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

Esther Rodriguez
Arizona State University, Breakpointing Bad
earodr32@asu.edu

Lobsang Gyatso
Tibet Action Institute
lobsang@tibetaction.net

Tenzin Thayai
Tibet Action Institute
thayai@tibetaction.net

Jedidiah R. Crandall
Arizona State University, Breakpointing Bad
jedimaestro@asu.edu

Abstract

A major use case for the use of VPNs by at-risk users is to put their web browsing activity outside the purview of potential attackers. State-sponsored attackers, specifically, can carry out various attacks against at-risk users who do not use a VPN within their country's borders. This can include censoring websites, performing surveillance of a user's web browsing activities, using this surveillance to build up censorship apparatus, or injecting malicious code into web traffic. The BAT Browsers study presented at FOCI 2016 demonstrated that common web browsers send web activity along with personally identifiable information (PII) to servers in China, often using poor or missing cryptography. Has this situation changed in the past 8 years? What does it mean for today's circumvention tools? How does it affect diaspora populations that are not in China? Do new incognito modes added by these browsers ameliorate the situation?

In this paper we examine security and privacy concerns associated with six prominent Chinese web browsers: Baidu Searchbox, UC Browser, QQ Browser, OPPO Browser, Redmi Browser, and VIVO Browser. Our analysis focuses on sensitive data collection, 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.

Keywords

reverse engineering, privacy, censorship circumvention

1 Introduction

In 2016 Knockel *et al.* [16] reverse engineered what were, at the time, three of the most popular web browsers in the world: Baidu's Baidu Browser, Alibaba's UC Browser, and Tencent's QQ Browser. Taking the first letter of each vendor's name, these became known as BAT browsers. 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.

In our work presented in this paper, we seek to update Knockel *et al.*'s results and analyze what they mean for at-risk users today. Specifically, web browsers from the Chinese market are heavily used by the following subpopulations of at-risk users: 1) users of VPNs and other circumvention tools; 2) members of the diaspora; 3) users who may be targeted by local actors on local networks (*e.g.*, corrupt local officials); 4) users of progressive web apps; and, 5) users whose data might be shared with the Chinese government by Chinese companies. Working in collaboration with a non-profit that works directly with these populations, our goal in this work is to collect up-to-date information and provide actionable advice to at-risk users.

We analyze recent versions of the three browsers from the original BAT browser study:

- Baidu Browser, the B in BAT, is no longer a major player in the browser market, but the Baidu Mobile Tongji (Analytics) and Baidu Push SDKs that were responsible for Knockel *et al.*'s findings with respect to Baidu Browser are included in many different apps as an interface to the Baidu search engine, which enjoys 45% of China's search engine market share [7]. Baidu Searchbox, which we reverse engineered instead of the defunct Baidu Browser, is the leading search engine in China, Hong Kong, Taiwan and many other countries.
- UC Browser, from Alibaba and forming the A in BAT, is the fourth-most used web browser in the world [5] and the third-most in China [4].
- QQ Browser, from Tencent and forming the T in BAT, has a little over 7% of China's browser market share [4].

We also analyze three built-in browsers. The three phone manufacturers were chosen according to their prevalent use by at-risk users served by the organization we worked with, but also happen to be large players in the Chinese market [6]:

- OPPO Browser had 6.81% of the Chinese mobile phone manufacturer market share in 2024.
- Redmi, also known as Xiaomi, had 13.37%.
- Vivo had 10.39%.

Free and Open Communications on the Internet 2025(1), 16–23

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. The browsers on these phones typically report their user agent as "Chrome" and have innocuous names in the user interface (*e.g.*, "Browser"), but often have manufacturer-specific features compiled into their binaries.

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, such as VPNs. As VPNs are increasingly used around the world by at-risk populations, a natural question is, do they actually protect a user's browsing activity? Furthermore, as VPNs are increasingly blocked in repressive countries and the VPNs themselves adapt by using advanced techniques to evade DPI, another question is, are there trivial ways to detect VPN use without DPI? Even for a VPN with perfect DPI evasion, if the browser is collecting data about the user (*e.g.*, their identity and GPS coordinates) and their web browsing activity (*e.g.*, the full URL and page title of every web page they visit) and uploading it to a company that shares it with the same state actor who the user is protecting themselves against with a VPN, the VPN is effectively useless.

Also, separate from VPNs, 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 [8], 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 the above-mentioned six browsers. Note that in this market choices of browsers are limited to various degrees by devices, app stores, or network conditions.

