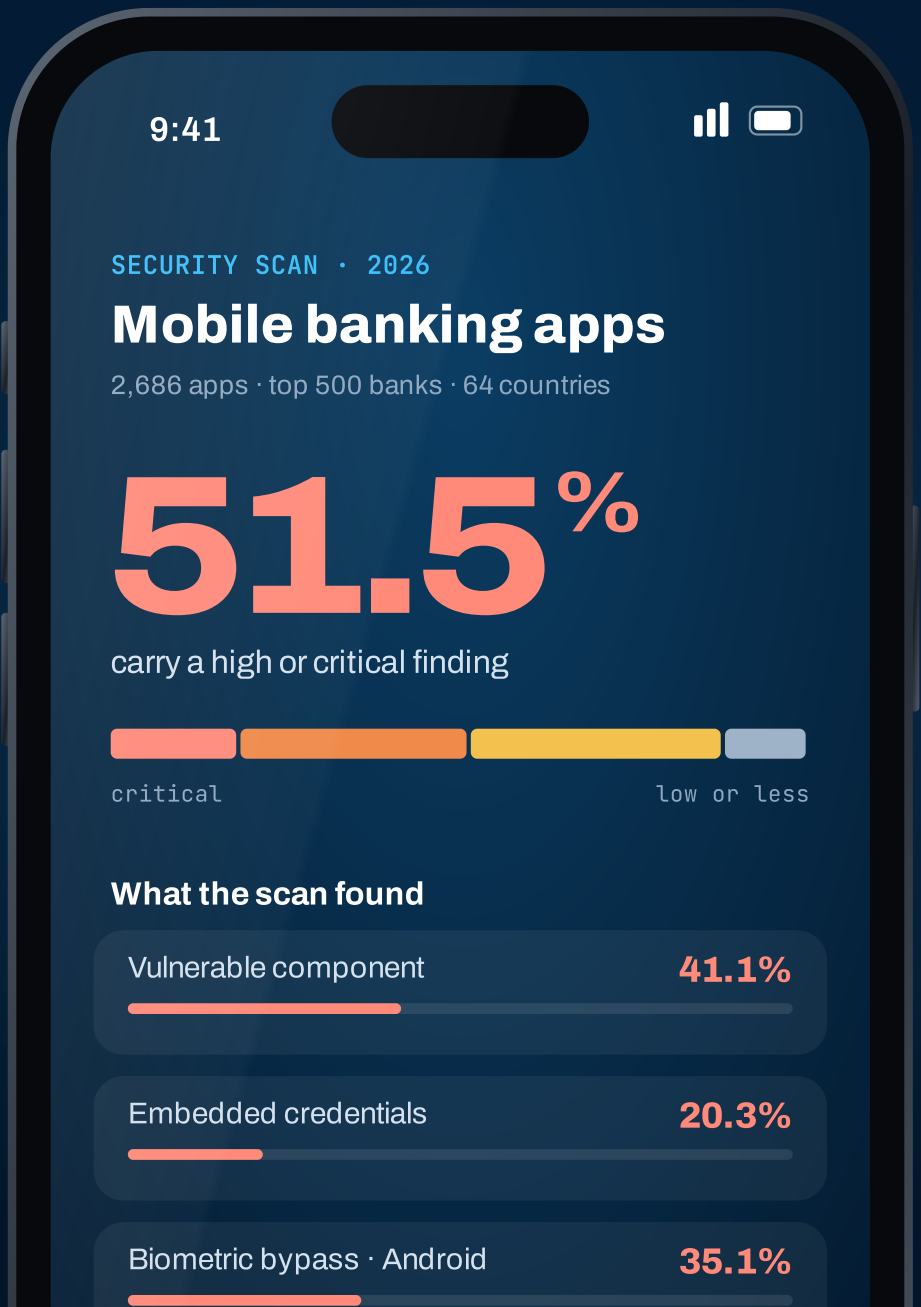
Ostorlab

Mobile Banking App Security Report 2026

What 2,686 banking apps from the top 500 banks in 64 countries ship on Android, iOS and HarmonyOS, and what that means for fraud.

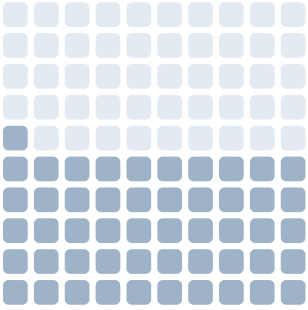


KEY FINDING

Independent neobanks carry the most serious findings.

Scans of neobank apps are rated high or critical more often than scans of traditional bank apps, 61.1% against 51.2%, and the whole gap comes from independent neobanks. Neobanks owned by a traditional bank look like the banks that own them.

51.2%

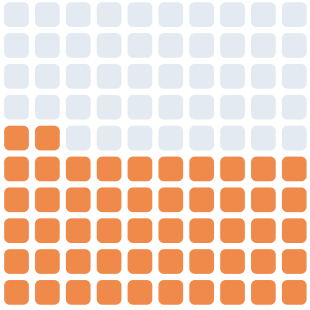


Traditional banks

High-street, retail and corporate banks

2,596 APPS

52.5%

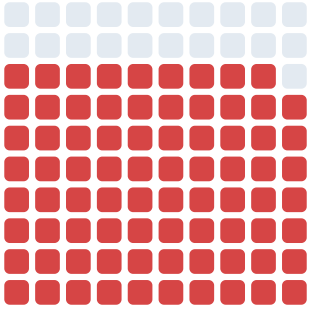


Bank-owned neobanks

Digital brands of a traditional bank

61 APPS

79.3%

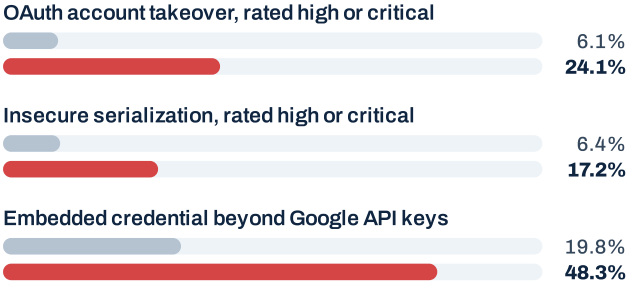


Independent neobanks

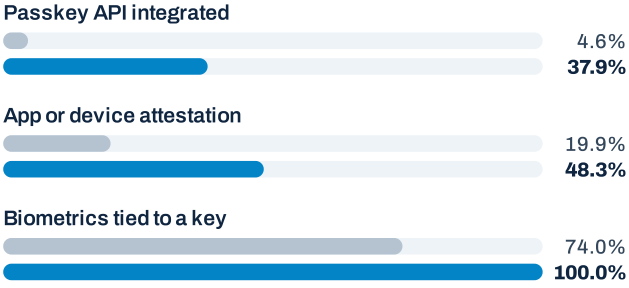
Digital-only banks with no traditional parent

29 APPS

What drives the gap



Yet they lead on modern defenses



SOURCE Ostorlab automated security analysis of 2,686 banking apps.

CONTENTS

2,686 apps, eleven chapters.

Every line links to its page.

Key finding: independent neobanks 02
Summary 04
The report in nine numbers 05

01 How we measured 06
2,686 apps, from every region 07
Three kinds of evidence, five rules for every number 08

02 How banking apps are built 09
Hybrid web apps carry the most risk 10
One platform, many banks 11
Updated often, built to run on old phones 12
A new release every 6 to 9 weeks 13

03 What ships inside 14
Analytics everywhere, fraud SDKs in few apps 15
The AI bank is not in the app yet 16

04 What the scans found 17
Half the apps have a serious finding 18
Code and privacy findings reach most apps 19
One app in five embeds a credential that should be secret 20
Four in ten apps ship a known vulnerability 21
Each platform fails in its own way 22

05 Authentication and payments 23
Passkey code ships, passkeys do not 24
Does the bank's website name the app? 25
Biometrics are often tied to a key, yet a third can be bypassed 26

06 Shielding: what apps ship 27
Protections found in the code 28
SDKs carry many of the checks 29
Few apps layer their defenses 30

07 Beyond the app 31
Bank domains: quantum-ready, hard to report to 32
Privacy labels that do not match the app 33
Some bank servers answer on more than the web 34
One bank in six has a public breach record 35
The rules banks answer to, and what they claim 36

08 The threat landscape 37
Four in ten banks are on a trojan's list 38
2026: attackers turn the bank's own app against it 39
Would a bank notice its app had been swapped? 40
Where the money goes 41

09 By region, bank type and country 42
No region has it all 43
Country results 44
The safest bank apps by region 45

10 Case studies 46
CyberDakait: seven trojans, one phone call 47
WindRelay: a loan from the victim's own app 48
Hirudo: the trojan that replaces your banking app 49
Inside the copy, every check can be made to lie 50
The Google API key that became an AI credential 51

11 What to do next 52
What banks should do in 2026: the top 10 52

A Definitions and data notes 53

The controls that stop modern fraud are still the rare ones.

This year we looked at 2,686 mobile banking apps from the top 500 banks in 64 countries. Banks update often and most apps ship the basics. The protections that matter against today's attacks, such as passkeys, attestation and runtime shielding, remain in a minority of apps.

Half the apps carry a serious finding

51.5% of apps have at least one high or critical finding from our automated scan. Only 11.8% have nothing above low. The most common causes are credentials embedded in the app (20.3% beyond Google API keys) and components with known vulnerabilities (41.1%).

Strong authentication is still emerging

Only 4.9% of apps integrate a passkey API, and 20.2% use app or device attestation. On Android, 35.1% have a biometric flow our scanner could bypass, even though most apps that bind biometrics do so to a key.

Attackers now target the app itself

Apps from 192 of the top 500 banks appear on public banking-trojan target lists. In 2026, campaigns such as Hirudo and Gigabud went further, replacing or cloning genuine banking apps on the victim's phone. In the UK, mobile banking carried 52% of authorised push-payment scam losses in 2025.

Shielding is common in pieces, rare as a whole

On Android, 69.9% of apps check for root, but only 26.9% ship a commercial or platform shielding product. Our data shows what apps contain, not how well it holds up; that needs device testing.

THREE FINDINGS FROM THE SCANS

51.5%



of apps have at least one high or critical finding

1,384 of 2,686 apps

35.1%



of Android apps have a biometric flow our scanner could bypass

Android apps

4.9%



of apps integrate a passkey API, the control that makes phished passwords useless

The report in nine numbers.

COMPONENTS 01

41.1%

of apps ship a component with a known vulnerability

1,103 of 2,686 apps

Chapter 04 · page 17 →

SECRET KEYS 02

20.3%

of apps embed secrets such as API keys, access tokens or private keys

544 of 2,686 apps; Google API keys excluded

Chapter 04 · page 17 →

OLD PHONES 03

96%

of apps install on versions that no longer get security patches

Android 99% · HarmonyOS 100% · iOS 72%

Chapter 02 · page 12 →

ATTESTATION 04

20.2%

of apps use app or device attestation

525 of 2,602 apps with readable code

Chapter 05 · page 23 →

SHIELDING 05

26.9%

of Android apps ship a runtime shielding product

Commercial or platform products, Android apps

Chapter 06 · page 27 →

AI 06

2 apps

reference a large language model API; none ships an LLM SDK, while 28.3% use on-device ML

Of 2,686 apps; barcode, text and face models lead

Chapter 03 · page 14 →

TROJANS 07

38%

of banks have apps on public banking-trojan target lists

192 of the top 500 banks; lists mostly from 2021 to 2023

Chapter 08 · page 37 →

POST-QUANTUM 08

71%

of reachable bank web hosts negotiate post-quantum TLS

732 of 1,038 completed handshakes

Chapter 07 · page 31 →

SCAM LOSSES 09

52%

of UK scam losses in 2025 went through mobile banking

£301.8m of £576.4m in authorised push-payment scams (UK Finance)

Chapter 08 · page 37 →

CHAPTER 01

01

How we measured

We downloaded 2,686 banking apps from the top 500 banks in 64 countries and ran each one through Ostorlab's automated security analysis, on Android, iOS and HarmonyOS.



On screen: the sample by country and region.

2,686

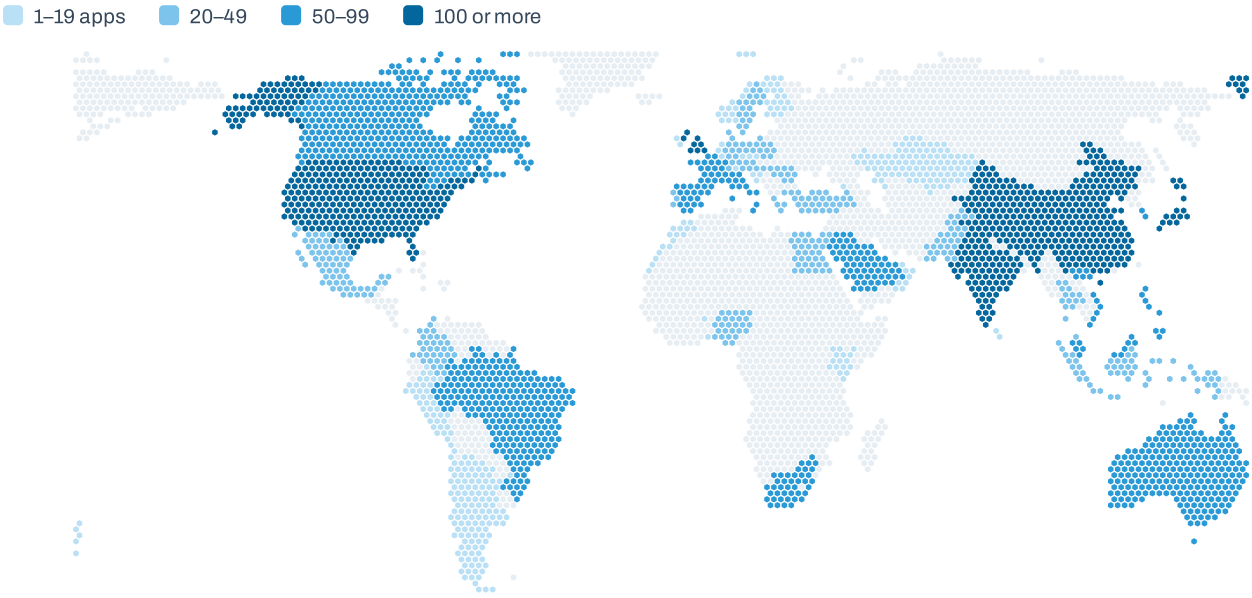
apps from the top 500 banks in 64 countries

2,686 apps, from every region.

The sample covers 2,686 apps from the top 500 banks, including 29 of the world's systemically important banks, on Android, iOS and HarmonyOS.

Banks from 64 countries, on every continent

FIG 1.1

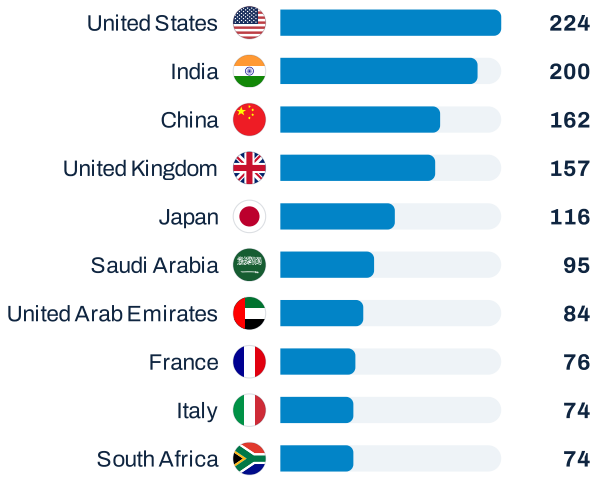


Apps analysed per country, by the bank's home market. 2,841 listings were collected; 155 could not be downloaded.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

The ten largest markets

FIG 1.2

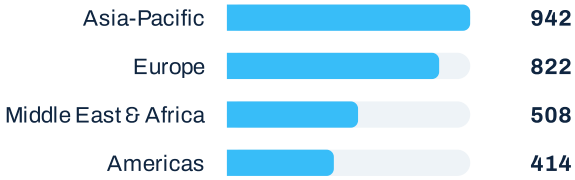


Number of apps, by the bank's home country.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Asia-Pacific is the largest region

FIG 1.3



Number of apps, by the bank's home region.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Three kinds of evidence, five rules for every number.

Each figure in this report comes from Ostorlab's analysis of the apps, and every number follows the same five rules, so that the charts, the text and the captions agree.

App inventory

Lists what each app is built with: framework, SDKs, permissions, protections and the hosts it talks to. All 2,686 apps.

Automated security scan

Ostorlab's scanner rules find vulnerabilities, rate their severity and map them to OWASP MASVS. Rules differ by platform. All 2,686 apps.

Control signals

Sixteen checks for authentication, payment and privacy controls, each with the code evidence that triggered it. All 2,686 apps.

HOW TO READ EVERY NUMBER

01 **Every share states its denominator**

Each percentage names the apps it was measured on, in the caption under the chart.

02 **Platform rules stay on their platform**

A rule that exists only on Android is reported as a share of Android apps, never of all apps.

03 **Unknowns are counted, not guessed**

Packed or unreadable code is left out of a check's denominator, and the number left out is shown.

04 **Google API keys are reported apart**

Google expects them in apps, so the headline credential figure excludes them.

05 **No bank is named**

Evidence comes from real apps, shown without the bank's name and with every secret masked.

