



The End of Mobile Security Complacency: DMA, Antitrust, and the Rise of Real API Protection

Ted Miracco, George McGregor
Approov

OWASP Los Angeles
June 25 2025



Open App Markets Act

Breaking News!

Blackburn, Blumenthal, Lee, Klobuchar, and Durbin Introduce Bipartisan Antitrust Bill to Promote App Store Competition

WASHINGTON, D.C. – June 24 2025, U.S. Senators Marsha Blackburn (R-Tenn.), Richard Blumenthal (D-Conn.), Mike Lee (R-Utah), Amy Klobuchar (D-Minn.), and Dick Durbin (D-Ill.) introduced the bipartisan **Open App Markets Act**, which would set fair, clear, and enforceable rules to promote competition and strengthen consumer protections within the app market. Google and Apple currently have gatekeeper control of the two dominant mobile operating systems and their app stores that allow them to exclusively dictate the terms of the app market, inhibiting competition and restricting consumer choice.

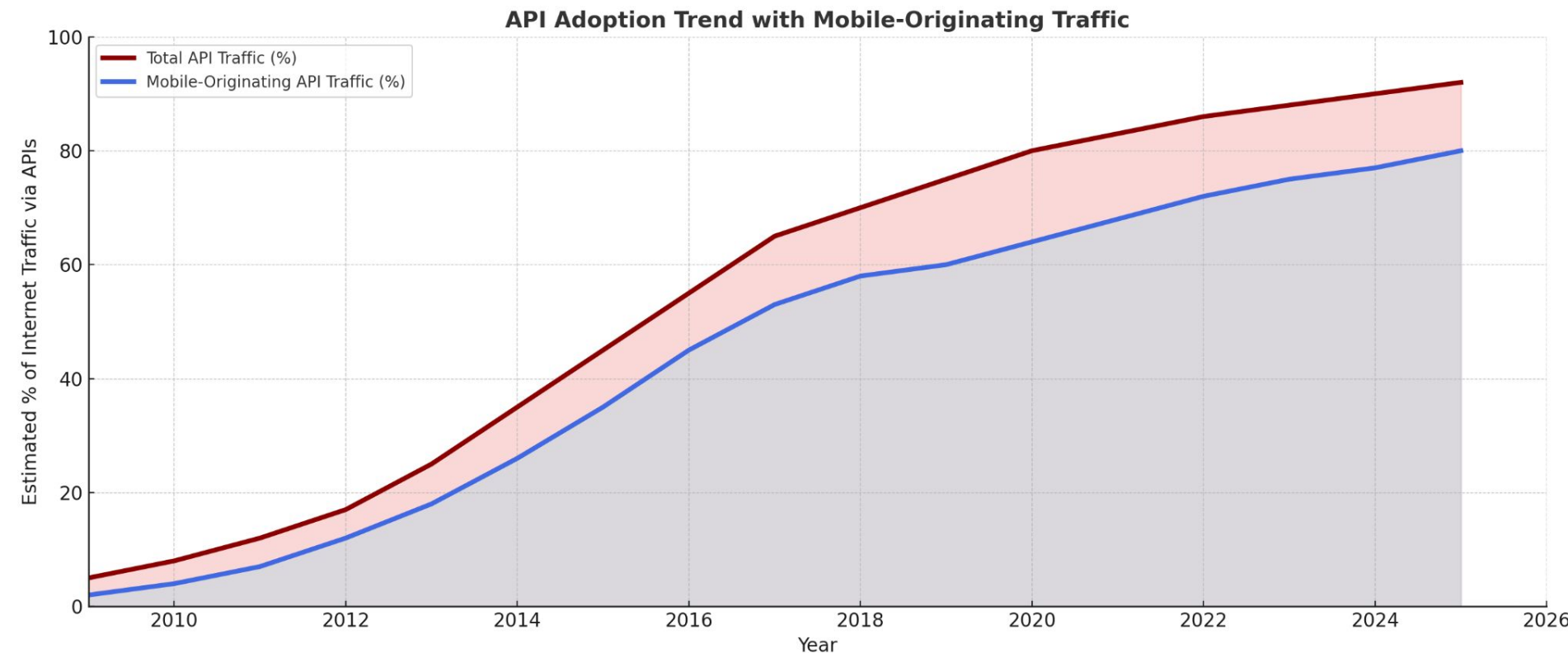
The Mobile Paradox

- The Reality: Mobile app traffic is a significant and rapidly growing piece of the global digital ecosystem, eclipsing traditional web traffic in many sectors.
- The Problem: Mobile security receives significantly less dedicated attention and budget from CISOs compared to web security.
- This creates a dangerous blind spot where organizations are most exposed.



The Mobile Issue

Common Mobile to API Attacks



- Repackaged apps bypass restrictions
- Emulators to scale attacks
- Man-in-the-Middle to extract API keys and secrets
- Automation tools for script-based abuse
- Credential Stuffing via the API

The threat is real: Kahoot!, Starbucks, T-Mobile, ...and VW...

The Great App Store Myth: A False Sense of Security

- A Walled Garden with Open Gates
- Consumers (and many developers) depend on the Apple App Store and Google Play, but this trust is misplaced.
- cursory Reviews: The review process is notoriously brief and automated, primarily focused on policy compliance, not deep security analysis.
- The Real Target is Unseen: Even a thorough review can't secure what it doesn't control: the backend APIs.
- The Crown Jewels: These APIs protect the truly valuable data:
 - Health information (HIPAA)
 - Financial data & cryptocurrency
 - PII, rewards points, vehicle access, and more.
- The app on the phone is just the key; the API is the lock on the vault.

