INTER SECLAB

The Max Messenger

An Analysis of Russia's State-Mandated Messaging Application

Authors:
- Anna Enders
- Lea Horne
- Marla Rivera

* The names of authors are displayed in alphabetical order.

Editor:
- Laura Schwartz-Henderson

Editorial Support:
- Curious Shapes

How to cite:
INTERSECLAB. The Max Messenger: An Analysis of Russia’s State-Mandated Messaging Application. InterSecLab, September 2026.

Available at: *interseclab.org/research/max*

Table of Contents

An introductory note from the authors

The Russian government has spent years building out extensive surveillance infrastructure. Under SORM (System for Operative Investigative Activities), operators and information services must give the FSB technical access to user data. Through the TSPU program (Technical Means of Countering Threats), Russia's internet and media regulator, Roskomnadzor, can run deep packet inspection equipment inside operators' networks, filtering and throttling all traffic. Between these two mechanisms, the state has had a very good view of who is connecting on the network, from where, to what, and when.

Despite these extensive monitoring and blocking capabilities, until recently, a gap still existed: messages sent through end-to-end encrypted tools, such as WhatsApp or Telegram's secret chats, could be seen as traffic but not read as content.

In September 2025, the Russian government mandated the pre-installation of a new app, Max, for all phones and tablets sold in the country. At the same time, all other messaging alternatives were removed (Signal blocked in August 2024, WhatsApp in February 2026, and Telegram mass-blocked from March 2026). Max is not just a new messenger for Russians. It is a replacement of all widely used end-to-end encrypted messaging options with an app built by a company controlled by Gazprom and run by the son of a senior Kremlin official.

Given what Max's rollout means for cybersecurity and internet freedom, researchers have scrambled to figure out how to better understand what the app actually does. However, Max was built to resist external examination. It doesn't use standard web encryption. Instead it runs a proprietary binary protocol, wrapped in Russian federal cryptography implemented by a vendor licensed by the FSB. It shuts itself down when it detects analysis tools, and it stores the addresses it contacts as obfuscated integer arrays instead of readable text. Researchers trying to analyze the app have run into this wall directly. RKS Global's September 2025 study monitored a phone with Max installed over 48-hour periods, identifying which hosts Max contacted but could not decrypt what was being exchanged. A separate technical analysis published in March 2026 on the Russian-language website, Habr, worked from the decompiled code. Both produced real findings, several of which we corroborate here, and both ran into the same limit: neither observed in detail what the app was actually doing.

For this research, we wanted to complement that static picture with dynamic analysis, watching Max run in real time rather than just looking at its code. This matters because code that is present, complete and readable can still never run. The routine that would start it might never get called. A permission it needs might never be declared. In this way, an inventory built from code alone counts those capabilities regardless of whether they run. We treated each capability found in the code as a question rather than as a finding, not counting the capability until dynamic testing either confirmed it or showed the path was not connected. Some capabilities that are fully present in the code never ran at all. And some of what Max does only became visible once we watched it: the application receives instructions from its servers that appear nowhere in the code we could read.

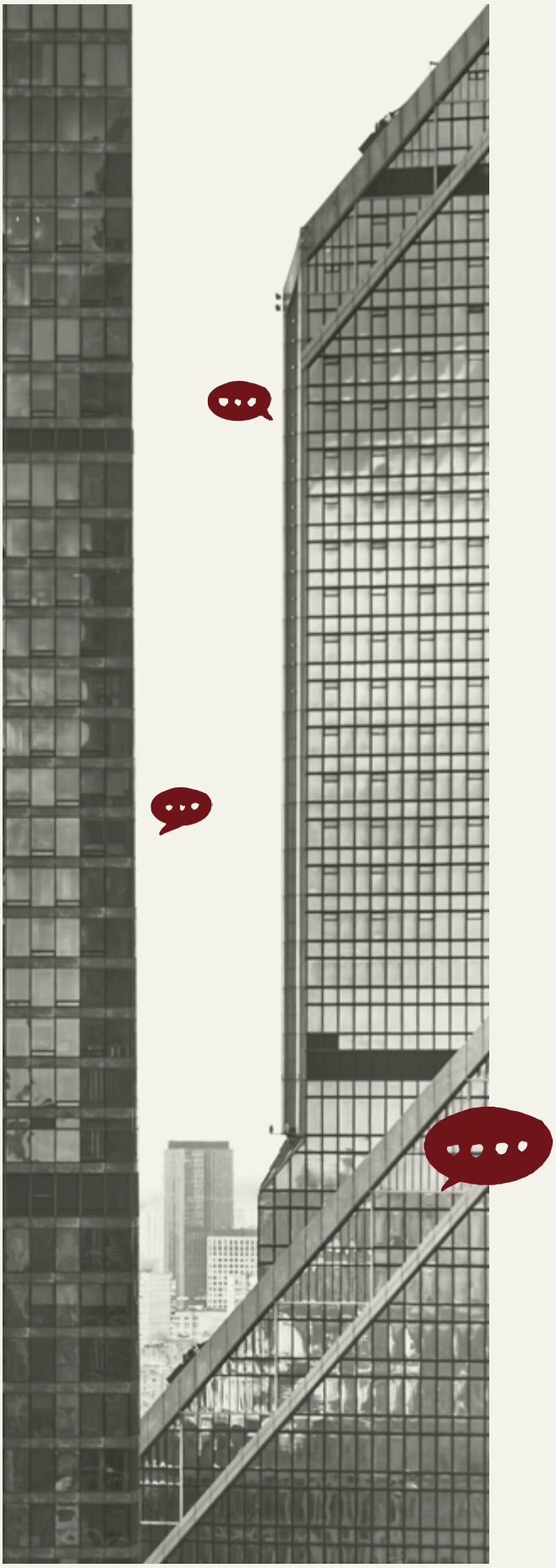
As we were finalizing this report, a team from four universities read Max's traffic by the same route we did and published its findings independently in September 2026. Their focus was different. They studied what Android's architecture allows any super-app to do to the mini-apps running inside it, using Max as a case study. Their five findings and ours don't overlap, as this research sought to understand what Max itself actually collects, from whom, and under what conditions.

We think this kind of analysis is necessary to understand how Max really works and what that means for users in Russia and in the diaspora who need to know what data is collected, where it goes, and what happens if they try to evade Max's controls or use it alongside circumvention tools. They need to know which capabilities are running and affecting them today, and which are dormant in the code but could be turned on tomorrow. These findings also matter for circumvention tool developers who need to understand how Max detects and blocks their tools in enough detail to respond to it. Additionally, lawmakers weighing sanctions or app store removals need claims backed by evidence that hold when the operator disputes them. All of this requires the same thing: separating what we observed from what we inferred, and saying which is which every time.

So we built a way to read Max's traffic, capturing it inside the app in the moment before it is encrypted, and we checked everything we saw against the code that produced it. This report is the result. It is not a vulnerability disclosure and it does not claim to be complete.

This research is also a snapshot of one moment of Max's deployment. We find that Max's capabilities are controlled by flags set on VK's servers, which can be changed per account, in real time, without updating the app and without any indication to the user. The capabilities can also change with the app itself, as each release potentially introduces new code or connects a path that was present but never called. What we found dormant in the code was dormant only for our accounts, in our build, during our testing window. A different user could receive a different Max, one where these dormant capabilities are active, and they would have no way to tell. The fact that capabilities can vary silently from account to account is itself a finding, and it means no single analysis can describe Max in full. It is also why this kind of work cannot be done only once. We have sought to describe our method in enough detail for other researchers to repeat it, extend it and test it against later builds, and we encourage continual work to examine how Max's deployment evolves.

Finally, we want to thank our partner RKS Global for all the support that made this research possible, including the inspiration and the motivation, but also the provisioning of Max accounts for performing the research, the sharing of internal briefings, the translation of this report into Russian, and their review of drafts of this work throughout its preparation.



CHAPTER 01

Executive Summary

Source: Pexels

In September 2025, Russia made Max, a messaging application developed by VK Group, compulsory. Under an order signed by Prime Minister Mikhail Mishustin, Max must be pre-installed on every smartphone and tablet sold in the country. At the same time, other messaging applications used by Russians were cut off: Signal was blocked in August 2024, WhatsApp was blocked in February 2026 with its domains stripped from Russia's national DNS servers, and Russia began mass blocking of Telegram in mid-March 2026.

Max is not just a messaging app. It is now reportedly required to access government services in Russia and in Russian-occupied areas of Ukraine, including enrolling a child in school. With other messaging apps removed and the app pre-installed by default, Max is designed to be universally used across the country. It is therefore necessary to examine how this application works, what information is collected, and what this means for internet users in Russia.

This report is the public summary of a nine-week technical investigation of Max conducted by InterSecLab between March and May 2026. We analyzed version 26.12.0, build 6664, of the Android application, distributed as the package *ru.oneme.app*.

How this analysis was done

Max is built to resist inspection. It does not use the standard web encryption that most applications rely on. The encryption it does use is a Russian federal standard implemented by a company licensed by the FSB (Federal Security Service of the Russian Federation), which puts its traffic beyond the reach of the tools researchers normally use. The application shuts itself down when it detects analysis software, and it hides the addresses it contacts inside the code rather than storing them as readable text. To read what Max sends, we captured it inside the application itself, in the moment before it is encrypted, and then checked every observation against the application's own code.

Checking observed behavior against the code matters because an application's code only tells you what it is capable of and not what it does. Max bundles a component that can list every application installed on a phone, but in the build we examined, that component was never switched on. When we say Max does something in this report, we mean we either watched it happen or traced the path that makes it happen. We always note whether it was watched or traced. Several of the findings below were reached independently by other researchers working on other builds while our analysis was underway. Where separate teams reached the same result, we highlight the corroboration. Where our evidence differs from theirs, we highlight that too and show our work. Where our findings differ from other accounts, that is not necessarily a contradiction. Because VK can enable and disable capabilities per account and per build, two analyses of Max can both be accurate and still disagree.

Key findings

01 **Max has no end-to-end encryption, and "secret chats" are not encrypted.** Every message is readable by VK's servers. We documented this directly, capturing a message in plaintext at the moment Max handed it to the cryptographic library. The "secret chat" feature is a disappearing-message timer and nothing more. The name seems misleading on purpose: in Russia, Telegram used the same term to mean specifically its end-to-end encrypted mode.

02 **Max has extensive surveillance capabilities.** These capabilities, as we documented them, include no end-to-end encryption, with message content readable by VK's servers; the entire address book uploaded in cleartext, names paired with numbers; voice messages transcribed on VK's servers; live call audio routed, according to the code, to an on-device model that VK supplies from its servers; network reachability probing reported alongside carrier and IP address; a second, concealed probing channel that takes an unrestricted list of targets from the server; VPN detection and the blocking of message sending; screen-by-screen behavioral telemetry; real-time presence for accounts that are not contacts; and lookup of any registered phone number without notification to the person looked up. One additional capability is present in the code but was not running in the build we examined: enumeration of every application installed on the phone.

03 **These surveillance capabilities are switched on and off from VK's servers, per account, in real time.** Network probing, VPN detection, voice transcription, elevated logging, the list of services allowed to receive a user's identity: all are gated behind server-pushed feature flags that can be flipped on for one individual user, with no application update and no indication in the interface. This is the single most important technical fact in the report. What Max does today is not what Max is. Any capability we found dormant can be live tomorrow for one person and no one else, and that person will not be able to tell.

04 **Max reports the user's network environment to VK on every session.** Each time the application opens or moves to the background, it looks up the device's public IP address through as many as six external services, checks whether a VPN is running, reads the mobile carrier, and tests the reachability of a list of internet services. All of it goes to VK in one report. The IP address alone gives VK an approximate location for every user, refreshed every time the application is opened. Telegram and WhatsApp were on that probe list until the practice was reported publicly in March 2026. Reachability results, paired with the carrier and the apparent IP address, are precisely the data needed to map which Russian networks are complying with blocking orders and which are not.

05 **VK can aim Max's network testing at anything it chooses, from the user's own device and network position.** Separately from the checks above, Max carries a reporting component that asks VK's servers for a list of addresses to test. Nothing in the application restricts what may be on that list. For each address the phone tries to resolve the name,

open a connection, and complete an encrypted connection, reporting each result separately alongside the user's account identifier, device identifier, apparent IP address, mobile operator, and VPN status. The server also sets how often this runs. The setting that starts this channel is off by default, but it was switched on for our test accounts. We found the endpoint only after decoding a hostname the application stores in scrambled form, then searched 26 days of recorded traffic: the phones contacted it nineteen times, on 21 April, 14 May, and 22 May 2026, completing an encrypted exchange each time. Immediately before each report, the device tested a short list of external addresses. Those addresses included the Russian state services portal, the push-notification services of Google, Apple and Huawei, and Max's own servers. One of them, Apple's push service, appears nowhere in the application in any character encoding we tested, so it can only have been supplied by the server. What it amounts to is a measurement capability, running from the user's device and network position, that VK can aim at anything it chooses, for one account or for many, with no application update and nothing shown to the user.

06 **Max stops working when it finds a VPN.** We observed this directly. An attempt to reply to a message produced a full-screen instruction to disable the VPN, with no way around it. Because the check looks for a VPN interface on the device itself, it can be evaded by moving the VPN to a router. The effect on a user who does not know that is a forced choice: keep the tool that reaches independent media and blocked services, or keep the application required to reach state services.

07 **Max uploads the user's entire address book to VK in cleartext.** Names are paired with numbers, with none of the privacy-preserving contact discovery that other messengers use. A change to any single contact triggers a real-time sync, and the total size of the address book is reported to the server every session, including contacts who have never used Max. We captured this on the wire during a fresh account registration: the phone numbers cross the network as plain digits. This is important because this is the clearest point where what we observed contradicts what has been published about Max. A widely cited analysis published on Habr in May 2026 by a researcher using the name "zarazaexe", working from the application's code, reported that the numbers are hashed before upload. We find that they are not. The routine that had been read as a hash is a formatting normalizer, and it invokes no cryptographic function at all. The finding is worse than the one in circulation, and we have the capture.

08 **Looking up any registered number returns that account's identifier, display name, and creation time, and the server acts on the query without telling users.** The query returns the target's identifier, display name, and account creation time. It requires no contact relationship, triggers no notification, and we could detect no rate limiting. It also does two things nobody asked for. Within 1.2 seconds, VK's servers began streaming the target's real-time online status to the querying account and kept streaming it for over fifteen minutes. In the same window, the server created a chat object linking the two accounts. The link between two people was established by looking one of them up, whether they know each other or not. We have not seen this reported in any other analysis of Max.

09 **The server reads content, and the machinery to read more is already in place.** Every text message carries a flag instructing the server to inspect it for links and shared content. Voice messages are transcribed on VK's servers, not on the device, incoming and outgoing alike. During voice calls, live audio is fed to an on-device machine learning model, downloaded at runtime from a server-provided URL and validated against a server-provided checksum. The pipeline will run any TensorFlow Lite model VK chooses to deploy. In the build we analyzed, the deployed model listened for one Russian phrase and reported a call-quality metric. What matters is not the model we found but the mechanism that delivers it: what the pipeline listens for is set by VK's servers rather than fixed in the application, and it can be changed without an update.

10 **Everything is inside one jurisdiction.** Max's backend sits in a block of 4,096 IP addresses allocated to LLC VK in Moscow on September 5, 2025, the month the pre-installation mandate took effect. Every server Max contacted for its core functions is Russian-operated, Russian-hosted, and reachable through Russian transit. Its telemetry goes to VK Group properties, including the second monitoring channel described above, whose destination is concealed in the application rather than written plainly. That concealment does not hide a third party. The destination, *trace-flow.ru*, resolves inside VK's own network, and the certificate the endpoint presented throughout the period we recorded was issued to VK LLC of Moscow, after the certificate authority had verified that company's identity. Both are matters of public record and can be checked independently of anything we observed.

11 **A fresh account is pre-loaded with state media before it is used.** A test account registered with a Russian number in April 2026, before searching or browsing anything, was given a list of twenty-five suggested channels. Ten were state media, sanctioned officials, state television correspondents, or pro-war military commentators, among them RIA Novosti, NTV, Izvestia, Vladimir Solovyov, and Dmitry Medvedev. There are no independent outlets, no opposition voices, no minority-language content, and no international news.