Reading the numbers

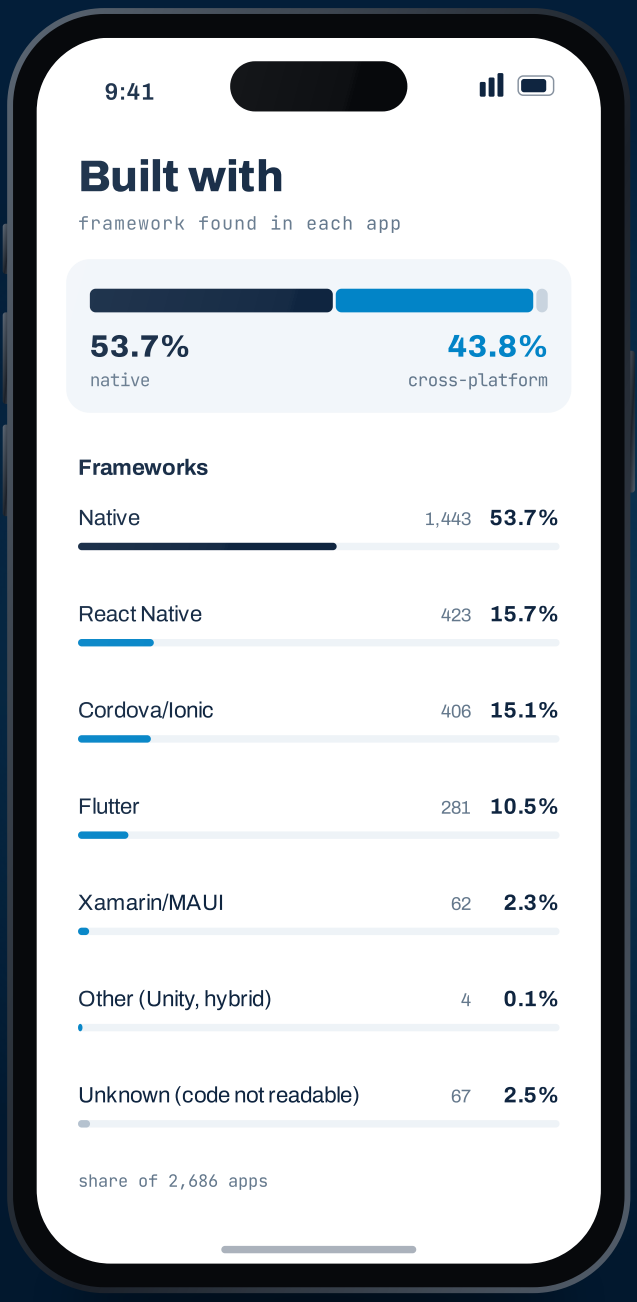
A finding means our scanner matched a rule in the shipped package. It is evidence that an issue is present, not proof that it is exploitable in production. Severity follows the scanner's ratings: critical, high, medium and low. The scanner also uses two lower ratings. Potentially marks a pattern that may be a real flaw but needs a person to confirm it, such as a string shaped like a key. Hardening marks a missing safeguard. An app is rated by its most severe critical, high, medium or low finding; potentially and hardening findings do not change that rating. Where a chart counts them, it says so.

CHAPTER 02

02

How banking apps are built

Most banks still build natively, but four in ten apps now run on a cross-platform framework. The choice shows up in the security results.



On screen: the framework found in each app.

43.8%

of apps are built on a cross-platform framework

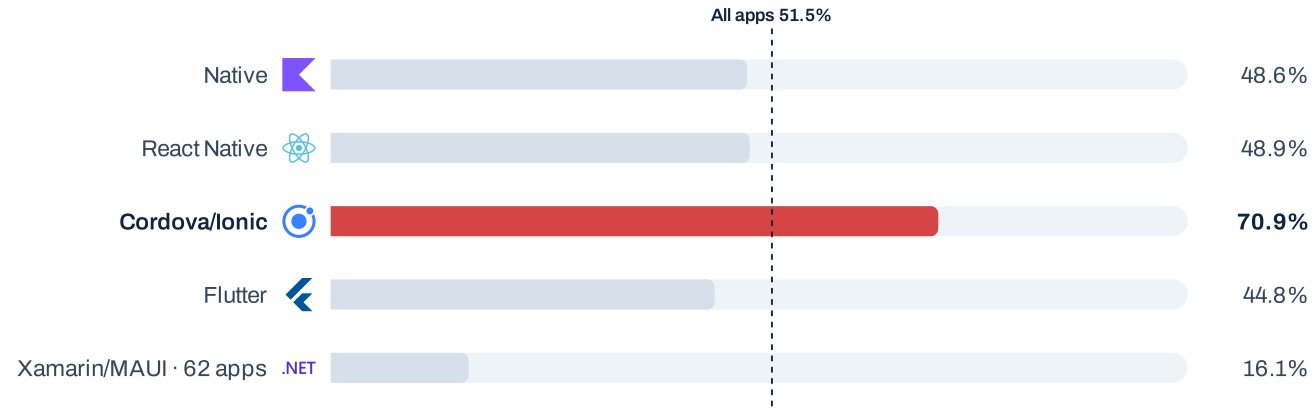
1,176 of 2,686 apps

Hybrid web apps carry the most risk.

Cordova and Ionic apps package a web app inside a native shell. They are 15.1% of the sample, and 70.9% of them have a high or critical finding, against 48.6% of native apps.

70.9% of Cordova and Ionic apps have a serious finding

FIG 2.1

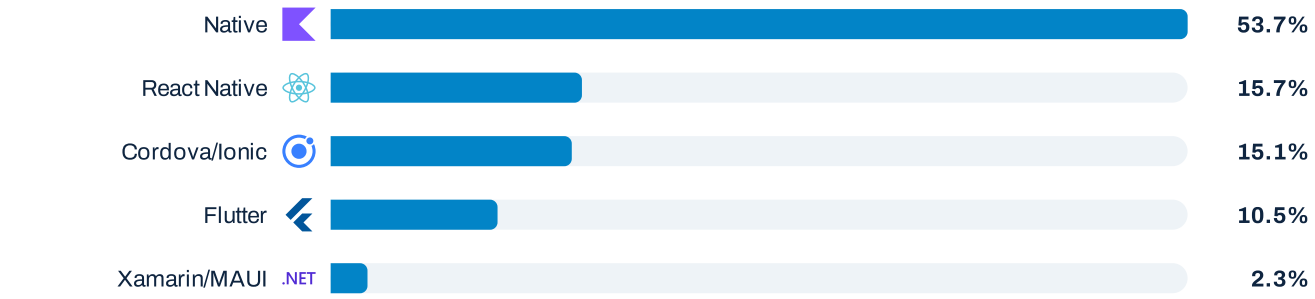


Share of each framework's apps with at least one high or critical finding. Dashed line: all 2,686 apps.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Native code still builds most banking apps

FIG 2.2



Share of 2,686 apps by framework; 67 with unreadable code not shown.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Cordova and Ionic apps also lead on vulnerable components (72.2%), mostly outdated JavaScript libraries bundled into the web layer.

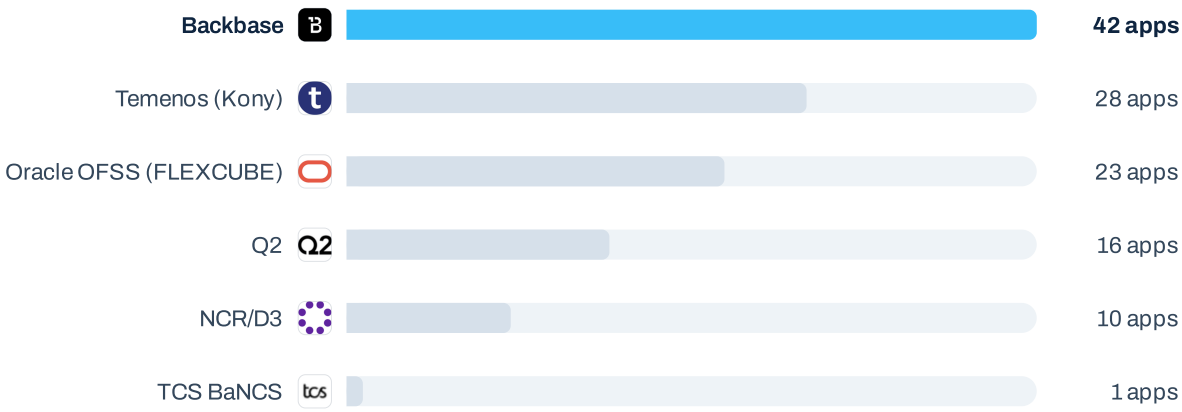
There is no clear difference between native (48.6%), React Native (48.9%) and Flutter (44.8%) apps. Xamarin and MAUI apps look best, but they are a small group of 62 apps, and they most often embed credentials (43.5%).

One platform, many banks.

Many banks do not write their app from scratch: they license a digital-banking platform and brand it. 120 apps in the sample carry the code signature of one of 6 such platforms.

Backbase is the most detected banking platform

FIG 2.3



Number of apps. 120 of 2,602 apps match a known platform signature. A match needs recognisable platform code, so these counts are a lower bound.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

> **Where a platform is shared, one vulnerability in it can reach every bank that uses it.**

Why it matters

A flaw in shared platform code is a flaw in every app built on it, and the fix depends on the vendor's release, then on each bank shipping it. One disclosure can expose many banks at once.

What good looks like

Ask the platform vendor for a software bill of materials, its security test results and an agreed time to patch. Track the vendor's advisories as closely as your own.

Updated often, built to run on old phones.

Banks keep their apps current with the platform: 72.1% of Android apps target the latest Android API. But on every platform, most apps still install on operating-system versions that no longer receive security patches.

On every platform, most apps install on unpatched OS versions

FIG 2.4



Share of each platform's apps with a readable minimum OS version. An app counts when its declared minimum OS version is older than the oldest version still receiving security patches (Android 14, iOS 18; HarmonyOS apps declare an Android version). This shows what the app allows, not what customers run.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Supporting old phones is a business choice: customers keep them. The risk is that the app runs on a system with known, unpatched flaws. Only 0.5% of Android apps refuse unpatched versions; on iOS, 28.5% require iOS 18 or newer. Banks can check the patch level at login and limit high-risk actions on devices that no longer get updates.

Oldest version each app can be installed on

FIG 2.5

Android and HarmonyOS · Android version

	Android	HarmonyOS
4 or older	0.8%	4.0%
5	4.4%	12.8%
6	5.3%	7.8%
7	17.4%	24.3%
8	19.5%	19.5%
9	13.1%	13.0%
10	16.5%	12.4%
11	7.0%	3.4%
12	12.8%	2.9%
13	2.7%	0.0%
14 or newer	0.5%	0.0%

iOS · iOS version

	iOS
11 or older	1.8%
12	2.1%
13	5.0%
14	6.5%
15	27.0%
16	23.1%
17	5.9%
18 or newer	28.5%

Share of each platform's apps by the oldest version they install on. A bar means exactly that version; such an app also installs on every newer one. Android 14 and iOS 18 are the oldest versions still patched.

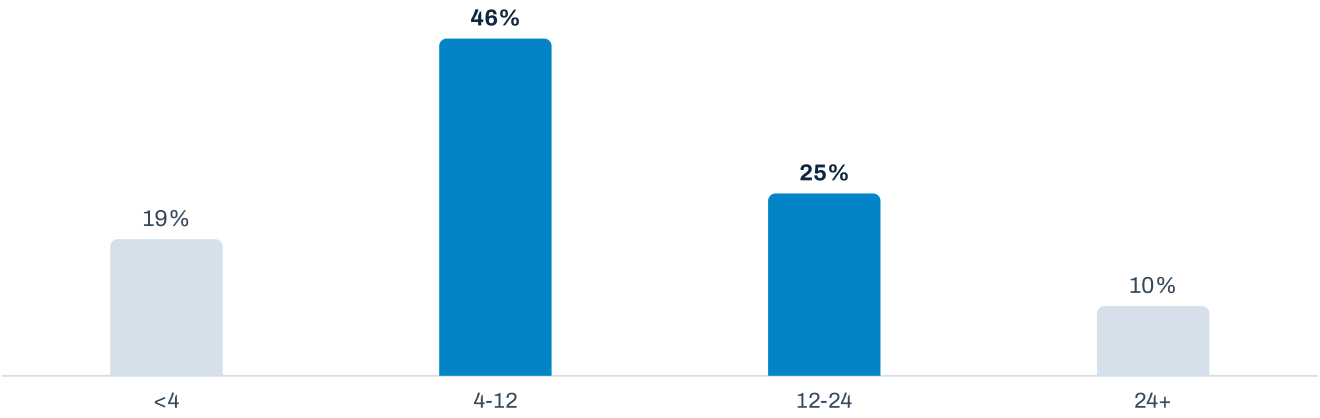
SOURCE Ostorlab automated security analysis of 2,686 banking apps.

A new release every 6 to 9 weeks.

The median Android app ships at least 8.9 releases a year (rebuilt from Google Play reviews) and the median iOS app 6.0 (App Store version history). The HarmonyOS store lists only the current version, so HarmonyOS could not be measured. Frequent releases are a strength: they are how fixes for the findings in this report reach customers.

Most apps ship between 4 and 24 releases a year

FIG 2.6



Share of Android apps with a release history rebuilt from Google Play reviews, by releases per year; median 8.9, a lower bound because releases nobody reviewed are missed. iOS: median 6.0, from App Store version history.

SOURCE Google Play and App Store listings and reviews, collected for this report.

75.1%
of Android apps with a Play update date were updated in the last 90 days

6.8%
were not updated in a year

3.8bn
Google Play installs across the Android apps, at least

Size

Median download size is 68 MB on Android, 38 MB on iOS and 129 MB on HarmonyOS. Half of the native libraries in these apps come from known third-party SDKs, and only 4% can be tied to the bank itself. Third-party code is the subject of the next chapter.

Fast releases, old components

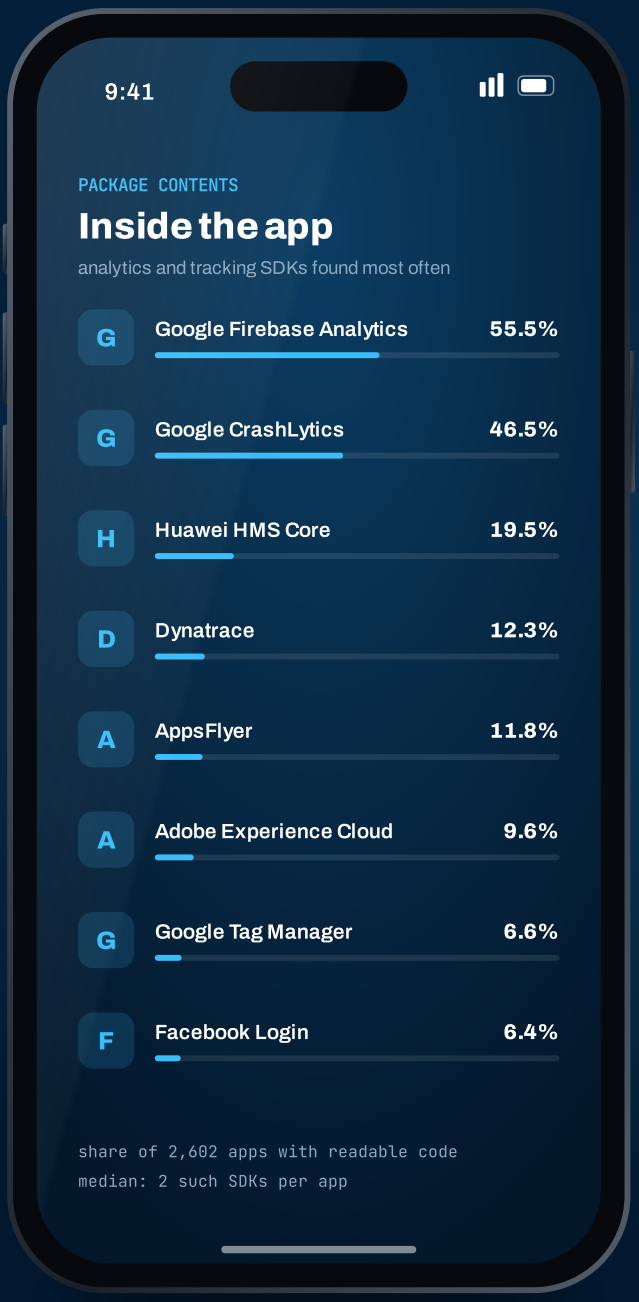
Shipping often does not keep every part current. As chapter 4 shows, 41.1% of apps still bundle a component with a known vulnerability.

CHAPTER 03

03

What ships inside

A banking app is mostly other people's code: analytics, crash reporting, fraud detection, maps, scanners. Each SDK brings its own data flows and its own vulnerabilities.



On screen: the analytics and tracking SDKs found most often.

80.6%

of apps with readable code ship at least one analytics or tracking SDK

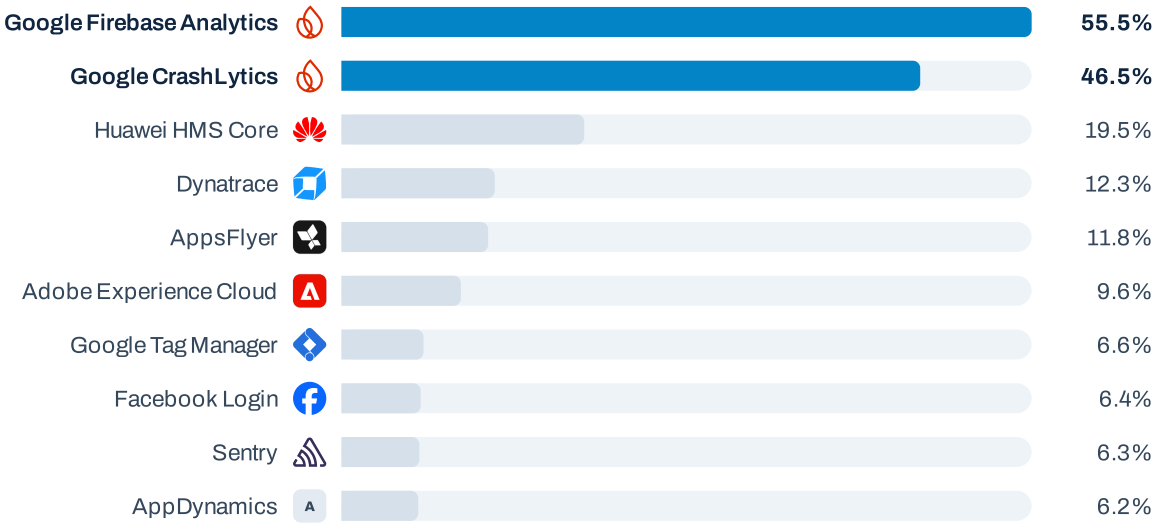
2,096 of 2,602 apps

Analytics everywhere, fraud SDKs in few apps.

Google's Firebase Analytics appears in 55.5% of apps. Dedicated fraud and device-risk SDKs, which banks use to spot account takeover in real time, appear in 12.1%.