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. In five out of six cases the data is transmitted with no cryptography or poor cryptography (*e.g.*, purely symmetric or with cryptography known to be vulnerable to chosen ciphertext attacks). Four of the six browsers offer incognito modes. All four leak PII in incognito mode, and three of those four still collect and transmit the user's web activity. All six grant potentially dangerous permissions to SDKs.

2 Background and Related Work

Our work is a follow-up to the BAT (Baidu, Alibaba, Tencent) browsers work performed by Knockel *et al.* [14, 16]. 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 of the 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.

Pradeep *et al.* [20] performed a large scale privacy analysis of Android browsers. In this analysis they added 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.* [18] 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 information 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.

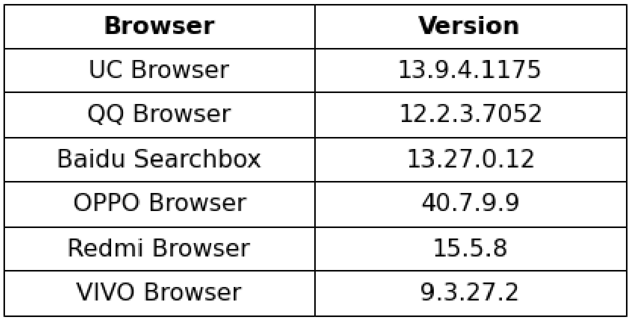
It is well documented [12, 15, 19] that UC Browser sends a considerable amount of PII during incognito mode using easily decryptable encryption. It has been found that the URLs of visited pages, the IP of the user and other PII were sent to the vendors servers using purely symmetric cryptography with hardcoded keys and different block cipher modes (with hardcoded initialization vectors when CBC is used).

It was reported [11] that Xiaomi devices recorded all the websites visited by users, even when using incognito mode. Xiaomi phones send user data to servers hosted by Alibaba. To collect some of this data, Xiaomi was using the services of a behavioral analytics company called Sensors Analytics [11]. 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 [22], 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.

Our work is aimed at answering questions about what it means for the results from the studies such as those above, especially the original BAT browser study, to be evaluated within the context of today's at-risk users in the Chinese market who use web browsers. The context of how web browsers are used has also changed in the past 9 years. VPNs and other circumvention tools are now targets of censorship themselves, and Progressive Web Apps (PWAs) are emerging as an attractive solution for offering functionality such as encrypted chat in repressive environments. Lastly, the diaspora that are connected to China but reside physically outside China, or any other country where use of these browsers is prevalent, was a major impetus for our revisiting of the BAT browser study. We hope that our results add to the existing body of evidence to convince circumvention technology developers that what real user traffic inside an encrypted tunnel looks like can affect the security and privacy of their tools.

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 analyzed are in Figure 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 to servers owned by Chinese companies (usually the vendor of the browser). Such data could be shared with state actors by the company, posing a threat to at-risk users. We categorize this data as PII, geolocation and browsing activity data. This app behavior is marked throughout the paper with a blue square (🟦).

(2) In many cases this data is transmitted with poor transport security, meaning it is easily decrypted or not encrypted at all and not encapsulated in TLS. So, in addition to the basic threat of the data being shared with state actors, this means that any attacker on the path from the user to the server has access to the data. If the user uses a VPN, only the routers between the VPN server and the remote server can access the data. This vulnerability is marked throughout the paper with a red square (🟥).

(3) For all four browsers with an incognito mode, private data is collected and transmitted even while in incognito mode, including browsing activity in three cases. This is the same threat as above, but is less expected by users because they assume incognito mode is more private. This app behavior is marked throughout the paper with a blue triangle (▼).

(4) Permissions granted by the browser applications to third party SDKs, which can then potentially access private data protected by these permissions. This app behavior is marked throughout the paper with a blue diamond (◆).

3.1 Environment Setup

We analyzed all six browsers in an analysis environment, and additionally tested the three built-in browsers and the thee others in separate validation environments. All Android instances in all environments were running Android 10.

We set up our analysis environment for reverse engineering mobile applications to perform static and dynamic analysis. For static analysis we used JADX [2]. For dynamic analysis, we set up a virtual environment with Genymotion [1], Frida [9] and Mobile Security Framework (MobSF) [10]. MobSF has an integrated httptool that aids with the capture, repeat and live intercept of HTTP requests with scripting capabilities, and is built on top of mitmproxy [3]. Since Genymotion acts as a rooted device, we installed a trusted certificate authority to strip SSL/TLS and perform analysis on HTTPS traffic. None of the browsers analyzed performed certificate pinning or anything else that would prevent machine-in-the-middle analysis of HTTPS. At no time did it become necessary to recompile APKs or any of the more advanced steps that may be necessary for analysis of some apps that have more advanced security and intellectual property protection features.

