A TOUGH NUT TO CRACK

A Further Look at Privacy and Security Issues in UC Browser

By Jeffrey Knockel, Adam Senft, and Ron Deibert

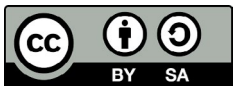
AUGUST 7, 2016
RESEARCH REPORT #77

munk school
OF GLOBAL AFFAIRS & PUBLIC POLICY

UNIVERSITY OF TORONTO

THECITIZENLAB

Copyright



Document Version: 1.0

Suggested Citation

Jeffrey Knockel, Adam Senft, and Ronald Deibert. "A Tough Nut to Crack: A Further Look at Privacy and Security Issues in UC Browser," Citizen Lab Research Report No. 77, University of Toronto, August 2016.

Acknowledgements

Thanks to Andrew Hilts, Sarah McKune, Jason Ng and Masashi Crete-Nishihata for assistance with this report. Jeffrey Knockel's research for this project was supported by the Open Technology Fund's Information Control Fellowship Program and Adam Senft's research from the John D. and Catherine T. MacArthur Foundation (Ronald J. Deibert, Principal Investigator). This material is based upon work supported by the U.S. National Science Foundation under Grant Nos. #1314297, #1420716, #1518523, and
#1518878.

About the Citizen Lab, Munk School of Global Affairs and Public Policy, University of Toronto

The Citizen Lab is an interdisciplinary laboratory based at the Munk School of Global Affairs and Public Policy, University of Toronto, focusing on research, development, and high-level strategic policy and legal engagement at the intersection of information and communication technologies, human rights, and global security.

We use a "mixed methods" approach to research that combines methods from political science, law, computer science, and area studies. Our research includes investigating digital espionage against civil society, documenting Internet filtering and other technologies and practices that impact freedom of expression online, analyzing privacy, security, and information controls of popular applications, and examining transparency and accountability mechanisms relevant to the relationship between corporations and state agencies regarding personal data and other surveillance activities.

Contents

Key findings

- **We identified Windows and Android versions of web browser UC Browser that transmit personally identifiable information with easily decryptable encryption and are or were vulnerable to arbitrary code execution during software updates.**
- **The Windows version analyzed transmits personal user data points, including a user's hard drive serial number and the full URL of all viewed pages, including HTTPS-encrypted sites, with easily decryptable encryption.**
- **The Android versions analyzed transmit personal user data, including a user's IMSI, IMEI, the full URL of all viewed pages (including HTTPs-encrypted pages) and the contents of the search bar, with easily decryptable encryption.**
- **The software update process for the Windows version analyzed is vulnerable to arbitrary code execution, which could allow an attacker to install malicious code without the user's knowledge. Older versions of the Android client are similarly vulnerable to arbitrary code execution during the software update process.**
- **These findings build upon prior findings of insecure data transmission by UC Browser, and are most likely the vulnerabilities identified by members of the Five Eyes intelligence alliance in documents leaked by Edward Snowden.**
- **After we notified the company of these issues they released updated versions of the Android and Windows versions of UC Browser. Please see the "Update: Analysis of updated Android and Windows versions of UC Browser" for our analysis of the changes made to these versions.**
- **We sent the company a letter containing a series of questions about their user data collection practices, and publish their response in full in the Appendix.**

Introduction

UC Browser is a popular mobile web browser developed by UCWeb, who are owned by the Alibaba Group. The application is widely used in Asia, making it by some metrics the second most used mobile browser in the world after Chrome.

In May 2015, we published a report, A Chatty Squirrel: Privacy and Security Issues with UC Browser, which described a number of security concerns we identified

with the application. Our research showed that UC Browser transmitted a number of personally identifiable data points, including a user's IMSI, IMEI, Android ID, geolocation data and search queries, without encryption. The insecure transmission of such data represented a serious privacy risk, since it allowed anyone with access to data traffic to identify users, their devices, and their search history.

This risk was not simply a hypothetical. Documents leaked by Edward Snowden showed that members of the Five Eyes intelligence community (Canada, the United States, United Kingdom, Australia and New Zealand) had already identified vulnerabilities in UC Browser and had actively used these vulnerabilities to target and surveil a set of users.

Our May 2015 report described technical analysis we performed on UC Browser to identify if, and what, personal user data was transmitted insecurely by the application. As described in the report, we identified a number of such vulnerabilities and disclosed them to Alibaba and UCWeb. Although we did not receive any confirmation from Alibaba that the issues identified would be fixed, prior to publication we analyzed the most recently released version of UC Browser (v10.4.1-576) and noted that while some of the security issues we identified were fixed, others remained unresolved.

This report seeks to both update our findings by analyzing a more recent version of the application, and to more precisely identify the vulnerabilities in prior versions that were identified in the publicly disclosed Snowden documents. Our analysis finds that all versions of the browser examined, both Windows and Android, transmit personal user data with easily decryptable encryption. In addition, the Windows version of the application does not properly secure its software update process, leaving it vulnerable to arbitrary code execution.

Our analysis of version 7.9.3.103 of the Android version, released in 2011, shows a number of previously unreported vulnerabilities that are likely the mechanism by which the Five Eyes intelligence community surveilled users.

We are publishing this report in coordination with the inaugural release from Net Alert, a collaborative project which seeks to translate new research on privacy and security into clear messages that explain online threats and what users can do about them.

This report is a continuation of prior Citizen Lab research on the privacy and security of mobile applications in Asia. We have published a series of reports documenting

privacy and security issues in mobile web browsers developed by China's big three Internet giants: Baidu Browser, Tencent's QQ Browser, and our prior research on Alibaba's UC Browser. We have also published a primer on the privacy and security of mobile devices, The Many Identifiers in Our Pockets.