What we do not claim

This report describes what Max can do. It does not claim that VK Group or any Russian state body has used any specific capability against any specific person, and we collected no evidence of such use.

We were also able to rule things out. The component that can enumerate every application installed on a phone is present in Max but was never initialized in the build we examined and could not function without a permission the application does not declare. A plaintext push-notification path reported in RKS Global's dossier did not exist in our build.

Our findings are a snapshot of one build, tested in our lab, at one moment. VK can reconfigure what Max does for each account from its servers, so no analysis of Max can be more than a snapshot. Any that claim otherwise should be read with care.

Why it matters

Under Federal Law 374-FZ and the System for Operative Investigative Activities (SORM), Russian information services must give the FSB technical access to user data. The legal framework and Max's technical architecture converge on the same point: message content in plaintext on servers under Russian jurisdiction. In its 2015 ruling in Roman Zakharov v. Russia[1], the European Court of Human Rights held that a system granting direct, unsupervised access to communications data violates the right to privacy in itself, independent of whether access is exercised in any particular case. Russia left the Council of Europe in 2022 and is no longer bound. The standard remains the reference point.

Under that standard, the harm does not depend on being singled out. A system that places the private communications of an entire population within reach of the state is itself the harm, whether or not any particular person is ever read. For journalists, opposition members, lawyers, religious and ethnic minorities, queer Russians, anyone with family abroad, and Ukrainians under occupation, it is not abstract. For them, a messenger is a helpline, a livelihood, a community, and a way to reach the people who keep them alive. The choice Max presents is not convenience against privacy. With no viable alternative communication platforms, the choice is to use a surveillance channel operated by a hostile state or have no ability to communicate at all.

What has changed while this report was being written

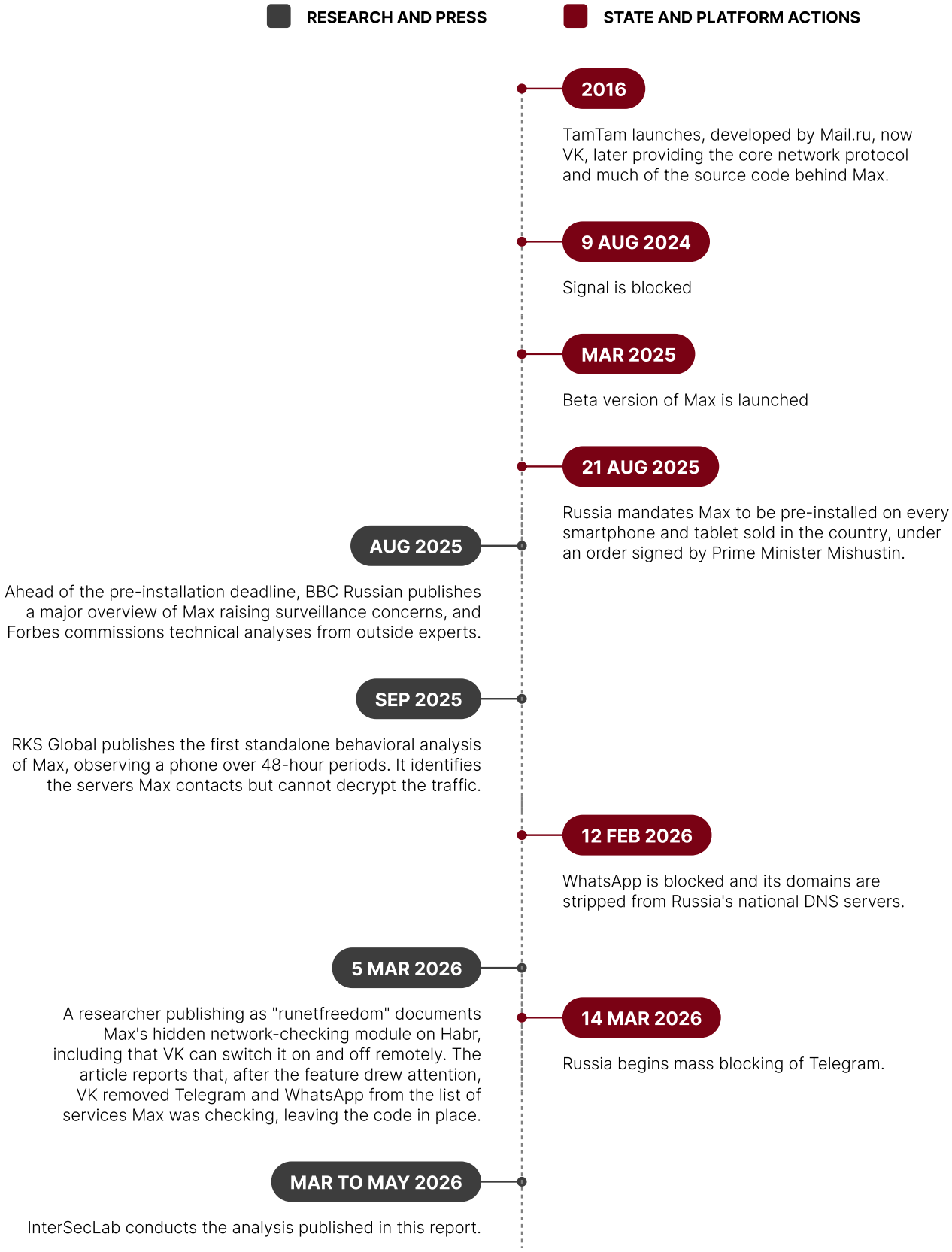
Our analysis ended in May 2026. Since conducting this analysis, several things have changed regarding the geopolitics of Max, as different actors responded to emerging findings about its capabilities. They point in the same direction as our findings.

On April 30, 2026, Cloudflare Radar briefly classified *max.ru* and *web.max.ru* as spyware, then removed the label a day later after VK attributed it to a misreading of analytics headers. Cloudflare has not publicly explained the reversal.

1 https://hudoc.echr.coe.int/eng?i=001-159324

Figure 1.

Timeline of events



11 APR 2026

A Telegram update restores access for many users.

APR 2026

RKS Global publishes a study of VPN detection across the thirty most popular Russian Android apps, finding that Max obfuscates its detection logic.

30 APR 2026

Cloudflare Radar classifies max.ru and web.max.ru as spyware. The label is removed a day later, with no public explanation.

18 MAY 2026

A researcher publishing as “zarazaexe” publishes a reverse-engineering analysis of Max on Habr. It is picked up internationally by The Moscow Times on 19 May and Meduza on 22 May.

3 JUN 2026

Apple removes Max from the App Store, stating it acted for sanctions compliance without specifying which. Push notifications stop working for more than 20 million iOS users, according to Russia's Digital Development Minister.

25 JUN 2026

Apple removes the rest of VK's applications, including VKontakte, Odnoklassniki, Mail.ru, VK Music, VK Video and VK Messenger.

13 JUL 2026

The Council of the European Union sanctions VK and its subsidiary Communication Platform LLC, renamed MAX LLC in January 2026, citing Max's surveillance capabilities and naming them responsible for developing and managing Max under FSB supervision. The same decision lists Citadel, VAS Experts and Norsi-Trans, the companies that build Russia's SORM interception systems.

16 JUL 2026

Google Play removes Max, VKontakte, Odnoklassniki, Mail.ru and around a dozen other VK applications, including Zen, VK Video, VK Music and Marusya. The same day, VK sells RuStore, Russia's mandatory app store, to the head of the company that develops it. No price is disclosed.

10 SEP 2026

A team from four universities, led from the University of Michigan, posts a preprint on what Max as a super-app can do to the mini-apps it hosts. Working independently, they read Max's traffic by the same route we did. Their questions concern the mini-app architecture, not what Max itself collects.

24 SEP 2026

InterSecLab publishes this report.

On June 3, 2026, Apple removed Max from the App Store, cutting off push notifications for iOS users; Apple stated it acted to comply with sanctions. Russia's Digital Development Minister, Maksut Shadayev, estimated that the affected population of iOS users is more than 20 million people. In late June, Apple removed the rest of VK's applications.

On July 13, 2026, the Council of the European Union sanctioned VK and its subsidiary Communication Platform LLC, along with the subsidiary's general director, under the human rights regime, naming them as responsible for developing and managing Max under FSB supervision. The same decision listed Citadel, VAS Experts, and Norsi-Trans, the companies that build SORM. The statement of reasons attributes its account of Max's surveillance capabilities to "numerous experts," without naming any or citing their work, and reports that information gathered through Max was used as the basis for fining a Russian citizen over content posted on the platform.

On July 16, 2026, Google Play removed Max, VKontakte, Odnoklassniki, *Mail.ru*, and the rest of the VK ecosystem.

Three observations follow:

1. First, the capabilities the Council attributes to Max are largely the capabilities documented here, and this report is generally more conservative in its conclusions: the Council asserts that Max's features were used for repression and cites one instance, and we make no such claim as it was not visible to our researchers from the client side.

2. Second, we find that one of the five capabilities cited in the Council's statement, the gathering of data on other applications installed on the phone, is present in Max's code but dormant in the build we examined: for our analysis, the enumeration routine was never initialized, the permission it would require was not declared, and we observed no execution of it. As we have found that Max's capabilities can be reconfigured from VK's servers per account, just because this capability was dormant in our build does not mean it is absent from every build. We do not speculate on what informed the Council's assessment.

3. Third, the removals from the App Store and Google Play don't fully address the problem for people living in Russia or Russian-occupied territories in Ukraine. Max is pre-installed by law on every phone sold in Russia and remains available through RuStore, which is itself mandatory. An app-store delisting limits Max's international spread and its reach into the diaspora, but it does not change the situation inside Russia, where, for example, a teacher in Kazan told to install the national messenger will still find it on their phone.

What should happen now

The technical countermeasures are narrow and we state them plainly: Max should not be used to communicate anything sensitive and when it is installed, a user should assume that anything sent through it can be read by VK. Where use of Max is unavoidable (due to lack of other options

to communicate as well as requirements for administrative services), it is possible to run a VPN on a router, isolate Max in a separate Android profile, or to keep a separate device for state applications in order to preserve circumvention access while keeping Max working. None of those measures, however, protect message content on Max, which VK stores unencrypted.

Our recommendations for sanctioning governments and policymakers are also not simple. The sanctions architecture now reaches VK and Communication Platform LLC, the entity the Council identifies as Max's developer, by tax identification number as well as by name, which is the durable hook. That is the entity operating Max today. It was renamed OOO "MAX" in January 2026, and Max's own user agreement[2] carries the same tax identification number that appears in the Council's listing. Before the publication of this report, we alerted the Council to this change. The Council's press office referred the question to the European External Action Service, whose Sanctions Division told us that it had noted the change of name and would update the Annex at the next review, and that the inclusion of tax identification numbers in the entry is what ensures the measures are properly implemented. CryptoPro, which supplies the FSB-licensed cryptography on which Max's transport depends, is not listed under any major Western sanctions regime. Beyond sanctions, there is a need to consider the relocation needs of the people most at risk. However, since 2024, the EU has been taking in fewer Russian human rights defenders.

From our perspective as part of the internet freedom research community, we encourage additional analysis to ensure we collectively monitor how this technology continues to be deployed in the country and its impact on users. Knowing how a system of control works is the precondition for resisting it. That work needs to distinguish what an application is capable of from what it actually does, because in Max the two do not necessarily coincide: a capability can sit dormant in the code and be switched on later from the server. Continued analysis is also necessary to avoid overestimating or underestimating its capabilities.

2 https://legal.max.ru/ps

CHAPTER 02

What Max is

Source: Pexels

Max is a messaging application developed by VK Group, the Russian internet conglomerate formerly known as *Mail.ru* Group. It is operated by OOO "MAX", a VK subsidiary registered until January 2026 as Communication Platform LLC[3]. It is distributed as the package *ru.oneme.app*. The version analyzed in this report is the 26.12.0, build 6664, released in Spring 2026 for the Android operating system.

VK Group makes up a significant part of the Russian sovereign internet project. Its other properties include the social networks VKontakte and Odnoklassniki (OK), the *Mail.ru* web portal and email service, the RuStore application store and the educational platform Sferum. Many of these appear in Max's infrastructure as supporting services.

VK Group's ownership has changed in recent years. Following a 2021 reorganisation, control of the company passed to entities closer to the Russian state. Sogaz, the Gazprom-founded insurance group in which Putin ally Yuri Kovalchuk holds a major stake, and the state-linked Gazprom Media hold substantial shares[4]. The company is, in practical terms, a strategic asset of the Russian state rather than an independent commercial actor.

In August 2025, the Russian government added Max to the list of applications that must be pre-installed on every smartphone and tablet sold in Russia. An order signed by Prime Minister Mikhail Mishustin that took effect on September 1, 2025. Max replaced VK Messenger, which had been on the mandatory pre-installation list since 2023. The requirement builds on a framework,

3 https://legal.max.ru/ps

4 https://www.themoscowtimes.com/2021/12/03/gazprom-gains-control-of-russias-top-social-network-a75724

established in 2020, for mandatory pre-installation of designated Russian applications on consumer devices[5].

Apple and Google complied with the relevant laws. The practical consequence is that since September 2025, every smartphone ships with Max already installed. Whether the user chooses to register and use it is a separate question, but the application is present on the device from first boot.

At the time of our analysis, Max was also available for direct download from the Google Play Store, the Apple App Store, the Huawei App Gallery, and RuStore, for users on devices that did not come with it pre-installed. Both Apple and Google have since removed the application from their stores (see the timeline of events later in this report).

In Russia and Russian-occupied areas of Ukraine, Max is reportedly required to access government services, including enrolling children in school[6].

Intensive propaganda accompanied the forced adoption of Max, appearing, for example, in a music video[7] by Russian singer Egor Kreed (Егор Крид) for his single “More” (Море) that starts with a scene of the musician lying in a rubber boat at sea, telling someone on the phone: “Bro, can you believe it, Max works even at sea!” (“бро. Прикинь, Макс ловит даже в море.”) The intro set off widespread mockery in the comments section, such as “He probably meant to say: Max leaks personal photos to the police even at sea” (“Наверное хотел сказать: Макс сливает личные фотографии в полицию даже на море”).

5 https://www.theguardian.com/technology/2025/aug/21/russia-max-app-phones
6 https://rsf.org/en/ukraine-s-occupied-territories-kremlin-s-messaging-app-max-building-digital-iron-curtain
7 https://www.youtube.com/watch?v=qrU4xhBvMhc

↑ Screen capture from the music video "More" (Mope) by Egor Kreed (2025). Source: YouTube, accessed 18 September 2026. Reproduced under the right of quotation for purposes of criticism and review.

CHAPTER 03

How we did this & what we could not see

Source: Pexels

A few important limitations frame this research. First, our findings are a snapshot in time: they reflect what the app was doing during a specific window of analysis, not a continuous picture of its behavior.

Second, everything we describe applies strictly to the specific version and build of the app we examined. Capabilities can be added, removed, or rewired between releases, and a different build may behave differently in ways large or small.