For the validation environment, we ran the three built-in browsers on actual phones from the respective manufacturers. The goal of this analysis was to compare packet captures from this validation environment and ensure that there were no discrepancies in how the browsers behaved as an APK *vs.* when they come pre-installed on a real device. For the three other (*i.e.*, not built-in) browsers, we ran them on a rooted Moto G7 Plus. We compared flows to various servers in the analysis *vs.* validation environments to ensure that all flows were accounted for in both environments. There were no notable discrepancies.

It is possible that we missed some additional behaviors beyond what we documented because we tested in mostly rooted environments. However, our static analysis in JADX and MobSF included analysis for root detection. The only anti-reverse-engineering behavior we found was that UC Browser sometimes will not transmit PII after some time has passed and an emulated environment is detected. Because the amount of time is on the order of minutes, we did not need to do anything special to mitigate this behavior.

We installed all of the official APKs on our Genymotion instance and allowed any permissions for which the apps asked. We did not sign into specific accounts, such as QQ or Weibo accounts. For the browsers supporting incognito mode, we used the default settings for incognito browsing.

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 certain 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 being transmitted. We also used existing genymotion scrips to dump uses of the crypto library for AES.

4 Results

We discovered many of the same privacy issues identified by Knockel *et al.* in the 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.

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 (Mobile Country Code) + MNC (Mobile Network Code), GPS coordinates, LAC (Location Area Code).
- 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 are 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. For example, for a URL of https://example.com/pagerequested/ with page title "Example Title", it is well known that example.com is exposed to all routers on the Internet path between the user and the server for example.com, in the Server Name Indicator field, unless Encrypted Client Hello (ECH) is in use. However, in the case of Chinese web browsers, the browser itself is sending, out-of-band, the full URL (https://example.com/pagerequested/, and sometimes with with page title ("Example Title") to the browser vendor's servers.

	Baidu	UC	QQ	OPPO	Redmi	Vivo
Activity	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
LAC			🟦🟥	🟦🟥		🟦🟥
GPS Coordinates	🟦	🟦🟥	🟦🟥	🟦	🟦🟥	🟦🟥
MCC + MNC		🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
Screen Size	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
Manufacturer	🟦	🟦🟥		🟦🟥	🟦🟥	
Phone Model	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
OS(Android)	🟦	🟦🟥		🟦🟥	🟦🟥	
OS Version	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
Client IP	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
IMSI		🟦🟥			🟦🟥	🟦🟥
WifiMac			🟦🟥		🟦🟥	
MAC	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
Android ID	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
AD ID	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥
IMEI	🟦	🟦🟥	🟦🟥	🟦🟥	🟦🟥	🟦🟥

Figure 2: Data sent by browsers when *not* in incognito mode. This PII is sent to remote servers in every case (🟦), and in most cases is unencrytped or poorly encrytped (*e.g.*, purely symmetric encryption) so that in- and on-path attackers can easily collect the PII (🟥).

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 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.

UC Browser

We found that UC Browser transmits sensitive data as detailed in Figure 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 such as 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 occurs whether or not the web page being visited uses HTTPS. This behavior is only observed when the browser is not in incognito mode, but other PII is leaked in incognito mode as described below. 🟦🟥

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 POST requests called WUP requests. The body of each request can be encrypted, partially encrypted or unencrypted. Different WUP requests send different information (WUP is a proprietary protocol that is not documented). Everything else is sent on WUP requests which are encrypted using AES and textbook RSA, which is known to be vulnerable [14]. The client uses the AES session key to encrypt the WUP request, in ECB mode.