UC Browser Background

UC Browser is a highly popular web browser developed by UCWeb, who are owned by the Alibaba Group. Alibaba is one of the largest tech companies in the world and alongside Baidu and Tencent is one of China's big three technology companies. The company posted total revenue of $12.2 billion USD in 2015, driven primarily by its massive online shopping platforms Taobao and Tmall. Alibaba purchased UCWeb in 2014, in what was at the time the largest ever merger of Chinese tech firms.

In December 2015, UC Browser pulled ahead of Safari to become the second most popular mobile browser globally, behind only Chrome, earning 400 million monthly active users. The company has planned to expand the functionality of the application, allowing media outlets to create "content stores" in the browser, as well as to create a mobile advertising platform called Huichuan that would incorporate collected user data from UC Browser and other Alibaba platforms, such as the Shenma search engine and Alibaba-owned location provider AutoNavi.

UC Browser is available in a number of different versions and for a variety of different operating systems. There are two basic versions of the application: the Chinese-language version and the 'International' version, which has support for multiple languages, including English. Varieties of both of these versions are available for Android, Windows, iOS, Blackberry, Windows Mobile and Symbian. The application advertises a number of features beyond those found in default browsers, such as gesture control, ad blocking, and a download manager.

The company's privacy policy discusses the collection of and transmission of personal user data, differentiating between personal data manually provided by users and automatically collected information. Regarding the latter, the privacy policy states that

> **"[t]o make UCWeb Services more useful to you, our servers (which may be hosted by a third party service provider) may collect information from you, including but not limited to: the IP address, device information, and location of your device or computer."**

The protection of such collected data is also discussed in the privacy policy:

> **"We use a variety of industry-standard security technologies and procedures to help protect your Personal Information from unauthorized access, use, or disclosure."**

Caveats about the limits of data transmission are also mentioned:

> **"No method of transmission over the Internet, or method of electronic storage, is 100% secure, however. Therefore, while UCWeb uses reasonable efforts to protect your Personal Information, UCWeb cannot guarantee its absolute security."**

These statements are important to consider closely and to compare to how the application actually performs in practice. Reverse engineering and closely scrutinizing how the application handles and communicates data back to servers allows us to make this comparison. We discuss these comparisons in more detail later.

Responsible Disclosure & Notification

On April 13, 2016, we submitted a written description of the issues identified in this report to Alibaba, and indicated that we would be publishing our findings no sooner than 45 days after this date, in line with international standards on vulnerability disclosure.

On May 17 and May 24, 2016, Alibaba sent us updated versions of the Android client (Chinese-language and international), which we analyze below in "Update: Analysis of updated Android versions of UC Browser"

We have documented all correspondence with Alibaba related to these security issues in an Appendix at the end of this report.

Technical analysis

We analyzed four different versions of the application for this report, which are summarized in Table 1:

Platform	Version	Version number	Source
Windows	Chinese	5.5.10106.5	http://www.uc.cn/ucbrowser/download/
Android	Chinese	7.9.3.103	http://wap.ucweb.com/verlist/chinese_999/ucbrowser/139
Android	Chinese	10.2.1.161	http://www.uc.cn/ucbrowser/download/
Android	Chinese	10.9.0.703	http://www.uc.cn/ucbrowser/download/

Table 1: Summary of versions of UC Browser analyzed in this report

We analyzed three different versions of the Chinese-language UC Browser for Android: a version downloaded in January 2016 (10.9.0.703), the version examined in our previous report (10.2.1.161), and a version (7.9.3.103) mentioned in slides leaked by Edward Snowden. Those slides suggested that a user running version 7.9.3.103 of the application had their communications surveilled by Western intelligence agencies as a result of the applications' leak of personally identifying data. As we had not previously examined this version of the application, we set out to analyze it to identify which vulnerabilities may have been exploited.

We analyzed both the Windows and Android versions of UC Browser using reverse engineering techniques. To analyze program behavior, we used machine code and bytecode disassemblers, decompilers, and debuggers including JD, JADX, smalidea, and IDA. To capture and analyze network traffic, we used tcpdump and Wireshark.

Our technical analysis is divided into three parts. In Part 1 of this report, we analyze the Chinese versions of UC Browser for Windows for security and privacy vulnerabilities. In Part 2, we similarly examine the Chinese version of UC Browser for Android. In Part 3, we set out which of the vulnerabilities we found in the Chinese versions exist in the international versions.

Part 1: Chinese-language Windows version

We analyzed version 5.5.10106.5 of the UC Browser for Windows, which was downloaded from http://www.uc.cn/ucbrowser/download/.

Our analysis shows three types of security and privacy concerns with this version of UC Browser: the application transmits personally identifying data and the full URL of all pages viewed in the browser with easily decryptable encryption, and has a vulnerable software update process which would allow an attacker to install malicious code.

Leaks identifying data with easily decryptable encryption

Our analysis shows that this version of the application transmits personally identifiable user data with easily decryptable encryption to UCWeb servers during its normal operation.

uc.ucweb.com

On startup, we observed UC Browser making an HTTP POST request to *uc.ucweb.com*. The body of this request is XML containing an encrypted then base64-encoded payload. The encryption algorithm used is an extremely simple, symmetric, and

easily decryptable encryption algorithm we call *UC-XOR* where plaintexts are encrypted by XOR masking with the following 8-byte key:

```
"\xee\xb9\xe9\xb3\x81\x8e\x97\xa7"
```

The resulting ciphertext is then appended with two bytes acting as a checksum. If *cksum* is the result of XOR'ing each byte of the ciphertext, then the two appended bytes are

1) *cksum* XOR'd with the first byte of the 8-byte mask (0xee) and
2) *cksum* XOR'd with the second byte of the 8-byte mask (0xb9).