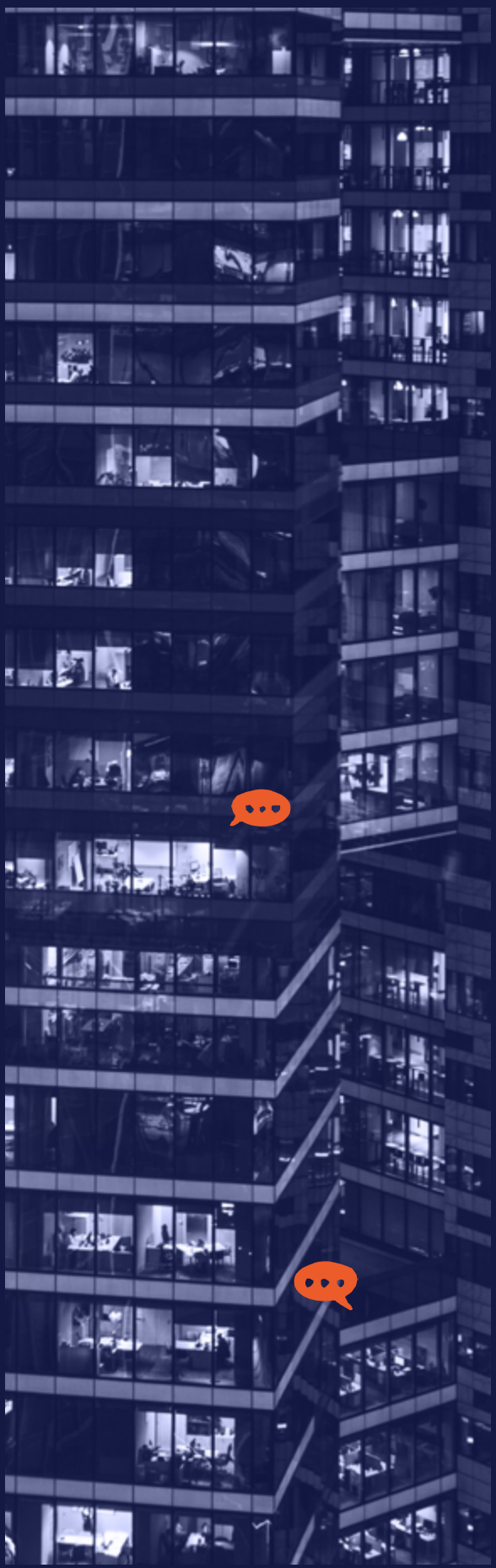
Additionally, our analysis was conducted under controlled laboratory conditions, using a specific test device, a specific network setup, and specific account states; behavior in the wild, across the diversity of real users, devices, networks, and geographies, may diverge from what we observed.

Another caveat is that many of the app's features can be switched on or off remotely by VK, meaning a capability that appears dormant or inactive today could be silently activated tomorrow without any update to the app itself. Conversely, a feature visible during our analysis could be quietly disabled at any moment.

Furthermore, parts of our analysis rely on searching the app's code for recognizable patterns and keywords. The app's extensive obfuscation, such as encoding sensitive values and heavy identifier renaming, could in principle hide behaviors from this kind of search, meaning the absence of evidence for a particular capability is not the same as evidence of its absence.

A distinction runs through this report that is worth stating plainly. Some of what we describe we watched happen, on a live device, with the traffic captured and decrypted. Other parts we established by reading the application's code and following a path from one component to the next. Both are evidence and neither is a substitute for the other: code can be present and never run, which we found more than once in this application, and behavior can be observed without its cause being visible. Where a finding rests on code rather than on observation we say so, and it should be weighed accordingly.

Finally, the app is actively evolving. Newer versions are already shipping changes, and some of the components, endpoints, and behaviors documented here may have shifted, been disabled, or been replaced by the time this report is read. For these reasons, our results should be understood as a careful, reproducible account of one specific build under one specific set of conditions, not as a permanent or universal description of how the app behaves.



CHAPTER 04

What was already known about Max

Source: Pexels

After Max was launched as a beta version in March 2025, concerns about the number of permissions Max requests circulated on Russian-language technology websites (such as Habr, *vc.ru*, and *ixbt.com*), along with various amateur and semi-amateur analyses. In July, publications like securitylab.ru raised surveillance concerns, and VK launched a public bug bounty program where researchers uncovered "several critical vulnerabilities within weeks." In August, in the run-up to the September 1 mandatory pre-installation deadline, BBC Russian published a major overview of Max with explicit surveillance concerns, Forbes commissioned multiple technical analyses by experts, and other media outlets (Moscow Times, TechRadar) covered the topic.

In September 2025, Russian internet-freedom research group RKS Global published its standalone behavioral analysis of the app[8], studying official store versions (25.9.2 on Android, 25.9.5 on iPhone) over 48-hour observation periods. Their public reports flagged *api.oneme.ru* and related hosts for further investigation but did not decrypt the traffic. A follow-up RKS Global publication in April 2026[9] examined VPN-detection behavior across the thirty most popular Russian Android applications, finding that all of them now actively detect VPN connections and that Max obfuscates its detection logic using XOR encoding.

A first detailed technical source was an article published in March 2026 by a researcher using the name "runetfreedom" on the Russian tech site Habr[10]. It examined an earlier version of the Max app (version 26.4.3, older than the 26.12.0 version we looked at). The article revealed that the app contained a hidden feature (the network reachability module) that quietly checks whether certain other apps and services are reachable over the internet. This check would kick in whenever a user minimized the app and reopened it. The researcher described exactly how this worked: which services the app was checking for, how the data was packaged and sent, that the data reported the user's mobile operator and VPN usage, the proprietary binary protocol on *api.oneme.ru*, and the fact that the app's makers could switch this feature on or off remotely from their own servers. Notably, the article reported that after the feature drew public attention, VK quietly removed Telegram and WhatsApp from the list of apps it was checking, but left the underlying code intact, meaning it could easily be switched back on.

In April 2026, RKS Global shared with us an unpublished dossier they had produced. Unlike the runetfreedom Habr article and the first RKS analysis, this dossier examined the same newer version of the app that this report analyzes, although a different build downloaded from Google Play. It flagged several additional things worth investigating: hidden flags that could expand how much user activity gets logged, an on-device system that listens for specific spoken keywords, speech-recognition components, methods for detecting whether a user is on a VPN, and a claim that push notifications were being sent in plain text.

As part of our analysis, we were able to independently verify three of the RKS Global dossier claims. First, the five hidden logging-elevation flags. Second, through static analysis, the on-

8 https://rks.global/research/max/
9 https://rks.global/research/vpn-detection/
10 https://habr.com/en/articles/1006666/

device keyword-spotting system that listens for a specific spoken phrase and routes live audio through a machine-learning model. Third, one of the VPN-detection methods the dossier identifies as a "kernel interface list check," which tests whether the system's network interface names show signs of VPN usage. We confirmed this third mechanism exists through static analysis, but also found it is a permission-gated fallback: it runs only when the app lacks the ACCESS_NETWORK_STATE permission, which Max always holds in practice. The primary VPN-detection path, which we confirmed firing in dynamic analysis, is the standard Android NetworkCapabilities check (see below). The interface-name method is therefore dormant in normal operation.

We were unable to verify two of the RKS Global dossier claims. We did not observe push notifications being sent in plaintext via Google's Firebase service in our build. The dossier identified a Firebase Cloud Messaging path carrying plaintext push content in their APK version. In our app build (downloaded from the Max website directly), notifications run through a different system (VKPNS), and message content travels separately through the proprietary protocol. We also found no FirebaseMessagingService component in our build. Because Firebase Cloud Messaging relies on Google Play Services, one possible explanation for the divergence is a difference in distribution channel between the build the dossier examined and our website-sourced build, although we have not confirmed this.

In May 2026, a further reverse engineering analysis of Max's Android application was published on Habr by a researcher using the name "zarazaexe"[11]. It was amplified in the international press, including by The Moscow Times on May 19 and by Meduza on May 22, and its account of the application entered the wider public record of what Max is understood to do. It examined a different build than this analysis and reported capabilities beyond those documented in earlier work. Several of its claims prompted us to re-examine our own work, and two of them changed findings in this report. We set out in the note at the end of this report where our evidence agrees with that account, where it contradicts it, where we got it wrong, and what we did not test. The same note covers the capability the EU sanctions listing attributes to Max.

Several shorter analyses also appeared on Habr over the same period. Between August and September 2025, M. Grishutin examined the decompiled package[12], and two posts by Chumikov compared Max's Android permissions with those of Telegram and WhatsApp[13] and observed its network traffic through a local proxy[14]. CyberB surveyed the application manifest[15]. In June 2026, a researcher using the name 3ntr0py published a reproducible evidence base assembled from static analysis[16]. These are static and observational studies. None of them dynamically instrumented Max to read its encrypted network traffic.

11 https://habr.com/ru/articles/1036222/
12 https://habr.com/ru/articles/938518/
13 https://habr.com/ru/articles/939006/
14 https://habr.com/ru/articles/939550/
15 https://habr.com/ru/articles/945306/
16 https://habr.com/ru/articles/1047490/

During our analysis window, on April 30, 2026, Cloudflare Radar briefly classified the *max.ru* and *web.max.ru* domains as spyware[17][18]. The label was removed one day later after VK's press service attributed it to a misinterpretation of standard analytics request headers. Cloudflare did not publicly explain why the label was removed.

On September 10, 2026, as this report was being finalized, a team from the University of Michigan, the University of Calgary, Georgia Institute of Technology and IIT Delhi published an analysis on arXiv titled "Don't Trust the Super-App: A Case Study of Russia's Max"[19].

Parts of their method match ours, arrived at independently. They instrument Max's encryption boundary inside the CryptoPro stack, for the same reason we did: the Russian cipher suites defeat conventional interception tools. They reconstructed the same binary protocol, a ten-byte header followed by a MessagePack body. They encountered the same VPN check and evaded it the same way, by routing traffic transparently on the network gateway rather than through an on-device VPN. Where separate teams reach the same result, that is corroboration, and we record it as such.

Their main subject is not Max though. It is the super-app architecture, in which a host application embeds third-party mini-apps inside itself. Their argument is that Android gives such a host powers over its mini-apps that the operating system cannot constrain or audit. Their five results all concern that relationship. The host can capture mini-app screen content, read and write mini-app storage, inject code into a mini-app's runtime, observe mini-app network activity, and control the identity a mini-app receives. They demonstrate these capabilities by injecting their own code into Max and invoking the interfaces themselves. They state that the capabilities are not specific to Max but inherent to any implementation of the architecture. On the screen capture specifically, they state that they found no evidence Max invokes those paths during normal use.

But when findings are presented as a list of things Max can do, that is how they end up reported. On 18 September 2026, the Guardian described the research as having found that Max "can screenshot users' devices – and the contents of its mini-apps – without their knowledge." The qualification that this was not observed in use is not in the article. Whether Max does it is a question the paper leaves open, and it is answerable from the application package.

So we answered it. In the build we examined, Max's own code invokes no screen-capture interface against a mini-app. The capture routines present in the application belong to the video player, the camera preview and bundled third-party libraries. The drawing-cache routines appear only inside those libraries and never in Max's own code. One screen-capture path is reachable. It exists to support screen sharing during calls, it captures the whole display rather

17 https://novayagazeta.eu/en/articles/2026/04/30/major-web-security-provider-flags-russian-super-app-max-as-spyware-en-news

18 https://meduza.io/en/news/2026/05/01/cloudflare-removes-spyware-label-from-russia-s-state-backed-messaging-app-max

19 https://arxiv.org/abs/2609.11814

than any particular view, and it runs through Android's own consent dialog, which selects the source. We found no calls to any capture routine made indirectly, and no equivalent entry point in the application's native libraries. Max also contains no way to detect screenshots: no screen-capture callback, no file watcher, and no monitoring of the device's image store for new screenshots. It does expose the opposite capability to the applications it hosts. Any embedded application can ask Max to block screenshots while it is open, and Max applies that block to its own window and lifts it when the application closes. The setting it writes is a user preference that has no reachable control in the interface of this build, so in practice the only thing that can change it is an embedded application.

Two interfaces their analysis describes are absent from our build. Max does not override the callback through which a host inspects a mini-app's resource requests, in any of its three WebView client classes, and it does not use Android's cookie or web-storage management interfaces. We record this as an observation about version 26.12.0 rather than a contradiction. Their paper does not say which version they analyzed, beyond noting that they began testing after version 25.15 in November 2025, and Max is assembled differently for different distribution channels.

None of this contradicts their argument, which is about what the architecture permits and what Android fails to constrain. We think that argument is correct, and the recommendations they draw from it are addressed to the right parties. Our disagreement is narrower, and it is about method. In the case of the screen capture specifically, a capability demonstrated by rooting a device and injecting code into an application establishes what the operating system permits. It does not establish what the application does, because the code that exercised it was not the application's. The authors say as much for the screen capture, and that caveat is correct. Our point is that the question it leaves open was answerable, from the package they already held, and that establishing whether a capability is used is not a footnote to establishing that it exists. It is the greater part of the work.

CHAPTER 05

What it is like to use Max

Source: Pexels

Registration on Max begins with a phone number, and only numbers from a fixed set of countries are accepted. VK publicly announced in March 2026 that registration was open to residents of 40 countries beyond Russia and Belarus[20]. In our own analysis, the list transmitted by Max's servers during registration contained 38 country codes (*full list in appendix A*). The set is dominated by post-Soviet states, countries with significant Russian diaspora populations, and additionally, a small number of governments with close diplomatic ties to Moscow. Users with phone numbers outside the list cannot complete registration.

After the number is entered and a verification code is confirmed, the user is asked to grant access to the contacts from their phone book. The dialog displays an "Allow" button, but it does not show a "Don't Allow" or "Skip" button. Tapping outside the dialog dismisses it, but Max re-presents it through the onboarding flow until permission is granted.

Granting the permission triggers an upload of the user's entire phonebook to VK servers. The phone numbers travel in cleartext, paired with names the user has saved for each contact. This is not a zero-knowledge exchange of the kind some other applications like Signal use to discover mutual contacts without revealing the full contact list.

Once registration completes, the user lands in the application's main view (the "chats" tab) with a list of suggested channels already populated. Of these twenty-five channels, ten were state media, channels of sanctioned officials, channels of state television correspondents, or channels of pro-war military commentators (see below, *full list in appendix B*).

On the surface, Max works as a normal messaging and communication app. Users can send text messages to one another and make calls, including voice or video calls. The app manages a user's contacts, matching their phone's address book against other people who use Max so they can find and connect with each other. But Max also positions itself as a broader platform that reaches well beyond simple chatting: a single "super-app" for messaging, payments, and government services[21]. It is designed to let users verify their identity through official Russian systems: either their mobile carrier, which can confirm a phone number automatically, or the national government services portal tied to a person's state-registered identity, though this state-services integration was still being rolled out during our analysis. The app can also launch smaller built-in apps and web services without making the user log in again each time. In our own analysis, this identity handoff happened without a prompt asking the user's permission, and the information passed to the receiving service included the user's IP address as Max's servers currently saw it. Among the services it connects to are Max's own digital-identity service and an educational platform, Sferum, used in Russian schools.

20 https://www.kyivpost.com/post/72439

21 https://www.themoscowtimes.com/2025/08/28/everything-you-need-to-know-about-max-russias-state-backed-answer-to-whatsapp-a90356

CHAPTER 06

How Max is built

Source: Pexels

Max is not a new application built from the ground up. Its network protocol and substantial portions of the source code appear to be inherited from TamTam, a messaging product VK (then *Mail.ru*) launched in 2016 on its Odnoklassniki platform that never grew into a serious competitor to Telegram or WhatsApp. References to TamTam are visible throughout the decompiled Max source code: protocol implementations sit in packages named *ru.ok*.tamtam, and an independent open-source implementation of the protocol (the openmax project) documents the similarities between the two apps' wire protocols.

