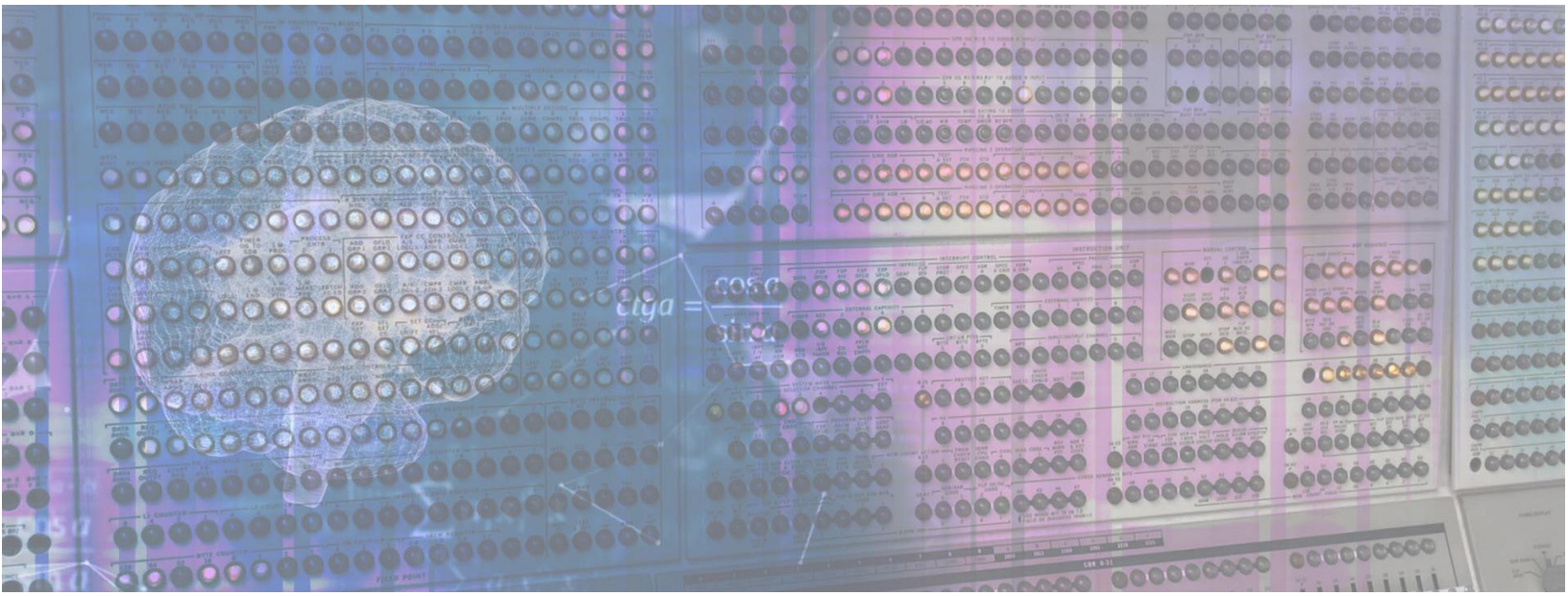

Why Architectural Security is the Key to Mobile App Hygiene

An Intellyx White Paper for Approov

by Jason Bloomberg, Managing Director

August 2026

Given the ubiquity of mobile devices and the global availability of AI-based tools and applications, mobile application hygiene has become a paramount concern of CISOs and their organizations.

Based upon this new reality, here are the key takeaways in this paper:

- **Assume attackers can compromise all mobile apps** – treat all mobile apps as untrusted. Assume attackers can reverse engineer them and extract any embedded secrets, including passwords, API keys, and more.
- **AI has made the mobile app hygiene problem worse** – attackers have easy access to increasingly powerful AI-based tools they can use to analyze apps, discover their vulnerabilities, and automate attacks.
- **Protect back-end services** – APIs are the primary security boundary between mobile apps and your back-end. Apply zero-trust principles to every request hitting your organization.
- **Don't rely on long-lived secrets** – since attackers can compromise any secrets no matter how well you hide them, don't use secrets that persist more than a few seconds. Use short-lived, tightly scoped tokens, app attestation, and continuous validation instead.
- **Adopt a shift-left, security by design architecture** – leverage Approov's Zero Secrets Architecture to ensure your mobile apps remain secure even when attackers know everything about their behavior and contents. And if they do compromise an app – make sure there's nothing of value for them to steal.

