URL that does not get sent to Baidu, it goes directly to the website. Furthermore, when using the browsers, Baidu gets location information by default (api.map.baidu.com, loc.map.baidu.com). This can be turned off by turning off location on the phone. Else, the data gets sent by the phone but code for this was not located in the browsers apk . Collecting both browsing history and PII can allow the browsers or third parties to match the history with a unique user and fingerprint them.

There is code (libcuid.so) to generate the CUID for Baidu.

VIVO Browser

Our research found that VIVO browser transmits sensitive data to various domains, which poses significant privacy risks for users.

The VIVO browser sends information such as IMEI, AD ID, Android ID, MAC, and IP to http://log.vivobrowser.com/upload/. This data can be utilized to uniquely identify users and their devices, potentially allowing for unwanted tracking and profiling. The transmission of such sensitive information to external domains raises questions about the privacy and security measures implemented by the VIVO browser.

Moreover, VIVO browser also sends data, including IMSI, OS version, phone model, and screen dimensions, to https://mlog.wangsu.com/sce/upload. While it is unclear if this data is sent to a VIVO domain, the sharing of such details can be used to gather insights about users' devices and preferences, further exposing users to privacy threats.

Additionally, the browser sends information like MCC, MNC, and GPS coordinates to Tencent's map service through the URLs "http://lstest.map.soso.com/loc?c=1" and "http://lbs.map.qq.com/loc?c=1". Sharing location data with external services can lead to users' real-time locations being tracked, which has serious privacy implications.

Similar to the OPPO Browser, VIVO browser also contains code (libcuid.so) to generate a CUID for Baidu. As previously mentioned, the CUID is an MD5 hash of the Android version information and the phone's IMEI number written backward, which can be utilized to uniquely identify users and their devices. The presence of this code raises further concerns about the commitment of VIVO browser to user privacy and security.

4.2 Incognito Mode

The version of OPPO we analyzed does not include an incognito mode. We could not determine if the built-in version of VIVO has an incognito mode.

When using incognito mode Xiaomi only guarantees that "Your browsing history, cookies, site data, and the information entered in forms won't be saved in incognito mode." However, we found that the Redmi browser leaks searched terms to Baidu, when using the default settings since it uses Baidu as the search engine.

We confirmed that UC Browser still sends encrypted sensitive information to "http://px-intl.ucweb.com/api/v1/crash/upload" while on incognito mode. Furthermore, the data is still encrypted using AES/CBC/PKCS5Padding with a zero IV and using the hardcoded AES key "Ine34@32b#jeRs2h". Other information is sent to px.ucweb.com, encrypted with the hardcoded key "1234567890abcdef" using AES/CBC/PKCS5Padding.

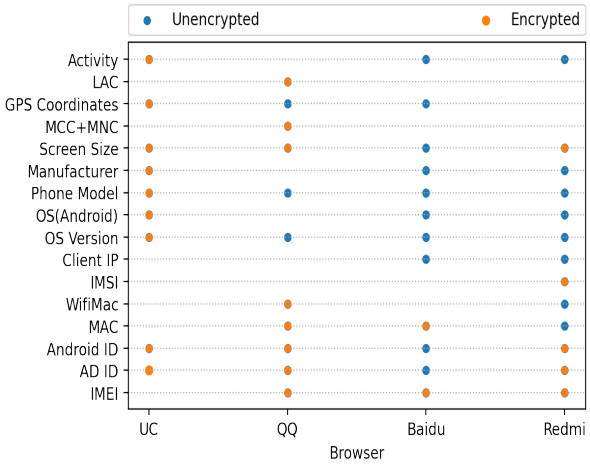

Figure 3

4.3 Permissions Granted to SDKs

All of the browser applications we analyzed were observed granting dangerous permissions to third-party SDKs, which can potentially put users' privacy at risk. In some cases, these permissions are automatically granted to the SDK without requesting user consent.

As mentioned in previous sections, Baidu Mobile Tongji (Analytics) SDK collects an extensive range of information by utilizing permissions such as READ_PHONE_STATE, INTERNET, and ACCESS_NETWORK_STATE. The collected information includes OS version, phone model, manufacturer, OS (Android), Baidu Browser version number, screen dimensions in pixels (width and height), IMEI number,

UUID, CUID, GAID, device MAC ID, device Bluetooth MAC, and package name.