Code to decrypt this algorithm is available here. When decrypted, these payloads contain miscellaneous details about the user's machine. A sample is below:

```
<assign platform="winnt" reassign="false" prd="UCBrowser"
sn="e2e63e260805aea910e1c2ce02b05211" version="5.5.10106.5" useragent=
"Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/44.0.2403.157 Safari/537.36" last_server="" lang="zh-cn" btype="UC"
pfid="101" bmode="WWW" />
```

These requests contain information about the operating system version and browser version. Most notable is the *sn* value, which is a generated value stored in the registry at *HKEY_CURRENT_USER\Software\UCBrowserPID\MachineID* that acts as a unique identifier to track the user. To calculate this value, the browser first takes the MD5 hash of the concatenation of

1) The string "CS"
2) CPU model
3) The string "DS"
4) The hard drive serial number
5) The string "MB"
6) The base board serial number

Then it takes the MD5 hash again of the resulting hash formatted in lowercase ASCII hexadecimal digits, e.g.,

```
md5(md5("CSIntel(R) Core(TM) i5-4300U CPU @ 1.90GHzDSVB3bb90c33-
fc547c89MB0")) → "e2e63e260805aea910e1c2ce02b05211"
```

mmstat.ucweb.com

We found frequent, unique HTTP GET requests sent to *mmstat.ucweb.com* during startup and during operation of the browser. Each of these requests contains a GET parameter named *encrypt_data*, which contains an encrypted payload which is

then base64-encoded. It is encrypted according to an algorithm we call UC-M9, as the algorithm is referred to internally as "m9" encoding. It is a more sophisticated algorithm than the UC-XOR algorithm in the previous section, but it is still non-standard, symmetric, and easily decryptable. It uses two hard-coded, ASCII-encoded keys "b59e216a" and "8067d108", or when combined into a single key:

```
"b59e216a8067d108"
```

Unlike the previous algorithm which XORs each 8-byte block with the same mask, this algorithm initializes the mask using the first 8 bytes of the key, and for each block the mask is modified as a function of the final 8 bytes of the key. Code to decrypt this algorithm is available here.

Most requests to *mmstat.ucweb.com* do not contain sensitive information. However, one request we found made during startup contains multiple sensitive data points. A sample of such a request is below:

```
bluesky.1.5.1.1.10?cache=3102618000&ka=&kb=e2e63e260805aea910e1c2ce02b0
5211&kc=3b5d366db90b1b60e22260a0278331f8v0000002e9952d46&firstpid
=0501&bid=800&ver=5.5.10106.5&defalutbrowser=UCHTML.AssocFile.HTML&flashver
=&hi=Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz&0&VB3bb90c33
-fc547c89&searchaddress=google&searchbar=google&
searchquick=google&openurltab=0&showsearch=1&showextension=1&applyall=
0&cloudspeed=0&autopage=0&autologin=0&theme_id=569&wallpaper_id
=207&autoclearhistory=0&service=1&sis_fool=5.1.2600_SP3_x86&tch
=0&ad_switch=10&lang=zh-CN
```

These requests contain a parameter named *kb*, which is identical to the *sn* value described in the previous section, stored in the *HKEY_CURRENT_USER\Software\UCBrowserPID\MachineID* registry key and generated using the same machine serial numbers. The *kc* value is similar to the *kb* value, except it is stored in *HKEY_CURRENT_USER\Software\UCBrowserPID\MachineIDEx* and is calculated with an additional MD5 hash and includes the string "DV" followed by the machine's file system volume serial number in the hash. E.g., using the values of our machine:

1) Let *hash* = *md5*(*md5*("CSIntel(R) Core(TM) i5-4300U CPU @ 1.90GHzDSVB3bb90c33-fc547c89MB0DV280522415-")) + "v0000002"
2) Then return *hash* + lastEightNibbles(*md5*(*hash*)) → "3b5d366db90b1b60e22260a0278331f8v0000002e9952d46"

These *mmstat.ucweb.com* requests also include, non-hashed, the machine's hard drive serial number (in this sample, *VB3bb90c33-fc547c89*), the machine's base board serial number (*0*), and the machine's CPU model (*Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz*). It also contains the version of the operating system and of UC Browser, and which browser is installed as the default Windows web browser.

Leaks pages viewed via mmstat.ucweb.com

While browsing, we observed that some of the HTTP GET requests to *mmstat.ucweb.com* also contained in their encrypted payloads the full URL of each page viewed. A sample of such a request is below:

```
bluesky.1.25.1.1.7?cache=3766412000&ka=&kb=e2e63e260805aea910e1c2ce
02b05211&kc=3b5d366db90b1b60e22260a0278331f8v0000002e9952d46&firstpid
=0501&bid=800&ver=5.5.10106.5&type=1&ssl=1&bandwidth=29.63&target_
ip=64.106.20.27&redirect_start=0&redirect_duration=0&dns_start=0&dns_du
ration=218&connect_start=218&connect_duration=251&request_start=469
&request_duration=916&response_start=1385&response_duration=1&dom_start
=1386&dom_duration=268&dom_interactive=234&dom_content_load_start
=1420&dom_content_load_duration=0&load_event_start=1654&load_event_dur
ation=26&t0=1385&t1=1719&t2=1719&t3=1420&total_requests=2&requests_via_
network=2&cloud_acceleration_enabled=0&average_of_request_duration=
809&average_of_t2_duration=859&private_data=host=www.cs.unm.edu|url=
https://www.cs.unm.edu/~jeffk/&lang=zh-CN
```

Note that even though HTTPS normally protects the path of the URL after the host name from being known by eavesdroppers, the full path appears in this easily decryptable request, revealing the individual pages the user browses to anyone eavesdropping on these requests. These requests also contain performance data on the time it takes to download and render the page, which may be their originally intended purpose.