Apple Testimony at DMA (2024) vs. ruling in US Court in 2025



Kyle Andeer, VP Products and Regulatory Law, Apple

“In stark contrast to Apple’s initial in-court testimony, contemporaneous business documents reveal that Apple knew exactly what it was doing and at every turn chose the most anti-competitive option,” Rogers wrote. “To hide the truth, Vice-President of Finance, Alex Roman, outright lied under oath.”

The EU Digital Markets Act (DMA)

The European Union's Digital Markets Act (DMA) aims to make digital markets in the EU fairer and more contestable. It does this by establishing rules for large online platforms, referred to as "gatekeepers," to prevent them from ***abusing their market power***.

Designates 6 “gatekeepers” and 22 core platform services

Explicitly requires Apple (AppStore) and Google (Play) to enable:

- Alternate app stores
- Sideloaded apps
- Alternative Payment Mechanisms

Google made some changes (Android already allows alternative apps stores and payment mechanisms)

Apple provided a detailed and complex response

- Heavily criticised by peers and app owners
- **In April 2025 the EU fined Apple for non-compliance**

The DMA is part of a larger worldwide trend.

The App Store Duopoly

App Store Commission Rates		Google Play Commission Rates	
Standard commission	30%	Standard commission	30%
Qualifying subscriptions (after 12 months)	15%	Qualifying subscriptions (after 12 months)	15%
App Store Small Business Program (annual revenue up to \$1 million)	15%	The first \$1 million in annual revenue	15%

Contentious Issues:

- App Store Exclusivity
- In-App Payment Restrictions
- Anti-Steering Provisions

Some platform dependent security “included”

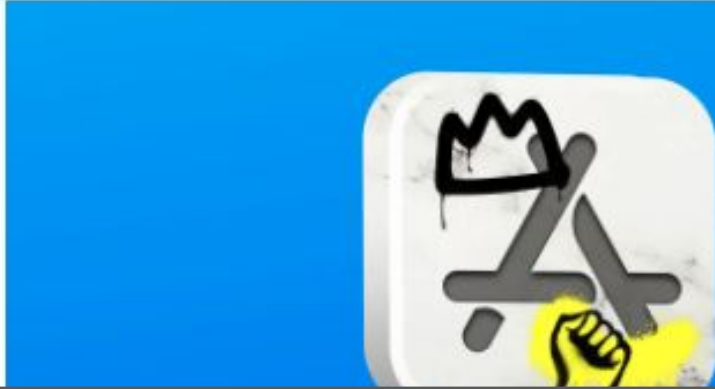
Response to the Apple Proposal

WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS MORE SIGN IN

MORGAN MEAKER BUSINESS FEB 12, 2024 4:00 AM

Developers Are in Open Revolt Over Apple's New App Store Rules

European app makers are seething, comparing Apple to "the Mafia" and piling pressure on lawmakers to act.



Exclusive: Apple faces 'strong action' if App Store changes fall short, EU's Breton says

By Martin Coulter, Foo Yun Chee and Supantha Mukherjee

January 26, 2024 1:08 PM PST · Updated 18 days ago



LONDON, Jan 26 (Reuters) - Apple faces strong action if changes to its App Store do not meet incoming European Union regulations, the bloc's industry chief said on Friday.

Proton

Proton news Privacy basics Privacy deep dives Privacy news Opinion For business



OPINION

Apple's DMA compliance plan is a trap and a slap in the face for the European Commission

 Andy Yen

Epic Games vs Apple and Google

Epic Games accused Apple and Google of monopolistic practices in their respective app stores app distribution and in-app payment systems

- Apple:
 - Court ruled not a monopoly but anti-steering illegal
 - In 2025, Apple found in contempt for not fully complying with the injunction
- Google:
 - Epic won (Google Play an illegal monopoly)
 - Google forced to allow alternative app stores and billing on Android



The Rise of HarmonyOS & Xiaomi

- Launched by Huawei in 2019, built on OpenHarmony
- in 2024 **HarmonyOS NEXT** removes any Android dependencies
- HarmonyOS now dominant in China smart device market
- Oniro from The Eclipse Foundation is built on Open Harmony, aimed at the global market



- Xiaomi is reportedly working on its own operating system, **HyperOS**, which could be developed in collaboration with Huawei and BBK Electronics to create a Google-free ecosystem.