The mobile app security hygiene problem in the age of AI

What do perpetrators of cybersecurity attacks love most of all?

Mobile apps.

They are everywhere, on every device, including the attacker's. All the code, all the binaries, and all the API calls, right in the attacker's hands.

Enterprises love to worry about their threat surface. It doesn't matter what's in that mobile app, it is outside your threat surface – exposed to the world for any attacker to break in and steal its secrets.

Secrets, of course, are only secrets if nobody knows what they are. But today, it doesn't matter how secret you think you are keeping your passwords or API keys or obscured security code.

Attackers can get them all.

Secrets are only secrets if nobody knows what they are. But today, it doesn't matter how secret you think you are keeping your passwords or API keys or obscured security code. Attackers can get them all.

Reverse engineering your app is now dead simple. It doesn't matter how obscure the code is or how hidden the secrets are. You can count on attackers finding – and exploiting – everything.

Not only can today's attackers reverse engineer your apps, they can also analyze your apps' runtime behavior on the fly.

And then, once they have full visibility into the inner workings of your app, they can use AI to automate attacks at scale – in particular, agentic AI-based attacks they fine-tune for the purpose.

Attacks that required sophistication a couple of years ago are now dead simple. Anyone can compromise your app at agentic AI speeds.

Instead of securing the app itself, therefore, you must retreat to a defensible position: the API. APIs are the primary security boundary separating potentially compromised mobile apps from the back-end services that support them.

In other words, securing mobile apps is a zero-trust exercise. Expect attackers to have compromised your mobile apps and act accordingly. Protect your back-end services regardless of the nature of the threats.

Securing mobile apps is a zero-trust exercise. Expect attackers to have compromised your mobile apps and act accordingly. Protect your back-end services regardless of the nature of the threats.

Proactive prevention is your essential posture. Practice mobile app security hygiene – all the time, every time.

Why mobile app security hygiene is a CISO-level concern

Mobile app security hygiene is about protecting APIs and the business-critical data they expose. Given the risks that mobile apps present, the entire security organization from the CISO on down must give mobile app security hygiene the top priority.

There are other points of compromise that CISOs must worry about, of course. What makes mobile apps so dangerous is the fact that when attackers compromise them, they obtain access to all the secrets and capabilities that the app contains.

Mobile apps don't only access back-end services. They also interact with the data they extract from those services. By understanding the internals of each mobile app, attackers can gain an understanding of the corporate data the app interacts with.

Mobile devices are also getting smarter with each generation – improving the experience for end users as well as attackers. As device vendors add large language models (LLMs) to smartphones, attackers will be able to deploy agentic attacks directly on the device – attacks with more sophistication than possible before.

What makes mobile apps so dangerous is the fact that when attackers compromise them, they obtain access to all the secrets and capabilities that the app contains.

In addition, the mobile app hygiene challenge is global in scope. As multinational enterprises focus their efforts on emerging markets, the diversity of device and mobile operating system manufacturers complicates the hygiene challenge.

If a bank with a global presence, for example, wants to deploy a mobile app for accessing accounts in multiple countries, then the bank must consider the diversity of technologies available in each country they do business in – even when those countries don't support the Apple AppStore and Google Play Store ecosystems.

Even within the Apple and Google ecosystems, furthermore, insecure handling of API credentials is a systemic problem – and with the increasing prevalence of AI capabilities on the devices, the problem is only getting worse.

Furthermore, no type of app is immune. Gaming, infotainment, B2B, travel, finance – all apps are vulnerable, regardless of the sophistication of the app developer or the popularity of the app itself.