Table 2 summarizes the personal identifiers transmitted by the application with easily decryptable encryption:

Personal data point transmitted	Source
Hard drive serial number	uc.ucweb.com leak
	mmstat.ucweb.com leak
Base board serial number	uc.ucweb.com leak
	mmstat.ucweb.com leak
File system volume serial number	mmstat.ucweb.com leak
Full URL of all pages viewed, including HTTPS full path URLs	mmstat.ucweb.com leak

Table 2: Vulnerabilities found in the Windows version of UC Browser

Vulnerable update process

While running, the browser silently checks for updates in the background. When an updated version is available, the update is downloaded without asymmetric cryptography. The update metadata itself is protected by an MD5 hash, as is each updated file; however, any active attacker could change both the files and their associated MD5 hashes in order to install malicious code during the update process.

The update process uses a complicated system for efficiently sending only the differences in each updated file without having to download the entire updated file. However, we avoided having to learn this system by using a patch type the update system refers to as *Cover*, which simply downloads the entire updated file.

We used this system to "update" ucagent.exe to an arbitrary executable, a benign program that displays "Oh Hai There", as shown in Figure 1. The ucagent.exe executable is run by a UC Browser system service, and so our injected executable runs with full SYSTEM privileges, a Windows user even more privileged than Administrator. Thus, instead of our benign program, an attacker could replace ucagent.exe with malicious code that could be installed and run without the user's knowledge. The update metadata itself is protected by an MD5 hash, as is each updated file; however, any active attacker could change both the files and their associated MD5 hashes in order to install malicious code during the update process.

The update process uses a complicated system for efficiently sending only the differences in each updated file without having to download the entire updated file. However, we avoided having to learn this system by using a patch type the update system refers to as Cover, which simply downloads the entire updated file.

We used this system to "update" ucagent.exe to an arbitrary executable, a benign program that displays "Oh Hai There", as shown in Figure 1. The ucagent.exe executable is run by a UC Browser system service, and so our injected executable runs with full SYSTEM privileges, a Windows user even more privileged than Administrator. Thus, instead of our benign program, an attack could replace ucagent.exe with malicious code that would have been installed and run without the user's knowledge.

Figure 1: Example man-in-the-middle attack on UC Browser's self-updater. When the updater silently runs in the background, we inject our own update that contains an arbitrary program that is then executed with Windows SYSTEM privileges. Our benign program displays "Oh Hai There," but any arbitrary program such as malware or spyware could have been injected.

Part 2: Chinese-language Android version

We analyzed three versions of the Chinese-language UC Browser for the Android platform: 10.9.0.7013, 10.2.1.161 (which was also analyzed in our previous report on the application) and 7.9.3.103 (released in 2011).

As described earlier, documents leaked by Edward Snowden showed that the Five Eyes intelligence alliance had identified vulnerabilities in UC Browser and had actively used those vulnerabilities to surveil a target. Those leaked documents included a screenshot (see Figure 4 below) of data collected through surveillance on targeted UC Browser users, and that data indicated that one of the compromised Android users was running version 7.9.3.103 of UC Browser. In order to more precisely identify the vulnerabilities noted in these leaked documents, we analyzed version 7.9.3.103 of the application.

The results of our analysis of version 10.9.0.703 are described below, and are followed by a comparison to the two older versions of the application. Like the Windows version, our analysis of version 10.9.0.703 of UC Browser for Android found a number of privacy and security concerns. Specifically, the application insecurely transmits personally identifiable user data and all pages viewed in the browser, and has vulnerabilities in the software update process which leave it susceptible to an attacker executing arbitrary code. Table 3 provides a summary of the vulnerabilities and the versions affected by them:

Vulnerability \ Version	**10.9.0.703**	**10.2.1.161**	**7.9.3.103**
uc.ucweb.com leak (IMEI, screen dimensions)	X	X	X
ucus.ucweb.com leak (IMEI, IMSI, screen dimensions)	X	X	X
applog.uc.cn leak (IMEI)	X	X	
sugs.m.sm.cn leak (address bar contents as typed, IMEI, IMSI, screen dimensions)	X	X	
ucsec.ucweb.com:8020 leak (full URL of pages viewed)	X	X[1]	
apilocate.amap.com leak (IMEI, IMSI, Android ID, cell tower)		X[2]	
utop.umengcloud.com leak (IMEI, IMSI, Android ID)	X	X[2]	
puds.ucweb.com leak (IMEI, IMSI, CPU type, screen dimensions)	X	X	X
Proxy leak (URLs and contents of pages visited, IMEI, IMSI)			X
Attack on update process (arbitrary code execution)			X

[1] Only HTTP, not HTTPS, sites
[2] Previously disclosed April 2015 during investigation for a previous report
Table 3: Vulnerabilities and which Android versions of UC Browser they appear in.

Leaks sensitive data

uc.ucweb.com

Like the Windows version, we also observed the Android version making requests to *uc.ucweb.com*. The Android version requests are similar to those made by the Windows version, as they are encrypted using UC-XOR and are just as vulnerable; however, the Android version transmits different data in the requests than the Windows version. Namely, the Android version transmits the IMEI number of the device, the device's screen dimensions in pixels, and the version of UC Browser. In response to this request, the server assigns the browser various parameters including the IP address of a UC proxy server to use for browsing web sites.

ucus.ucweb.com

We observed that the Android version makes requests to *ucus.ucweb.com* upon startup. (The Windows version also makes similar requests, but we did not observe the Windows versions' requests to this host to contain any sensitive data.) These requests are encrypted with the same UC-M9 algorithm as used in the Windows version. This algorithm uses the following hard-coded, ASCII-encoded key:

```
"e19237a3a933f7eb"
```

The code to decrypt these requests is available here.