The Rise of Cross Platform Development

Framework	Approx Share of Cross Platform Apps	Example Apps
Flutter	45%	Google Pay, BMW, eBay
React Native	30%	Facebook, Bloomberg, Walmart, Pinterest, Wix
Cordova	10%	Slack, Coinbase, Duolingo, Mint

Security should be cross-platform too

Security Implications

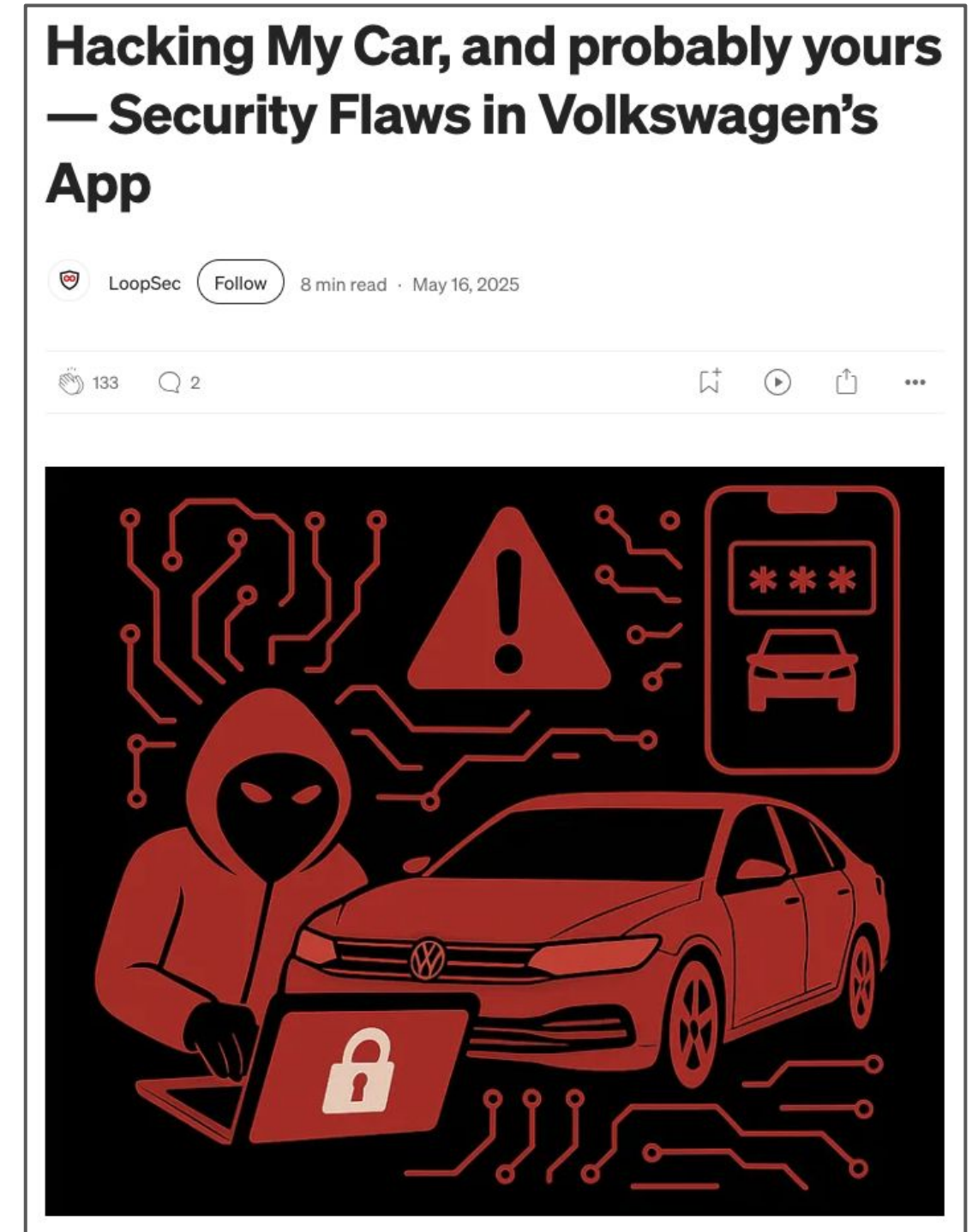
- Complexity and costs of having separate security approaches for Google and Apple
- New platforms such as Harmony OS
- Sideloaded apps and alternative app stores **will** happen
- Use of cross-platform development tools
- **Less control over apps means shift to API protection**
- App developers should be able to opt out of high taxes in return for incorporating their own or third party security

Mobile app providers must

- Make no assumptions about how apps are distributed
- Urgently seek alternative security approaches

Volkswagen Hack - May 2025

- Accessed any VW using the VIN number and a simple script to find the right 4 digit code
- Obtained internal app keys and tokens, owner personal info, service data for the vehicle (VIN)
- ... and any other Volkswagen via a BOLA issue