Google's Firebase Analytics and Crashlytics lead the trackers

FIG 3.1

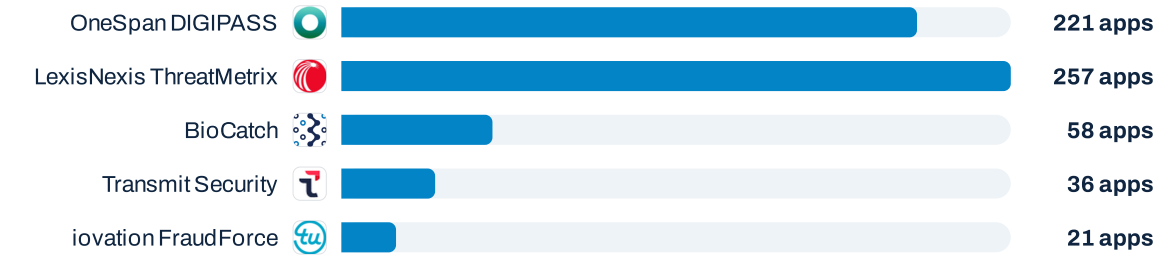


Share of 2,602 apps with readable code. Tracker signatures from the Exodus Privacy database. Huawei HMS Core is found in 64% of HarmonyOS apps; it covers Huawei push, location and analytics services.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Specialist security SDKs reach few apps

FIG 3.2



Number of apps. Authentication SDKs: 9.3% of apps. Fraud and device-risk SDKs: 12.1%.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

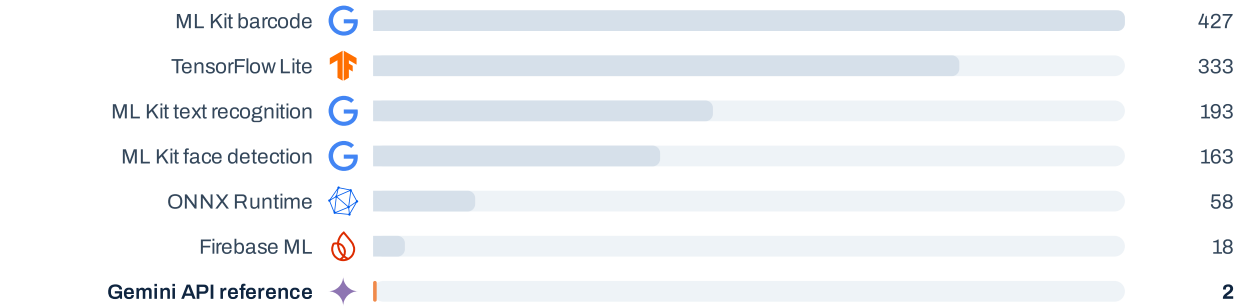
Specialist SDKs are concentrated in iOS apps: 35.3% of iOS apps ship a fraud or device-risk SDK, against 8.8% on Android. Several vendors serve many banks, so their security matters well beyond any single app.

The AI bank is not in the app yet.

28.3% of apps ship on-device machine learning, almost all of it for reading barcodes, documents and faces. No app ships a large language model SDK: 2 of 2,686 apps reference an LLM API, and we found no AI provider keys embedded in any app.

On-device ML reads barcodes, text and faces

FIG 3.3

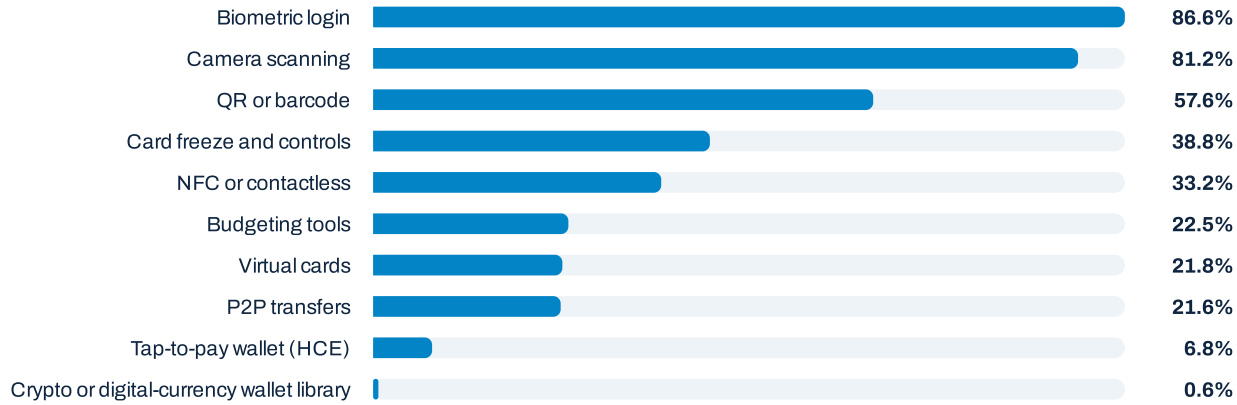


Number of apps. Gemini API reference: the app names the Gemini API endpoint; no app ships an LLM SDK. LLM features may run entirely on the bank's servers and would not appear in the app.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Biometric login and camera scanning are near universal

FIG 3.4



Share of 2,686 apps. Detected from code, resources and permissions; a feature present in code may be switched off. Crypto or digital-currency wallet library: counted when the app's code ships a wallet or blockchain library; it does not show that the app offers crypto trading.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Apps with a chatbot or virtual-assistant SDK embed credentials more often

FIG 3.5



Share of apps with at least one embedded credential beyond Google API keys. A chatbot SDK is a vendor chatbot or virtual-assistant library such as Dialogflow, Kore.ai, Kasisto or Watson Assistant, not a large language model; only 2 apps name the Gemini API, and neither ships a chatbot SDK. Larger banks more often run one and ship more SDKs: a correlation, not a cause.

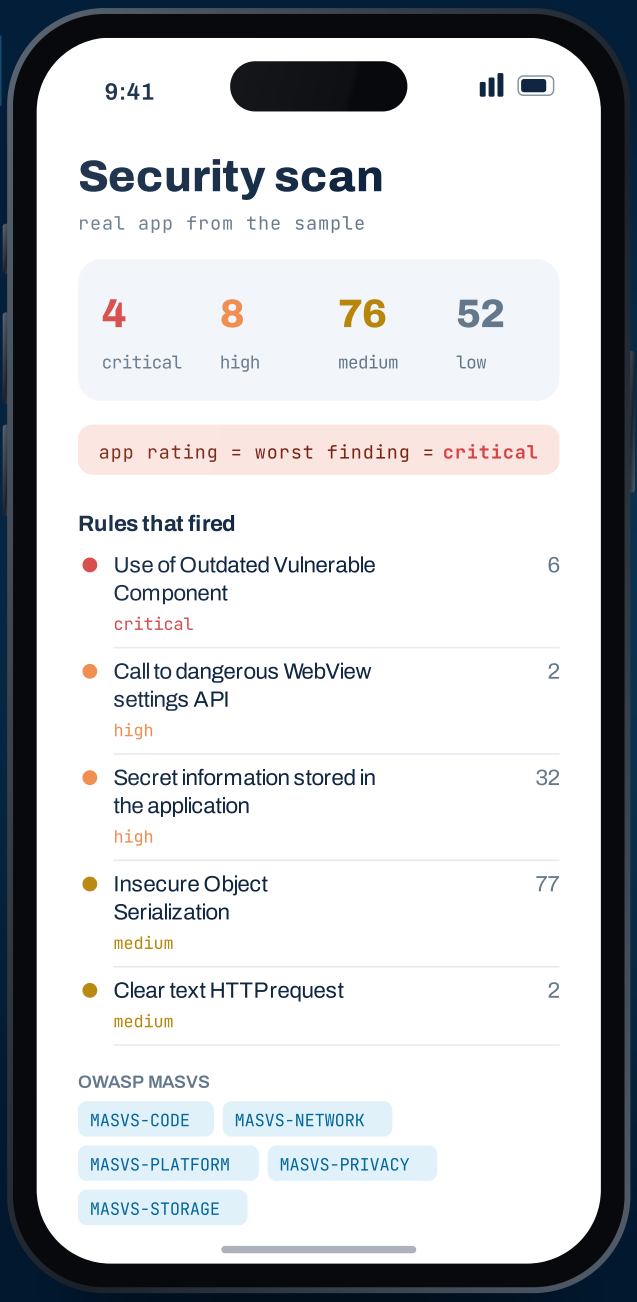
SOURCE Ostorlab automated security analysis of 2,686 banking apps.

CHAPTER 04

04

What the scans found

Ostorlab's automated scan found at least one high or critical issue in half the apps. Embedded credentials and outdated components lead, followed by issues specific to each platform.



On screen: the scan report of a real app from the sample.

18.4%

of apps have at least one critical finding

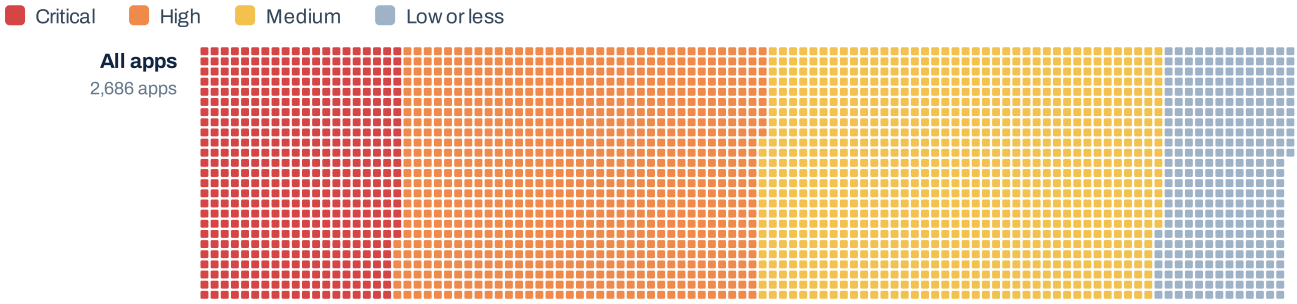
494 of 2,686 apps

Half the apps have a serious finding.

We rate each app by its most severe finding. On Android, 54.2% have at least one high or critical issue; on iOS, 36.2%. HarmonyOS apps have the most critical findings, at 31.6%, driven by outdated components.

Every one of the 2,686 apps, by its most severe finding

FIG 4.1

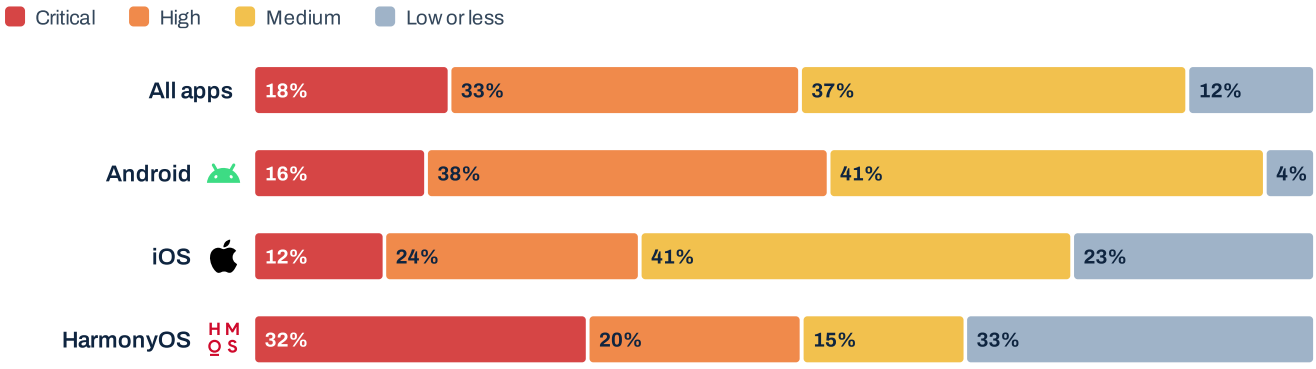


One square per app, sorted by severity. Low or less includes apps whose worst finding is low, potentially or hardening, and apps with no finding.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

HarmonyOS apps have the most critical findings

FIG 4.2



Share of apps by most severe finding. All 2,686 apps. Low or less includes apps whose worst finding is low, potentially or hardening, and apps with no finding. The larger low-or-less share on iOS and HarmonyOS partly reflects fewer scanner rules on those platforms.

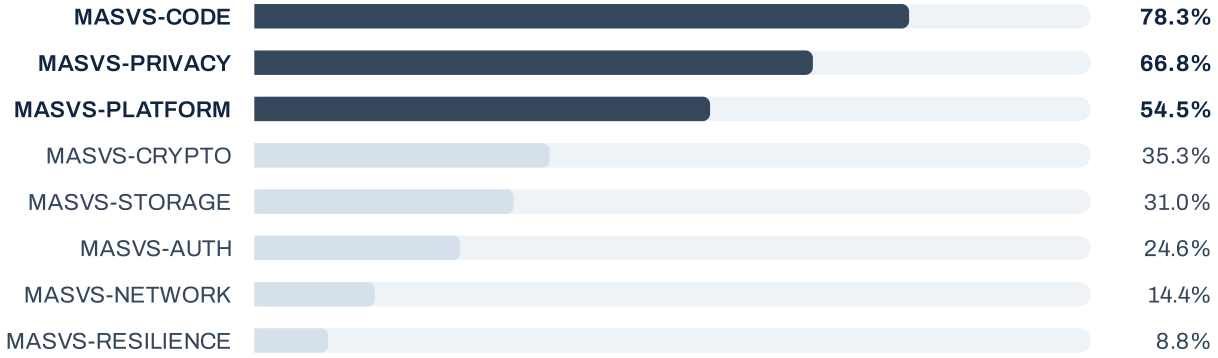
SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Code and privacy findings reach most apps.

Most findings map to a control group of OWASP MASVS, the industry standard for mobile app security. Counting critical, high, medium and low findings, code findings appear in 78.3% of apps and privacy findings in 66.8%. iOS and HarmonyOS apps show no crypto or authentication findings, which pulls those groups down.

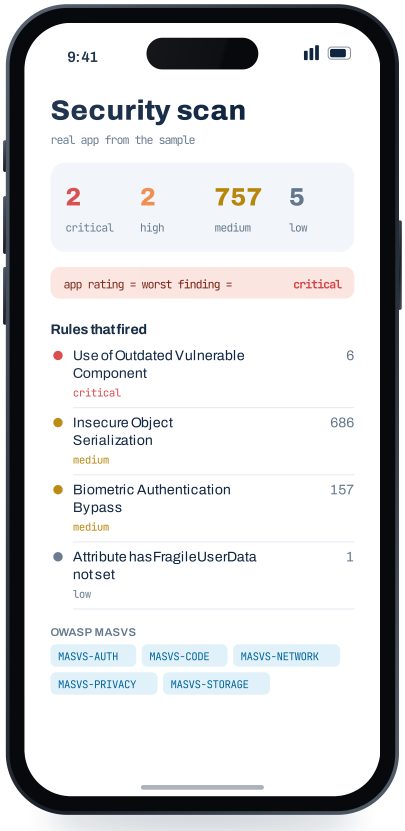
Share of apps with a finding in each OWASP MASVS control group

FIG 4.3



Share of 2,686 apps with at least one critical, high, medium or low finding in each group. “Potentially” and “hardening” findings are left out. 15% of Android findings map to no group.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.



A real app scan from the sample

ONE APP, AS THE SCANNER REPORTS IT

766 findings, one rating

This Android app from the sample has 2 critical, 2 high, 757 medium and 5 low findings. Like every app in this report, it is rated by its worst finding: critical.

A rule can fire many times in one app. The medium count is large because one serialization rule matched hundreds of call sites; it is still one issue to fix.

The scanner also reported 98 “potentially” and 2 “hardening” items. Potentially marks a pattern that may be a real flaw but needs a person to confirm it; hardening marks a missing safeguard. Neither changes the app’s rating.

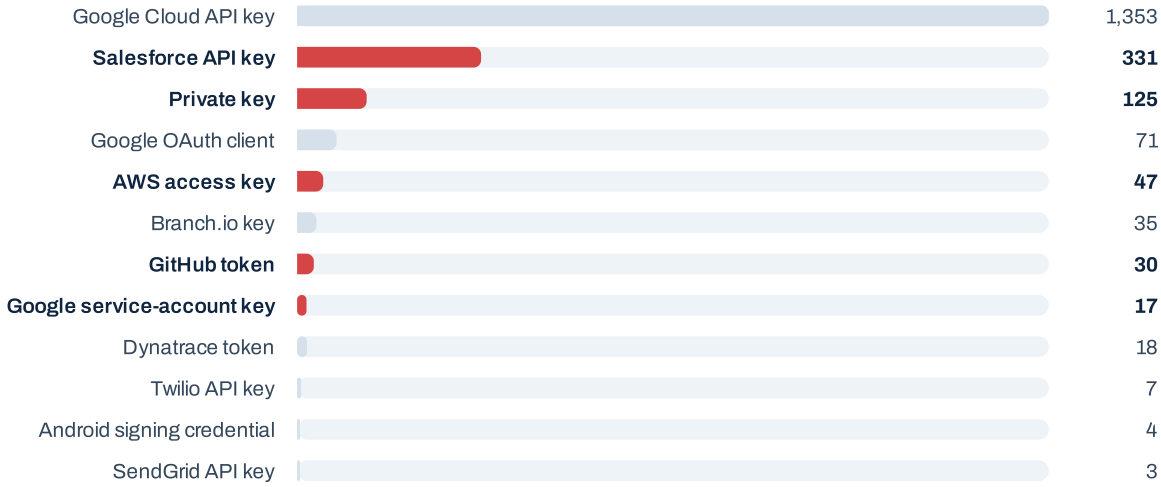
MASVS-AUTH-2 MASVS-CODE-1 MASVS-CODE-3 MASVS-CODE-4 MASVS-NETWORK-1 MASVS-PRIVACY-1 MASVS-PRIVACY-3 MASVS-PRIVACY-4 MASVS-STORAGE-1 MASVS-STORAGE-2

One app in five embeds a credential that should be secret.

59.8% of apps contain at least one credential in a recognisable format. Most are Google API keys, which Google designs to ship inside apps: they identify a project for services such as Maps and Firebase, and are protected by restrictions set in the cloud project (the app's signature, named APIs), not by secrecy. We did not test whether any key is restricted. Leaving those aside, 20.3% of apps still ship private keys, cloud keys and access tokens (Android 18%, iOS 21%, HarmonyOS 28%).

After Google API keys, Salesforce keys and private keys lead

FIG 4.4



Number of apps where the scanner's credential rules flagged each type; an app can contain several. A search of the whole package, a different method, finds a Google API key in 80% of Android apps, 71% of HarmonyOS apps and 31% of iOS apps (page 51).

SOURCE Ostorlab automated security analysis of 2,686 banking apps.



From a real app scan in the sample; values masked

Anyone can download a banking app and read what is inside it.

ONE APP'S PACKAGE

This Android app ships 4 kinds of candidate secret, among them a Google API key, a Sentry DSN, a Firebase database address and signed tokens. Each is readable by anyone who downloads the app.

A Google API key can gain new access after the app ships: see the case study on page 51.