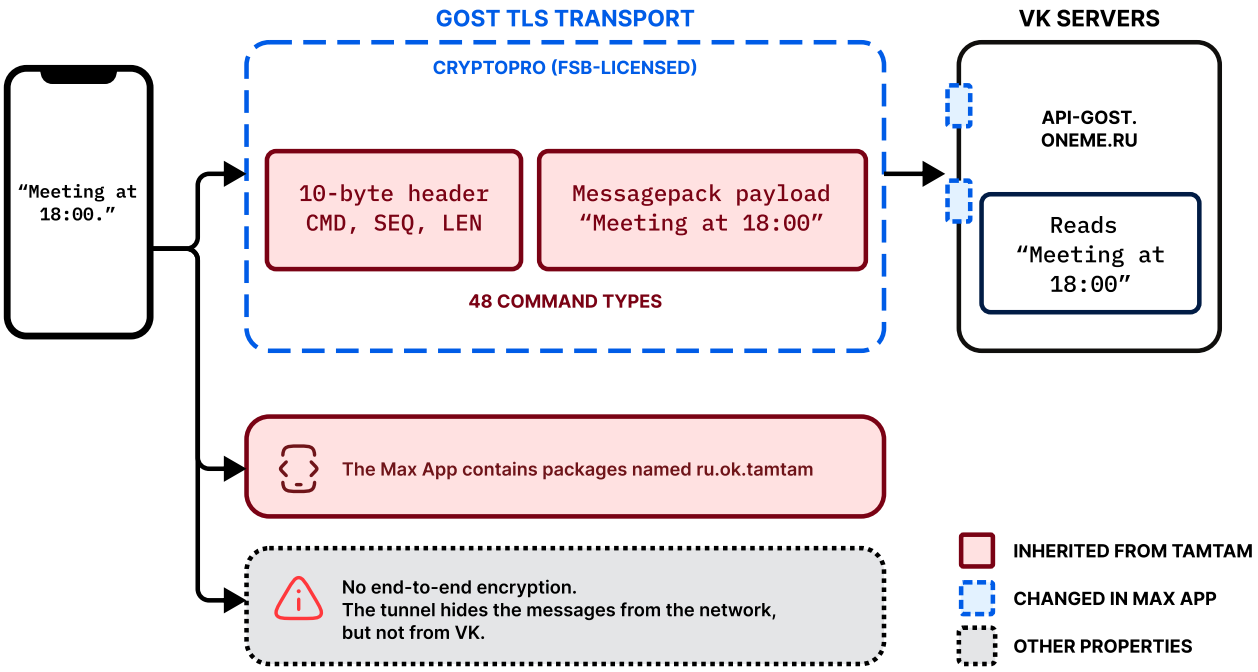
What has changed between TamTam and Max is the server endpoints and the cryptography. The protocol itself, a proprietary MessagePack-based format carried over TLS, is largely the same as in TamTam. In Max, the transport encryption has been migrated to GOST TLS, the Russian federal cipher family, and the primary server endpoint has been changed to *api-gost.oneme.ru*. Max is, in effect, an existing VK messaging product rewired to use state-approved cryptography and pushed into the role of a national messenger.

Max's backend servers sit inside a block of 4,096 IP addresses allocated to LLC VK in Moscow on 5 September 2025, the same month the pre-installation mandate took effect. VK provisioned a substantial dedicated network range specifically for the nationwide rollout.

Figure 2.

How Max is built

Max's network protocol and portions of the source code appear to be inherited from TamTam (VK, then mail.ru, 2016)



Every server Max communicated with for its core functionality is operated by a Russian entity, hosted on Russian infrastructure, and reachable through Russian transit networks. The only traffic to non-Russian providers we observed during our analysis sessions was directed to several public IP address lookup services.

Max's network traffic does not use HTTPS, instead it uses a proprietary binary protocol carried over a GOST-encrypted transport. Each message consists of a ten-byte header followed by a payload encoded in MessagePack, a compact binary serialization format. The header identifies the command type, a sequence number, and a length, while the payload carries the actual content.

We identified fifty-one distinct command types in use, covering session management, messaging, telemetry, presence tracking, contact synchronization, web application single-sign-on handoff, and voice transcription (*see complete list in appendix C*).

Message content travels in plaintext within the encrypted transport, meaning VK servers see every message readable as plain text. We documented this directly by capturing a sent message at the moment Max handed it to the GOST TLS cryptographic library.

GOST TLS is a variant of TLS, negotiated with cipher suites from the Russian federal cryptography standards. The specific cipher Max uses is Kuznyechik, a symmetric block cipher specified in Russian federal standard GOST R 34.12-2015 and published internationally as RFC 7801[22]. The cipher is implemented by CryptoPro[23], a commercial library licensed by the FSB.

Modern Android applications are required by default to use TLS for all network traffic. Applications can opt-out for specific domains by listing them in a network security configuration file. Max's configuration lists eight such exceptions.

Six are mobile carrier domains: endpoints operated by MegaFon, MTS, Tele2, and Beeline, the four major Russian mobile operators. These support Mobile ID, a carrier-based identity lookup and verification system. When a user authenticates to a service using Mobile ID, the carrier confirms their identity directly through their cellular network connection rather than through SMS.

The remaining two exceptions are wildcards for **.gov.ru* and **.voskhod.ru*. The first covers Russian government domains, while the second relates to NII Voskhod, the state research institute that operates components of ESIA, Russia's Unified System of Identification and Authentication. ESIA is the identity layer that underpins Gosuslugi, the Russian government services portal.

None of these domains are contacted by Max's own code in normal operation. Analysis of the decompiled application shows the government and carrier Mobile ID endpoints may be

22 https://www.rfc-editor.org/info/rfc7801/

23 https://cryptopro.ru

reached through a WebView, a component that allows an application to display web content inside its own interface, when the user interacts with specific “bot” accounts on the Max platform that handle government and carrier services. The exceptions therefore likely apply to traffic generated by web apps loaded inside Max rather than by Max itself. We were unable to initiate the Mobile ID flow during testing because doing so requires a Russian SIM card.

What these exceptions mean in practice is that network connections to the listed services are potentially sent in plain text, visible in full to any network equipment along the path between the user’s device and destination. This includes the user’s mobile carrier, any transit network the traffic crosses, and any surveillance equipment installed at those networks, including the SORM equipment Russian telecommunication providers are required to host. Authentication tokens, identity-confirmation responses, and any other data transmitted to these endpoints can be read by anyone with visibility to the network path.

If Max is permissive about exposing user data to carriers and the state, it takes deliberate steps to make research and analysis difficult. The application binary uses aggressive identifier obfuscation, with most class and method names replaced by short alphanumeric strings. The host names the application probes (covered in later sections) are not stored as readable text in the binary, they are encoded as integer arrays in a format that defeats standard string-search tools. The network interface names the VPN detection code checks for are encrypted. Max detects the most common dynamic analysis framework, Frida, and terminates itself within seconds of detection. The backend servers reject any connection that doesn’t negotiate GOST cipher suites, which prevents traffic interception proxies from connecting.

CHAPTER 07

What Max collects

Source: Pexels

The data collected by Max fall into four broad categories: information about the user's network connection, information that identifies the user and ties them to others, the content of their communications, and a record of their behavior inside the application.

Network probing and VPN detection

When the setting that controls it is enabled for the account, every time the user opens Max or moves it into the background, the application probes the network the device is connected to. It looks up the device's public IP address through up to six external services, checks whether a VPN is active on the device, reads the mobile carrier the device is connected to and tests the reachability of a fixed list of internet services. The result is packaged into a single report and sent to VK's servers.

The current list of services Max probes includes Max's own servers, Google's static content delivery servers, Google Cloud Messaging, the Odnoklassniki content delivery network, and *gosuslugi.ru*, the Russian government services portal. An older version of the application also tested the reachability of Telegram and WhatsApp servers, but those checks were removed from the active list after the practice was reported publicly in March 2026[24]. The relevant code remains in the binary.

The earlier inclusion of Telegram and WhatsApp on the probe list is worth examining for what it could have been doing. At a minimum, the reachability of those two services from a given device is a strong

24 https://habr.com/en/articles/1006666/

signal about where the device is located. If Telegram and WhatsApp are reachable, the device is most likely outside the Russian network blocks that restrict access to them. Reachability probing of competing applications can therefore function as a passive geolocation check.

A more concerning reading is that as Russian internet providers are required to enforce the restrictions on Telegram and WhatsApp imposed by Roskomnadzor, the probing turns the millions of devices on which Max is installed into a distributed network measurement platform that can alert the government to which ISPs are complying with the restriction order and which are not. The data Max collects from the reachability module includes the user's mobile carrier and apparent public IP address alongside reachability results, which is precisely the information needed to map non-compliant networks.

The network reachability testing is controlled by a server-pushed feature flag that VK can toggle for any individual account without an application update. The flag controls only whether the module is active: it carries a true-or-false value and cannot carry addresses. The list of domains this module probes is present in the application itself, obfuscated so that it does not appear in a standard inspection of the file, and the server cannot change that list without shipping a new version of the application. That is not the whole picture, however. A second and separate monitoring channel, described later in this report, does take its list of targets from the server, and that list is unrestricted. The claim that the set of addresses Max probes is chosen by the server is correct, although not for the module described here.

The service reachability probing is paired with a VPN check. Max determines whether a VPN is active by querying built-in Android system APIs. This is a standard Android API that reliably detects any active on-device VPN. We have observed this VPN detection method in traffic analysis.

The app contains a second VPN detection method that was dormant in the app build we analyzed (present in code but not active in traffic). This fallback detection tests if any network interface name contains the keywords "tun", "ppp", "tap", or "ipsec". It is only activated if Max cannot use its primary VPN detection method because it doesn't have the permission ACCESS_NETWORK_STATE. As this permission is in the app's manifest and granted at install time, Max de facto always has this permission and the second VPN detection method is never used.

The VPN detection result is included in the same report as the network probing results described above. The combination gives VK's servers, on every session, a record of where the user is connecting from, whether they are using a VPN, which mobile carrier they are on, and which services are currently reachable from their device.

When Max determines that a VPN is active on the device, it sometimes blocks the user from sending messages until the VPN is disabled. We observed this directly during testing. An attempt to reply to a message produced a screen instructing the user to disable their VPN, with no other option to proceed. Switching from an on-device VPN to a VPN router (which routes the device's traffic through a VPN tunnel at the network gateway, with no VPN interface visible on the device itself) restored normal messaging.

Blocking messaging when a VPN is detected presents the user with a forced choice: disable the protection that is keeping their network traffic private and helps them circumvent censorship or lose access to a state-mandated communication channel.

Identity and social graph

Max ties each user account to a phone number, a physical device, and to the user's social graph based on the content of the device's contact book.

The user's contacts are uploaded to VK's servers in cleartext shortly after registration. Phone numbers are transmitted paired with the names the user has saved associated with the contact. The upload happens through a single network call and is not protected by any privacy-preserving contact-discovery methods (hash exchange, private set intersection) that some other messengers use to determine mutual connections without revealing the full contact list.

Once a phone number is registered to a Max account, any other user can look that number up. The query returns the target user's internal identifier, their chosen display name, the exact timestamp when their account was created, and the timestamp of the most recent profile update. The query does not require a mutual contact relationship, and it does not trigger a notification for the person being looked up.

The lookup also triggers two side effects on the server side without any further action from the user. First, within 1.2 seconds of the query, the server began pushing real-time online-presence updates for the target to the querying account. The pushes continued for fifteen and a half minutes, tracking when the target was active, when they were idle and when they went offline.

Second, in the same sub-second window, the server auto-materialized a chat object linking the two accounts: the querying client received metadata for a chat entity that already existed on the server side, and the application silently subscribed to it without any message having been sent and without any contact-add action having been performed. The server-side linkage between the two accounts was therefore established by the lookup itself, observable to VK even if neither party ever messaged the other.

Max integrates with two separate Russian state identity systems. The first, Mobile ID, is a phone-number verification system operated by the four major Russian mobile carriers, which allows confirmation of a user's phone number through the cellular network connection. The second is ESIA, the central Russian government identity system that underlies the Gosuslugi portal. A user can authenticate with either system when accessing government services through Max. Phone numbers in Russia are bound to real identities through mandatory SIM card registration.

When a user launches an embedded web application from inside Max, for example a VK ecosystem service like Sferum (the education platform deployed in Russian schools), Max

passes a signed identity token to the destination. The token contains the user's account identifier, display name, language preference, and current IP as Max's servers currently see it. The receiving service can use this token to authenticate the user without requiring a separate login. The authentication handoff happens without a consent prompt for the user. The list of services that receive the authentication handoff is controlled by a feature flag and can be modified at any time without requiring an application update. Max also exposes a second, separate interface to embedded web applications, *PrivateWebViewHandler*, which is switched on for individual services by a server-pushed setting named webapp-pr. That setting holds a list of service identifiers, and it was empty in our build, so the interface was available to none of them. A service on that list gains one capability the ordinary interface does not provide, named *WebAppVerifyMobileId*: it can hand Max a web address, and Max will fetch it over the phone's cellular connection specifically, rather than over whatever connection the device is using, and return the result. The name refers to mobile identity verification, and the mechanism matches the way Russian carriers confirm a subscriber's phone number from the network side. We did not observe it in use. Its error responses include one for the case where a VPN is active, but that response is never produced anywhere in the code we examined.

Content

The content of a user's communications on Max is accessible to VK's servers in a form suitable for searching and indexing. This is true for both text and voice communications.

Max provides no end-to-end encryption for text messages. The "secret chats" feature, despite its name, is only a disappearing-message timer, it adds no cryptographic protection. The naming is particularly misleading in the Russian context, where Telegram (the country's most widely used messenger prior to its restriction by Russian authorities) uses the same term, "secret chat," to specifically refer to its end-to-end encrypted mode. On Max, messages travel to the server in plaintext within the encrypted transport regardless of whether the secret chat feature is enabled.

The server also actively reads message content. Every text message carries a flag instructing the server to perform share detection, and the server inspects text messages for links and shared content as a routine processing step. This differs from how some competing messengers handle the same feature. Signal, for example, generates link previews on the client side. The sending device fetches the URL, builds the preview locally, and includes it in the encrypted message. The server never sees the URL.

Max transcribes voice messages on its servers, not on the device. When transcription is requested, it can apply to both incoming and outgoing voice messages, so VK can produce searchable text both for what a user received and for what they themselves recorded. The user can trigger transcription through a settings toggle labeled "Отображать распознанный текст аудиосообщений" ("Display recognized text of audio messages"). At the protocol level, transcription is handled by two dedicated commands, TRANSCRIBE_MEDIA and NOTIF_TRANSCRIPTION. We found no evidence of on-device transcription. In everything we

observed, transcription was triggered by the user rather than performed automatically; we could not determine whether an automatic transcription pathway also exists.

Live audio during voice calls is fed into an on-device machine learning model for keyword detection. The current model is configured to detect a single Russian phrase, “не слышу” (“I can’t hear”), as a signal of poor call quality. When the model triggers, an analytics event is sent to VK servers with a confidence score.

This machine learning model is not bundled with the application. It is downloaded at runtime from a server-provided URL, validated against a server-provided checksum and executed against the call audio. Changing the keyword the detector listens for requires only a feature flag push from the server and no application update. The pipeline supports any TensorFlow Lite model the server chooses to deploy.

Regarding the device location, Max requests both precise and approximate location permissions on Android, and it periodically reports to its servers whether the user has granted or denied that permission.

The app includes both user-initiated location-sharing features and background location retrieval. It offers two user-facing location features: a one-time “share my location” option (accessed through a chat’s attachment menu and a map picker) and a separate live-location sharing system that lets two users exchange their positions in real time. When the user shares their location in a chat, the device’s GPS coordinates are transmitted to VK’s servers as part of the message. Live location sharing, where a user’s position is updated continuously, runs through a dedicated set of five protocol commands devoted entirely to location: one to request another user’s location, one to send a location, one to stop sharing, and two server-push messages that deliver a location or relay a request to share. This live-location data includes a device identifier alongside the coordinates, meaning the server and the recipient can tell which of a person’s devices is sharing. We did not test the live-sharing flow end to end, and none of these five commands appeared in our captured traffic; we documented them from the application’s source code.

However, in the build of Max we analyzed (obtained directly from the Max website) the user-triggered functionality for reading the device’s actual GPS location is non-functional. The application attempts to use a vendor-specific location service, finds none configured, and defaults to coordinates of (0,0), a point in the Gulf of Guinea sometimes called “null island”. Neither the “use my current location” button nor the live-sharing functionality returns the user’s location. The only way the app reliably captures a location is when the user manually drags the map and drops a pin, in which case it sends whatever spot was chosen, which may or may not match where the person is located. Other builds (the version pre-installed on Russian phones, or the version distributed through RuStore) may behave differently.

Nevertheless, Max does process the user’s location in the background, without user interaction. The app constantly learns the user’s approximate location simply by having its servers observe the user’s IP address, something it does every time the app moves to the foreground or background (as part of the network reachability module described above). The

IP-derived location data is much less precise than a GPS-based location. Other location-related indicators the app retrieves are the user's mobile carrier (indicating their country) and a language/locale code that Max transmits as part of the user-identity data it hands off to connected web services.