MASVS-CRYPTO
MASVS-NETWORK
MASVS-RESILIENCE

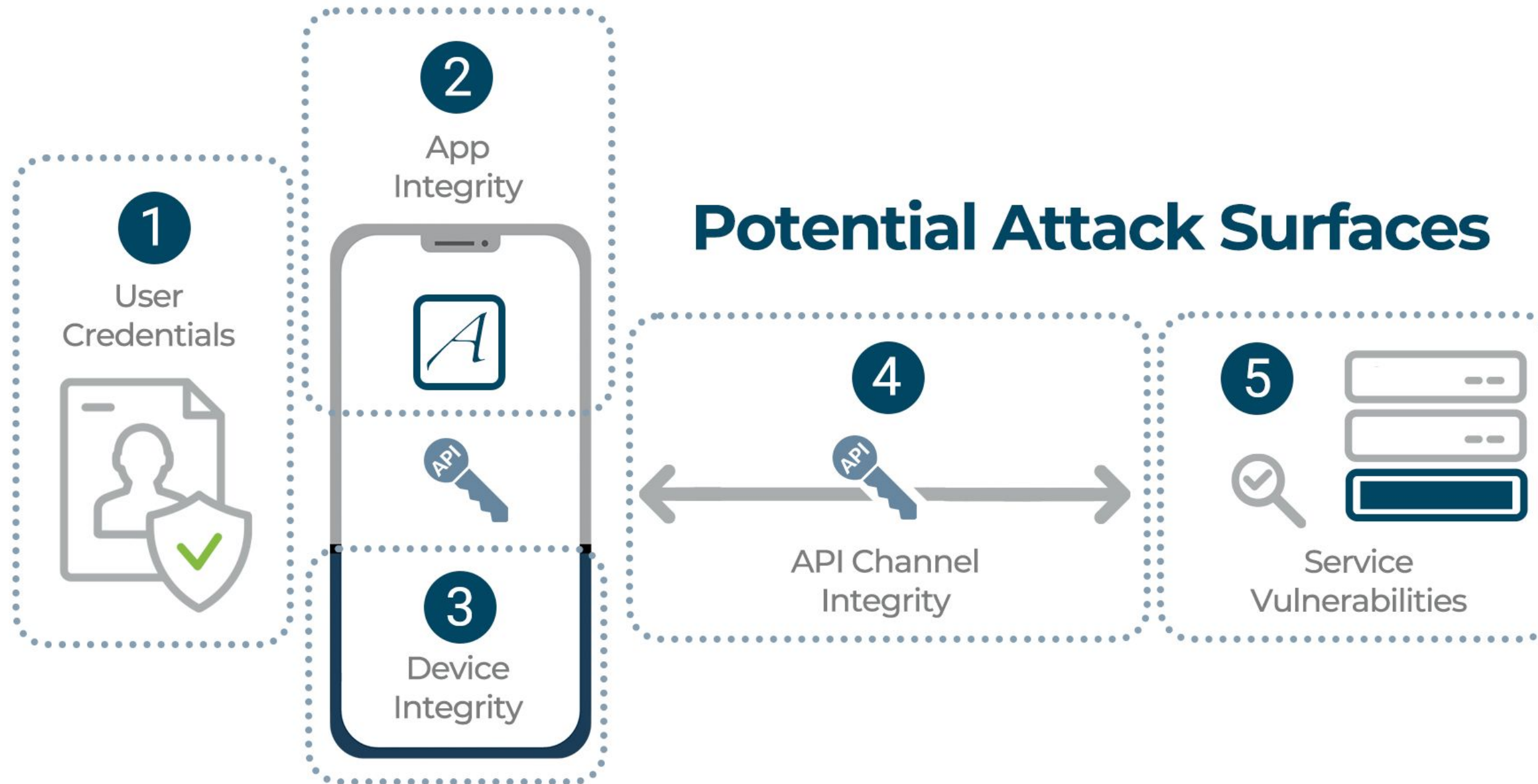
Worst Mobile Security Breaches by OWASP Mobile Top 10 Category

OWASP Category	Breach Example(s)	Impact Summary
M1: Improper Credential Usage	- Uber (2016) : Leaked hardcoded credentials in mobile code- Toyota (2022) : API keys exposed in mobile apps	Secrets stored insecurely in apps led to unauthorized access to backend APIs
M2: Insecure Authentication/Authorization	- Instagram (2020) : Logic flaw allowed unauthorized access to user DMs- WhatsApp clones : Bypassed auth to hijack sessions	Incomplete session/token validation allowed account hijacking
M3: Insecure Communication	- Kaspersky (2021) : Insecure HTTP endpoints in Android app- TikTok (2022) : TLS validation bypass risks	Data exposure in transit due to lack of HTTPS or certificate validation
M4: Insecure Data Storage	- Facebook (multiple) : App caches exposed private data- Health apps : Local logs saved sensitive user data in plain text	Local device storage leaks through logs, backups, or accessible app files
M7: Code Tampering	- Banking malware (e.g., Teabot, Hydra) : Inject malicious code into legitimate apps- Modded apps (APKMirror clones)	Reverse-engineered or tampered apps allowed attackers to bypass security or steal data
M8: Security Misconfiguration	- Grindr (2021) : Debug settings and exposed backend endpoints- Retail mobile apps : Excessive permissions or debug APIs active	Misconfigured apps leaked sensitive info or enabled abuse of backend systems

Worst Real-World API Breaches by OWASP API Top 10 Category

OWASP Category	Breach Example(s)	Impact Summary
API1:2023 Broken Object Level Authorization	- Parler : Public user metadata via predictable IDs- Johns Hopkins : IDOR in internal system	Exposed user profiles, medical or academic data; easily scriptable attacks
API2:2023 Broken Authentication	- Facebook (540M records exposed)- Uber (57M user details leaked)	Poor token handling and stolen credentials led to mass data compromise
API4:2023 Unrestricted Resource Consumption	- GitHub/Twitter : Data scraping via search/email enumeration- T-Mobile : Attackers extracted personal info in bulk	Automated scraping and DoS-style abuse through lack of rate limiting
API6:2023 Unrestricted Access to Sensitive Business Flows	- Ticketmaster bots - Nike SNKRS app : Abuse of limited-offer flows	Business logic flaws led to unfair access, fraud, and revenue loss
API9:2023 Improper Inventory Management	- Snapchat : Exposed user data via debug API- Panera Bread : Customer data leaked for months	Untracked, undocumented APIs exposed critical user data silently

Mobile Threats at Runtime



Why Our Old Walls Are Crumbling - Traditional Defenses

- Current solutions for mobile security are often based on outdated concepts.
- Static Security: Scans code before it's compiled, but is blind to runtime behavior where attacks actually happen.
- Code Obfuscation: Increasingly a "speed bump, not a roadblock."
- It's easily defeated by determined attackers.
- Modern AI-powered deobfuscation tools can automate the process of reverse-engineering, rendering it ineffective as a primary defense.
- We are trying to solve a dynamic, runtime problem with static, pre-deployment solutions.