What good looks like

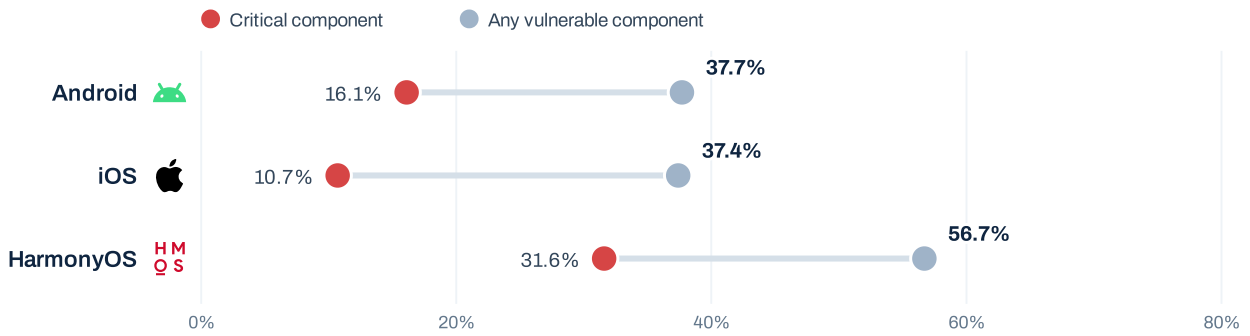
Keep secrets on the server and issue short-lived tokens to the app. Where a key must ship, as with Google Maps, restrict it to the app's signature and the APIs it needs, then monitor its use.

Four in ten apps ship a known vulnerability.

41.1% of apps include a third-party component with a published vulnerability, and 18.2% include one rated critical. HarmonyOS apps fare worst.

HarmonyOS apps ship the most vulnerable components

FIG 4.5

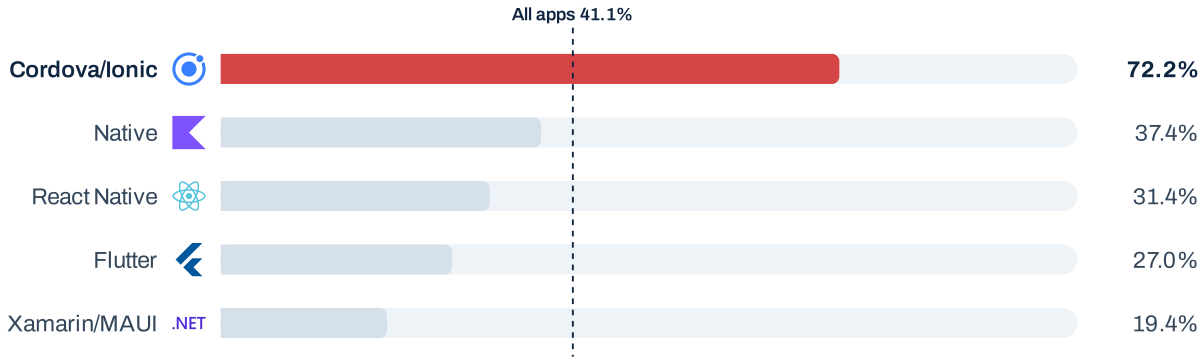


Share of apps on each platform. The label gives the share with any vulnerable component.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

72.2% of Cordova and Ionic apps ship a vulnerable component

FIG 4.6



Share of each framework's apps. Dashed line: all 2,686 apps.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Why so persistent

Components are pulled in by other components. A bank can update its own code every month and still ship a years-old library inside an SDK it does not control.

What good looks like

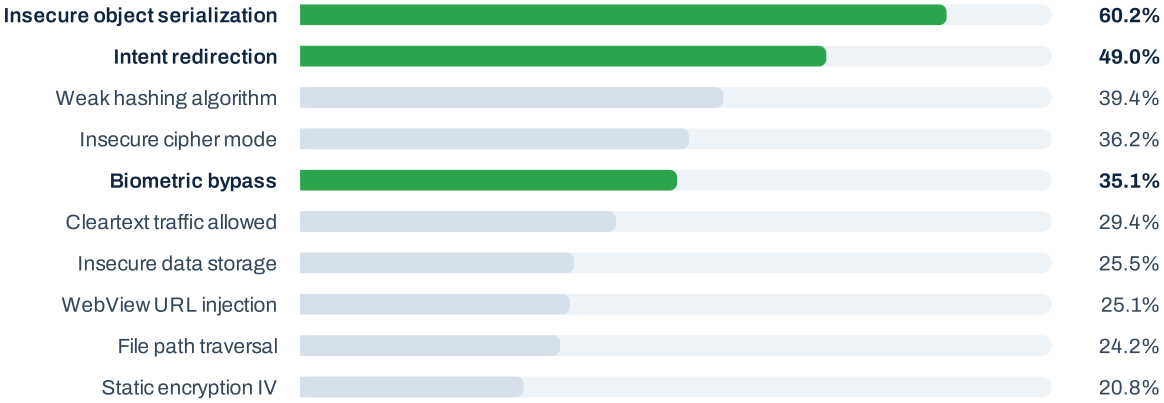
Keep a software bill of materials for every release, track advisories against it, and set a deadline for fixing critical components. Ask SDK vendors for the same.

Each platform fails in its own way.

Many scan rules exist for one platform only, so we compare apps within a platform. On Android, code-level issues dominate. On iOS, the leading issues are configuration. On HarmonyOS, old components and web code lead.

Android: serialization and intent redirection lead

FIG 4.7



Share of Android apps. Android-only rules.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

iOS: configuration issues lead

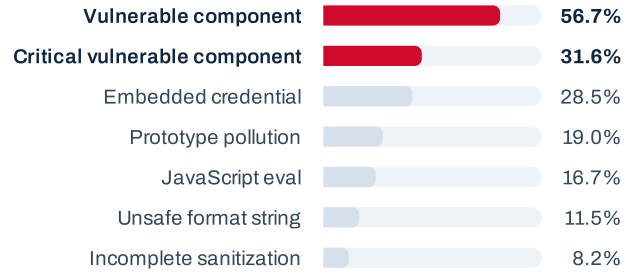
FIG 4.8



Share of iOS apps. Universal links prevent the OAuth takeover, where another app claims the login redirect.

HarmonyOS: old components and web code lead

FIG 4.9



Share of HarmonyOS apps.

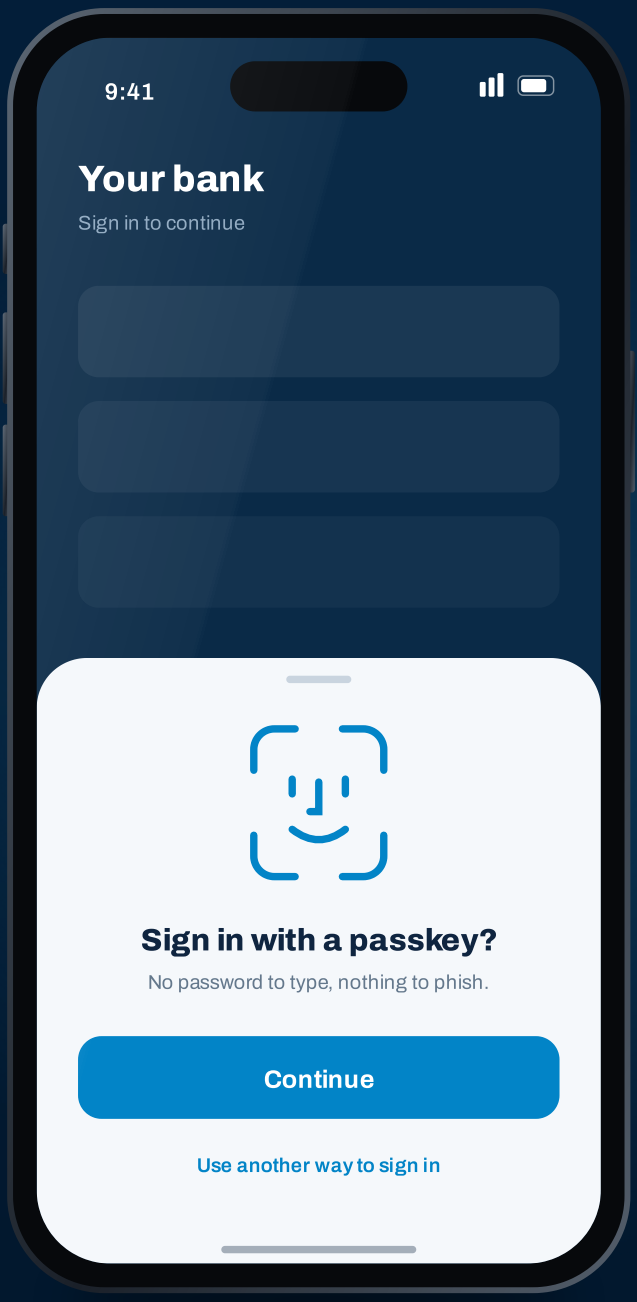
SOURCE Ostorlab automated security analysis of 2,686 banking apps.

CHAPTER 05

05

Authentication and payments

Passkeys, attestation and key-bound biometrics make stolen passwords and tampered apps far less useful. They are spreading, slowly.



On screen: an illustration of a passkey prompt, not a real app.

4.9%

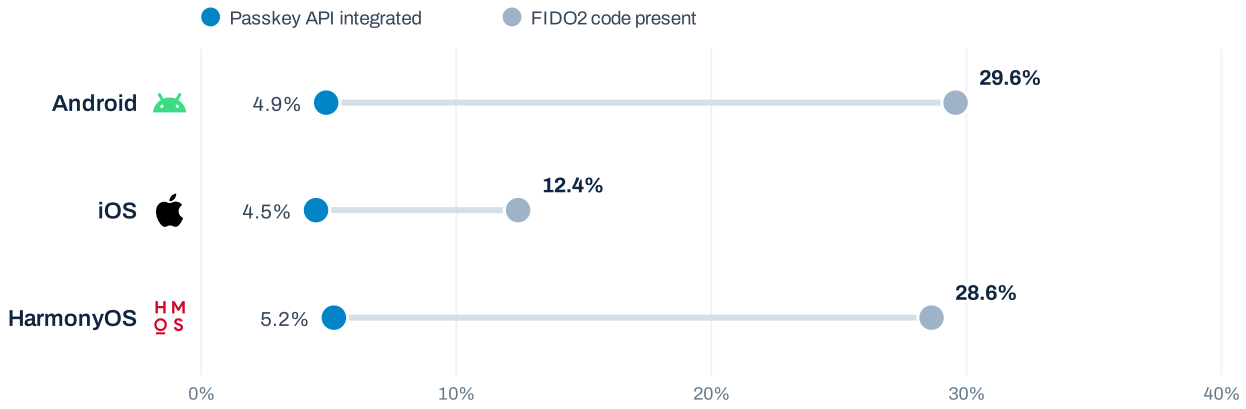
of apps integrate a passkey API

Passkey code ships, passkeys do not.

Nearly a third of Android apps contain FIDO2 or WebAuthn code, but only 4.9% call a passkey API. On iOS, 4.5% do. Code can arrive inside other libraries without the feature being switched on.

FIDO2 code is common; passkey APIs are rare

FIG 5.1



Share of apps on each platform. Passkey API integrated: the app calls Credential Manager, the Google FIDO2 API or Apple's passkey APIs.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

79.8% of apps use no attestation at all

FIG 5.2

	Strong	Weak only	None
Android	16.0%	2.8%	81.2%
Strong: Play Integrity · Weak: SafetyNet only (deprecated)			
iOS	3.6%	17.2%	79.2%
Strong: App Attest · Weak: DeviceCheck, no confirmed App Attest			
HarmonyOS	20.9%	4.5%	74.5%
Strong: Play Integrity 13.4%, HMS Safety Detect 7.5% · Weak: SafetyNet only (deprecated)			

Share of each platform's apps with readable code. Strong: the bank's server can verify the app and the device. Weak only: an older or device-only check, with no strong one beside it. App Attest and DeviceCheck are Apple services; Play Integrity and SafetyNet are Google's; HMS Safety Detect is Huawei's.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Attestation lets the bank's server check that a request comes from its genuine app on a genuine device. It is the main defense against repackaged apps like Hirudo, yet 79.8% of apps do not use it. Whether servers enforce the result cannot be seen from the app.

Does the bank's website name the app?

Passkeys and app links only work if the bank's website names the app. We read each bank's public association file and checked whether it lists the app.

Few bank websites name their app for saved logins and passkeys

FIG 5.3



Share of apps on each platform whose bank's website names the app as allowed to share saved logins and passkeys. Read from the bank domains' apple-app-site-association and assetlinks.json files.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

16%

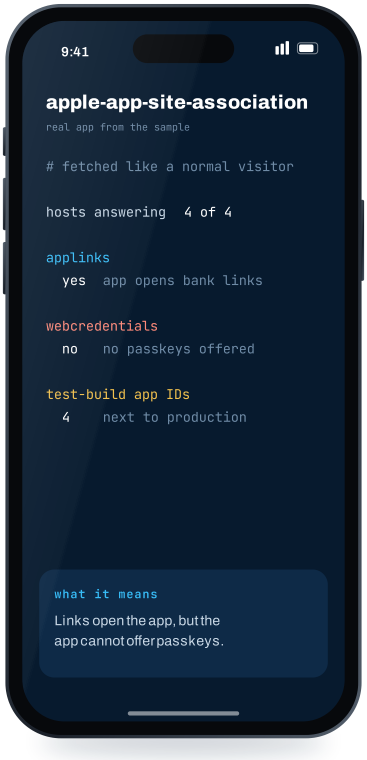
of iOS apps belong to a bank whose file also lists test-build app IDs

92

of 226 bank domains with an Apple association file list non-production app IDs

88

of 217 bank domains with an Android asset-links file list non-production packages



One bank's file, as our check read it

ONE BANK'S ASSOCIATION FILE

The bank's file names its iOS app for universal links, so links to the bank's site open the app. It does not name the app for web credentials, so the app cannot offer the bank's passkeys. It also lists 4 test-build app IDs beside the production app.

Hygiene, not a vulnerability

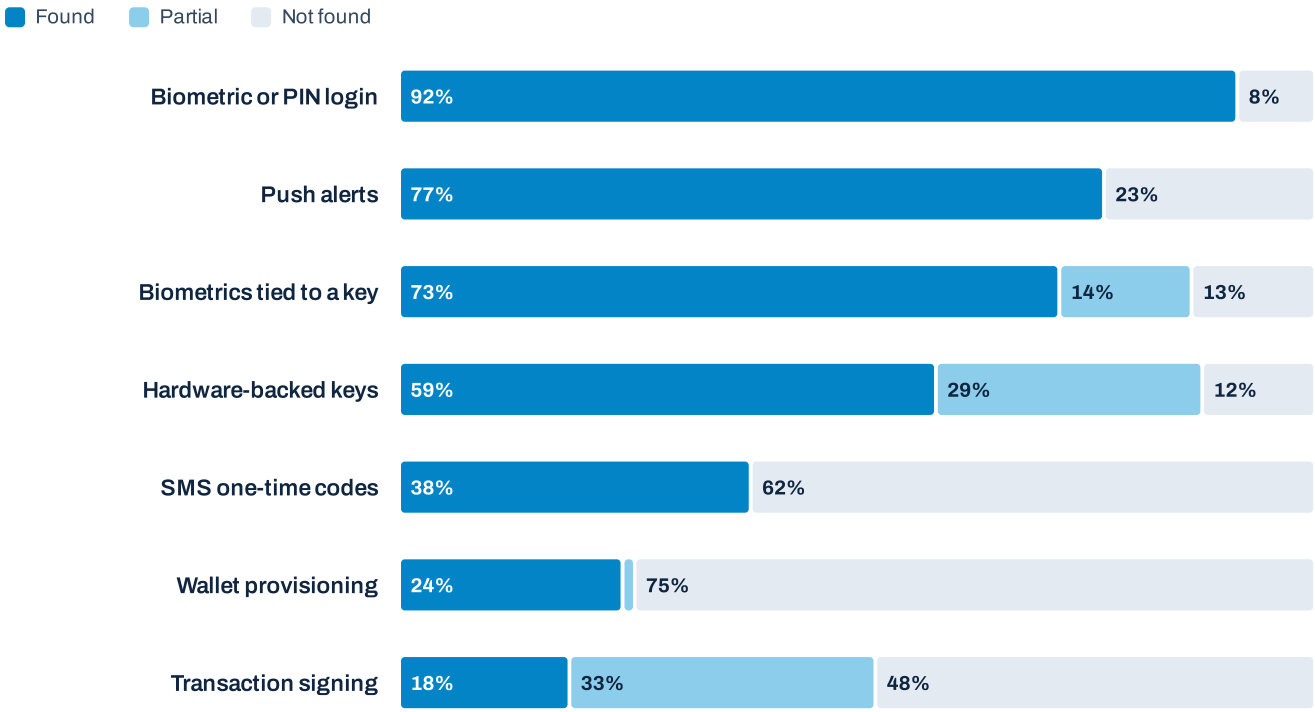
Only apps signed by the bank can use these entries. But a production file that names test builds tells anyone the bank's internal build identifiers, and keeps links and credentials working for builds customers should never run. The bigger gap is the missing entry: without it, an app cannot offer passkeys at all.

Biometrics are often tied to a key, yet a third can be bypassed.

Where we could tell how an Android app uses biometrics, 73.2% tie the check to a cryptographic key, and 11.4% treat it as a yes or no answer that malware can fake. Our scanner also found a biometric flow that can be bypassed in 35.1% of Android apps. Meanwhile, a third of Android apps and most iOS apps support SMS one-time codes, the target of SIM-swap attacks.

How apps sign users in and approve payments

FIG 5.4



Share of 2,686 apps across platforms. Found: clear code evidence. Partial: some but not all of the evidence. Attestation and fraud SDKs are covered on pages 24 and 15.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Bypassable biometrics

A bypassable flow can sit in an app that uses key-bound biometrics elsewhere. Every sensitive action should go through the key-bound path, so that a faked “yes” unlocks nothing.