OPPO has denied their association with with BBK Electronics, however they include a BBK SDK in their browser APK. The presence of such SDKs in browser applications can further exacerbate privacy concerns and raise questions about the overall security and privacy practices of these companies.

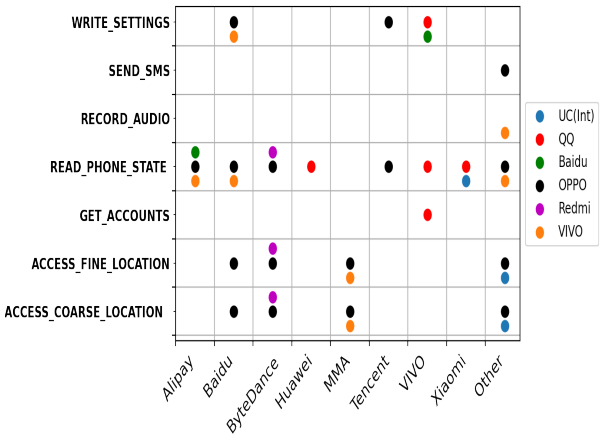

Figure 4

When analyzing built-in browsers compared to those available on Chinese app stores, it becomes evident that built-in browsers tend to request more permissions, as shown in 4, many of which are considered dangerous and can jeopardize users' privacy.

For the version of the Chinese built-in Mi browser, we also found that it no longer uses Sensors Analytics and it does not send information to Sensors Analytics domains. The SDK is no longer present in the Mi Browser APK we analyzed.

References

[1] Download Tor Browser. `https://www.torproject.org/download/`.

[2] Jeffrey Knockel, Adam Senft, and Ronald Deibert. Privacy and security issues in BAT web browsers. In *6th USENIX Workshop on Free and Open Communications on the Internet (FOCI 16)*, Austin, TX, August 2016. USENIX Association.

[3] Available at `https://www.similartech.com/compare/baidu-search-box-vs-google-programmable-search-engine` (2023/03/24).

[4] Jeffrey Knockel, Thomas Ristenpart, and Jedidiah R. Crandall. When textbook RSA is used to protect the privacy of hundreds of millions of users. *CoRR*, abs/1802.03367, 2018.

[5] Amogh Pradeep, Álvaro Feal, Julien Gamba, Ashwin Rao, Martina Lindorfer, Narseo Vallina-Rodriguez, and David Choffnes. Not your average app: A large-scale privacy analysis of android browsers, 2022.

[6] Haoyu Liu, Douglas J. Leith, and Paul Patras. Android os privacy under the loupe – a tale from the east, 2023.

[7] Available at `https://netalert.me/uc-browser-leaks-personal-data.html` (2023/03/24).

[8] Available at `https://hookgab.medium.com/ucbrowser-privacy-study-ecff96fbcee4` (2023/03/24).

[9] Available at `https://tspace.library.utoronto.ca/bitstream/1807/97086/3/Report%2377--ucbrowser.pdf` (2023/03/24).

[10] Thomas Brewster. Exclusive: Warning over chinese mobile giant xiaomi recording millions of people's 'private' web and phone use. Available at `https://www.forbes.com/sites/thomasbrewster/2020/04/30/exclusive-warning-over-chinese-mobile-giant-xiaomi-recording-millions-of-peoples-private-web-and-phone-use/?sh=39ce2a311b2a` (2023/03/24).

[11] Copyright © 2010-2022 xiaomi. All rights reserved. Miui privacy white paper. Available at `https://trust.mi.com/docs/miui-privacy-white-paper-global/3/3` (2023/03/24).

[12] Jadx. Available at `https://github.com/skylot/jadx` (2023/03/24).

[13] Genymotion. Available at `https://www.genymotion.com/` (2023/03/24).

[14] Frida. Available at `https://frida.re/` (2023/03/24).

[15] Mobile security framework. Available at `https://github.com/MobSF/Mobile-Security-Framework-MobSF` (2023/03/24).

[16] httptools. Available at `https://github.com/MobSF/httptools` (2023/03/24).

[17] mitmproxy. Available at `https://mitmproxy.org/` (2023/03/24).