Why Mobile SDKs are Critical for API Security

- Mobile apps are easily modified, run in hostile environments.
- Automated tools can mimic valid traffic.
- Backend API security has no visibility into mobile threats.

A Mobile SDK Can Add the Missing Context

- Verify that the app has not been modified or repackaged.
- Ensure the device is not rooted/jailbroken, running on an emulator, or tampered with.
- **Continuously attest the runtime environment** using trusted hardware or integrity checks.
- **Bind requests to the genuine app** using cryptographically signed tokens (e.g., JWTs).
- **Block automated tools like Frida, Magisk, Xposed** before they even touch the API.

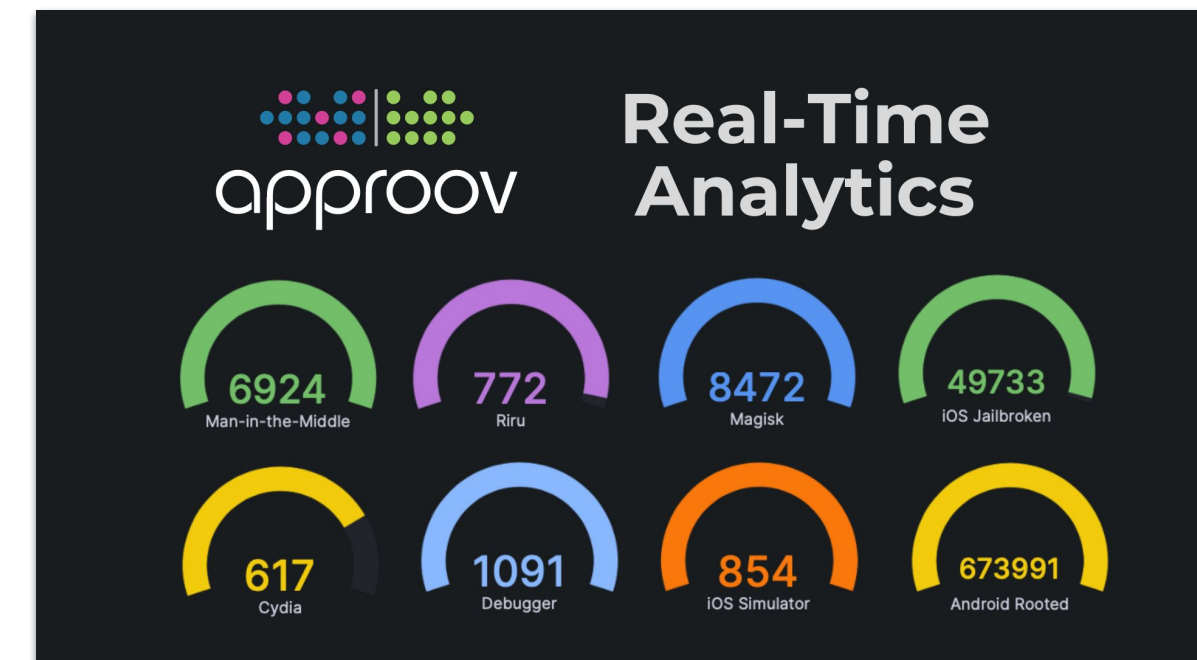
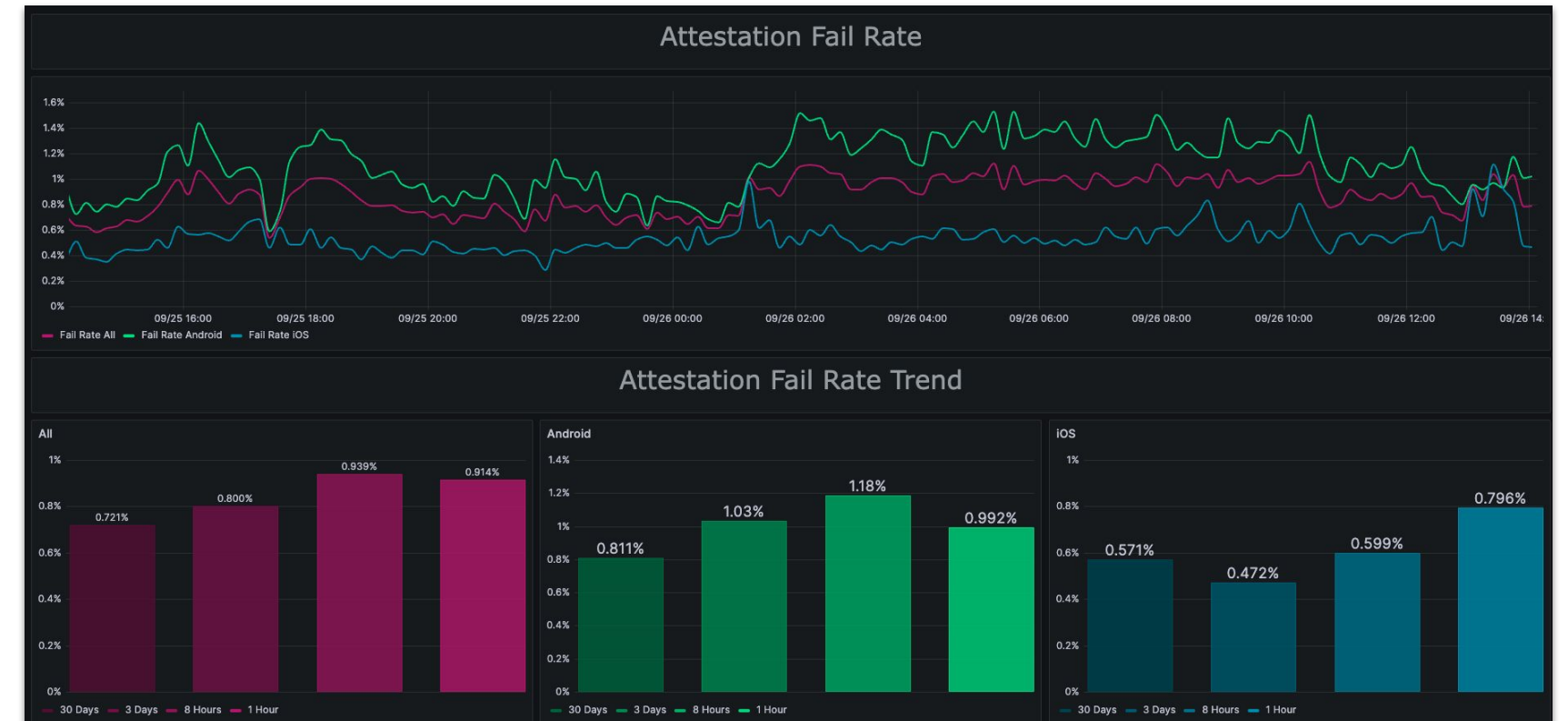