SMS codes and SIM swap

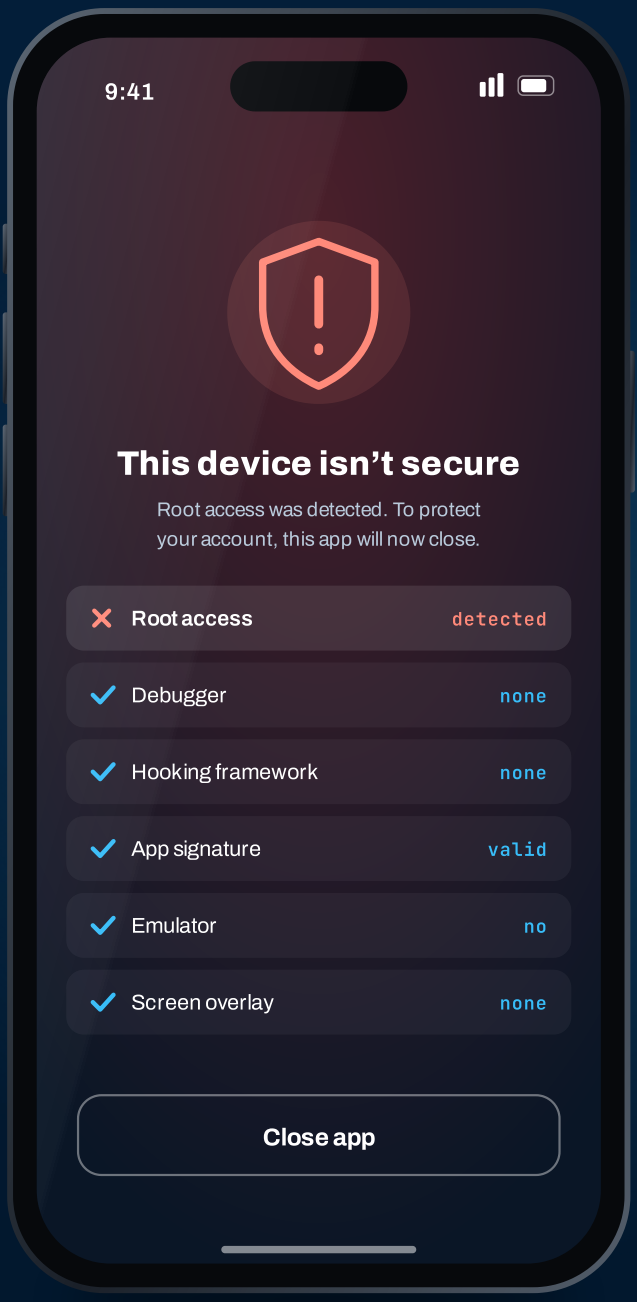
In a SIM-swap attack, a fraudster persuades the mobile operator to move the victim’s number to a SIM they control. Every SMS one-time code then reaches the attacker, not the customer. Codes bound to the device and the transaction do not travel by SMS.

CHAPTER 06

06

Shielding: what apps ship

Runtime shielding detects rooted phones, debuggers, hooking tools and tampered copies of the app. We looked for these protections in the code of every app, on all three platforms.



On screen: an illustration of a shielded app on a rooted phone.

26.9%

of Android apps ship a commercial or platform shielding product

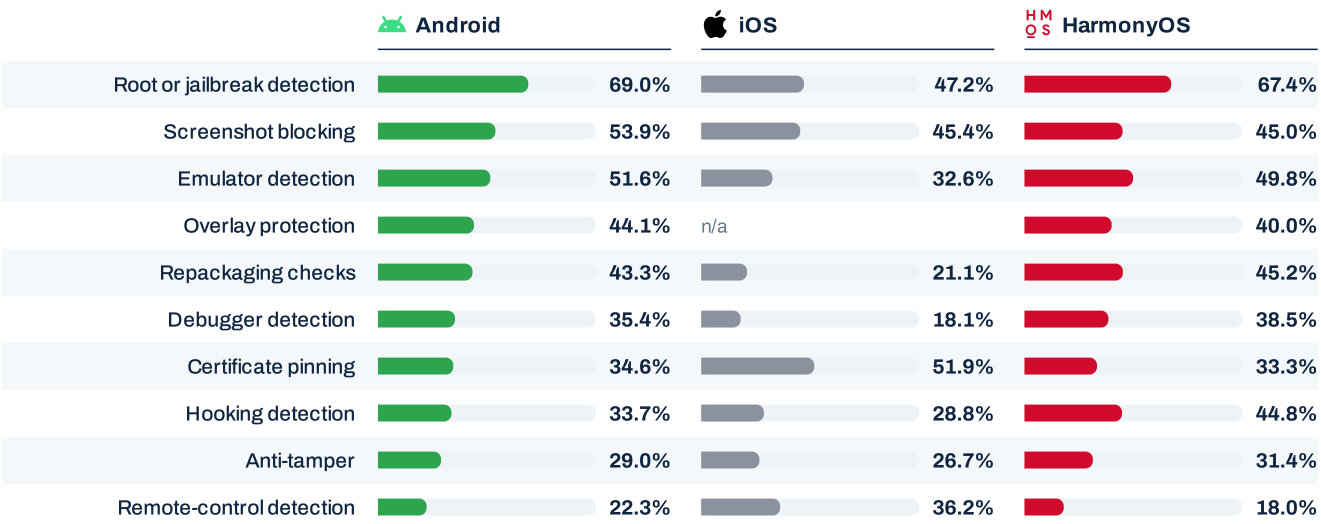
Android apps

Protections found in the code.

Most apps on every platform check for at least a rooted or jailbroken phone. Far fewer detect hooking tools, tampering or remote control: 22.3% of Android apps and 36.2% of iOS apps show code that detects screen sharing or remote control, which trojans use to commit fraud from the victim's own phone.

Root checks are common; hooking and remote-control checks are not

FIG 6.1



Share of apps with the check called from the bank's own or unattributed code, on each platform. Sorted by the Android share. n/a: overlay protection does not apply on iOS, where no app can draw over another.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Present is not the same as effective

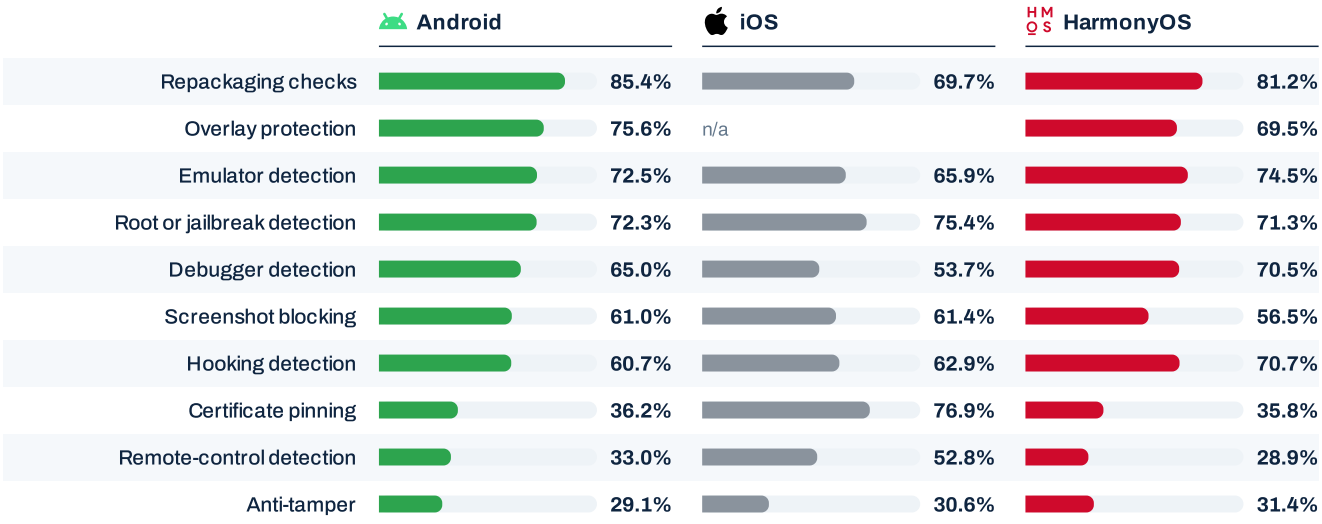
These results show that a protection exists in the app. They do not show that it resists an attacker who knows it is there. Testing that needs real devices and is planned for a later edition. Treat these figures as an upper bound on protection.

SDKs carry many of the checks.

Counting checks that sit inside a bundled SDK as well as the bank's own code, 85.4% of Android apps carry repackaging checks, but only 33% carry remote-control detection, the check that matters most against remote-access trojans.

Repackaging checks are widespread; anti-tamper checks are rare on every platform

FIG 6.2



Share of each platform's apps. Found: the check is in the app, in the bank's own code or inside a bundled SDK. n/a: overlay attacks do not apply on iOS. A check inside an SDK protects the app only if the bank turns it on.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Few apps layer their defenses.

Shielding works best in layers, so that bypassing one check does not disable the rest. The median Android app shows clear evidence of 4 of the 10 protections we look for; 10.7% show 8 or more.

The 10 protections we look for

01 **Root detection**
Notices a rooted phone, where app data is exposed.

02 **Emulator detection**
Notices the app runs in an emulator, not on a real phone.

03 **Debugger detection**
Notices a debugger attached to the running app.

04 **Hooking detection**
Notices tools such as Frida that change the app while it runs.

05 **Repackaging checks**
Notices a modified copy of the app, signed by someone else.

06 **Anti-tamper**
Checks that the app's own code has not been changed.

07 **Certificate pinning**
Accepts only the bank's own server certificate.

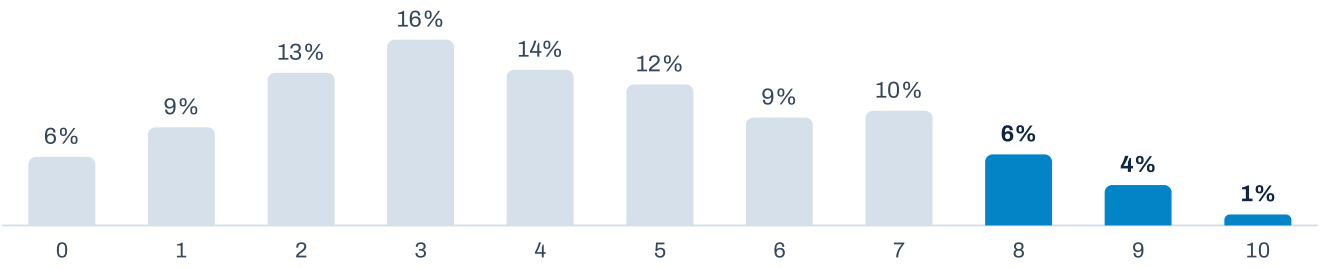
08 **Overlay protection**
Stops another app drawing a fake screen over the bank app.

09 **Screenshot blocking**
Stops screenshots and screen recording of sensitive screens.

10 **Remote-control detection**
Notices screen sharing or a remote-access tool.

The median Android app shows 4 of 10 protections

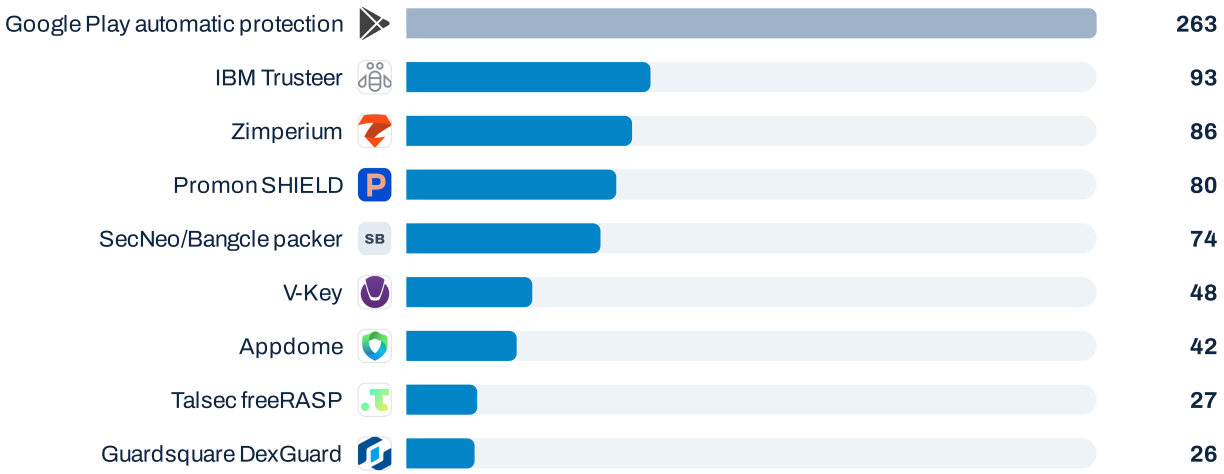
FIG 6.3



Share of Android apps by the number of protections with clear evidence, out of 10. Highlighted: 8 or more.

Google's automatic Play protection is the most common shield

FIG 6.4



Number of apps using each product, all platforms. Google's automatic Play protection (pairip) is applied by the store, not chosen in the code.

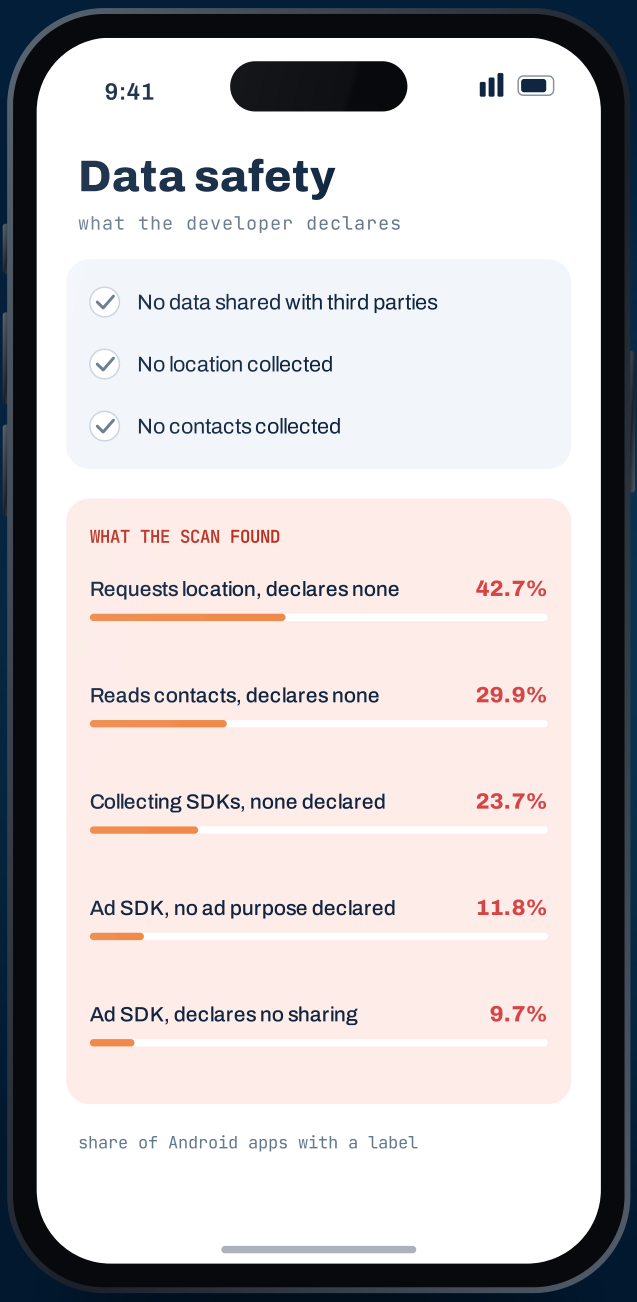
SOURCE Ostorlab automated security analysis of 2,686 banking apps.

CHAPTER 07

07

Beyond the app

An app is only as trustworthy as the domains, servers and institution behind it. We checked bank web and mail domains, the servers the apps talk to, privacy labels, public breach records and the rules each bank answers to.



On screen: a typical label, and the mismatches our scan found most often.

67%

of Android apps have a Google Play privacy label that contradicts their own permissions or SDKs

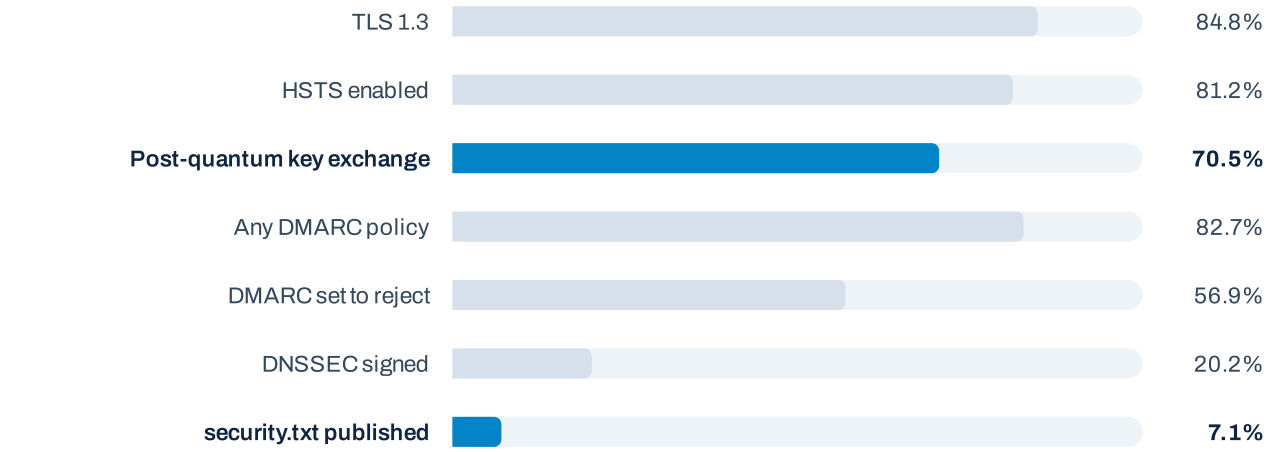
Android apps with a Google Play privacy label

Bank domains: quantum-ready, hard to report to.

70.5% of bank web hosts we reached already negotiate a post-quantum key exchange. Basic hygiene lags: only 7.1% of domains publish a security.txt file telling researchers where to report a flaw.