Max's five telemetry systems (described below), on the other hand, do not pull GPS coordinates. MyTracker does contain a setting related to location tracking, but nothing in the app ever uses it. The only piece of the app that directly reads the device's GPS is an unrelated night-mode feature that calculates local sunrise and sunset times to decide when to switch the screen to dark mode.

Behavior

Every screen transition inside Max is reported to the server with a millisecond-granularity timestamp, an identifier for the screen the user came from, an identifier for the screen they moved to, and (where relevant) an identifier for the specific chat or contact they interacted with. The application defines 108 distinct screens for telemetry purposes. Among them are screens for the privacy settings like hiding one's online status, the two-factor authentication setup, and the account deletion flow. The server is notified every time a user visits any of these.

From this stream of telemetry, VK can reconstruct, for any user, which conversations they entered, in what order, from what entry point, and how long they spent in each.

Max ships with five separate analytics and reporting systems running in parallel: OneLog for video-player events, AppTracer for crash reports, MyTracker for general analytics, a fourth system used for in-application event tracking, and a fifth, DPS, which reports on the reachability of other internet services. Each has its own SDK (software development kit), its own batching logic, and its own server endpoint. All five are VK Group properties under Russian jurisdiction. Three of the five are active in the build we analyzed. MyTracker is bundled but appears dormant in this specific build, though its capabilities include enumerating the user's full list of installed applications and reporting it back, which would be a substantial addition to the data collected if activated.

Max maintains a real-time activity status log and broadcasts it to other users without those users requesting it. We observed this functionality directly. A test account with one contact (the official Max support bot) received continuous online-status pushes for eighty-three different user identifiers over the course of several network capture sessions. None of the eighty-three were in the account's contacts or had ever exchanged a message with the account. The mechanism which causes VK's servers to push these status updates is not visible from client-side analysis.

The Max application does not display these broadcast presence updates to the user. The pushes arrive at the device, are received by the application, but are not surfaced in the interface,

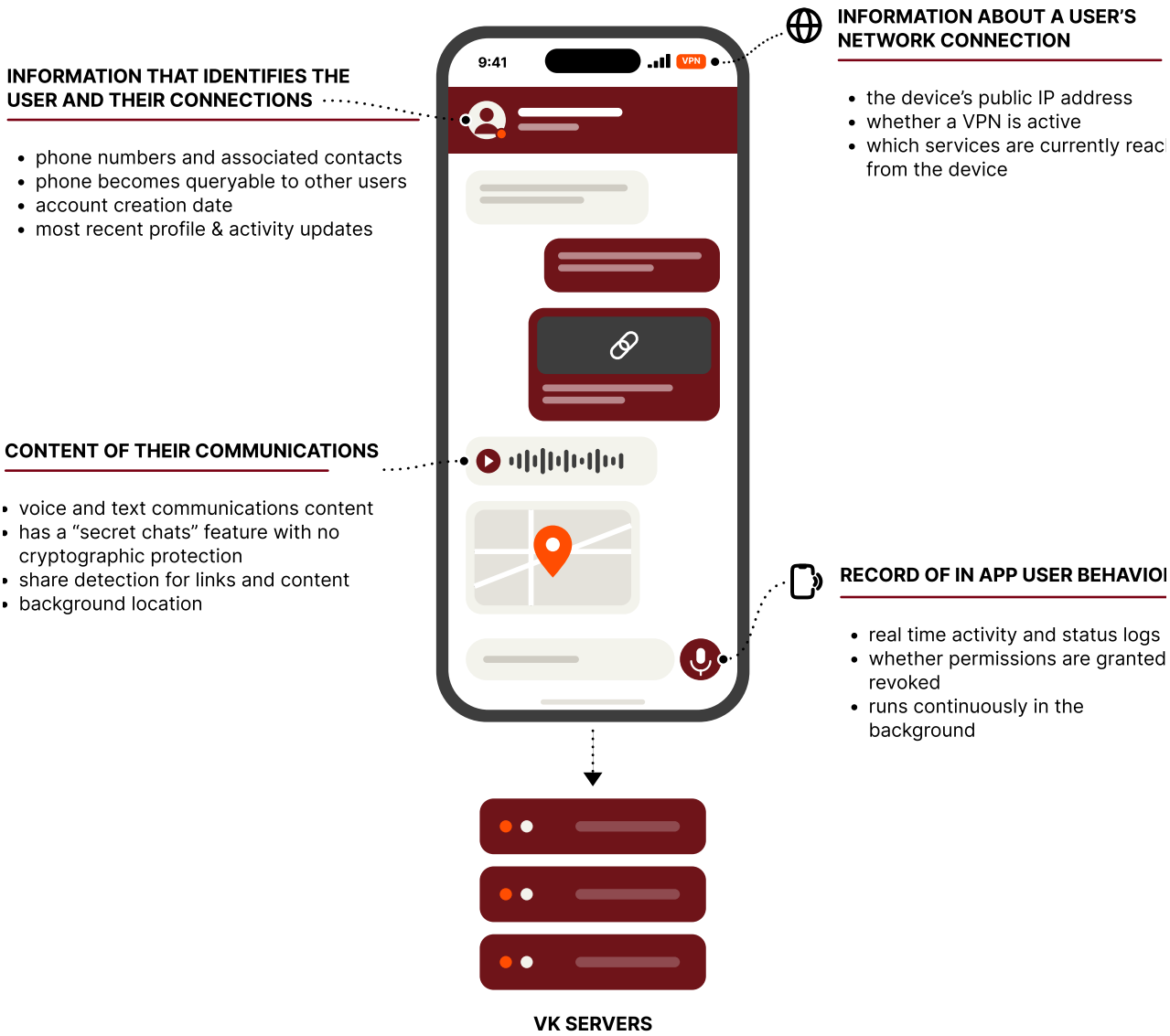
unless the user is present in the contact list. Observing them at all required a modified client that logged the raw protocol traffic.

Furthermore, Max reports the current state of seven permissions to its servers, both periodically and whenever a permission is granted or revoked. The permissions covered are notifications, contacts, full-screen interruptions, photo gallery access, camera, microphone, and location.

Max is engineered to run continuously. It declares twenty-one background services, starts automatically when the device boots, requests exemption from Android's battery-saving features, and uses multiple other redundant mechanisms to keep itself alive in the background.

Figure 3.

What Max collects



CHAPTER 08

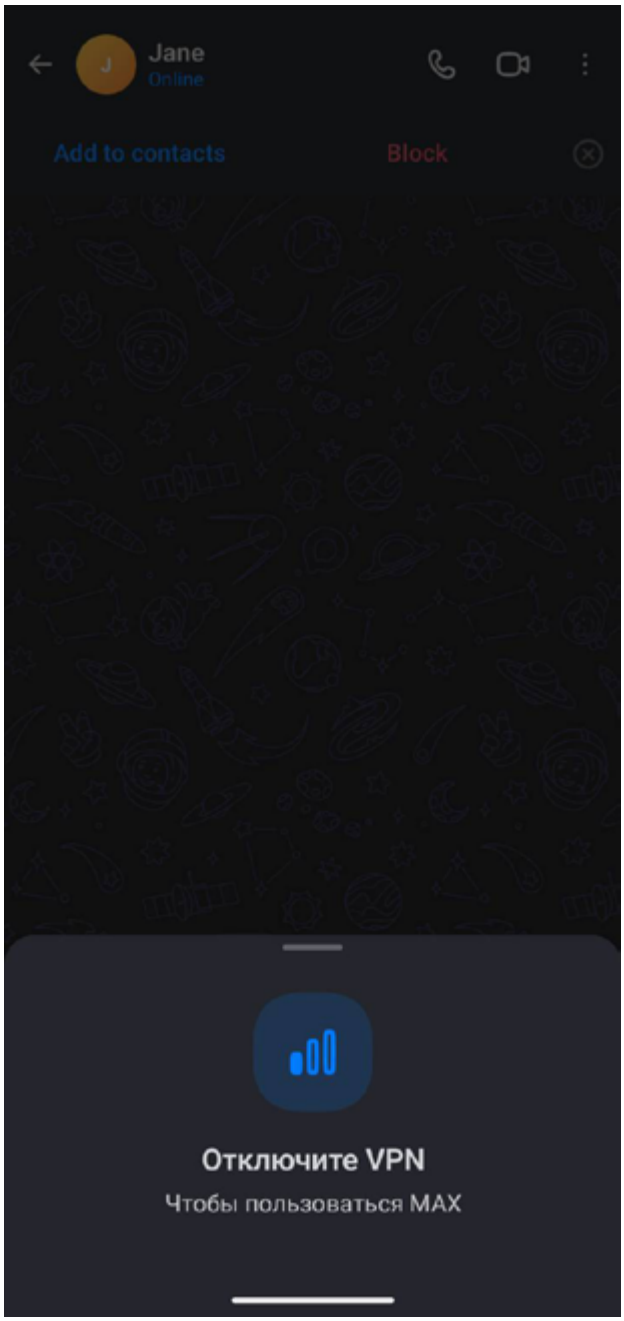
Active VPN blocking

Source: Pexels

During a two-account interaction test in April 2026, the user on one device attempted to reply to a message. Max displayed a full-screen instruction to disable the VPN before continuing. Sending messages was blocked entirely. The user could not proceed by tapping past the screen or by switching chats. The only way to proceed was to disable the VPN.

We confirmed this by switching from an on-device VPN client to a VPN router which routes the device's traffic through a VPN tunnel at the network gateway rather than on the device itself. The VPN detection in Max works by checking for the presence of a VPN interface on the device itself, rather than inferring VPN use from the network properties (i.e. IP address, MTU) of the traffic. The check happens locally and can be evaded by moving the VPN off the device. VPN routers require the purchase of a router that supports them, knowledge of how to configure it, and control over one's own network setup.

In Russia and in Russian-occupied areas of Ukraine, Max is increasingly required to access government services. A user who loses access to Max loses access to administrative life. A user who disables their VPN loses the ability to reach blocked websites, read independent media, and communicate privately.



↑ Screenshot of Max blocking the messaging function due to VPN usage.

CHAPTER 09

The onboarding feed

Source: Pexels

After registration, the app suggests a list of channels to the user in the “chats” tab before they have searched for anything, browsed anything, or indicated any interests.

We documented one such list (*see appendix B*), assigned to a new account registered with a Russian phone number in April 2026. It contained twenty-five channels. Of these, ten were state media, channels of sanctioned officials, channels of state television correspondents, or channels of pro-war military commentators. Among them: RIA Novosti, NTV, Izvestia, the personal channel of Vladimir Solovyov, the channel of Dmitry Medvedev, and the channel of a sitting State Duma deputy under EU and US sanctions. The remainder were lifestyle, entertainment, and general-interest accounts. There were no independent media outlets, no opposition voices, no minority-language content, and no international news sources.

CHAPTER 10

Telemetry & analytics

Source: Pexels

The Max app contains several different telemetry and analytics systems to report app performance and user actions back to VK. It measures a vast amount of user behavior: every screen the user visits and how long they stay on it, every application-level network call and its timing, every video the user watches and how they interact with the player, the application's foreground-and-background lifecycle, and which other Max users are online at any given moment. These data flow into the proprietary network protocol's telemetry pipeline. Each individual element is plausibly defensible as ordinary application analytics. Their combination, mediated by the per-account configuration mechanism described above, is comprehensive behavioral surveillance.

Next to the network protocol's telemetry, the app has five telemetry systems integrated in the code as separate SDKs, and contacts a further set of analytics endpoints, all of them VK-internal.

Integrated telemetry systems

As mentioned, Max ships with five different analytics and reporting systems running in parallel and integrated into the app as separate SDKs: OneLog for video-player events, AppTracer for crash reports, MyTracker for general analytics, a fourth system used for in-application event tracking, and DPS, a separate reachability reporter. All five are VK Group properties under Russian jurisdiction, although the fifth conceals its destination in a way the others do not.

1. **OneLog.** The OneLog tracker is primarily used for video-player telemetry, capturing granular events: which parts of which videos a user watched, at what playback quality, with quality-change events, seeks, and latency measurements. It sends this telemetry data as JSON POSTs containing application metadata, after being compressed (gzip) for better performance. They are uploaded to the domain "*api.ok.ru/api/log/externalLog*".

2. **AppTracer.** AppTracer handles crash reporting and performance monitoring for Max, informing VK how well the app works. Crash uploads are sent to the domain "*sdk-api.apptracer.ru/api/crash/upload*" as multipart-encoded, compressed (gzip) minidumps. The minidump format follows the Microsoft minidump specification. Importantly, minidumps captured at the moment of a crash can contain in-memory data including thread stacks, heap regions, and pointers to user data. This means that session tokens, user identifiers, and recently handled message content could in principle be present in an uploaded dump. We did not inspect a captured minidump to determine its precise scope. The AppTracer version used in the Max build we analyzed is TracerSDK/1.1.7.

3. **OK.ru product analytics.** The Odnoklassniki (OK) tracker system has a similar architecture to OneLog, and uses the same URL for uploading telemetry data (*api.ok.ru/api/log/externalLog*). What OneLog does for video playing, the OK tracker does for calls: it measures battery statistics during calls, call analytics, and call session management.

4. **MyTracker.** This system, one of the five included in the code, appears dormant in this specific Max build. Its entry point "MyTracker.initTracker()" is never called. By default, MyTracker uploads its telemetry data to the domain "*tracker-api.vk-analytics.ru*" (this endpoint can be customized in the app's configuration). According to the code, MyTracker can collect the user's Google Advertising ID or the Huawei OAID (Open Anonymous Device Identifier) if the device runs Google Play Services or Huawei Mobile Services. In the build we examined, neither is collected because the vendor-services component returns a placeholder for "no vendor". Its capabilities also include enumerating the user's full list of installed applications and reporting it back, which would be a substantial addition to the data collected if activated. However, the app does not declare the QUERY_ALL_PACKAGES permission in its manifest. Since Android 11, that permission is necessary to retrieve a list of all installed apps on the device.

5. **DPS**[25]**.** The fifth reporting system, carried in the application under the package name *ru.trace_flow.dps* and reporting to the domain "*trace-flow.ru*", sits apart from the other four: it has its own initialization, its own network client, its own credentials and its own destination, and it shares no code with AppTracer, the system we had previously grouped it with. It is also the only system whose destination is concealed. Where the other endpoints appear in the application as readable text, this address is stored in scrambled form and must be decoded before it can be read. It transmits the user's identifier, application version, mobile operator, apparent IP address and a per-address reachability report, authenticated with a fixed 256-bit key compiled into the application. Its targets are not fixed. When the channel runs, it asks the server for a list of addresses and the server may name any address it likes, including a port; nothing in the application restricts or checks what it is given. For each address the device then performs up to three tests, whether the name resolves, whether a connection opens, and whether an encrypted connection completes, and reports each result separately, labeled with the identifier the server assigned to that target. The server also sets how often this runs.

The effect is a network measurement capability, running on the user's device and from the user's network position, that VK can point at anything it chooses, for one account or for many, without an application update and with nothing shown to the user. The flag that gates this channel is off by default, but it was evidently switched on for the accounts we tested. Searching 26 days of recorded traffic across two devices, we found the application contacting the channel's reporting address nineteen times, on 21 April, 14 May, and 22 May 2026, completing an encrypted exchange on each occasion.

The probes themselves are visible in the same captures and have a distinctive shape: for each target, two connections milliseconds apart on adjacent source ports, one that opens and closes

25 This channel was first described publicly in May 2026, in the analysis discussed in the note at the end of this report. That analysis identified the package and its obfuscated destination. RKS Global later reviewed it, working from the application's code rather than from observation of it running. What follows is what we found in the build we examined, and what static analysis alone could not establish: that the channel runs, how often, and what it was told to probe.

without sending anything and one that carries a full encrypted handshake. Reconstructing those bursts recovers the addresses the server asked the devices to test: Max's own API and calls infrastructure, the Russian state services portal *gosuslugi.ru*, and the push-notification endpoints of Google, Apple, and Huawei. Apple's push endpoint appears in every one of the fourteen bursts recorded on 21 April, each time at a different Apple address, and it appears nowhere in the application itself in any character encoding we tested. It can only have come from the server. We were not able to read the contents of the reports themselves, which were encrypted and not intercepted, so the list of targets is recovered from the connection pattern rather than from the messages, and it may be incomplete rather than exhaustive.