No amount of backend analysis can detect if the device was rooted, if the app has been modified, or if sensitive secrets are being exfiltrated.

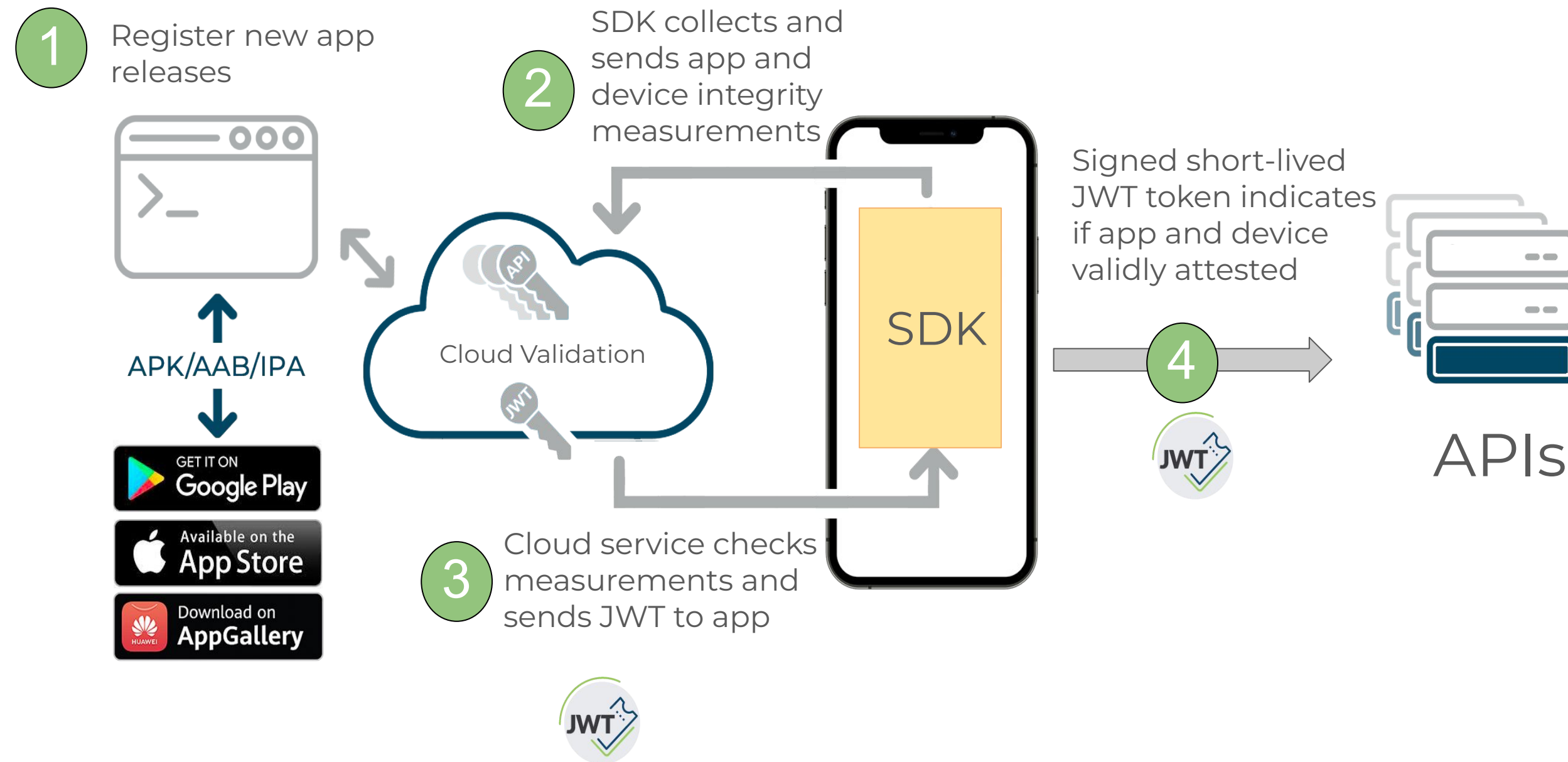
What is really needed

**Continuous runtime,
transaction-level security:
multi-platform, easy to manage**

- RASP (Runtime App Self Protection)
- App Attestation
- Secure Client Validation
- Real Time Analytics
- Dynamic Secrets (Not embedded)
- Certificate Pinning (Dynamic, OTA)
- Over-the-Air (OTA) Updates



Effective App and Device Attestation



Google Runtime Security Limitations

- Google **PlayIntegrity** and **SafetyNet** are also “free” to App Developers
- Android only, needs Play Services
- Limited device checks
- Slow and complex to implement
- No MitM protection or dynamic secrets management

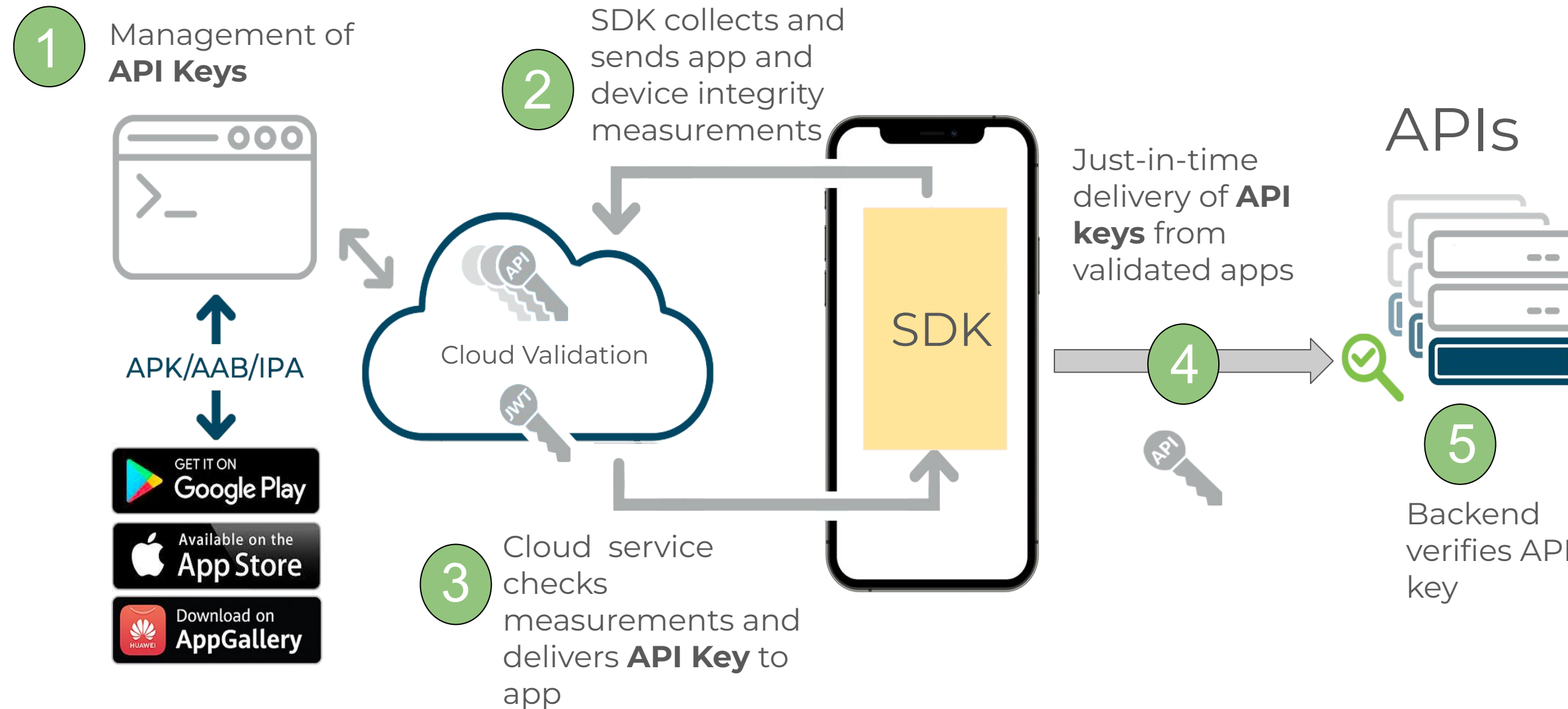


Apple Runtime Security Limitations

- Apple **DeviceCheck** and **AppAttest** are “free” to Mobile App Developers
- iOS only
- Limited device checks
- Limited analytics
- Non-published service rate limits
- Complex to implement
- No MitM protection or dynamic secrets management



Extending Attestation to Secret Protection



- Just-in-time delivery of secrets to mobile apps, only when needed and only if app is safe
- Dynamic and secure cloud management of secrets
- Prevents abuse of secrets stolen from any source
- Must works with owned and 3rd party APIs

Backend Integration

JWT Validation settings

Token configurations

Define the token key and location in a request. Configurations are used with JWT validation rules. [Rotating keys with Workers](#)

You have used 0 out of 4 available configurations.