When decrypted, we found that these requests are protobuf serializations containing the device's IMEI number, the SIM card's IMSI number, and the screen dimensions of the device in pixels, as well as various version information.

applog.uc.cn

Upon startup, we also observed the browser making requests to *applog.uc.cn*. These requests are encrypted with the same UC-M9 algorithm and key as the requests sent to *ucus.ucweb.com*, but the decrypted message is in plain text instead of being serialized into protobuf. These requests contain the device's IMEI number, CPU type, screen dimensions in pixels, and Android OS version.

sugs.m.sm.cn

As a user types into the address bar, the browser sends requests to *sugs.m.sm.cn*. These requests are similar in format to the requests sent to *ucus.ucweb.com* in that they are serialized using protobuf and encrypted with the same UC-M9 algorithm and key. Moreover, these requests include all of the sensitive data that is also sent to *ucus.ucweb.com*, and in addition include the contents of the address bar in order to get auto-suggestions.

utop.umengcloud.com

We found that the browser still leaks information originally reported in April 2015 during investigation for a previous report. These leaks still include the IMEI, IMSI, and Android ID.

Leaks pages viewed via ucsec.ucweb.com:8020

Upon page view, we observed requests being sent to *ucsec.ucweb.com*. These requests are similar in format to the requests sent to *ucus.ucweb.com* in that they are serialized using protobuf and encrypted with the same UC-M9 algorithm and key. However, instead of containing hardware serial numbers, they contain the full URL of each page viewed, including those of HTTPS pages.

Proxying

Unlike the Windows version, the Android version of the browser proxies HTTP requests through UC servers that perform data compression and accelerated browsing. The browser proxies HTTP requests through a server assigned to the browser in the response to the request to *ucus.ucweb.com* described earlier. We found that when the requested URL is unencrypted HTTP, the communication with the proxy is unencrypted; however, when requesting an HTTPS URL, the communication does not go through the proxy.

Update process

The browser checks for updates by making an HTTP POST request to *puds.ucweb.com*. These requests are similar in format to the requests sent to *ucus.ucweb.com* in that they are serialized using protobuf and encrypted with the same algorithm and key. Moreover, they also contain the phone's IMEI number, the SIM card's IMSI number, and the screen dimensions of the phone in pixels. The requests additionally contain Android OS version and CPU type.

The server's responses to the update checks are similar in format and also M9-encrypted except they are encrypted with the following hard-coded, ASCII-encoded key:

```
"aa171021f9438cb2"
```

These responses do not contain any personal user data, but they do contain the URL for the Android Application Package (APK) that the browser will download to upgrade the application, alongside that APK file's MD5 hash. Since this URL and the

MD5 hash are encrypted using only symmetric encryption, it is possible to perform a man-in-the-middle attack to cause the browser to download an arbitrary APK file. However, the version of UC Browser we analyzed verifies the digital signature of the APK downloaded before the system prompts the user to install it, meaning that an attacker can only prompt the user to install an APK signed by UCWeb. The Android system prevents users from downgrading apps, so this attack could not be used to downgrade UC Browser to an older version, but there may be other UCWeb applications signed with the same key as UC Browser, and an attacker may still be able to trick a user into installing and running a different UCWeb app, even one that contained a vulnerability. More work is required to investigate this attack.

Comparing with older Android versions

In this section, we compare the newest version that we analyzed (10.9.0.703) to older versions of the browser. In particular, we compare it to version 10.2.1.161, the version analyzed in our previous report on UC Browser and version 7.9.3.103, a version whose vulnerabilities were shown to have been exploited in slides released by Edward Snowden.

To obtain version 7.9.3.103 of the APK, we Googled for "7.9.3.103", the exact version referenced in the slides, and found a page on ucweb.com offering historical versions of the browser for download. As a precaution, we also checked the digital signature on the APK and verified that it was untampered and signed using the same private key as the other Android versions of the browser that we analyzed.

Our analysis found that one of the two vulnerabilities described in the previous report is not present in version 10.9.0.703, and we found that three of the newly discovered vulnerabilities described here date back all the way to at least version 7.9.3.103 released in late 2011. We also found two additional vulnerabilities in 7.9.3.103, a privacy leak in the browser's proxy implementation and a vulnerability in the browser's update process. Both of these vulnerabilities had been fixed before version 10.2.1.161 and 10.9.0.703. We describe these vulnerabilities in the following two sections.

Proxy information leak in version 7.9.3.103

Unlike version 10.9.0.703 which only uses the proxy for unencrypted HTTP requests, we found that version 7.9.3.103 of the Android version of UC Browser also sends HTTPS requests through the proxy. The requests to the proxy are encrypted using *UC-XOR*, as described earlier. When decrypted, in version 7.9.3.103, each request includes the full requested URL, even if it is HTTPS, along with the IMEI and IMSI

numbers of the device. In 7.9.3.103, we found the response from the proxy is not encrypted and contains the full URL of the request. Moreover, by using binwalk, we were able to automatically extract lzma-compressed sections of the response. We found that these contained the contents of the page, including the full text of the page, thus eliminating the expected encryption of HTTPS.

Vulnerable update process in version 7.9.3.103

We also found that version 7.9.3.103 checks for updates in a process similar to the one we describe in version 10.9.0.703. However, in version 7.9.3.103, the server's responses are not encrypted at all, and, crucially, the update process does not perform digital signature verification of the downloaded APK file. This allows an attacker to perform a man-in-the-middle attack to trick a user into installing an arbitrary APK file that may contain malware or spyware (see Figure 2).