Post-quantum TLS is ahead of security.txt

FIG 7.1

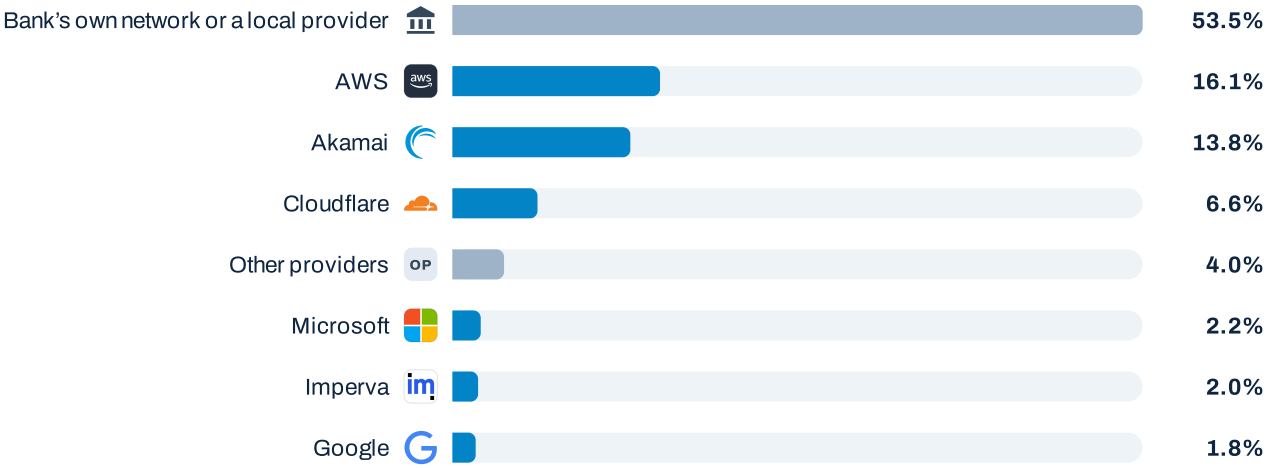


TLS and HSTS: share of 1,140 reachable web hosts. Post-quantum: 1,038 completed handshakes. DNS checks: 1,252 domains.

SOURCE Ostorlab TLS handshakes and public DNS records for the banks' domains.

Most backends run on the bank's own network; AWS leads the clouds

FIG 7.2



Share of 5,101 bank-owned backend IP addresses referenced by the apps on all three platforms, resolved through public DNS.

SOURCE Hosts referenced in the apps, resolved through public DNS; IP ownership registries.

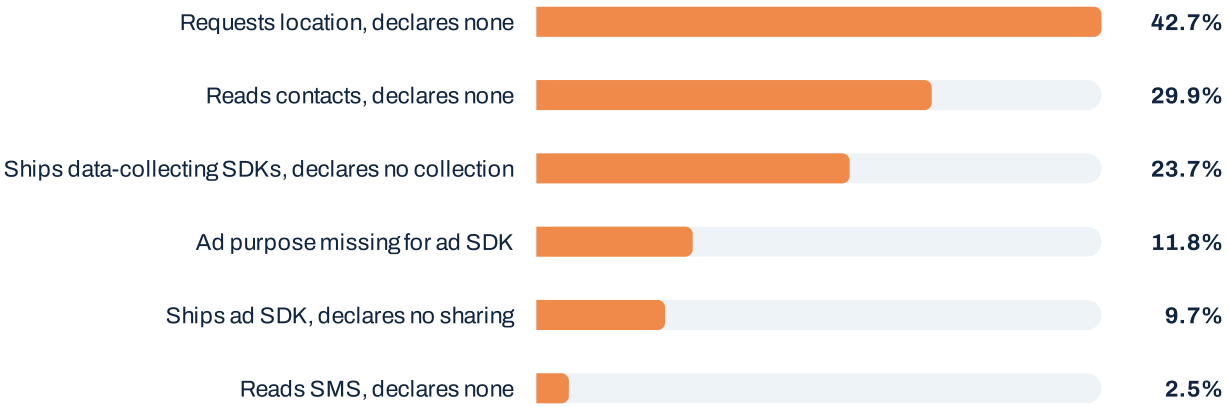
AWS, Akamai, Cloudflare and Microsoft carry 38.7% of the backend addresses. An outage or breach at one of them reaches many banks at once. The EU's DORA asks banks to weigh exactly this before signing, including whether they would be "contracting an ICT third-party service provider that is not easily substitutable" (Regulation (EU) 2022/2554, Article 29(1)).

Privacy labels that do not match the app.

App stores ask developers to declare what data their app collects. For 67% of Android banking apps, the declaration contradicts what the app's own permissions or SDKs show.

Undeclared location is the most common label mismatch

FIG 7.3



Share of Android apps with a Google Play Data safety label. An app can have several mismatches.

SOURCE Google Play Data safety labels, compared with the apps' permissions and SDKs.

Why it matters

A permission in the manifest does not prove the app uses it, so some mismatches are over-cautious manifests. But regulators read labels as promises. A label that understates collection is a compliance risk, and it misleads customers comparing apps.

iOS labels

Of iOS banking apps with a privacy label, 19% declare data used to track users across other companies' apps, most often identifiers and usage data.

Some bank servers answer on more than the web.

Of the 5,101 bank-owned server addresses the apps point to, 4,113 appear in public internet-wide observation data. 251 of those answer on ports beyond web and CDN traffic, at 113 banks, and 79 run software whose version matches a known vulnerability.

251

bank IPs with open ports beyond web and CDN ports, at 113 banks

79

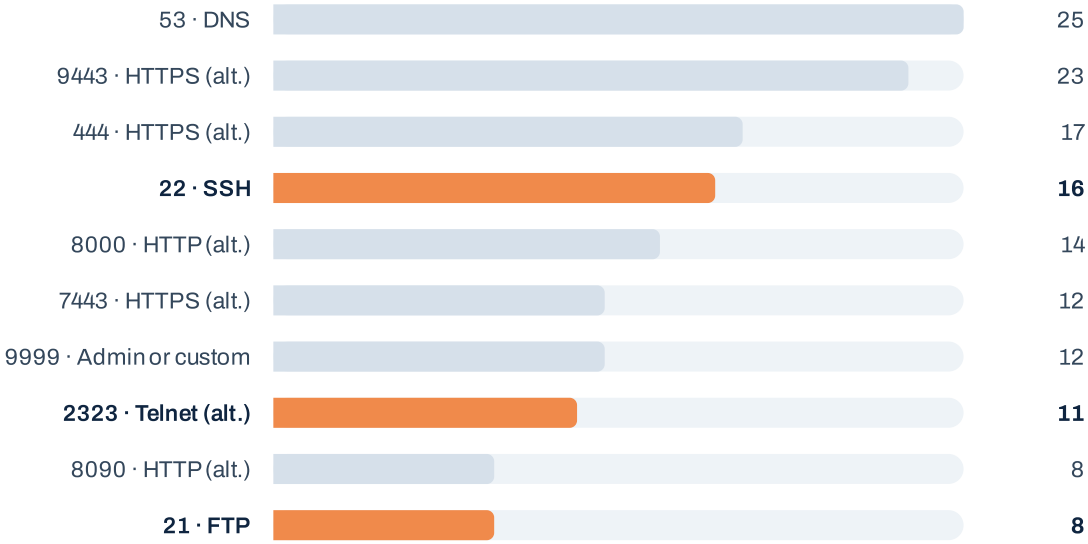
bank IPs with CVEs inferred from software versions, at 74 banks

25

bank IPs running a product past its end of life

DNS and alternative HTTPS ports lead the extra open ports

FIG 7.4



Number of bank-owned IPs answering on each port, of 4,113 indexed by Shodan InternetDB. Ports 80 and 443 and CDN proxy ports are excluded; 151 IPs that answer on more than 40 ports, typical of firewalls and tarpits, are not counted.

SOURCE Shodan InternetDB, passive index; no request was sent to any bank server.

We did not scan any bank server

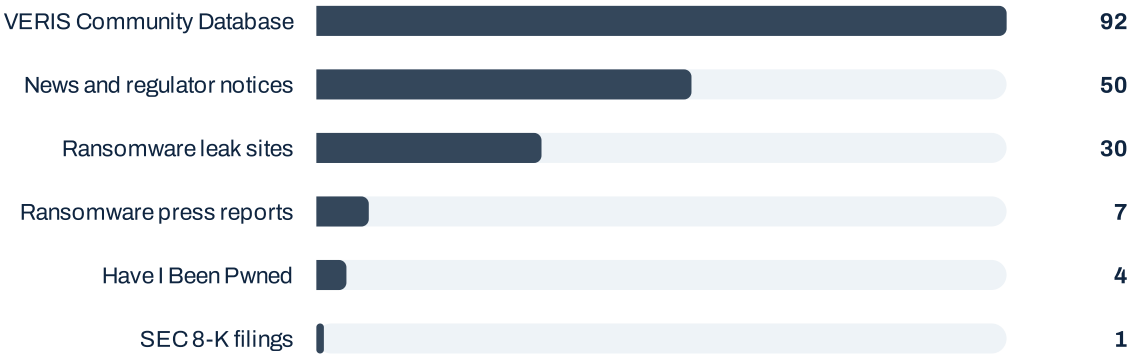
Open ports and software versions come from public internet-wide observation data (Shodan InternetDB), so no bank system was touched. The other 988 addresses are not in that data: unknown, not closed. CVEs are matched from the software version seen, so each is a lead for the bank to confirm.

One bank in six has a public breach record.

Across breach databases, ransomware leak sites, regulator notices and SEC filings, we found at least one public breach record for 83 of the top 500 banks (16.6%). 55 of them have a record from 2020 or later.

Most breach records come from the VERIS database

FIG 7.5



Number of records for the top 500 banks; a bank can have several. Every match was reviewed by hand. News and regulator checks covered the 29 G-SIBs only.

SOURCE VCDB, Have I Been Pwned, ransomware.live, SEC 8-K filings, news and regulator notices.

What a record is

A record is a publicly reported incident: an attack, an insider misusing access, an error that exposed data, or a ransomware group's claim. It is not a finding about the bank's app, and many records are old: the VERIS database reaches back more than a decade.

CVEs that name a bank

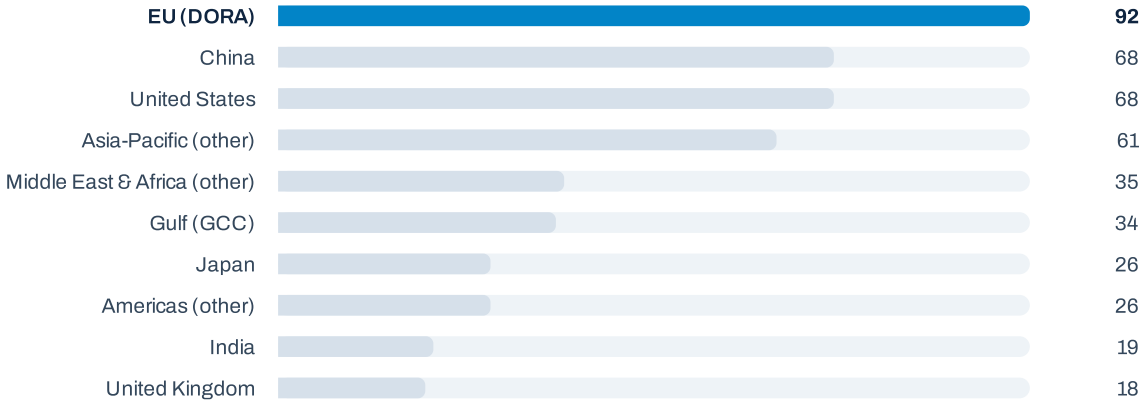
51 published CVE records match the brand of 37 banks, 14 of them since 2025. Most describe old mobile app versions, and brand matching includes false positives, so we treat them as leads rather than findings.

The rules banks answer to, and what they claim.

Every bank in the sample sits under a national or regional regime for ICT risk. 95 of the top 500 banks fall under the EU's Digital Operational Resilience Act, which applies since January 2025 and makes dependencies such as the hosting providers on page 32 a management duty.

The EU's DORA covers the largest group of banks

FIG 7.6



Number of the top 500 banks, by the regime of their home country. Ten largest groups shown; 40 more banks sit in smaller groups.

SOURCE Regulator registers and the banks' home markets.

332

of the top 500 banks are Apple Pay issuers

44

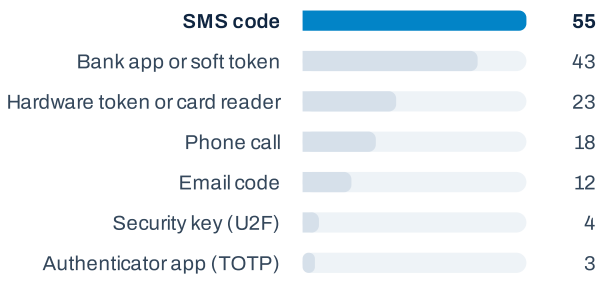
banks state on their website that they hold a certification, most often ISO 27001

19

banks run a public vulnerability disclosure program, 5 with a bug bounty

SMS is the most advertised second factor

FIG 7.7



Number of the 73 banks whose websites describe their sign-in methods; a bank can list several. 8 advertise passkeys.

SOURCE Bank websites and the 2fa.directory list.

Claims, not certificates

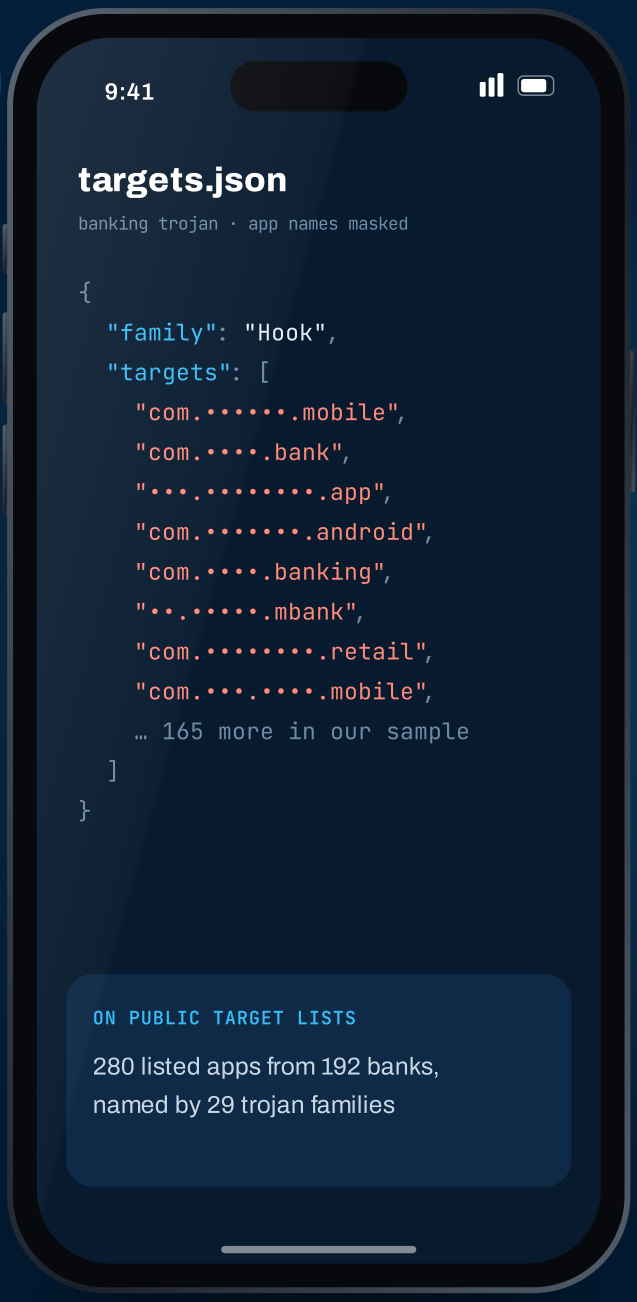
Certifications are what a bank's website says, not verified certificates. 62 banks publish a security.txt file on at least one domain, and 19 name a disclosure program. SMS remains the most advertised second factor, even as regulators such as the UAE's central bank phase out SMS one-time codes.

CHAPTER 08

08

The threat landscape

Banking trojans keep lists of the apps they attack. Phishing kits copy banks' login pages. In the past year, attackers have also replaced genuine banking apps with patched copies.



On screen: the shape of a published target list, app names masked.

15%

of the analysed Android apps appear on at least one public banking-trojan target list

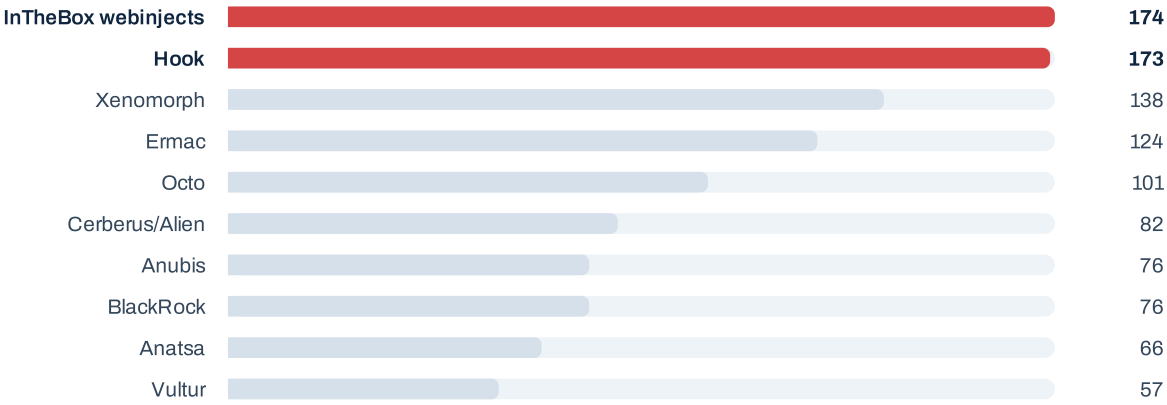
Android apps

Four in ten banks are on a trojan's list.

Researchers publish the target lists found inside banking trojans. Matching them to the Android apps in our sample, 280 apps from 192 banks (38.4%) are named by at least one of 29 families.

Hook and InTheBox target the most apps in the sample

FIG 8.1



Number of apps in our sample on each family's published target lists: 47 lists, mostly published 2021 to 2023 by ThreatFabric, Cleafy, Zimperium and others. Android only, because the lists name Android packages.

SOURCE Published banking-trojan target lists (ThreatFabric, Cleafy, Zimperium and others), 2021–2023.

Europe and the Americas draw 93% of phishing hosts and all nine of the most-targeted banks

FIG 8.2

54%
Europe
5 of the 9 most-targeted banks

39%
Americas
4 of the 9 most-targeted banks

5%
Asia-Pacific
none of the 9 most-targeted banks

1%
Middle East & Africa
none of the 9 most-targeted banks



Colour: each region's share of the live phishing hosts that name a bank. Pins: the number of the nine most-targeted banks based in that country; together those nine draw 57% of all hosts. Grey: not in the sample. Banks are not named.

SOURCE PhishTank, OpenPhish, Phishing.Database and URLhaus.

2026: attackers turn the bank's own app against it.

For ten years, banking trojans drew fake screens on top of the real app. The campaigns of the past year go further: they patch, clone or replace the genuine app, so the bank's own code ends up working for the attacker.

SINCE 2016

Overlay

Draw on top

A trojan draws a fake login or payment screen over the real app and reads what the customer types.

