Prevent Mobile Bot Abuse

A Guide for App Security Teams

MAPS™

ZIMPERIUM®

What we get wrong about Mobile Bots

The primary misunderstanding about mobile bots that leads to inadequate security is viewing them merely as network traffic anomalies. The truth is, a **mobile bot** isn't just one thing; it's a tactic that can be implemented in many ways.

At its core, a mobile bot is an automation built to imitate human behavior on a mobile device. Its goal is to trick your app into thinking it's interacting with a legitimate human user.

What protections can they bypass

Mobile bots are attractive because once they trick an app into thinking they're real users, most protections never even trigger. They operate inside the client, so they bypass checks meant for outside traffic.

Examples of protections bypassed:

1. **CAPTCHA**
   Once solved or bypassed in a session, the result can be stored and reused, removing the challenge for future requests.

2. **Rate Limits / Throttling**
   Bots imitate human pacing or spread activity across many devices and IPs to stay under detection thresholds.

3. **Multi-Factor Authentication (MFA)**
   If the bot runs in an authenticated session, MFA is already satisfied, giving it full account access.

4. **API Security**
   Traditional API gateways and WAFs focus on server-side traffic inspection. Bots operating inside a real app send requests that look legitimate, so these tools rarely flag them.

5. **Traditional Bot Detection**
   Network-based bot detection relies on traffic patterns or device fingerprinting. Bots on actual devices or emulators can spoof device signals, rotate identities, and blend into normal mobile traffic.

Mobile bots combine automation with trusted app environments to bypass these layers at scale, enabling account takeovers, transaction abuse, and large-scale fraud.

How Are They Built and Deployed

As we mentioned earlier, mobile bots are a tactic that attackers package and deploy in various deceptive ways. Understanding the tools used to manipulate the device and app will help you apply the right protections to defend against them.

Here are the common ways we see attackers create and run mobile-bots:

1. ### Emulator or Rooted Device Scripts

Attackers run automated scripts on emulators, rooted/jailbroken devices, and device farms. Tools like **Android Debug Bridge (ADB)** or scripting frameworks (e.g., **MonkeyRunner, Appium**) simulate user actions such as creating accounts, logging in, or making purchases at high volume.

2. ### Runtime Injection with Tools like Frida or Xposed

Frida and **Xposed** are dynamic instrumentation tools that let attackers hook into a running app's code. They can inject scripts at runtime to alter logic, skip security checks, or automate workflows from inside the app. This means the bot logic runs invisibly during a real user session, making it harder to detect through normal traffic inspection.

3. ### Repackaged Apps with Built-in Automation

Attackers take a legitimate app, **reverse-engineer** it, and modify its code to embed bot logic directly. They then repackage and redistribute this "cloned" version. Tools like **apktool** or **smali/baksmali** are used to decompile and reassemble the app. The automation is hardcoded, allowing it to bypass client-side restrictions like rate limits or redemption caps without depending on external scripts.

4. ### Malware-Controlled Bot Activity

Malware on the device acts as a command-and-control client. Once installed, it can intercept app traffic, automate in-app actions, or forward commands from a remote attacker. This is common with **banking trojans** or spyware that exploit Android's background service capabilities to trigger bot activity through legitimate apps.

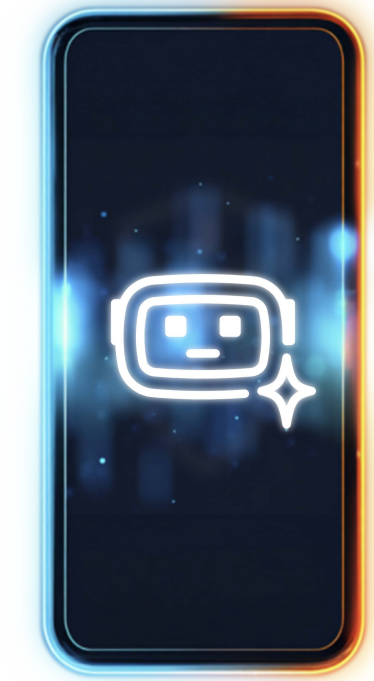
5. ### Accessibility Permission Abuse

A malicious app requests Android Accessibility Service permissions, which are meant for assisting users with disabilities. Once granted, it can programmatically "tap" buttons, enter text, and navigate inside other apps — including your app — without changing its code. Tools like **AutoInput** or custom accessibility scripts make this automation simple and stealthy.

How Can We Proactively Stop Mobile Bots on the Device

To effectively prevent mobile-bot abuse, you must enhance your app security approach with strong **in-app protection** that can do the following:

- Detect when your app is running in an emulator or a rooted/ jailbroken device.
- Detect when your apps are being run on malicious device farms.
- Detect when the app is being exercised without normal device sensor activity.
- Block runtime code injections.
- Prevent app repackaging and tampering.
- Identify malicious apps or malware controlling device behavior.
- Flags when a device is remotely operated, allowing another party to tap and interact with apps during a FaceTime session.
- Flags when an iPhone is operated from another Apple device, meaning actions can be performed without the user holding the phone.
- Flags when screen sharing is active via Zoom, exposing app data to others.
- Flags when screen sharing is active via Teams, risking credential or workflow exposure.
- Flags when screen sharing is active via Whatsapp, enabling social engineering or data leakage.



This is precisely what Zimperium's Mobile Application Protection Suite (MAPS) delivers, continuous, in-app protection that keeps mobile-bots from exploiting your business logic and APIs. The MAPS platform empowers your mobile apps to detect and protect themselves against these bot-related tactics on the device in real-time.

Learn more about how MAPS protects against mobile-bots and other threats:
https://zimperium.com/maps/

ZIMPERIUM®

Learn more at: zimperium.com
Contact us at: 844.601.6760 | info@zimperium.com
Zimperium, Inc
4055 Valley View, Dallas, TX 75244