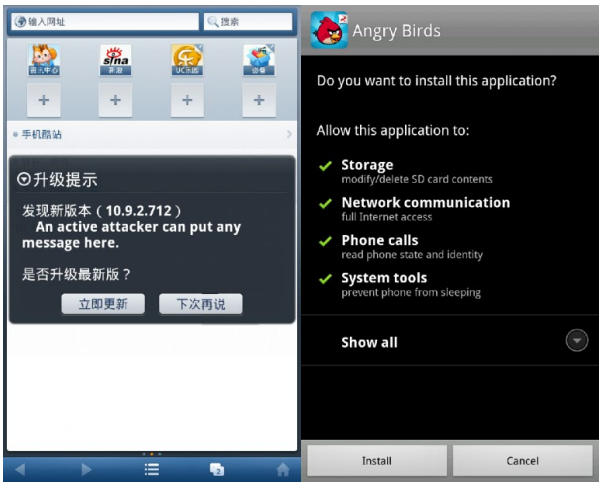


Figure 2: *Example man-in-the-middle attack on UC Browser's updater. On the left, we injected a custom update description. On the right, after the update is downloaded, the browser prompts the user to install the Angry Birds APK (an actual attacker might instead craft an app called "UC Browser" with an icon similar to that of UC Browser to further convince the user to install it).*

Connection with leaked intelligence agency slides

In this report, we have discovered numerous sensitive data leaks in the latest version of UC Browser. Both data leaks discovered in the previous report appear to have been introduced into the browser after the version 7.9.3.103 of the browser analyzed by Western intelligence agencies in 2012. However, we found that many of the vulnerabilities that we outline in this report also exist in version 7.9.3.103, and the contents of the leaks match the description in the intelligence agencies' slides, as they also describe the leaks as divulging IMEI and IMSI numbers in addition to "device characterics." Moreover, the slides reference finding vendor update servers and having the capability of pushing "malware" to victims' devices. This is consistent with the man-in-the-middle vulnerability we discovered in the update

process of version 7.9.3.103. Thus, many of the vulnerabilities outlined in this report are strong candidates as the same ones referenced in the leaked intelligence agency slides.

Part 3: Comparison to International Versions

Our analysis in this report thus far concerns Chinese versions of UC Browser. However, UCWeb also produces "international" versions of the browser intended for users outside of China. We examined two of these international versions, one for Windows and one for Android, to determine if the same security and privacy vulnerabilities found on the Chinese versions were also present on the international versions. We looked at the following versions:

Platform	Version	Version number	Source
Windows	International	5.5.9936.1231	http://www.ucweb.com/ucbrowser/download/
Android	International	10.9.0.731	http://www.ucweb.com/ucbrowser/download/

Our analysis showed that this international version of the Windows client contained the same vulnerabilities as those found in the Chinese version (5.5.10106.5).

Our analysis of the international Android version showed that it shared some, but not all, of the data leaks compared to the Chinese version (10.9.0.703). Although it has many of the same sensitive information leaks, we did not observe any traffic going to *applog.uc.cn*, *sugs.m.sm.cn*, or *ucsec.ucweb.com*. Like the Chinese version, we did observe it using a proxy when connecting to HTTP sites but not with HTTPS sites.

Table 4 summarizes the types of vulnerabilities found in the most recent versions of the application analyzed in this report:

Vulnerability/Version	**WindowsChinese** (5.5.10106.5)	WindowsInternational (5.5.9936.1231)	AndroidChinese (10.9.0.703)	AndroidInternational (10.9.0.731)
Leaks personally identifiable data	X	X	X	X
Leaks pages viewed	X	X	X	
Vulnerable software update process	X	X		

Table 4: Summary of vulnerabilities in most recent version of client

Discussion

The privacy and security issues we identify in this report are concerning for all users of UC Browser. The browser leaks the personal data of users, likely without their knowledge and in a manner which we know leaves it subject to surveillance. However, the issues identified here extend beyond this particular application and beyond this one developer. As our prior research on UC Browser, Baidu Browser and QQ Browser has shown, even within the narrow category of web browsers developed by large Chinese companies, a remarkably similar set of privacy and security issues have emerged.

It is important to note that the privacy and security risks identified here are not merely hypothetical. As documents disclosed by Edward Snowden show, western intelligence agencies identified and developed a surveillance plugin exploiting the data leakage in UC Browser -- the second most widely used mobile browser in the world -- no later than 2012, and have in all likelihood been collecting the private data of users since that time. The slide shown in Figure 3 shows the specific data points that intelligence agencies had collected from the application, and our analysis here reveals the vulnerabilities that leak this data.

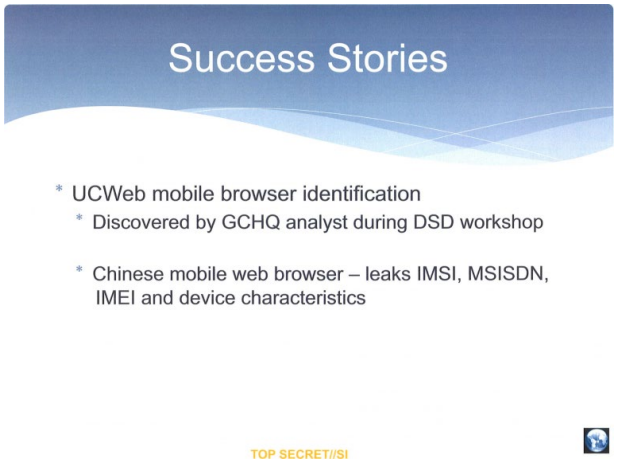


Figure 3: Slide of Five Eyes intelligence agencies presentation describing UC Browser vulnerability

The discovery of this application's leakage of data allowed the NSA to create a plugin for XKEYSCORE, the spy organization's massive, comprehensive searchable database of collected Internet traffic. The slide shown in Figure 4 shows a screenshot of this UC Browser XKEYSCORE plugin, illustrating that the IMEI, IMSI, device model and e-mail address of users are actively collected and searchable:

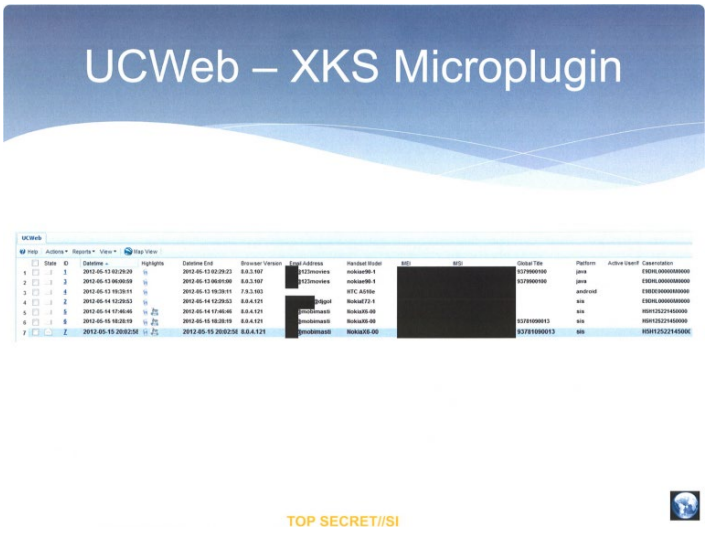


Figure 4: Slide showing XKEYSCORE microplugin screenshot, showing leaked data from UC Browser users

Further, if Five Eyes intelligence agencies have identified these vulnerabilities as a way to track and target users, it is highly likely that other actors, such as other national intelligence agencies or non-state criminal actors, could have done the same. Although there are no known reports of similar targeting of users of QQ Browser and Baidu Browser, signals intelligence agencies and criminal organizations do not, as a rule, openly disclose their tradecraft; public leaks, such as those coming from the Edward Snowden disclosures, are rare. We can, however, safely assert that users of UC Browser have been open to *potential* exploitation by such actors for many years based on the privacy and security vulnerabilities we have identified.

As discussed in Part 2, the data leakage exploited by intelligence agencies to collect this data has likely remained unfixed since at least 2012. Although we notified the company in April 2015 of vulnerabilities which leaked data points including a user's IMSI, IMEI and geolocation data, the issues we identify in this report have remained unaddressed until our April 2016 disclosure. In other words, despite the company's knowledge that vulnerabilities in UC Browser were actively being exploited by intelligence agencies to target users, they do not appear to have examined beyond the subset of problems we initially disclosed to identify similar vulnerabilities elsewhere in the application.

This case illustrates the 'whack-a-mole' nature of identifying and fixing these kinds of vulnerabilities. Applications like UC Browser are continually updated with new features and services, across numerous different platforms and localized versions. Unless developers make the privacy and security of user data a more fundamental part of their development process, security researchers will continue to play 'catch-up' in efforts to find these problems and report them.

UC Browser's security concerns are not limited to the insecure transmission of personal user data. The vulnerabilities found in the browser's update process would allow an attacker with a privileged network position to install malicious software onto the device, potentially with minimal warning to the user. Like the leak of sensitive data, we know from slides disclosed by Edward Snowden that this vulnerability was known and likely exploited by intelligence agencies, as shown in Figure 5:

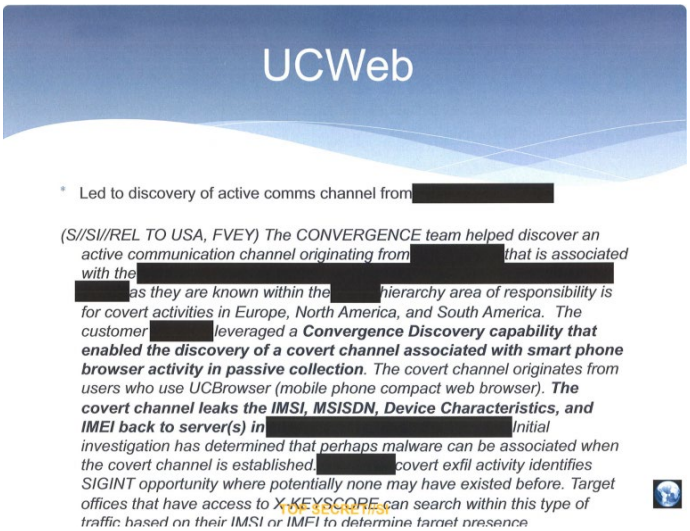


Figure 5: Slide describing the installation of malware onto the devices of UC Browser users

One of the causes of the software update vulnerability described in our report is China's own system of Internet censorship. Specifically, the blocking of the Google Play Store in China forces developers to create home-grown methods for updating their code, exposing another vector which can be exploited by attackers. The problem of vulnerable software update processes has been previously discussed. Underscoring the complexity of securely updating software, other documents disclosed by Snowden indicated that the Five Eyes intelligence alliance also sought to deliver targeted malware by hijacking software updates delivered to the Google and Samsung app stores.

While the challenge of ensuring the security and integrity of user data and devices is complex, there are established practices that developers can follow. The secure transmission of sensitive data should follow industry-standard practices, like the use of asymmetric encryption and well-tested implementations such as OpenSSL. Although Alibaba's Privacy Policy states that the company uses "a variety of industry-standard security technologies and procedures to help protect your Personal Information", the use of non-standard, symmetric encryption algorithms suggests otherwise.

However, transmitting user data in a more secure manner is only the first step of the data processing chain. Companies who collect the abundant data generated by our devices will store, analyze and potentially share that data; additional steps that must also be performed in a more secure manner. While implementing more secure methods of transmission in UC Browser may limit the risk data is intercepted in transit, it does not secure such data as it is stored, used and shared by Alibaba.

Perhaps the most straightforward way of ensuring the integrity of user data is simply not to collect it in the first place. While many companies, including Alibaba, seek to leverage the vast quantity of user data they have available for their own competitive purposes, minimizing what data points are collected is the simplest way of ensuring such data is not misused or exposed to unnecessary risk. As the ecosystems of companies such as Alibaba continue to expand, incorporating new applications and features, without a more diligent approach focused on user privacy, the risks to users will only continue to grow.