The application itself gives no indication of who operates the receiving domain, but public records identify it without ambiguity. The domain resolves to three addresses, all within autonomous system AS47764, registered to LLC VK in Moscow. The certificate the endpoint presented throughout our capture window, valid from 24 March to 9 October 2026, names VK LLC, Moscow, and was issued by GlobalSign, a Belgian authority whose organization-validated certificates require a company's legal identity to be verified before issue. We confirmed both facts against the public Certificate Transparency logs and public internet scanning data, and neither depends on anything we observed on the device. The obfuscation therefore conceals not a third party but one of VK's own reporting channels, from anyone who reads the application.

Telemetry destinations beyond the SDKs

Live network captures revealed that telemetry from Max does not flow only to the SDK endpoints. Four further VK-ecosystem destinations were observed receiving data: a *Mail.ru* tracking pixel (*top-fwz1.mail.ru*), *Mail.ru* Cloud Solutions analytics (*mcs.mail.ru*), a VK batch telemetry endpoint, and the Odnoklassniki content-delivery network (*okcdn.ru*) with a Max-specific client identifier. All eight endpoints observed in traffic sit inside the VK Group corporate perimeter and within Russian jurisdiction. So does the ninth, the concealed destination used by the DPS channel described above, which resolves into VK's own network and presented a certificate issued to VK LLC of Moscow. The telemetry flows are therefore better understood as an integration mechanism within a single commercial entity than as a dispersed set of third-party analytics relationships.

Behavioral events carried over the proprietary protocol

Beyond these application-level analytics, the app uses different commands embedded in its proprietary protocol to send behavioral events to VK's servers. One of these is the telemetry command (LOG, 0x0005) that includes the network-reachability checks described earlier, and several other sub-event types that together produce a detailed behavioral log of the user's activity:

- **Screen-by-screen navigation (NAV events):** every screen transition inside the app is reported to the server with millisecond-precision timestamps, the screen the user came from, the screen they went to, and the dwell time on each. The app defines 108 distinct screens in an internal list, including privacy-sensitive ones such as the privacy settings panel, the active-devices list, two-factor-authentication setup, contact-search-by-phone, and account deletion. When the user opens a chat or a contact, the identifier of that chat or contact is also transmitted, meaning the server can reconstruct not only which screens were viewed but which specific conversations were entered and for how long.

- **Per-call network metadata (NET events):** for instrumented internal network requests, the app reports back the connection type (WiFi or mobile), the bytes sent and received, the round-trip response time, and a flag indicating whether the app was in the foreground or background at the time. This gives the server an independent activity log annotated with whether the user was actively interacting.

- **Permission reporting (PERMISSION events):** reports the grant status of seven permission categories back to Max's servers: push notifications, contacts, full-screen intents, gallery access, camera, microphone, and location. It does this in two ways: periodically broadcasting the status of all seven at once and firing a change event the moment any single permission is granted or revoked. The server is therefore informed not only of what the app can access at any given time but of every change in those permissions as it happens.

- **Interface and recommendation events:** lower-volume events report which prompts, banners, channel recommendations, and invite surfaces are shown to the user and in which contexts.

Real-time presence

Another protocol command, NOTIF_PRESENCE, pushes real-time online and offline status for a server-determined set of other user accounts, without the client ever requesting them. During the analyzed sessions, the test account received presence updates for 83 distinct user identifiers, none of which were its contacts, and the pool rotated over the course of each session and differed across accounts. The mechanism that decides whose presence is pushed to whom is opaque from the client side and remains an open question, but it indicates that VK maintains a real-time activity log of every Max user on its servers and selectively distributes the online status of a server-chosen subset of users to each client.

Always-on engineering

Both the telemetry and presence functionalities depend on the app remaining alive in the background. To this end, the app declares 21 background services and three boot receivers that start the app the moment the device powers on. It also requests a battery-optimization

bypass permission and declares seven foreground-service permissions, including one that enables screen-recording capability. The aggregate effect is an app that does not fully stop: it starts on boot, exempts itself from Android's power-saving modes, and uses multiple redundant mechanisms to remain running.

CHAPTER 11

Where Max sits in the industry

Source: Pexels

While many of Max's characteristics are standard for a messaging application, others are unusual. To understand what is abnormal and what is standard, it is useful to place Max alongside the communication applications its users would otherwise use, Signal, WhatsApp, and Telegram (all blocked or under blocking). Separately, we can compare Max's characteristics to China's WeChat, which is the closest existing model for what Max appears to be becoming: a single state-aligned application that combines messaging with identity, payments and access to government services. Max's in-application platform of service accounts, which the platform calls "bots," is modeled on the same "super-app" pattern that WeChat pioneered and Telegram later adopted. Both of those applications built out the pattern toward e-commerce and payments: WeChat through WeChat Pay, and Telegram through an integrated cryptocurrency wallet tied to the TON blockchain. Max extends the pattern in a different direction, one neither of those applications has taken. It is the only one through which a user can authenticate to and interact with state government services as a core function. Where WeChat and Telegram grew into payment platforms, Max is growing into a state-services platform.

We compared Max with similar apps along six dimensions: whether message content is protected from the operator, what cryptography the application uses, what jurisdiction the operator falls under, how the application discovers users' contacts, whether the user has any choice in using the app, and if/how the application detects VPN usage.

On end-to-end encryption, Max differs the most from other messengers. It offers none, which puts it level with the oldest channel of all, SMS. Text messages are readable by the mobile carrier and by anyone with access to the carrier's network, which in Russia means the SORM apparatus by design. iMessage encrypts messages end to end between Apple devices, though it silently falls back to unencrypted SMS when one party is not an Apple device, and messages may be recoverable from iCloud backups unless the user has enabled Apple's Advanced Data Protection. Signal encrypts all messages end to end by default and has no mode in which the operator can read content. WhatsApp also encrypts all message content end to end by default, using the protocol Signal developed. Telegram encrypts messages end to end only in its "Secret Chat" mode, which is not the default. Ordinary Telegram chats are encrypted in transit but readable by Telegram's infrastructure. WeChat provides no end-to-end encryption. Messages are encrypted in transit but readable by the operator and subject to server-side monitoring and censorship. Like WeChat, Max provides no end-to-end encryption at all. Every message is readable by VK's servers, and the "secret chats" feature is only a client-side disappearing-message timer. Max provides no protection for message content from the operator and, through the operator, none from the state.

On cryptography the application uses to protect traffic in transit, again Max stands out. SMS relies on the carrier's network-level protections, which are weak and, in Russia, deliberately accessible to the state. Signal and WhatsApp use the Signal Protocol, built on well-established, peer-reviewed cryptographic constructions[26][27]. iMessage uses Apple's own protocol built on standard primitives. Telegram, by contrast, uses MTProto, a protocol of its own design that

26 https://signal.org/docs/

27 https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

cryptographers have repeatedly criticized for departing from established practice. WeChat similarly uses a non-standard, in-house construction. Max's core traffic is encrypted using GOST, the Russian federal cryptography family, implemented by CryptoPro, a library licensed by the FSB. It ties the application's transport security to a cryptographic regime certified and licensed by the same security service that Russian law entitles to access user data. No other application in this comparison builds its transport security on state-controlled national cryptography.

Jurisdiction determines who can compel the operator to hand over its data, and on what terms. SMS is handled by the user's mobile carrier, which in Russia is directly subject to SORM. Both Apple's iMessage and Meta's WhatsApp fall under US laws. The same applies to Signal, being operated by the US non-profit Signal Foundation. Telegram is Russian-founded but operates today through Telegram FZ-LLC, a Free Zone company based in Dubai, with a multi-jurisdictional structure. Its founder, Pavel Durov, left Russia after refusing to hand over user data from VKontakte, the social network he founded and then lost control of, and which is now part of the same VK Group that operates Max. WeChat is operated by Tencent, a Chinese company subject to Chinese law, which requires cooperation with state security and data-access requests. Max is operated by VK Group, subject to Russian law, including the SORM access requirements described earlier. Compared with the Russian and Chinese systems, the US framework requires judicial or quasi-judicial authorization for most data-access requests and allows operators to litigate or publicly challenge demands they consider overbroad, though protections weaken substantially for foreign-intelligence and non-US-person targets. The practical privacy outcome still depends on what the operator actually holds: Apple, Meta, and Signal differ significantly in how much message content and metadata they retain, and end-to-end-encrypted content remains beyond the operator's reach regardless of the legal demand. For Max, WeChat, and SMS, the operator or carrier holds all message content and metadata, and in the case of Russia and China, operators are obliged to provide user data to government institutions as a default.

The fourth dimension for comparison is contact discovery, which determines how much the operator learns about a user's social graph. WeChat discovers contacts through phone numbers, WeChat IDs and QR codes, leaving Tencent positioned to observe the resulting social graph. SMS has no discovery step, it simply addresses messages to phone numbers, and the carrier sees every number contacted. iMessage checks Apple's servers to determine whether a given number or email address is registered with iMessage, which reveals the queried contacts to Apple. Signal uses a privacy-preserving method which lets Signal tell a user which of their contacts are on Signal without Signal learning the user's full address book[28]. WhatsApp historically allowed any number to be checked against the service, a property researchers used to enumerate billions of accounts before Meta introduced rate-limiting in late 2025. Max sits at the most permissive end. It uploads the user's full address book to the server in cleartext, and it allows any user to look up any registered phone number and receive that number's profile and real-time online status, with no contact relationship required and no rate-limiting we could observe.

28 https://signal.org/blog/private-contact-discovery/

Additionally, when compared with other apps, Max is unique contextually in that it is presented as the only messaging option, is pre-installed by law, and is required for access to government services. SMS and iMessage are built into the phone, the user does not install them, but users are not coerced into using these functions. Signal, WhatsApp, and Telegram are applications a user chooses to install. WeChat, in China, is close to unavoidable in daily life, but it is not formally pre-installed by law. Max, on devices sold in Russia, is pre-installed by law and effectively required for access to Russian government services and inside occupied areas of Ukraine. At the same time, users no longer have access to other messaging options. Instagram and Facebook have been blocked in Russia since 2022 when the Russian government designated their parent company, Meta, as an "extremist organization." Signal has been blocked since August 2024, Discord since October 2024, and Viber since December 2024. Through 2025, Russian authorities throttled WhatsApp and Telegram in stages, limiting calls, then blocking new registrations, until by the end of the year the great majority of connection attempts to WhatsApp from inside Russia were failing. In February 2026, WhatsApp was formally blocked, and its domains removed from Russia's national DNS servers. Mass blocking of Telegram began across the country in mid-March 2026. Roskomnadzor justified the blocks by citing the platforms' failure to store Russian user data in Russia and to police prohibited content. The blocking has been openly tied to promoting Max as the state-sanctioned replacement.

The last dimension, related to Max's VPN blocking function, is also distinct and important. None of the other communication applications prevents a user from running a VPN. As Signal, WhatsApp, and Telegram are now blocked in Russia, accessing them from inside the country requires a VPN or another circumvention tool. In other circumstances, this kind of blocking would lead to an increase in VPN use, as new users set up a VPN "just" to reach a blocked messenger while also gaining unrestricted access to the rest of the blocked internet, independent media, international news, and everything else Roskomnadzor filters. As Max refuses to operate when a VPN is present, it removes the incentive that would otherwise push users toward the very tool that frees the rest of their browsing. The user is left with a choice between a working state messenger and no VPN, or a VPN that unlocks the open internet but breaks the state messenger. The only relevant communication channels that still work without a VPN are SMS, which is fully exposed to the carrier and the state, and Max, which is no better.

Table 1.

Comparison with other messaging channels

	SMS	iMessage	Signal	Whatsapp	Telegram	WeChat	Max
End-to-end encryption	None	By default (Apple devices); fallback to SMS	All messages, always	All message content, by default	Secret Chats only, not default	None	None
Transport cryptography	Carrier network only	Standard	Standard	Standard	MTProto	MMTLS	GOST (FSB-licensed)
Operator	Mobile carrier	Apple (US)	Signal Foundation (US)	Meta (US)	Telegram FZ-LLC (UAE)	Tencent (China)	VK Group (Russia)
Compelled state access	Direct via SORM	US legal process	US legal process, minimal data to compel	US legal process	Contested	Required under Chinese law	FSB access required (SORM)
Contact discovery	Carrier sees all numbers	Queried contacts visible to Apple	Private set intersection	Number lookup (rate-limited)	Restricted to own contacts by default	Operator-visible	Full address book in plaintext
Blocks VPN use	No	No	No	No	No	No	Yes
Government-services integration	No	No	No	No	No	Regional, optional	State-mandated
Status in Russia	Available, carrier-monitored	Available	Blocked (Aug 2024)	Blocked (Feb 2026)	Largely blocked (Mar 2026)	Not widely used	State-mandated

CHAPTER 12

What Max means for society

Source: Pexels

Max's nation-wide rollout is not just a change to the country's communication architecture. This analysis finds that surveillance is built into Max's code. The architecture treats monitoring as an ordinary and integral function of the app, running in the background of everyday use. Anyone who owns a phone in Russia or the Russian-occupied territories in Ukraine is faced with a trap. Refusing to use Max, to avoid the normalized and ubiquitous surveillance of private and professional life that comes with it, means being excluded from most formal and informal social contexts: children can't be enrolled in school, neighborhood chats and other groups like sports clubs become inaccessible, contact with friends and family is harder, and government services become out of reach. The pressure to install it is also real, as teachers, employers, neighbors, and colleagues all push non-users to install the national app[29]. The threats to human rights are not just about the individual but also about the collective: the forced adoption of the super-app effectively undermines all forms of community organizing that are not supervised and ideologically supported by the government. This kind of isolation serves to shrink the social, emotional, and political space in which resistance, solidarity, and ordinary human connection can exist. Max is not simply a messenger that happens to be insecure, it is the infrastructure of that isolation, deployed at scale and enforced by law.

Furthermore, the Max server-pushed feature flag mechanism may lead to a panopticon-like chilling effect on society: as VK can (de) activate certain features such as keyword detection in voice calls, or elevated logging of user behavior for all or some specific accounts, remotely, anytime, and without the

29 https://novayagazeta.eu/articles/2025/12/12/academic-rigour-en

users' knowledge, Max users can never know for sure what exactly the state can see and when. In this way, population control is achieved with minimal financial and political cost, as people adjust their behavior to the ever-present threat of surveillance even when no data is extracted.

This paints a bleak picture. It is also the reason for doing this work. Building knowledge about how an authoritarian system operates is an essential condition for resisting it. This is why devoting resources and expertise to detailed analysis of this state-mandated surveillance app is not a hopeless academic exercise. Detailed information about surveillance and censorship implementation ensures actors neither underestimate nor overestimate the regime's capacities. This report attempts to provide some of this necessary knowledge.

CHAPTER 13

Who is most at risk

Source: Pexels

For some users, the forced install of Max is an existential problem. Journalists and opposition members rely on safe communication channels to be able to stay active in a repressive environment and fulfill their vital role of upholding some degree of transparency and accountability.

Religious and ethnic minorities, as well as queer Russians, have been under heavy political and social attack in the country for many years. Ethnic and religious groups are frequently targeted through 'anti-extremism,' 'anti-terrorism,' and 'foreign agent' laws. Targeted groups include Crimean Tatars, Muslims (particularly in the North Caucasus), Indigenous peoples, and Ukrainians both in occupied territories and within Russia.

For people who are excluded and targeted, community is vital. For these marginalized users, a messenger is far more than a phone app. It is where people find each other. It is often the channel for psychological reassurance, crowdfunding, international connection, local community, and a haven of normality within an often-brutal social context. This is why religious, ethnic, gender, and sexual minorities are most impacted by the vast Max rollout since 2025.

Additionally, the rollout of Max, coupled with the ban on alternative messengers, heavily affects anyone communicating with people outside Russia. With no other communication platforms and Max only operating in some forty countries, people within the country are limited in their ability to communicate directly with the rest of the world. Russian citizens living outside those countries lose the ability to easily communicate with friends and family at home. Max deepens the "digital iron curtain" that separates Russia and countries under its influence from the millions of Russians abroad, whether they left for politics, work, or study.