Token configurations

No JWT configurations added.

Schema Validation settings

Default action

Select the default action for schemas to take on non-compliant requests. Changing this setting will effect all endpoints set to the “default” action.

No action

Select configurations

Validate incoming JSON Web Tokens against one or more of your configurations. [Manage configurations](#)

Configuration	Token Location	Last Updated	If token is missing
<input checked="" type="checkbox"/> Shapes 38cfc61d-776f-46c...0b8-a44e813244d2	Header:approov-token	05/24/2024 6:26 PM	Mark as non-compliant

Validation behavior for multiple configurations

Validate at least one configuration

Expression Preview [Edit expression](#)

(is_jwt_valid("38cfc61d-776f-46c2-90b8-a44e813244d2"))

Set action

Specify what should happen when non-compliant requests to this hostname are found.

☐ Log

Non-compliant requests will be shown in Security Events and are allowed to access the requested endpoint.

☒ Block

Prevent non-compliant requests from accessing the requested endpoint.

Conclusion - The Need for End-to-End Mobile App and API Protection

1. **Shift Focus:** Prioritize mobile API security as much as, or more than, client-side app security. The data lives on the backend.
2. **Distrust the App Stores:** They are distribution platforms, not a security control.
3. **Embrace Runtime Security:** Static analysis and obfuscation are not enough. You need runtime protection and the agility of OTA updates.
4. **Prepare for the New World:** A de-monopolized, global, multi-store ecosystem is coming. Your threat model must expand to include it.





Thank You

ted.miracco@approov.io

george.mcgregor@approov.io



Approov Promotional Stuff...

- Follow us on [X.com](#), [LinkedIn](#), [BlueSky](#)
- Approov [blog](#)
- Upwardly Mobile API & App Security Podcast
 - 67 Episodes
 - Security News and Information
- WE ARE [HIRING!](#)
www.approov.io



Extra Resources

On Apple and Google security:

<https://approov.io/blog/limitations-of-apple-devicecheck-and-apple-app-attest>

<https://approov.io/blog/limitations-of-google-play-integrity-api-ex-safetynet>

Alignment with OWASP

<https://approov.io/download/Achieving-OWASP-App-Resilience.pdf>

<https://approov.io/info/how-to-use-the-2024-owasp-mobile-top-ten>

Other whitepapers and videos

<https://approov.io/resource/>



Revisiting the OWASP Mobile Top 10

Category	Title	Description
M1	Improper Credential Usage	Insecure handling of passwords, API keys, tokens, or certificates.
M2	Insecure Authentication/Authorization	Weak or flawed user identity or session management mechanisms.
M3	Insecure Communication	Unprotected transmission of sensitive data (e.g., via HTTP, weak TLS).
M4	Insecure Data Storage	Improperly secured sensitive data at rest on the device.
M5	Insufficient Cryptography	Use of broken or improperly implemented encryption.
M6	Insecure Code Quality	Bugs and unsafe coding patterns leading to vulnerabilities.
M7	Code Tampering	Lack of protections against reverse engineering or code modification.
M8	Security Misconfiguration	Incorrectly set permissions, exposed debug services, etc.
M9	Insecure Dependencies	Use of vulnerable third-party libraries or SDKs.
M10	Insufficient Security Controls	Missing runtime protections, lack of defense-in-depth mechanisms.

OWASP API Top 10

Category	Title	Description
API1:2023	Broken Object Level Authorization	APIs expose endpoints that handle object identifiers, creating a wide attack surface for unauthorized access.
API2:2023	Broken Authentication	Authentication mechanisms are improperly implemented or absent, enabling attackers to compromise accounts.
API3:2023	Broken Object Property Level Authorization	APIs allow access or modification to properties that should not be exposed.
API4:2023	Unrestricted Resource Consumption	APIs don't impose limits on resource usage, enabling DoS attacks.
API5:2023	Broken Function Level Authorization	Access control checks are missing or inconsistent across functions.
API6:2023	Unrestricted Access to Sensitive Business Flows	Lack of access controls on high-value business actions (e.g., purchases, transfers).
API7:2023	Server Side Request Forgery (SSRF)	APIs fetch remote resources without validating the URL, enabling attackers to access internal systems.
API8:2023	Security Misconfiguration	Poorly configured security headers, CORS, or default settings expose APIs to risks.
API9:2023	Improper Inventory Management	Lack of visibility into API versions and exposed endpoints leads to shadow APIs and outdated versions being exploited.
API10:2023	Unsafe Consumption of APIs	Trusting external APIs without validation can result in data leaks or unexpected behavior.

Part 2 - The Cesspool- Mobile App and API Exposure

- Volkswagen example
- Back to basics - the mobile app and API attack surfaces
- Real world Attack scenarios
- Obfuscation and a false sense of security
- Revisit Apple and Google Security - more about the shortcomings
- Non-negotiable -> Security should be runtime, transaction-level, multi-platform, easy to manage