Even the type of device doesn't present limits to determined attackers. While most attacks focus on smartphones, other mobile endpoints like automobile infotainment systems and Internet of Things endpoints from logistics to manufacturing all suffer from mobile app hygiene weaknesses.

For all these reasons, CISOs should consider mobile app security hygiene as an essential component of the enterprise's protections against AI-based attacks.

Instituting adequate mobile app security hygiene is not a Band-Aid fix. It requires a rethink of the architecture that comprises the entire app-to-back end service ecosystem.

Instituting adequate mobile app security hygiene is not a Band-Aid fix. It requires a rethink of the architecture that comprises the entire app-to-back end service ecosystem.

Protecting the mobile-to-API ecosystem

This architectural rethink of the mobile app security challenge begins with the realization that simply protecting the app itself is insufficient. Instead, organizations must protect the entire mobile-to-API ecosystem – not simply the app, but the devices, the network, the APIs, and the back-end services they provide access to.

Given the fact that attackers can not only reverse-engineer apps but can also inspect runtime behavior to automate attacks at scale, architects and developers must design systems that remain secure even under attackers' full analysis of each app.

This ecosystem-centric approach to mobile app security begins with secrets: passwords, API keys, credentials, and other bits of information that organizations wish to keep out of the hands of attackers.

The mobile app ecosystem architecture must focus on keeping long-lived secrets off devices entirely. Simply hiding or otherwise obscuring the secrets is insufficient. Always assume the attacker has a sophisticated, AI-driven hacking tool that can uncover even the best hidden secrets.

Always assume the attacker has a sophisticated, AI-driven hacking tool that can uncover even the best hidden secrets.

The alternative is to use short-lived, tightly scoped tokens instead, so even if an attacker is able to obtain a token, its usefulness is limited both in scope and time.

Simply authenticating users is insufficient. Instead, it's essential to verify each app, each device, and individual API requests via attestation, integrity checks, and replay protection.

It's also important to treat the APIs themselves as the primary security boundary. Assume everything outside the API is hostile and implement zero-trust principles accordingly. At no time should any element of the mobile app ecosystem assume that anything outside this boundary is trusted.

Taking this zero-trust approach to implementing mobile app security requires constant vigilance. As a result, organizations must implement continuous security testing including automated scanning, dependency analysis, and API testing, with a focus on runtime attacks and software supply chain security.

The bottom line: every organization should implement an architecture that minimizes the need to trust the client while protecting backend services, even when an attacker compromises the app itself.

Architectural security in the face of agentic AI threats

Many attackers are certainly inexperienced, relying upon dark web downloads for tools that may or may not give them a foothold in your organization.

Nevertheless, every organization must assume that some of the attackers trying to breach their mobile apps are smart, experienced, and have access to the most powerful AI-driven hacking tools available.

Such AI-assisted tools can quickly decompile mobile apps, identify hard-coded API keys, map API endpoints, understand application logic, and generate scripts to automate attacks or replay captured requests.

As a result, security techniques that relied on slowing attackers, such as code obfuscation, embedded secrets, or basic behavioral detection, are no longer sufficient and at worst, can provide a false sense of security that increases the threat level.

It is for these reasons that an architectural approach to mobile app security hygiene is the better approach. Organizations must shift mobile app security to the left by designing systems to remain secure even when attackers have full visibility into the workings of each app.

Best practices for implementing such architectures include removing long-lived secrets from apps, delivering credentials only to trusted app instances at runtime, using short-lived, app-bound access tokens that are difficult to replay, adopting stronger app and device attestation, and zero-trust API security that verifies each request.

The goal is to both discourage attackers from attempting a compromise and to minimize the value of anything an attacker might obtain, thus encouraging them to attack someone else.

Approov's Zero Secrets Architecture

Approov offers its customers technology for improving their mobile app security hygiene, not only via technology, but more importantly, via architecture.