WHY IT WORKS

The genuine app is untouched; the trojan only needs the accessibility permission.

WHAT STOPS IT

Overlay protection, accessibility-service checks

GOLDFACTORY, 2025

Patch

Change the app

The attacker takes the genuine app, injects hooking code and re-signs it, so the bank's own screens run the attacker's code.

WHY IT WORKS

The copy looks and behaves like the real app, including its integrity checks, which the hook switches off.

WHAT STOPS IT

Signature check against the bank's key, server-side attestation

GIGABUD, 2026

Clone

Run a second copy

The attacker runs the genuine app inside a hidden work profile or virtual space that it controls.

WHY IT WORKS

Fraud tools see the real app on a known device, so the session looks normal.

WHAT STOPS IT

Work-profile and virtual-space detection, device binding

HIRUDO, 2026

Replace

Swap the app

A dropper uninstalls the genuine app and installs a patched copy with the same name, icon and version number.

WHY IT WORKS

The customer sees no change. Only the signing key differs, and only the bank's server can check it.

WHAT STOPS IT

Server-side attestation that checks the signing key

How an app swap works, step by step

1. A fake update

The victim installs a dropper from a link or a fake store page.

2. The real app goes

The dropper uninstalls the genuine banking app.

3. A copy comes back

It installs a patched copy: same name, icon and version, different signing key.

4. Credentials leak

The copy captures passwords and one-time codes while it looks normal.

5. The server decides

Only attestation on the bank's server can tell the copy from the app.

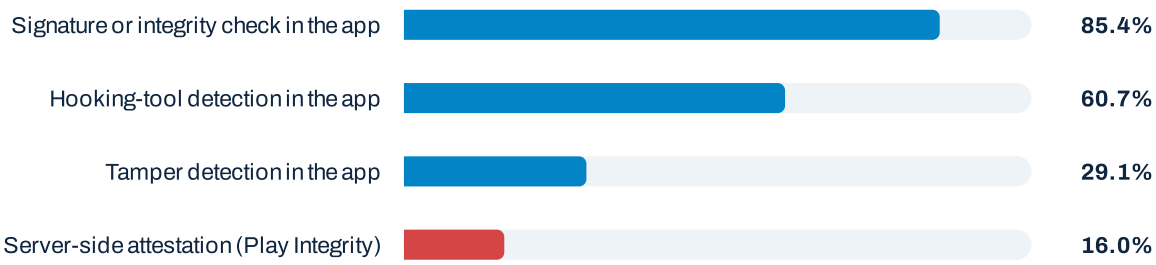
SOURCE Group-IB (GoldFactory, Gigabud), ThreatFabric (Hirudo); Ostorlab analysis.

Would a bank notice its app had been swapped?

A patched copy can switch off any check that runs on the phone. The server can do better: it can refuse requests that do not carry a fresh proof, signed by Google or Apple, that the app is the bank's genuine build. Only 16% of Android apps contain the code to ask for that proof, and we cannot see whether their servers enforce it.

Checks inside the app are common; checks on the server are rare

FIG 8.3



Share of Android apps. In the app: found in the bank's own code or inside a bundled SDK. Server-side attestation: the app contains the code to request the proof; whether the server enforces it cannot be seen from the app, and a genuine but compromised device can still pass.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

The 2026 campaigns

When	Campaign	Type	What happened	Reported by
Dec 2025	**GoldFactory**	Patch	Genuine banking apps in Thailand, Vietnam and Indonesia injected with hooking code that bypasses their integrity checks: 27 modified apps, over 11,000 infections reported.	Group-IB
Feb–Jul 2026	**Gigabud**	Clone	Banking apps run inside an Android work profile to evade fraud detection; about 1,469 compromised devices reported.	Group-IB
May 2026	**Lab test**	Patch	58% of the banking apps tested across eight markets could be repackaged.	Promon
2026	**Overlay families**	Overlay	34 active banking-malware families targeting 1,243 financial brands in 90 countries.	Zimperium
Aug–Oct 2026	**Hirudo**	Replace	A dropper swaps a real Indonesian banking app for a patched copy signed with a public test key; eight apps from seven banks had ready-made replacements. Case study, page 49.	ThreatFabric

What to do

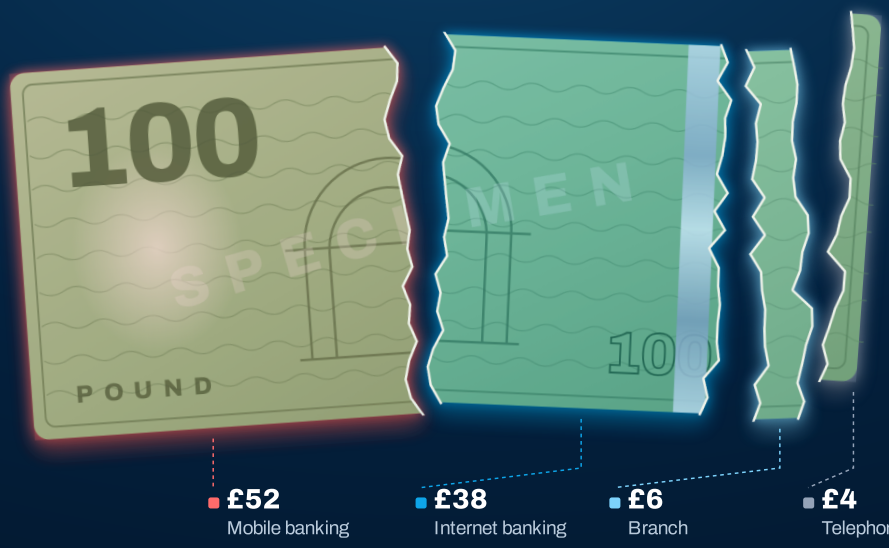
Treat any check that runs on the phone as a speed bump. Make the server decide: require attestation for login and payments, reject requests from apps signed with any key but the bank's own, and bind the customer's session to the device it started on.

Where the money goes.

Published fraud data is patchy: of 101 markets we checked, 83 publish any comparable figure. Where data exists, it points the same way: fraud has moved to payments the customer is tricked into making, and to the phone.

FIG 8.4

Of every £100 lost to scams in the UK, £52 left through a mobile banking app



£576.4m
lost to authorised push-payment scams in the UK in 2025

79%
of scam payments, by number, were made in mobile banking

88%
of reported scam losses were paid back under the UK refund rule

A £100 note torn by the channel the scam payment was made in; each piece is as wide as its channel's share of the value lost. In authorised push-payment (APP) scams the customer is tricked into sending the money. Authorised losses rose 19% in the year, while unauthorised losses fell 5%.

Source: UK Finance, Annual Fraud Report 2026; UK Payment Systems Regulator

Europe

In 2024, 74% of fraudulent credit-transfer value in the EU came from manipulating the payer into authorising the payment. Card fraud is 13 times more common, by value, when the card is not present. Source: ECB and EBA payment fraud report, 2024 data.

Reimbursement works

Since UK reimbursement became mandatory in October 2024, banks returned 88% of reported APP scam losses by value over the following 18 months. Liability is moving to banks, which raises the value of controls that stop scams in the app. Source: UK Payment Systems Regulator.

Context, not cause

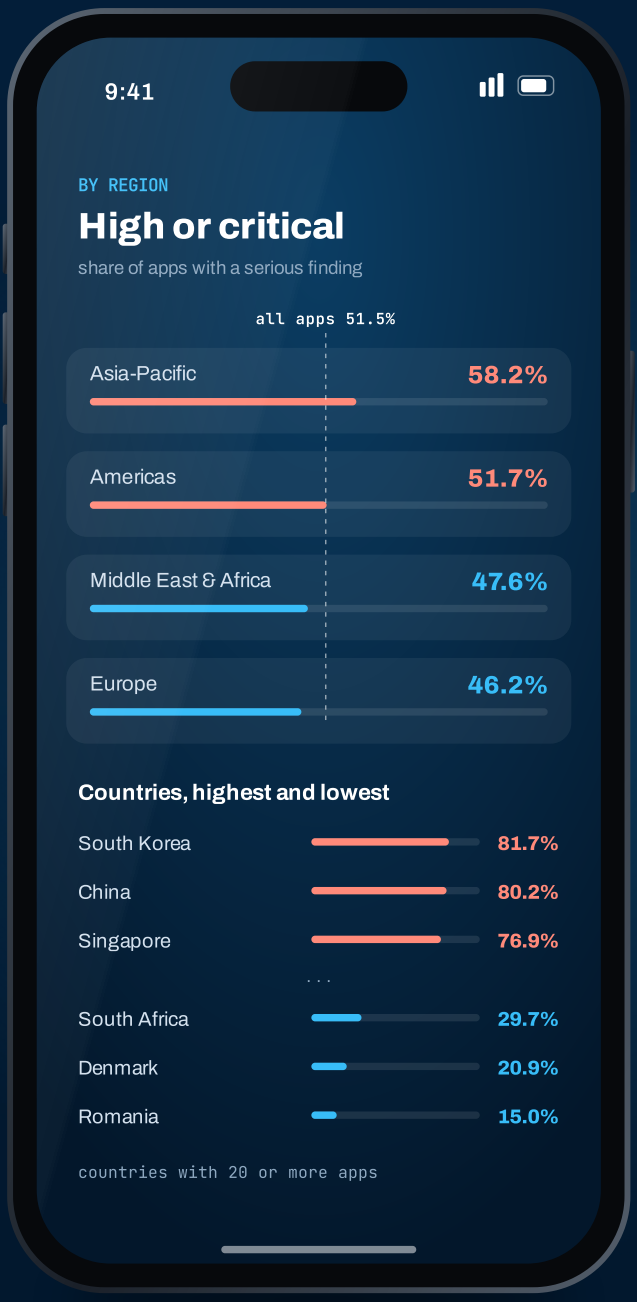
These statistics describe fraud in each market. They do not show that the app weaknesses in this report caused the losses.

CHAPTER 09

09

By region, bank type and country

Results vary more by country than by platform. Some markets lead on shielding, others on attestation, and no market leads on everything.



On screen: apps with a high or critical finding, by region and country.

64

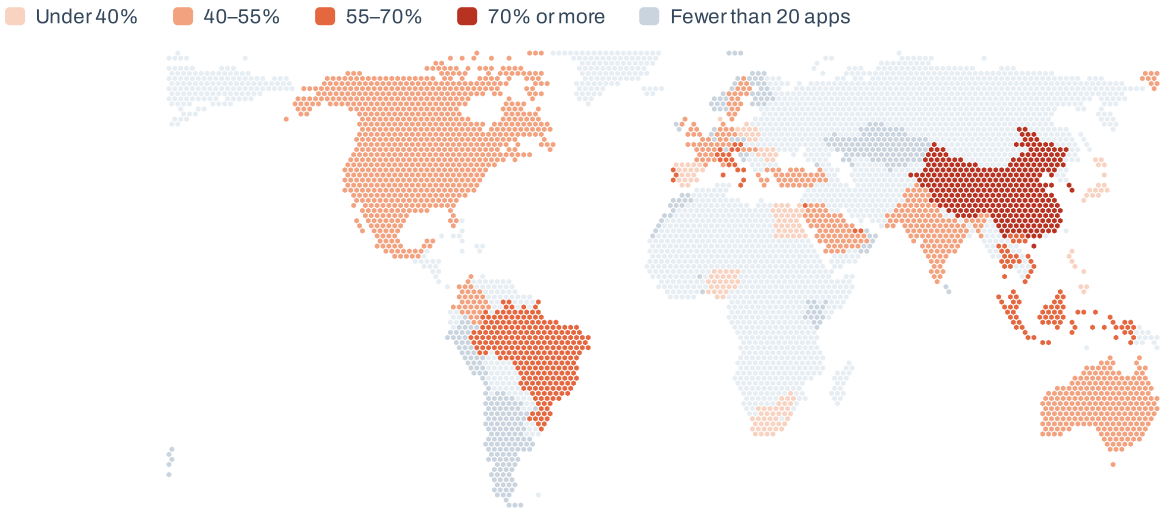
countries in the sample, from the United States and China to Togo and Liechtenstein

No region has it all.

Asia-Pacific apps have the most serious findings and the most shielding at once. European apps have the fewest high or critical findings. Americas apps lead on passkeys, at 9.9%.

Serious findings vary widely from country to country

FIG 9.1



Share of each country's apps, for the 40 countries with at least 20 apps; grey: in the sample with fewer than 20 apps.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

No region or bank type leads on every measure

FIG 9.2

Group	Apps	High or critical	Vulnerable component	Embedded credentials	Shielding product	Attestation	Passkey API
Asia-Pacific	942	58.2%	46.3%	24.5%	33.7%	20.0%	2.8%
Europe	822	46.2%	40.0%	15.7%	21.3%	16.9%	4.7%
Middle East & Africa	508	47.6%	39.0%	23.0%	19.9%	19.9%	4.3%
Americas	414	51.7%	33.8%	16.2%	29.5%	23.4%	9.9%
By bank type							
Traditional	2,596	51.2%	40.9%	19.8%	26.4%	19.3%	4.4%
Bank-owned neobank	61	52.5%	47.5%	24.6%	37.7%	16.4%	4.9%
Independent neobank	29	**79.3%**	44.8%	**48.3%**	24.1%	**48.3%**	**37.9%**
All apps	*2,686*	*51.5%*	*41.1%*	*20.2%*	*26.6%*	*19.6%*	*4.8%*

Share of apps in each group. Independent neobanks have a high or critical finding far more often than other apps (79.3% against 51.2%; Fisher exact test, p = 0.002), while bank-owned neobanks look like the rest. Independent neobanks also lead on attestation and passkey APIs. They are only 29 apps, a third from one bank, so the size of the gap is indicative. Cells shaded as on the next page. Embedded credentials exclude Google API keys. Only cross-platform rules are compared, because the platform mix differs by region.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

Country results.

The 18 countries with the most apps in the sample. Each cell is shaded by how far it sits above the all-apps share: red for a risk, blue for a defense. Unshaded cells are at or below it.

Each country has its own pattern of risks and defenses

FIG 9.3

Above the all-apps share: Risk +5 pts +12 +20 Defense +5 pts +12 +20

Country	Apps	High or critical	Vulnerable component	Embedded credentials	Shielding product	Attestation	Passkey API
United States	224	52.2%	35.3%	12.9%	33.9%	28.1%	12.9%
India	200	50.0%	30.5%	21.0%	**43.0%**	**32.5%**	0.5%
China	162	**80.2%**	**77.8%**	**43.8%**	**51.9%**	15.4%	5.6%
United Kingdom	157	47.1%	**57.3%**	31.2%	**43.9%**	18.5%	14.0%
Japan	116	38.8%	33.6%	13.8%	6.9%	4.3%	3.4%
Saudi Arabia	95	47.4%	32.6%	31.6%	17.9%	13.7%	6.3%
United Arab Emirates	84	**64.3%**	**59.5%**	25.0%	23.8%	28.6%	0.0%
France	76	43.4%	35.5%	3.9%	21.1%	9.2%	0.0%
Italy	74	58.1%	**58.1%**	21.6%	5.4%	29.7%	2.7%
South Africa	74	29.7%	17.6%	**37.8%**	20.3%	**54.1%**	9.5%
Malaysia	71	57.7%	42.3%	25.4%	31.0%	25.4%	1.4%
Spain	69	37.7%	24.6%	8.7%	**42.0%**	27.5%	7.2%
Brazil	66	60.6%	31.8%	22.7%	27.3%	21.2%	10.6%
Australia	63	54.0%	33.3%	19.0%	4.8%	23.8%	3.2%
South Korea	60	**81.7%**	**71.7%**	21.7%	3.3%	3.3%	0.0%
Vietnam	56	**64.3%**	51.8%	**33.9%**	**41.1%**	25.0%	3.6%
Qatar	55	41.8%	40.0%	21.8%	10.9%	25.5%	0.0%
Canada	50	48.0%	26.0%	14.0%	12.0%	14.0%	4.0%
All apps	*2,686*	*51.5%*	*41.1%*	*20.2%*	*26.6%*	*19.6%*	*4.8%*

Share of each country's apps. Shading: 5, 12 and 20 percentage points or more above the all-apps share. China's apps include many HarmonyOS apps, which lift its component and critical figures.

SOURCE Ostorlab automated security analysis of 2,686 banking apps.

The safest bank apps by region.

Three banks per region whose apps have fewer and less severe weaknesses than their peers', in alphabetical order. No scores are shown.

North America

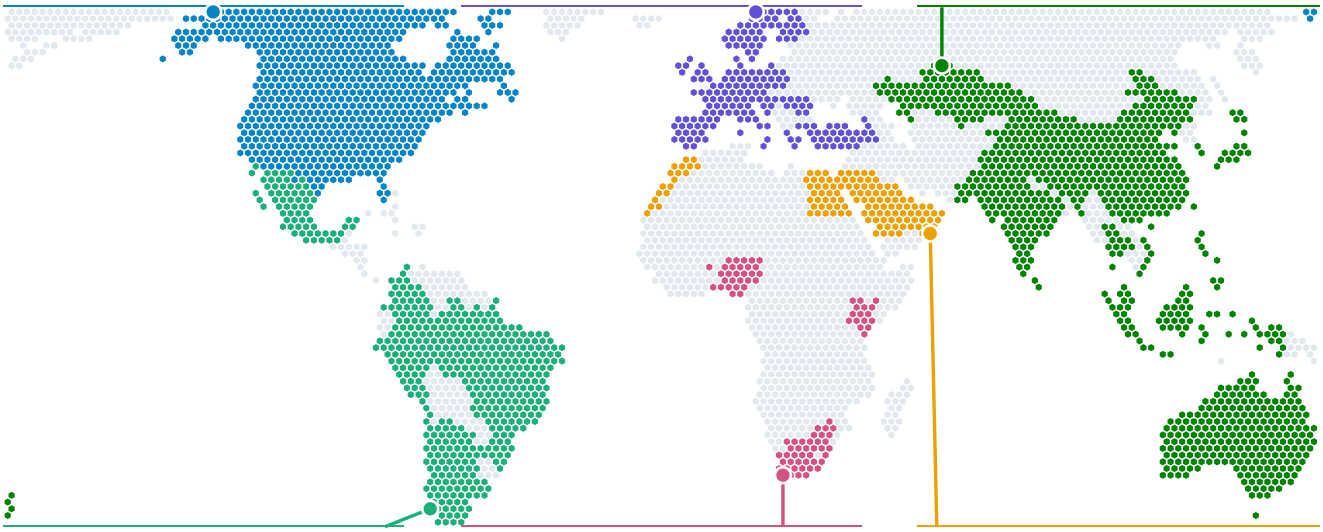
- **Scotiabank** — Canada
- **WaFD Bank** — United States
- **Zions Bancorporation** — United States

Europe

- **Ibercaja** — Spain
- **Komerční banka** — Czechia
- **TSB** — United Kingdom

Asia-Pacific

- **Bandhan Bank** — India
- **HBL** — Pakistan
- **RCBC** — Philippines



Latin America

- **Banorte** — Mexico
- **BCP** — Peru
- **Itaú** — Brazil

Sub-Saharan Africa

- **Absa** — South Africa
- **Co-Operative Bank of Kenya** — Kenya
- **Rand Merchant Bank** — South Africa

Middle East & N. Africa

- **Gulf Bank** — Kuwait
- **Qatar Islamic Bank** — Qatar
- **Saudi Investment Bank** — Saudi Arabia

Colour: countries whose banks were compared, one colour per region Grey: not compared

How banks were compared

Banks are compared, not single apps. Each scanned app gets a value from its scan rating (70%) and from the defenses and weaknesses found in its code (30%), and is placed among the apps of its own store. A bank's app score is the average over all its scanned apps, pulled toward the middle when it has few.