Additional identifiers not present in Figure 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. 🟦🟥

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 *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 does not send data to sa.api.intl.xiaomi.com, according to [18] they did not find this to be the case for all the apps on Xiaomi'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 uses Baidu as the search engine by default. For this reason it shares all of the search term information with 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 browser's 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 such as 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 in 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. 🟦🟥▼

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 includes a BBK Electronics 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. ◆

	Baidu	UC	QQ	OPPO	Redmi	Vivo
Activity	▼🟦	▼🟦🟥		N/A	▼🟦🟥	N/A
LAC			▼🟦🟥	N/A		N/A
GPS Coordinates	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
MCC + MNC			▼🟦🟥	N/A	▼🟦🟥	N/A
Screen Size	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
Manufacturer	▼🟦	▼🟦🟥		N/A	▼🟦🟥	N/A
Phone Model	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
OS(Android)	▼🟦	▼🟦🟥		N/A	▼🟦🟥	N/A
OS Version	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
Client IP	▼🟦			N/A	▼🟦🟥	N/A
IMSI				N/A	▼🟦🟥	N/A
WifiMac			▼🟦🟥	N/A	▼🟦🟥	N/A
MAC	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
Android ID	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
AD ID	▼🟦	▼🟦🟥	▼🟦🟥	N/A	▼🟦🟥	N/A
IMEI	▼🟦		▼🟦🟥	N/A	▼🟦🟥	N/A

Figure 3: Data sent by browsers in incognito mode (▼). UC Browser adds purely symmetric encryption in Incognito mode, but we still mark it as poor transport security (🟥) because purely symmetric cryptography puts users at risk and is easy to decrypt.

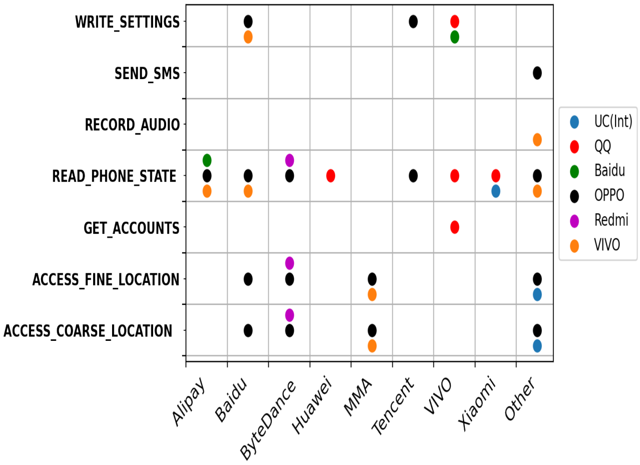


Figure 4: SDK permissions.

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 Figure 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.

5 Discussion and Conclusion

The original BAT browser work by Knockel *et al.*, presented at FOCI 2016 [16], was an eye-opening look into what the Internet traffic for at-risk users in China actually looks like and what various actors might do with access to that traffic. Our work is a follow-up to their work that considers both the newest versions of BAT browsers and the built-in browsers that users are likely to actually use.

The privacy problems we highlight apply to users in China, diaspora populations outside China (such as the Tibetan diaspora), and users of these browsers throughout the world. A simple solution to the problems highlighted in the original BAT browser work and in our own work is for at-risk users to simply not use these browsers. However, there are two considerations that are important to highlight in this context:

- Unless they are informed about the background activities of these browsers and told to do otherwise, users will tend to use the browser that comes built-in with their phone or a browser that is regionalized to suit their needs. Regionalization includes not only visible elements like language, built-in search engines, regionalized suggested content, *etc.*, but also invisible configurations such as large timeouts to work properly on high-delay networks.
- Browsers are not the only apps that collect PII and send it to servers (such as ad servers or telemetry servers) with poor or missing cryptography.

We list specific recommendations for users and developers in the next section. More generally, we recommend that circumvention and privacy tool developers (whether their tools are based on VPNs, proxies, or progressive web applications) consider the applications that users will use with their tools. Even for a VPN with perfect server information distribution, traffic obfuscation, and cryptography, if the app being tunneled is a web browser that sends PII (including user identity, GPS coordinates, network information, web activity, *etc.*) back into the censored domain then identifying and tracking users by colluding with the browser companies is trivial. Poor transport layer security of this PII makes it possible to identify and track users without colluding with the company, *e.g.*, using a national firewall. The same is true for other types of apps, such as Chinese input methods [13, 17] where everything the user types in any application can be easily decrypted at a national firewall while being transmitted from the circumvention server back into China. Also, even apps with better transport security, such as WeChat [21], need to be taken into consideration because users can be linked through the revealed PII, *e.g.*, a journalist outside the country could have their browsing behavior linked to their WeChat contacts.

The lack of transport layer security for some of these browsers should not be discounted. Even though it is true that the data is being sent to companies that possibly collude with state actors anyway, state actors in the Internet backbone can solve many principal–agent problems by harvesting easily decryptable data rather than developing working relationships with every possible software vendor within a country. Also, some actors within the Internet backbone are affiliated with local governments or criminal organizations. Furthermore, not all users of these browsers reside within China, and they may face threats from state actors in the country that they are in. All of this means that poor transport security greatly amplifies threats to at-risk users.