This division is especially difficult for Ukrainians in occupied and non-occupied territories: friends and families separated by the front lines and scattered across the country now face an additional barrier to staying in touch. In Russian-occupied parts of Ukraine, where mobile networks have been forcibly integrated into Russian infrastructure and access to Ukrainian and international services is increasingly restricted, Max is being pushed as the default communication tool. This means that Ukrainians under occupation who want to remain in contact with relatives in free Ukraine, in the diaspora, or with humanitarian organizations are forced onto a state-controlled channel, run by the government waging war against them.

Max's rollout is not only a privacy concern but a human rights concern as well.

Figure 4.

Who is at risk

Max's rollout restricts the communication channels that marginalized communities and cross-border families depend on, with alternatives banned and the app itself available in just 38 countries.

CHAPTER 14

What can be done

Source: Pexels

In the light of our analysis of Max's surveillance functionality, civil society, governments, and businesses who care about human rights in Russia can act on these findings through the following actions.

Circumvention

In Russia, VPNs are widely used to reach blocked websites and services. Max's messaging functionality is restricted when a VPN is active on the device itself but continues to function when traffic is routed through a VPN on the router. Users who rely on a VPN for routine access to restricted content therefore face a tradeoff between keeping Max usable and keeping their device-level VPN running.

Several workarounds allow them to do both:

A VPN travel router sends all phone traffic through a router configured to use a VPN. On the build we examined, Max does not currently detect this because its VPN check looks only for a VPN on the device itself. Max contains a second detection method, which matches network interface names; it is dormant in this build because it runs only when the application lacks the ACCESS_NETWORK_STATE permission, which Max always holds. Both methods check the device itself, so neither detects a VPN running on a router. On Android, it is also possible to isolate Max and other domestic applications that restrict VPN use to a dedicated secondary user profile or work profile (via the open-source app Shelter for example). Another possible solution that offers the strongest separation, although at additional cost, is to use a separate device used exclusively for Max and other domestic apps.

Another circumvention path that could be developed in the future is an open source or modified Max client with the VPN detection and restriction logic removed. This is technically feasible because the detection happens on the client side. The relevant checks can be identified and patched out of a modified build.

These approaches let users continue using Max where it is socially or practically required, while allowing for VPN use and access to restricted content. It is important to note as well that even in these scenarios a VPN only protects network-level metadata such as the user's IP address and approximate geolocation. The content of messages, images, audio, and calls remains visible to the VK Group in cleartext regardless of which workaround is used.

For this reason, Max should never be used to directly communicate sensitive or private information. Max lacks end-to-end encryption and presents a range of additional privacy and surveillance concerns documented elsewhere in this report. Vulnerable communities should continue to rely on trusted solutions such as Signal accessed through a VPN for any communication that requires confidentiality.

Research on circumvention and development of tailored technical advice

Further efforts into Max-specific surveillance and circumvention research are urgently necessary. The app's rapid mandatory rollout, its deep integration with state services, and its function as a chokepoint for nearly all digital communication inside Russia mean that the cost of leaving users without viable alternatives is measured not in inconvenience but in human safety.

This necessary research includes threat modeling tailored to specific at-risk groups. A queer teenager in a small Russian city, a Crimean Tatar journalist, an opposition activist in exile staying in touch with family, and a Ukrainian under occupation: each faces different adversaries, different risks, and different practical constraints. Generic technical advice is much less effective than group-specific guidance, co-developed with members of those communities.

International coordination and pressure

Democratic governments and (tech) businesses worldwide have leverage points against the Max surveillance system that have so far been used inconsistently. This report, clearly documenting Max's internet control functions, provides a coordinated technical documentation that makes it harder for Max to credibly claim legitimacy in any international forum.

In this field of action, one of the most effective measures is the removal of the Max application from the Google Play Store and Apple's App Store, supported by regulatory action by the EU or US. Removal of an application from the major app stores serves to make clear that it is not a "normal" messenger and limits its spread to unsuspecting users abroad.

It is important to note that between the completion of our analysis and the publication of this report, Max was removed from Apple's App Store on June 3, 2026, the Council of the European Union sanctioned VK and its subsidiary Communication Platform LLC on July 13, 2026, and Google removed Max from the Play Store on July 16, 2026. Despite these updates, we feel it is important to discuss what app-store removal requires and what it does and does not achieve to set out why these measures were warranted and identify where gaps remain.

The history of removal of Russian apps: The applications for Russia Today (RT) and Sputnik were removed from the Google Play Store and the Apple App Store in March 2022 following EU sanctions as a result of Russia's invasion of Ukraine. The apps became unavailable in the Google Play Store across the EU, the UK, Ukraine, Iceland, Liechtenstein, Norway and Switzerland, while Apple banned the apps in all locations except Russia itself.

In September 2022, the app VKontakte was temporarily removed from the global Apple App Store after the UK government enacted new sanctions on 23 executives at Gazprombank.

However, the removal lasted only about three weeks. In October 2022, Apple reinstated the app after VK provided documentation showing it was not directly controlled by a sanctioned entity. This example illustrates how app-store removal largely depends on sanctions law: the companies act when compliance requires it, and enforcement is largely based on how the company interprets which entities are covered.

Figure 5.

Imperfect circumvention tactics

People at risk face a tough decision because there is no silver bullet solution. Given the strength of Max's surveillance capabilities, circumvention strategies can only provide partial protection.

Your number is your identity in Max, and anyone can look up any registered number without being noticed. Keep in mind that your phone number is your identity in Max, and anyone can look up any registered number without being noticed.

VPN Travel Router

Using a travel router to send all phone traffic through a VPN

- ☑ allows Max to function
- ☑ Max currently does not detect router-based VPN connections

- ⚠ hides where you connect from, not what you send through Max
- ⚠ Max still reads your messages, contacts and activity inside the app
- ⚠ future Max updates may affect the viability of this strategy

Secondary Profile

Create a dedicated "work profile" or secondary user

- ☑ available on most Android phones
- ☑ lets you run Max in a separate space from your main apps

- ⚠ linked to your phone number, device and internet connection
- ⚠ other can still expose you: if someone has your number saved, Max will upload their entire address book to VK

Separate Device

A completely separate physical device for restricted applications

- ☑ offers the strongest separation
- ☑ Max cannot see your main phone's contacts or activity

- ⚠ requires extra hardware
- ⚠ needs its own phone number and its own internet connection
- ⚠ others can still expose you: if someone has your number saved, Max will upload their entire address book to VK

It is also important to note that company compliance cuts both ways. Reporters Without Borders has criticized Apple's compliance with Russian censorship demands[30]. In July 2024, Apple removed nearly 100 VPN apps from its Russian App Store[31], including 25 at Roskomnadzor's request[32].

These companies have acted when legally obliged to. The following are possible routes to create that obligation.

- **The EU's Digital Services Act:** The EU's Digital Services Act (DSA) could offer a legal solution. The DSA requires systemic risk assessments from Very Large Online Platforms (VLOPs), a category that includes the Google and Apple app stores. The DSA gives civil society some concrete pressure points. These include the potential for filing complaints with national Digital Services Coordinators (DSCs), submitting evidence directly to the European Commission, and requesting that the Commission investigate whether Apple's App Store and Google Play are adequately mitigating the systemic risks (under Articles 34 and 35 DSA) posed by hosting a state-mandated surveillance messenger.

- **The European Commission:** Russian disinformation campaigns are an explicit priority of the European Commission. The app's default onboarding media feed that includes Russian state sanctioned media channels (*see media list from our analysis in appendix B*) can be designated as disinformation. If the Max app is available for download within Europe the argument can be made that it leads directly to the spread of disinformation.

- **More Comprehensive Sanctions:** Sanctions coverage also remains incomplete for several entities materially involved in the development and deployment of Max. The EU listed VK Group and Communication Platform LLC in July 2026, but Max's commercial partner CryptoPro, which supplies the FSB-licensed cryptography on which Max's transport depends, remains unlisted on every major Western sanctions regime. It is clear through this analysis and prior research on Max that it is a state-aligned surveillance instrument, and its rollout is being coerced through institutional pressure on students, teachers, public-sector employees and ordinary citizens. It is already standard practice to not only sanction the operators but also the suppliers of the enabling infrastructure for this kind of surveillance technology.

30 https://rsf.org/en/apple-s-complicity-russian-censorship-rsf-calls-tech-giant-refuse-compliance-regulators-do-not

31 https://applecensorship.com/news/unveiling-the-extent-of-vpn-app-removals-by-apple-from-the-russia-app-store-an-analysis-of-silent-removals-and-the-need-for-transparency

32 https://techcrunch.com/2024/07/08/apple-removes-vpn-apps-at-request-of-russian-authorities-say-app-makers/

Relocation support

As surveillance and repression tighten, life in Russia for several at-risk groups becomes increasingly unbearable. Living under constant surveillance and political attack, while safe communication spaces disappear, can take a significant toll on mental and physical health. The technical countermeasures described above cannot address these threats alone.

Increased offers of relocation, both temporary and permanent, are therefore necessary to support the most vulnerable groups, including LGBTQ+ Russians, active political dissidents, independent journalists, lawyers, and members of persecuted religious and ethnic minorities.

However, the trajectory across the EU since 2024 has been one of narrowing rather than expanding relocation pathways for Russian human rights defenders and at-risk groups. For example, Germany's humanitarian admission program under Section 22 of the Residence Act has relocated approximately 2,500 Russian human rights defenders, journalists, and at-risk activists since 2022 through a fast-track procedure. However, this pathway was effectively closed in summer 2025, with humanitarian visas now issued only in unspecified 'exceptional cases.'[33] In 2025, Lithuania, Latvia and Estonia narrowed their support for Russian human rights defenders.

Some recent measures have gone the other way. The Czech Republic now grants citizenship to Russian applicants only if they renounce their Russian citizenship, a measure its interior ministry frames as strengthening internal security. Additionally, while European institutions have urged member states to offer protection to Russians facing persecution and have actively engaged with Russian opposition figures, these positions have not been backed by binding policy[34].

Non-governmental organizations doing this work should be supported. Reporters Without Borders supported 72 Russian journalists with relocation in 2024 and continues to provide emergency support and relocation grants worldwide[35]. Front Line Defenders runs protection grants as well as an emergency 24-hour multilingual phone line, including in Russian, and is lead partner in ProtectDefenders.eu, the EU Human Rights Defenders mechanism, a consortium of twelve organizations[36]. These programs are valuable but severely limited in scale.

Any serious human rights and anti-surveillance policy toward Russia must recognize that relocation support for the most targeted groups is not an optional supplement but a central obligation.

33 https://novayagazeta.eu/articles/2025/10/03/germany-dramatically-scales-back-humanitarian-visa-programme-for-russians-and-belarusians-en-news
34 https://ecfr.eu/publication/life-in-exile-a-new-approach-to-russian-democrats-in-europe/
35 https://rsf.org/en/seventy-cent-rsf-emergency-funds-allocated-resettling-exiled-and-displaced-journalists-2024
36 https://www.frontlinedefenders.org/en/who-we-are

This includes the expansion of relocation capacity, broadening eligibility criteria to cover people experiencing intersecting grounds of persecution, and simplifying documentation requirements for applicants whose identifying documents may themselves expose them to risk.

Addressing the super-app surveillance research gap

The Max application, and other state-mandated surveillance apps such as WeChat, illustrate a distinct and growing category of mobile surveillance: "super-app surveillance," implemented by openly distributed, nominally voluntary, state-mandated, or state-aligned applications in which surveillance functionality is concealed inside an otherwise legitimate-seeming consumer product. This category differs from former threat models that focus on covert commercial stalkerware installed by a private abuser without the target's knowledge on one side, and high-end commercial spyware such as Pegasus or Predator deployed by state actors through sophisticated exploit chains on the other side.

Super-app surveillance sits at the intersection of several existing bodies of work, including broader digital-authoritarianism literature, technical case studies of specific state-deployed surveillance applications, and the growing body of work on super-apps themselves. This emerging field has focused on WeChat as the canonical example of how a consumer super-app can function as both an everyday utility and a node in a state surveillance and policing infrastructure.

Max fits naturally into this lineage: like WeChat, it is positioned as a consumer messenger evolving toward a super-app role, integrates with state identity and service infrastructure, and operates within a legal and regulatory environment that obliges its operator to cooperate with state security. Adding robust analysis of Max will strengthen the broader field's ability to recognize, document, and respond to this category as it spreads.

CHAPTER 15

A note on this report's scope & on the EU listing

Source: Pexels

Everything in this report describes one build of Max, version 26.12.0, examined on our test devices between March and May 2026, under a specific configuration our two test accounts received. That sentence is not a disclaimer. It is the most important thing to understand about any analysis of this application, including our own and those that disagree with it.

Max's capabilities are gated behind feature flags set by VK's servers, per account, and changeable in real time without an application update. A capability can be present in the code and switched off. It can be switched on for one user and no one else. It can be off on the day one researcher looks and on the day another does. This is documented earlier in this report as a finding about Max. It is also a constraint on the report itself: what we observed is what this build did, for these accounts, in this window, and we have tried to be careful throughout to say which claims rest on something we watched happen and which rest on code we read.

Comparing our analysis with the statement from the Council of the European Union

On July 13, 2026, the Council of the European Union sanctioned VK, its subsidiary Communication Platform LLC, and that subsidiary's director, over Max. In its statement on the decision to sanction those responsible for Max, the Council listed five capabilities, citing the assessment of numerous experts. One of them was that Max can gather data on other applications installed on a user's phone.

We want to be precise about what our analysis found here, because we examined this capability closely and the answer is more specific than a yes or a no. The code for this capability is present in Max: the bundled MyTracker component can list installed applications, record the install time, hash the list, and upload it. On a reading of the code alone, the capability is unquestionably possible. In the build we examined, however, it does not run. The routine was never initialized, the component that would supply it the list of packages was never set, and the application did not declare the permission Android has required since version 11 to retrieve a full list of installed applications. We looked for this capability operating, and in our build, for our accounts, it did not.

These two statements do not contradict each other, and our analysis does not contradict the Council's statement. A capability can be shipped and remain dormant. Given that we found that features can be activated per account, what is dormant in our build is not necessarily dormant for every user. We cannot say the capability never operates. We can say, with evidence, that we did not observe it operating, and that in the specific build and accounts we examined the paths that would make it operate were not connected. We cannot say what analysis informed the Council's assessment, and we do not speculate about it. We offer only what we saw.

We put this observation to the European Union before publication. The EEAS Sanctions Division replied that the Statement of Reasons should be read first and foremost as an explanation of how designated persons and entities meet the listing criteria set out in the legal

acts, and that both the intended features and the documented functions of Max are relevant to that assessment. On our specific question, whether the description records capabilities present in the application or behavior observed in operation, the answer is that both count. Our observation refines the description rather than contradicting it, which is how we have presented it. The full response appears at the end of this report.

Where the record diverges, and where the evidence sits

Several researchers have independently examined other builds of Max over the same period. Their work is credited earlier in this report. Our findings largely align with theirs: on the VPN detection, on the cleartext transmission of the mobile subscriber identifier, on the on-device audio model and its server-delivered payload, on the structure of the network-reachability report, and on the mechanism by which Max collects the address book. Two teams reaching the same result by different methods on separate builds is corroboration, and it should be read as strengthening those findings rather than as a contest over who reached conclusions first. Below we set out where our evidence contradicts a published claim, where we ourselves got it wrong, and which claims we did not test.

Of the points where our findings differ from other accounts, this is the one where our evidence is strongest: zarazaexe's May 2026 analysis of Max's code on the platform Habr reported that Max hashes the phone numbers in a user's address book before uploading them. In the build we examined it does not. We captured the upload on the wire during a fresh account registration, and the numbers cross the network as plain digits. In the code, the routine that reads as though it might hash is a formatting normalizer: it strips non-digit characters, removes an international prefix, rewrites a leading 8 to a 7, and invokes no cryptographic function of any kind. Here reading the code and watching the traffic give different answers, and the traffic is dispositive. This matters because it shows Max's privacy protections are weaker than previously reported.