In fact, Approov's Zero Secrets Architecture (ZSA) assumes attackers have full visibility into the inner workings of mobile apps – via reverse engineering, AI-based tooling and any other technique that exposes secrets that developers may have embedded in the apps.

Approov's ZSA combines several architectural best practices, including:

- **Cross-platform mobile app attestation** that verifies that API requests originate from genuine apps on trusted devices – apps that attackers haven't tampered with.
- **Runtime secrets protection** that delivers short-term secrets at the point of use where the app never embeds or stores the secrets.
- **Dynamic certificate pinning** that allows the infrastructure to update trusted server certificates automatically without requiring users to update their apps.
- **Short-lived JSON web tokens (JWTs)** that the infrastructure ties to specific instances of each app, ensuring that even if attackers steal a token, they would be unable to use it to achieve their goals.
- **Rapid credential rotation** that reduces the changes that an attacker is able to compromise a security credential.

The goal of these architectural best practices isn't to make apps impossible to analyze (unfeasible in the age of agentic AI), but rather to ensure they contain nothing of usable value.

By following these practices, ZSA reduces the risks of app impersonation, credential theft, and unauthorized access to backend services.

In other words, Approov is reinforcing the shift toward architectural security: building security best practices into the design of individual applications as well as the entire mobile app ecosystem, including back-end services and their APIs.

This 'shift left' approach to mobile app security hygiene, therefore, goes beyond simple Band-Aid fixes that lead to never-ending cat and mouse games with attackers.

Instead, even as attackers improve their tools and attack methods, there remains little worth stealing. The apps themselves are free of useful secrets, and impersonating either legitimate clients or back-end services gives them nothing of value.

The Intellyx take

Keeping cybersecurity threats from compromising an organization has always been a cat and mouse game, as each side improves their technology.

Keeping the bad actors out, therefore, has always been hit or miss. Any perimeter-based approach to security always has its holes – holes attackers are only too happy to compromise.

This failure of perimeter-based security has always been the impetus behind zero-trust approaches, but even zero-trust falls short if the secrets each party uses to verify the other are themselves the point of compromise.

Approov's ZSA takes zero-trust to the next level – not only assuming each endpoint is untrusted, but also assuming that attackers can and will compromise all the secrets.

We would say, therefore, that ZSA is a modern reinvention of zero-trust. Where zero-trust was authorization-centric, ZSA is no-secrets centric.

How, then, do you know you need no-secrets centric ZSA vs. the more traditional authorization-centric zero-trust approaches?

Ask yourself this question: if attackers reverse-engineered your mobile app today, could they impersonate it or compromise your backend services tomorrow?

If your answer is 'yes' – or if you even think it might be 'yes' on a single occasion – then your mobile app hygiene is insufficient.

In contrast, with Approov and ZSA, rest assured your answer will always be 'no.'

About the Author

Jason Bloomberg is founder and managing director of enterprise IT industry analysis firm Intellyx BV. He is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

Mr. Bloomberg advises, writes, and speaks on a diverse set of topics, including artificial intelligence, cloud native computing, cybersecurity, and legacy modernization. He currently writes for SiliconANGLE, The New Stack, CIO, and other publications, and he is the author or coauthor of five books.

About Intellyx

Intellyx is the change agent analyst firm focused on customer-driven, technology-empowered enterprise transformation. Rather than categorizing vendors, our thought leadership distills insights across the rapidly evolving enterprise IT landscape; from agentic AI to quantum-safe security, cloud native development to modern mainframes. Our advisory helps you and your customers see through the hype and get beyond the fear of technology disruption to take action and realize value through change. Read and learn more at www.intellyx.com or follow us on LinkedIn.

About Approov

Approov is a comprehensive runtime security solution for mobile apps and their APIs, protecting against automated attacks, credential stuffing, and API abuse. By verifying that every request comes from a genuine, untampered app running on a safe device, Approov stops attackers — human or AI — in their tracks while ensuring a frictionless experience for legitimate users.