6 Recommendations

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 in some of these browsers 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 has access to 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 sent to servers in China while using these browsers.
  - The poor transport-layer security of some 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 activities can be tied to the individuals they chat with (*e.g.*, in WeChat).

Acknowledgments

This work was primarily supported by the Open Technology Fund. This material is based upon work supported by the National Science Foundation under Grant Number CNS-2141547. We would like to thank the anonymous reviewers for their helpful feedback. No certificate pinning was harmed in the making of this paper.

References

[1] 2023. Genymotion. Available at https://www.genymotion.com/.

[2] 2023. JADX. Available at https://github.com/skylot/jadx.

[3] 2023. mitmproxy. Available at https://mitmproxy.org/.

[4] 2024. statcounter GlobalStats, Mobile Browser Market Share China, 2024. https://gs.statcounter.com/browser-market-share/mobile/china/2024.

[5] 2024. statcounter GlobalStats, Mobile Browser Market Share Worldwide, 2024. https://gs.statcounter.com/browser-market-share/mobile/worldwide/2024.

[6] 2024. statcounter GlobalStats, Mobile Vendor Market Share China, 2024. https://gs.statcounter.com/vendor-market-share/mobile/china/2024.

[7] 2024. statcounter GlobalStats, Search Engine Market Share China, 2024. https://gs.statcounter.com/search-engine-market-share/all/china/2024.

[8] 2025. Download Tor Browser. https://www.torproject.org/download/.

[9] 2025. Frida. Available at https://frida.re/.

[10] 2025. Mobile Security Framework. Available at https://github.com/MobSF/Mobile-Security-Framework-MobSF.

[11] Thomas Brewster. 2020. 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).

[12] Gabi Cirlig. 2021. The 4th largest mobile browser exfiltrates users' data even in Incognito mode. Available at https://hookgab.medium.com/ucbrowser-privacy-study-ecff96fbcee4 (2025/02/13).

[13] Jeffrey Knockel, Zoë Reichert, and Mona Wang. 2023. "Please do not make it public": Vulnerabilities in Sogou Keyboard encryption expose keypresses to network eavesdropping. https://citizenlab.ca/2023/08/vulnerabilities-in-sogou-keyboard-encryption/.

[14] Jeffrey Knockel, Thomas Ristenpart, and Jedidiah R. Crandall. 2018. When Textbook RSA is Used to Protect the Privacy of Hundreds of Millions of Users. *CoRR* abs/1802.03367 (2018). arXiv:1802.03367 http://arxiv.org/abs/1802.03367

[15] Jeffrey Knockel, Adam Senft, and Ron Deibert. 2016. A Tough Nut to Crack: A Further Look at Privacy and Security Issues in UC Browser. Available at https://utoronto.scholaris.ca/server/api/core/bitstreams/2a2dccb4-e704-4721-aaf8-0c96801a0f44/content (2025/02/13).

[16] Jeffrey Knockel, Adam Senft, and Ronald Deibert. 2016. Privacy and Security Issues in BAT Web Browsers. In *6th USENIX Workshop on Free and Open Communications on the Internet (FOCI 16)*. USENIX Association, Austin, TX. https://www.usenix.org/conference/foci16/workshop-program/presentation/knockel

[17] Jeffrey Knockel, Mona Wang, and Zoë Reichert. 2024. The Not-So-Silent Type: Vulnerabilities in Chinese IME Keyboards' Network Security Protocols. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 1701–1715. https://doi.org/10.1145/3658644.3690302

[18] Haoyu Liu, Douglas J. Leith, and Paul Patras. 2023. Android OS Privacy Under the Loupe – A Tale from the East. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM Association for Computing Machinery, 31–42.

[19] Net Alert. 2016. UC Browser Leaks Personal Data. Available at https://netalert.me/uc-browser-leaks-personal-data.html (2025/02/13).

[20] Amogh Pradeep, Álvaro Feal, Julien Gamba, Ashwin Rao, Martina Lindorfer, Narseo Vallina-Rodriguez, and David Choffnes. 2022. Not Your Average App: A Large-scale Privacy Analysis of Android Browsers. https://doi.org/10.48550/ARXIV.2212.03615

[21] Mona Wang, Pellaeon Lin, and Jeffrey Knockel. 2023. Should We Chat? Privacy in the WeChat Ecosystem. https://citizenlab.ca/2023/06/privacy-in-the-wechat-ecosystem-full-report/.

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