Another capability reported in zarazaexe's analysis, which we examined and could not reproduce, is the covert recording of calls. The claim describes raw audio being captured during a call and uploaded with no indication to the user. The pieces this would require are present: the calling component contains a recording function, and the reporting component can upload a file. What we could not find is the connection between them. The recording function is never called anywhere in the build we examined, the folder it would write to is never set, it returns without writing anything, and the upload component transmits only files it is explicitly handed rather than sweeping a directory. The command named in the claim only travels from the application to the server, and not the other way. Its one handler accepts replies, so an instruction arriving under that name from the server would go unrecognized. This looks like the local audio logging that ships with WebRTC, the library Max is built on, for debugging purposes. It appears to have been left in place and unused, and not a surveillance capability in operation. We found no path from a server message to recorded audio. RKS Global reached

the same conclusion about the underlying audio-writing function independently, by a different method.

One setting we could not fully resolve: A server-controlled flag, labeled "log local audio" in the application's own developer menu, is read at the start of every call and passed to the calling library. We could not determine from the code what it makes the library write, or where, and we didn't test it on a live device. What we can say is that it was never switched on while we were watching: the application reported the setting as off on all 112 occasions it appeared in captured diagnostic traffic between 2 April and 14 May 2026, one hundred of them for one test account and twelve for the other. That is the application's own account of its state rather than an instruction we saw the server send, and we did not capture any configuration message carrying this setting. We therefore report what the setting does as unresolved. It is neither a confirmation of the claim above nor does it undercut it, and a reader should treat our finding as bounded by it.

What we did not test

Our analysis does not cover everything that has been reported about Max. Where we did not test something, we say so, rather than treat our silence as a verdict. zarazaexe's analysis describes two things we did not examine: a server-controlled flag that would disable validation of the encrypted session, and a device identifier derived from the phone's hardware media-DRM module that would survive a reinstall or a factory reset. We can neither confirm nor dispute either claim, and we flag them as live questions that this report does not close.

We have since checked two further points on which our account and zarazaexe's differed. On the first, the analysis is right: the component we had treated as part of the application's crash-reporting system is a separate reporting channel with its own address, credentials and payload, and it shares no code with the system we had grouped it under. That same channel bears on a further disagreement. He reported that the list of services Max probes is supplied by the server rather than fixed in the application. He named the wrong mechanism, but this second channel does take an unrestricted list of targets from the server. We record both halves, because correcting the mechanism while dropping the finding would be the more misleading error. On the second point it is mistaken: the permission he reports as an unsubstituted placeholder left in the application's manifest is present in full and correctly formed, although the application never uses it.

What this is for

Max will be argued about for a long time: in sanctions proceedings, in app-store compliance reviews, in courts, and in the ordinary choice by a user of whether to install it. Those arguments will turn on technical claims, and technical claims about Max are unusually hard to make definitively because the application is built to resist inspection and because what it does is a

setting on a server rather than a fixed property of the software. That is a reason for care, not for silence. We have tried to mark every claim in this report for what it is: observed, inferred, tested and not reproduced, or left open. We have published our negative findings alongside the positive ones, including the ones that make Max look less alarming than the current public picture, because a report that only reports in one direction is not evidence. Reverse engineering a state-mandated application from inside or near the country that mandates it is done by people taking real risks, and their work gave this investigation its leads. We are not correcting them. We are adding one more careful reading to a record that will need every careful reading it can get.

This report is not the last word on Max and cannot be. Any analysis of it, this one included, is a photograph of a moving target.

CHAPTER 16

Responses

Source: Pexels

Before publication we put our findings to the parties named in this report and offered each the opportunity to respond. We wrote to VK Group and to MAX LLC, the entity that operates Max, setting out the technical findings that concern them. We wrote to Apple and to Google regarding the removal of Max from their application stores. And we wrote to the Council of the European Union regarding Council Decision (CFSP) 2026/1709. Each was asked to reply by September 1, 2026, and each was told that we would publish any response in full or record that they had declined to comment. The EEAS Sanctions Division replied on behalf of the European Union, and its response is reproduced in full below. VK Group, MAX LLC, Apple and Google did not respond.

Response from the European Union, received from the EEAS Sanctions Division, reproduced in full:

First, let us underline that the designations related to Max App were made under the restrictive measures regime in view of the situation in Russia, which aims to target persons responsible for serious violations or abuses of human rights, repressions of civil society and democratic opposition, or actions undermining democracy and the rule of law in Russia, as well as persons who provide support for, or are otherwise involved in, these acts. The dramatic situation of civil society in Russia and broad scale of repressions, including restricting freedom of expression and freedom of association, is the central point, as was underlined in the press release accompanying the adoption of this package[37]*. The Statement of Reasons should be read first and foremost as an explanation of how the designated individuals and companies fulfil the listing criteria set out in the legal acts (Council Decision (CFSP) 2024/1484 and Council Regulation (EU) 2024/1485). Both the intended features and documented functions of the Max App are relevant for us.*

As Council's deliberations are confidential, we cannot confirm or deny whether an individual or entity was considered for listing. We can confirm that the listings in the July FAC package related to surveillance were adopted by the Council acting in unanimity.

The Council conducts regular (usually annual) reviews of sanctions regimes. The review is an opportunity to update information contained in the Annex, including the name of the company, other identifying information, and the Statement of Reasons. We took note of the change of the name of the mentioned company, and we will update the Annex accordingly during the next review. Including other identifying information in the entry (such as the TIN and INN numbers) ensures that the measures are properly implemented.

We would also like to draw your attention to similar listings of tech entities, who provide products or services that are used by various regimes to run surveillance and repress their citizens, who were listed under the EU Global Human Rights Regime (NtechLab, Tevian, and others) and Belarus sanctions regime (Synesis).

Best regards,
EEAS Sanctions Division

37 https://www.consilium.europa.eu/en/press/press-releases/2026/07/13/human-rights-violations-in-russia-council-lists-four-individuals-and-five-entities-over-abusive-surveillance/

CHAPTER 17

Appendices & supporting material

Source: Pexels

Appendix A: Registration countries

The list of countries whose SIM card holders can register with Max has been obtained from the app's traffic (it appears in the payload of every SESSION_INIT response from *api-gost.oneme.ru*). The list contains thirty-eight country codes that always appeared in the same order. No per-account or per-session variation was observed. All captures originated from a Russian-residential exit (Moscow), so whether the server varies the list by requesting-IP geolocation was not tested.

Country names have been looked up for readability. The wire payload carries only the two-letter codes.

#	CODE	COUNTRY
1	AZ	Azerbaijan
2	AM	Armenia
3	KZ	Kazakhstan
4	KG	Kyrgyzstan
5	MD	Moldova
6	TJ	Tajikistan
7	UZ	Uzbekistan
8	GE	Georgia
9	TH	Thailand
10	TR	Turkey
11	TM	Turkmenistan
12	AE	United Arab Emirates
13	LA	Laos
14	MY	Malaysia
15	ID	Indonesia
16	CU	Cuba
17	KH	Cambodia
18	VN	Vietnam
19	AF	Afghanistan
20	BO	Bolivia
21	CD	Congo (Democratic Republic of the)

#	CODE	COUNTRY
22	CG	Congo (Republic of the)
23	CO	Colombia
24	GD	Grenada
25	GM	Gambia
26	IN	India
27	IQ	Iraq
28	KN	Saint Kitts and Nevis
29	KW	Kuwait
30	LB	Lebanon
31	MM	Myanmar
32	NI	Nicaragua
33	PK	Pakistan
34	PW	Palau
35	QA	Qatar
36	SA	Saudi Arabia
37	VE	Venezuela
38	TZ	Tanzania

Appendix B: Channels proposed to newly registered Max users (onboarding feed)

The list is not a popularity ranking.

#	CHANNEL	SUBSCRIBERS	DESCRIPTION (SERVER-SUPPLIED)	VERIFIED ATTRIBUTION
1	РИА Новости	1,020,072	Official *ria.ru*	State news agency, Rossiya Segodnya
2	VK ПРО СПОРТ	25,946	Sports media on VKontakte	VK platform sports channel
3	СОЛОВЬЁВ	664,018	Vladimir Solovyov on Max	Host of state TV's "Evening with Vladimir Solovyov"; EU-sanctioned
4	Чемпионат	48,431	"Highest-quality sports channel"	*championat.com* sports portal
5	Дмитрий Медведев	301,856	Official channel of Deputy Chairman of the Security Council of the RF	Former president of Russia, deputy chairman of the Security Council
6	Mikhail DELYAGIN	16,676	"Doctor of economics, Deputy Chair of Duma Committee on Economic Policy. Wanted by Interpol, under all sanctions"	State Duma deputy (A Just Russia); EU/US-sanctioned
7	Третьяковская галерея	14,956	Official channel of the State Tretyakov Gallery	State art museum, Moscow
8	savelenok	19,095	Lifestyle/interior design blogger	Yulia Savelenok
9	Украина.ру	148,275	"Top expert on Ukraine. News and analysis from Rossiya Segodnya media group"	Operated by Rossiya Segodnya state media group
10	Мир Михаила Онуфриенко	186,396	"Reviving Great Rus!"	Mikhail Onufrienko, pro-Russian military commentator based in Crimea since 2014

#	CHANNEL	SUBSCRIBERS	DESCRIPTION (SERVER-SUPPLIED)	VERIFIED ATTRIBUTION
11	Военкор I Z I Лисицын ZOV-TV	17,716	“News from the front”	Evgeny Lisitsyn, pro-war military correspondent; ZOV-TV registered as media outlet
12	Коммерсантъ	86,431	“Official channel of the Kommersant Publishing House”	Russian business daily, owned by Alisher Usmanov
13	Мы везде. Николай Долгачёв	8,971	“VGTRK military correspondent for the Vesti program”	State TV (VGTRK) war correspondent
14	Kostenko	4,035	Lifestyle/motherhood blogger	Not independently verified
15	Поделкин	18,759	Children’s crafts channel	Not independently verified
16	ТУТ ЖИВЕТ ВАДИК	133,053	“Blogger. Content for good mood”	Vadim Spiridenkov, comedy creator
17	IZ.RU	133,064	“Everything important to know”	Izvestia daily, National Media Group (Yuri Kovalchuk); state-affiliated
18	BEARWOLF	5,327	(no description)	Valeriya Vasilevskaya, fitness and modeling
19	НТВ	107,709	“News, analysis, reports”	Federal TV network, owned by Gazprom-Media (86.52%)
20	Антитеррор Урал	7,845	(no description)	Official channel of Anti-Terrorism Commission of Sverdlovsk Oblast
21	Пикабу	56,917	“Official channel of Pikabu. Warm and lamp-style”	Russian social-news platform
22	Домохозяйка со стажем Галина	8,089	Cooking blog	Not independently verified

#	CHANNEL	SUBSCRIBERS	DESCRIPTION (SERVER-SUPPLIED)	VERIFIED ATTRIBUTION
23	ЖАСМИН	3,346	Singer Jasmin’s official channel	Sara Manakyan, Russian pop artist
24	Марьяна Локель	197,170	Teen content creator	Maryana Lokel, songwriter and creator
25	Mash	788,439	“Channel of fast, exclusive, important news”	News outlet, News Media Holding (Aram Gabrielyanov heritage)

Appendix C: Max's proprietary binary protocol — 51 command types

The 51 distinct protocol command codes observed across all captures, grouped by functional category. Codes marked with an asterisk are server-only push commands; codes without an asterisk are client-initiated request-response pairs. Codes marked with a dagger are client-initiated commands for which only the server's response was captured.

Session and authentication

CODE	HEX	NAME	DIRECTION
1	0x0001	PING	Both
6	0x0006	SESSION_INIT	Both
18	0x0012	AUTH	Both †
19	0x0013	LOGIN	Both
23	0x0017	AUTH_CONFIRM	Both †
96	0x0060	SESSIONS_INFO	Both
158	0x009E	OK_TOKEN	Both †

Sync

CODE	HEX	NAME	DIRECTION
21	0x0015	SYNC	Both †

Telemetry

CODE	HEX	NAME	DIRECTION
5	0x0005	LOG (carries NAV, NET, PERF, HOST_REACHABILITY, DEV, VIDEO_STATS, CHANNEL_RECSYS_FOLDER, ERROR, INVITE_MAX_BANNER, BANNER, SHARE_TO_MAX, PERMISSION, CALL, AUDIO_STATS, CONTACT_RENAME_BANNER, CONTACT_OR_BLOCK sub-events)	Both

Chats

CODE	HEX	NAME	DIRECTION
48	0x0030	CHAT_INFO	Both
49	0x0031	CHAT_HISTORY	Both
50	0x0032	CHAT_MARK	Both
51	0x0033	CHAT_MEDIA	Both
61	0x003D	CHAT_PERSONAL_CONFIG	Both
75	0x004B	CHAT_SUBSCRIBE	Both
144	0x0090	CHAT_BOT_COMMANDS	Both
145	0x0091	BOT_INFO	Both
198	0x00C6	CHAT_SEARCH_COMMON_PARTICIPANTS	Both
272	0x0110	FOLDERS_GET	Both
300	0x012C	CHAT_SUGGEST	Both

Messaging

CODE	HEX	NAME	DIRECTION
64	0x0040	MSG_SEND	Both
65	0x0041	MSG_TYPING	Both
74	0x004A	MSG_GET_STAT	Both
128	0x0080	NOTIF_MESSAGE	*
129	0x0081	NOTIF_TYPING	*
130	0x0082	NOTIF_MARK	*
136	0x0088	NOTIF_ATTACH	*
156	0x009C	NOTIF_MSG_YOU_REACTED	*
178	0x00B2	MSG_REACTION	Both
179	0x00B3	MSG_CANCEL_REACTION	Both
180	0x00B4	MSG_GET_REACTIONS	Both
258	0x0102	REACTIONS_SETTINGS_GET_BY_CHAT_ID	Both

Contacts

CODE	HEX	NAME	DIRECTION
32	0x0020	CONTACT_INFO	Both
34	0x0022	CONTACT_UPDATE	Both
35	0x0023	CONTACT_PRESENCE	Both
46	0x002E	CONTACT_INFO_BY_PHONE	Both

Search

CODE	HEX	NAME	DIRECTION
60	0x003C	PUBLIC_SEARCH	Both
68	0x0044	CHAT_SEARCH	Both

Presence

CODE	HEX	NAME	DIRECTION
132	0x0084	NOTIF_PRESENCE	*

Identity handoff

CODE	HEX	NAME	DIRECTION
160	0x00A0	WEB_APP_INIT_DATA	Both †

Voice transcription

CODE	HEX	NAME	DIRECTION
202	0x00CA	TRANSCRIBE_MEDIA	Both
293	0x0125	NOTIF_TRANSCRIPTION	*

Account management

CODE	HEX	NAME	DIRECTION
162	0x00A2	COMPLAIN_REASONS_GET	Both
200	0x00C8	PROFILE_DELETE_TIME	Both

Calls

CODE	HEX	NAME	DIRECTION
78	0x004E	VIDEO_CHAT_START_ACTIVE	Both
82	0x0052	VIDEO_UPLOAD	Both
83	0x0053	VIDEO_PLAY	Both
301	0x012D	AUDIO_PLAY	Both

Assets

CODE	HEX	NAME	DIRECTION
27	0x001B	ASSETS_UPDATE	Both
28	0x001C	ASSETS_GET_BY_IDS	Both †
302	0x012E	BANNERS_GET	Both †

† Client-initiated command for which only the server's response was captured. The sessions carrying these commands recorded server-to-client traffic only, so the request side is absent from our captures rather than absent from the protocol.

INTER SECLAB

Authors:
- Anna Enders
- Lea Horne
- Marla Rivera

* The names of authors are displayed in alphabetical order.

Editor:
- Laura Schwartz-Henderson

Editorial Support:
- Curious Shapes

How to cite:
INTERSECLAB. The Max Messenger: An Analysis of Russia's State-Mandated Messaging Application. InterSecLab, September 2026.

Available at: *interseclab.org/research/max*