Security history counts

The app score is then lowered for each security incident at the bank that was reported publicly in the past three years: more for a severe incident, less for a minor one. Scams aimed at customers do not count. An empty history means nothing was found, not that nothing happened.

What a place means

The three best-placed banks of each region, among banks with at least two scanned apps. A place means fewer and less severe findings than the bank's peers. It does not mean the apps are free of findings.

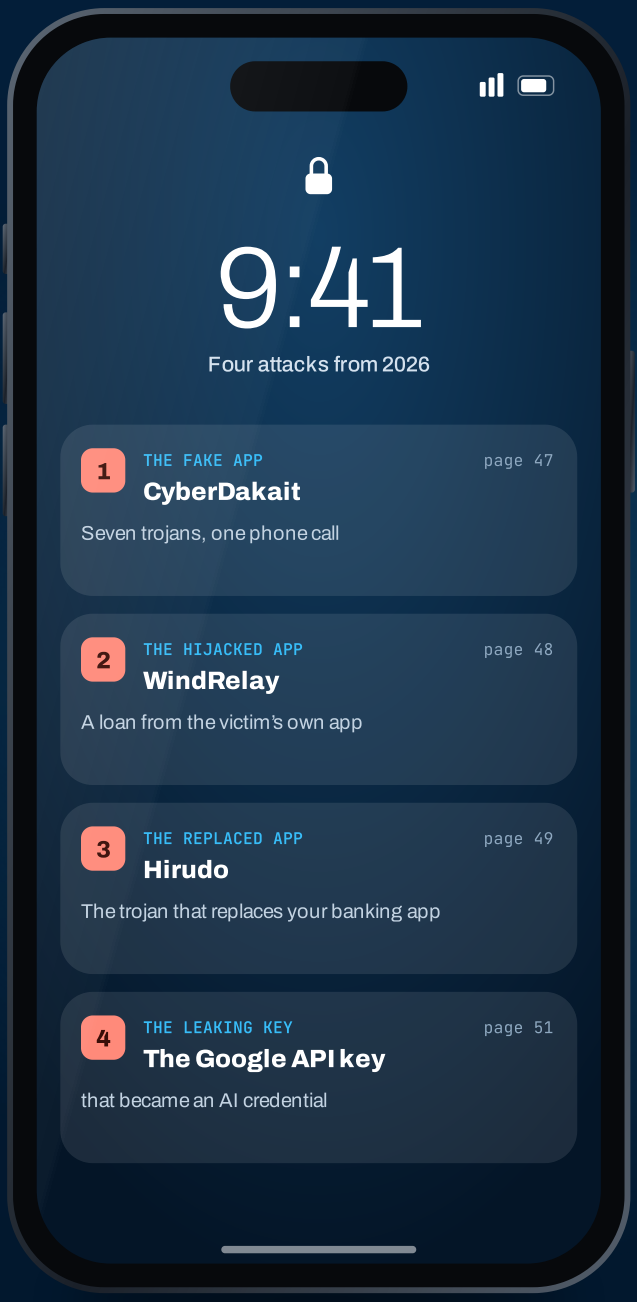
SOURCE Ostorlab scan ratings and security controls for the apps of the top 500 banks; public incident reports. Logos are the banks' website icons and trademarks of their owners, shown for identification only.

CHAPTER 10

10

Case studies

Four public attacks from 2026, each set against what our scan found in the apps they target. They run from a fake app installed during a phone call, to the genuine app being replaced, to a cloud key that gains new powers after the app ships.



On screen: the four cases and the page each one starts on.

4

public attacks from 2026, each set against what our scan found in the apps they target

CASE STUDY 1 OF 4 · THE FAKE APP

CyberDakait: seven trojans, one phone call.

Seven Android trojan families aimed at customers of Indian banks. The malware is ordinary. What makes it work is the call that comes first: a scammer posing as the bank talks the victim into installing it.

7
trojan families in the cluster

17,960+
infected devices, counted from the operators' own servers

7.6m
SMS messages intercepted from victims' phones

Reported by Group-IB, 28 October 2026

How the attack works

1

The call

A caller posing as the bank offers a "Credit Card Protection Plan" or a reward.

Tell customers the bank never sends an app during a call

2

The link

The victim gets an app file posing as one of 11 Indian banks or 2 government programmes.

3

The fake update

A Google Play style "update" screen gets the app installed and its permissions granted.

4

The overlay

A copy of the bank's screens asks for card details, login and PIN.

5

The SMS

The trojan reads incoming SMS and forwards one-time codes to the operators.

Bind codes to the device and the transaction, not plain SMS

6

The theft

With credentials and codes, the operators empty the accounts. Reported losses exceed $30,500.

WHAT OUR SCAN FOUND

Overlay protection in the app **75.6%**

Support SMS one-time codes **33.7%**

Share of Android apps. Overlay protection guards only the genuine app: a fake app sent during a call never runs it. The attack wins when the bank accepts an SMS code as proof of identity.

WHAT WOULD HAVE STOPPED IT

- Tell customers, in the app and on every channel, that the bank never sends an app or asks for one during a call.
- Replace SMS codes with approvals bound to the device and the transaction.
- Hold new devices and new payees for review when they appear during a call.

Group-IB, *False Protection, Emptied Accounts: The Seven Android Trojan Families Targeting Indian Banking Apps*, 28 October 2026.
No bank is named: Group-IB does not name the impersonated banks.

CASE STUDY 2 OF 4 · THE HIJACKED APP

WindRelay: a loan taken out from the victim's own banking app.

Two tools used together in Central Europe: SpyNote, a remote access trojan that hands the operator the victim's phone, and WindRelay, which relays the victim's payment card over NFC to a phone held by a mule.

2
tools combined: a remote access trojan and an NFC relay

3
countries targeted: Czechia, Slovakia and Slovenia

23
samples collected from November 2025 to July 2026

Reported by Group-IB, 12 August 2026

How the attack works

1

The call

A vishing operator contacts the victim, posing as the bank or another trusted party.

2

The install

The victim installs an app; each SpyNote build is labelled with the victim's name.

3

The card tap

The relay app has the victim hold their card to the phone and type the PIN, and forwards it to a mule.

4

The cash

The mule pays or withdraws cash with the relayed card.

5

The takeover

Through SpyNote, the operator drives the phone and opens the genuine banking app.

Detect that the screen is shared or remotely controlled

6

The loan

In at least one case, the operators took out a loan in the victim's name from that app.

Step-up checks on the server for loans, payees and limits

WHAT OUR SCAN FOUND

Remote-control detection in the app **33.0%**

Server-side attestation (Play Integrity) **16.0%**

Share of Android apps. The takeover happens inside the real app on the victim's real device, so device binding and SIM checks pass. Only the app can see that its screen is being driven from elsewhere.

WHAT WOULD HAVE STOPPED IT

- Detect screen sharing and remote control, and send the signal to the bank's risk engine.
- Require a fresh, key-bound approval for loans, new payees and higher limits.
- Treat a card tap followed by a PIN request inside an app as a fraud pattern.

Group-IB, *WindRelay and SpyNote: an NFC and RAT combination in fraud*, 12 August 2026.
No bank is named: Group-IB does not name the affected banks.

CASE STUDY 3 OF 4 · THE REPLACED APP

Hirudo: the trojan that replaces your banking app.

An Android banking trojan that does not stop at drawing fake screens. It uninstalls the real app and installs a patched copy whose hidden hooks feed forged answers to the app's own security checks.

42
financial apps in Hirudo's target list

8
banking apps from 7 Indonesian banks with a ready-made patched copy

1
line of code added to each copy to load the hooking engine

Reported by ThreatFabric, 14 October 2026

How the attack works

1 **The lure**
A working shopping, short-drama, football or adult video app.

2 **The fake update**
The lure installs the trojan, which asks for Accessibility and notification access.

3 **The secrets**
Fake checkout, bank and wallet prompts collect PINs and passwords; a fake pattern screen gives up the unlock code.

4 **The target**
The trojan finds one of eight targeted banking apps and checks its signing certificate.

5 **The swap**
A fake update prompt opens the uninstaller, then the trojan installs the patched copy itself.

6 **The takeover**
The copy runs with forged security signals while the operator watches the screen live.

Check the Play Integrity verdict on the server for every protected action

Inside the patched app

The copy keeps the original name and version number but is signed with a public Android test key, which is why the genuine app has to be removed first. One added line loads a hooking engine that switches on only when a file written by the trojan exists.

Remote access and collection

Live screen viewing and camera sessions, a 90-day scan of SMS, and credential targets in China, India, Mexico, Morocco and Thailand. An unsecured operator server held thousands of stolen credentials and financial profiles.

ThreatFabric, *Hirudo: from fake updates to patched banking apps*, 14 October 2026.
BleepingComputer, *Hirudo Android malware replaces banking apps to evade security checks*, 14 October 2026.
No bank is named: ThreatFabric does not name the Indonesian banks. What our scan found: next page.

Inside the copy, every check can be made to lie.

Hirudo runs the bank's own checks inside a process the attacker controls, and answers them itself. 61% to 85.4% of Android banking apps carry the in-app checks it forges.

THE QUESTION THE APP ASKS	WHAT HIRUDO ANSWERS	ANDROID APPS THAT ASK IT
Signing certificate "Is this the bank's signing key?"	HIRUDO'S FORGED ANSWER Returns the genuine certificate and hides the test-key signature.	Signature or repackaging check **85.4%**
Accessibility services "Is another app controlling the screen?"	HIRUDO'S FORGED ANSWER Returns an empty list, hiding the trojan's controlling service.	Overlay or accessibility-abuse protection **75.6%**
Screenshot blocking "Can this screen be recorded?"	HIRUDO'S FORGED ANSWER Strips the flag, so banking screens can be viewed and recorded remotely.	Screenshot protection **61.0%**
Hardware key attestation "Is the key in secure hardware?"	HIRUDO'S FORGED ANSWER Answers with a leaked hardware keybox whose certificates Google has revoked.	Not counted: cannot be told apart from other attestation calls
Play Integrity "Is this the genuine app on a genuine phone?"	HIRUDO'S FORGED ANSWER Sends the request to the attacker's server, which returns a passing verdict that the app accepts.	Play Integrity call **16.0%**

Hooks: ThreatFabric. Shares: Android apps with the check in the bank's own code or inside a bundled SDK; Play Integrity: apps with readable code. A check that exists is not one that resists a hook.

What still holds: answers the bank's server checks on its own side

- Verify Play Integrity on the server, bound to the session and the protected action.
- Check key-attestation certificates against Google's revocation list.
- Keep transaction-risk controls on the server at login, credential recovery, new payees and payment approval.
- Feed accessibility and screen-capture signals into risk scoring instead of using them as standalone blockers.

Sources: ThreatFabric (hooks); Ostorlab analysis (apps).

CASE STUDY 4 OF 4 · THE LEAKING KEY

The Google API key that became an AI credential.

Google long told developers that its API keys, the strings starting with `AIza`, are not secrets: they identify a project for Maps or Firebase and are meant to ship in apps. But what a key can reach is set in the cloud project, often by another team, long after the app ships.

2,863
live Google API keys with Gemini access found in public web data

32
live keys with Gemini access in 22 popular Android apps

$82,314
Gemini bill reported by one victim of an abused key

Reported by Truffle Security, February 2026, and CloudSEK, April 2026

How a harmless key turns into an AI credential

1

The key ships

The app carries a Google API key for Maps or Firebase, as Google's documentation allows.

Restrict every shipped key to the APIs and the app signature that need it

2

Gemini is switched on

Another team turns on the Generative Language API in the same cloud project.

Keep AI APIs in a project whose keys never ship to a device

3

The key gains AI access

Every existing key of the project can now call Gemini. Nothing in the app changes.

4

The abuse

An attacker lifts the key from the app, reads files uploaded to Gemini and runs up the bill: one victim was charged $82,314.

WHAT OUR SCAN FOUND

Android apps	80.3%
iOS apps	30.6%
HarmonyOS apps	71.4%

Share of apps with at least one Google API key anywhere in the package; 72.5% of all apps. Keys were not tested: most are probably limited to Maps or Firebase, and the app team cannot tell from the key.

WHAT WOULD HAVE STOPPED IT

- Restrict each key to the APIs it needs and to the app's signing certificate.
- Keep Gemini and other AI APIs in a separate cloud project.
- Rotate any key that has shipped in an app before turning on a new API.

Truffle Security, *Google API keys weren't secrets, but then Gemini changed the rules*, 25 February 2026.
CloudSEK, *Hardcoded Google API keys in top Android apps now expose Gemini AI*, 7 April 2026.
No bank is named: neither source names the financial institutions whose keys were exposed.

What banks should do in 2026: the top 10.

Ten actions, each tied to a finding in this report and to an OWASP MASVS control group, ordered by how directly they reduce fraud.

01 Verify the app on the server

Use Play Integrity and App Attest, check the verdict on the server and reject requests that fail. It is the one defense a patched or swapped app cannot switch off.

79.8% of apps use no attestation

MASVS-RESILIENCE

02 Tie every sensitive action to a key

Route logins and payment approvals through biometrics bound to a hardware-backed key, never a yes or no check.

35.1% of Android apps have a biometric bypass

MASVS-AUTH MASVS-CRYPTO

03 Offer passkeys

Add passkey sign-in and publish the matching asset links and associated domains, so a phished password is no longer enough.

4.9% integrate a passkey API

MASVS-AUTH

04 Take secrets out of the app

Move private keys, cloud keys and service tokens to the server and issue short-lived tokens. Restrict the keys that must ship to your app's signature and the APIs it needs.

20.3% embed credentials beyond Google API keys

MASVS-STORAGE MASVS-CRYPTO

05 Close the login redirect

Return logins through universal links and app links that only your app can claim, never custom URL schemes another app can register.

21.1% of iOS apps allow an OAuth account takeover

MASVS-PLATFORM MASVS-AUTH

06 Detect remote control and overlays

Combine root, hooking, tamper, overlay and screen-sharing detection, turn on the checks your SDKs already carry, and test them on real devices.

33% of Android apps carry remote-control detection

MASVS-RESILIENCE

07 Know what you ship

Keep a software bill of materials for every release, including SDKs and vendor platforms, and fix critical components within a set deadline.

41.1% ship a vulnerable component

MASVS-CODE

08 Encrypt every connection

Turn off cleartext traffic, pin certificates for your own APIs, and keep WebViews to your own pages.

29.4% of Android apps allow cleartext traffic

MASVS-NETWORK MASVS-PLATFORM

09 Watch old devices

Check the operating system's patch level at login and limit high-risk actions on devices that no longer get updates.

99.5% of Android apps run on unpatched Android

MASVS-CODE

10 Make labels and contacts honest

Align store privacy labels with what the app does, and publish a security.txt so researchers can reach you.

7.1% of bank domains publish security.txt

MASVS-PRIVACY

Definitions and data notes.

Sample

The study covers the mobile apps of the top 500 banks. 2,841 app listings were collected and 2,686 were downloaded and analysed, on Android, iOS and HarmonyOS. Countries and regions follow the bank's home market.

Severity

An app's severity is its most severe critical, high, medium or low finding. "Potentially" (a pattern that needs a person to confirm it) and "hardening" (a missing safeguard) do not change that rating; a chart that counts them says so. High or critical means at least one finding at either level; low or less means no critical, high or medium finding.

Platform-specific rules

Biometric bypass, intent redirection, serialization, cryptography, cleartext traffic and storage rules run on Android only; App Transport Security and privacy-manifest rules run on iOS only. Their shares use the platform's apps as the denominator.

Unknown values

When an app's code is packed or unreadable, checks return unknown. Those apps are excluded from that check's denominator and the count is shown beside the figure. Shielding charts show unknown as its own segment.

Embedded credentials

Secrets in a recognised format, such as API keys, access tokens or private keys. Google Cloud API keys are reported separately. Keys were not tested for validity.

Passkeys and attestation

Passkey API integrated means the app calls Android Credential Manager, the Google FIDO2 API or Apple's AuthenticationServices passkey APIs. FIDO2 classes bundled by other libraries do not count. Attestation means Play Integrity, SafetyNet, HMS Safety Detect, App Attest or DeviceCheck.

Shielding

Protections found in code or bundled products, on all three platforms. On iOS, jailbreak detection replaces root detection and overlay protection does not apply. Presence was not tested on devices.

Unpatched OS versions

Android and HarmonyOS: below API level 34 (Android 14), without Android Security Bulletin patches since March 2026. iOS: below iOS 18, the oldest major version Apple still fully patches.

Bank-side data

Backend exposure and CVEs come from Shodan's passive InternetDB index and are version-inferred, not tested. Breach records come from VCDB, Have I Been Pwned, ransomware.live, SEC 8-K filings and, for G-SIBs, news and regulators; each match was reviewed by hand. Certifications are website claims.

External data

Web and mail checks cover 1,252 bank domains. Post-quantum support was read from recorded TLS handshakes. Trojan target lists come from 47 published lists. Phishing hosts are from a single snapshot of four public feeds. Privacy labels are from Google Play and the App Store. For the 142 bank websites we could not open, we read the latest public archive copy (Internet Archive) from 2025 or later; archived copies can be older than the live site.

Fraud statistics

536 published figures from regulators, central banks and industry bodies, each kept with its period and population. They describe fraud in each market, not losses caused by the weaknesses in this report.

Case studies

Summaries of public research, credited on each page. Scan figures beside them are ours.



Mobile Banking App Security Report **2026**

Version 3 · 12 November 2026