Update: Analysis of updated Android and Windows versions of UC Browser

On May 17, 2016, Alibaba sent us two updated Android APK's for testing: versions 10.9.9.739 and 10.9.0.731 of the Chinese and international editions, respectively, of UC Browser. Our analysis of these APKs found that most of the previously identified data leaks had been fixed in these versions by using SSL to transmit sensitive data. The two exceptions were transmissions to *sugs.m.sm.cn* and *puds.ucweb.com*. While our analysis found that the browser no longer sends sensitive data to *puds.ucweb.com*, we did find that search terms entered into the address bar were still sent to *sugs.m.sm.cn* in an insecure manner. For transmission to both of these domains, the updated versions switched to a different symmetric algorithm which was equally as insecure as the version it replaced. The company identified that they would eventually switch transmission of data to these two sources to SSL by an unstated future date. In addition to the problems noted above, these versions still had unfixed data leaks to *applog.uc.cn* and *utop.umengcloud.com*.

After we notified the company of these issues, on May 24, 2016 they sent us two further updated Android APKs for testing: version 10.10.0.800 (Chinese) and 10.10.0.796 (international). These versions no longer leaked data to *applog.uc.cn* and *utop.umengcloud.com*; however, the use of the easily decrypted symmetric algorithm to transmit data to *sugs.m.sm.cn* and *puds.ucweb.com* was still present.

On May 27, 2016, Alibaba notified us that they had fixed all the reported issues with the Windows versions of the browser, and that updated versions were available on their website. On June 3, 2016, we downloaded and analyzed the latest Windows versions of the Chinese (5.6.12860.10) and international (5.6.12265.1017) editions of UC Browser. We found that all leaks we reported were fixed by switching to the use of SSL. Moreover, the software update vulnerability was also fixed by using SSL to protect the downloaded update metadata.

Appendix

On June 3, 2016, we sent a letter to Alibaba with additional questions about the security vulnerabilities we identified. The letter is reproduced here.

On June 8, 2016, a representative from Alibaba sent us the following response:

> Hi Ronald,
> Nice to meet you by e-mail. We appreciate your team's collaboration and interest in raising security matters for our consideration and discussion. We are quite happy to work with you, and we consider your recommendations to be helpful and in parallel with our ongoing product development process. Our technical team has confirmed as of end of May 2016 that the data security items raised by Citizen Lab have been fixed. As you know, our team has forwarded both the Android and PC upgraded versions to you at the end of May for your team's review and we have been pleased that your team confirmed our revisions to the Android sufficiently addressed your concerns. We look forward to your feedback on the upgraded PC version.
>
> While we appreciate and worked to address the items you have raised, we do not believe that consumer personal information has been placed at risk by use of our browser.
>
> We take our users' privacy seriously and we rely on universal privacy principles to guide our business. We strive to adhere to the principle of data minimization by collecting and storing data that is actually used to provide our services Our data collection and use practices are consistent with general industry norms. We will of course continue to review our practices and to revise and improve our policies when appropriate.
>
> UCWeb was acquired by Alibaba in 2014 and operates independently as a subsidiary company. We do not have insight into the data security or privacy practices of Alibaba generally, or of other products which are offered by it. We are not able to respond to issues you have raised about other Alibaba products.
>
> Thank you again for contacting us. We will keep working to improve our product and service, and we welcome constructive conversations with third parties that may lead to additional improvements and innovations at UCWeb.

The following table lists our all communications with Alibaba related to the security and privacy issues we identified in UC Browser:

Date	Contact
April 12, 2016	We contacted Alibaba in order to determine who to submit our disclosure to.
April 13, 2016	We submit written description of technical findings via email
April 14, 2016	We have an initial call with UCWeb/Alibaba security engineers
April 18, 2016	Alibaba request a follow-up call to discuss the vulnerabilities
April 19, 2016	We have another phone call with Alibaba engineers. They commit to implementing a series of fixes on the Android client within 6 weeks: They will no longer collect the IMEI or IMSI of users; they will use asymmetric encryption to transfer some collected user data. They also commit to encrypting the transmission of the URLs of all pages viewed in the browser at a later date. They commit to sending us an email describing these proposed fixes in more detail. There are no commitments regarding the Windows version of the client.
April 25, 2016	Alibaba sends an email outlining the fixes they will make to the Windows version of UC Browser. They will begin using HTTPS to correct the issues we identified in Part 1 of this report: the leaking of identifying data with easily decryptable encryption, the leaking of pages via mmstatt.ucweb.com and the vulernable update process. They also note that they do not consider the sn value (which contains numerous hardware serial numbers) to be personally identifiable information.
April 26, 2016	Alibaba confirms that the fixes described on April 25th apply to both the Chinese and international versions of the Windows client. In addition, they state they will be contacting us at a future date to discuss fixes for the Android client.
April 29, 2016	Alibaba sends a summary of proposed fixes for both the Windows and Android versions of the client, and commits to implementing these fixes by May 15, 2016. They outline a series of fixes for the Android client, including using HTTPS in place of HTTP, using a “new encryption algorithm”, and no longer sending the IMEI/IMSI in some, but not all, of the data leaks we identified. The proposed fixes indicate that personal data points such as the user IMSI and IMEI will continue to be sent in some cases, but with HTTPS encryption. In addition, the software update process will be protected by a “new encryption algorithm”.
April 30, 2016	Alibaba confirms that the they will implement a new encryption algorithm in the Android version to replace the algorithms used during the software update process and during the transmission of data to sugs.m.sm.cn.
May 17, 2016	Alibaba provides APks of updated Chinese and International versions of the Android client for review.

Date	Contact
May 21, 2016	We identified a number of problems which remained in the updated APKs sent May 17, 2016. User data including IMEI was sent to applog.uc.cn via HTTP.
May 24, 2016	Alibaba provided two updated APKs in response to the issues we identified on May 21. They state that they intend to switch the transmission of some data points to HTTPS at a future date.
May 27, 2016	Alibaba notifies us that all the issues we identified in the Windows version have been fixed and the updated installers were available for download.
June 3, 2016	We sent Alibaba a letter with additional questions about privacy and security issues in UC Browser.
June 8, 2016	Alibaba representative responds to our letter