BlueCedar

WHITE PAPER

Mobile App Security Methods

By: Edward Amoroso

Founder & CEO of TAG Cyber
Former SVP & CSO of AT&T

With security threats lurking around every device, OS, and software update, in-app security is a necessity for enterprises looking to protect their data while boosting employee productivity. Mobile security no longer ends with device security, but enterprises must now secure the data being pushed to and transferred from their apps. In-app security makes managing security threats possible for every endpoint, even in a “Bring Your Own Device” environment.

General versus tailored security

One of the great tenets of modern cybersecurity is that protections work best when tailored to the specifics of the targeted resource. That is, if some database is to be secured, then the optimal safeguard solution would involve designing controls that are specific to the function, protocol, and accesses used for that database. Such fine-tuning of policy, detection, mitigation, and response actions is the most powerful means for protecting any asset.

The primary challenge to this method, however, is the high cost of tailoring cybersecurity controls to the myriad of different resources and assets in an organization. Any complex enterprise might have, for example, thousands of endpoints, servers, and applications—so the idea of shrink-wrapping controls to match their unique functions has been so impractical as to render the approach largely impossible.

This practical barrier is beginning to dissolve with the introduction of enterprise mobility. In contrast to the more rigid approach supported on traditional LAN-based desktops, mobile devices and apps create a new, more virtual approach to enterprise computing. With this evolution comes the ability to tailor protections more closely to the specifics of the app. Mobile app security thus becomes pinnacle in a taxonomy of methods starting with the perimeter.

With mobile apps, the ability arises that software controls can be virtualized into the actual application code to increase security protection. This approach allows for such controls to work in proximity to the valued app, which reduces dependency on avoiding vulnerabilities in the surrounding environment.

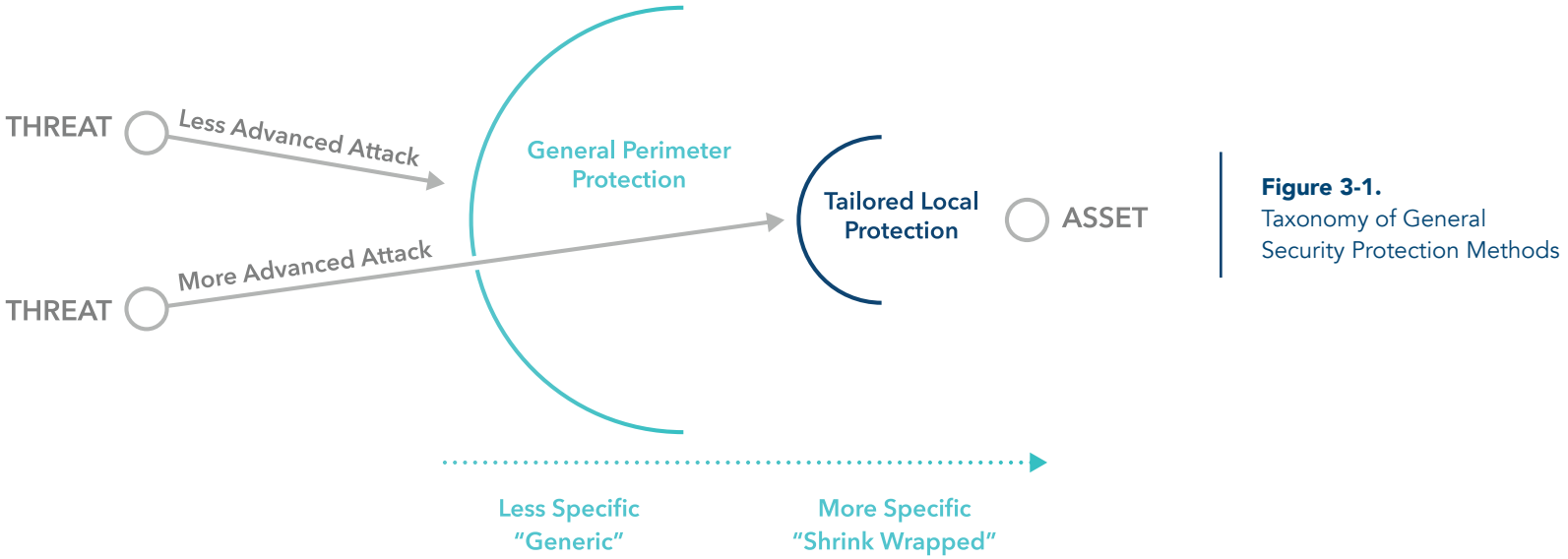


Figure 3-1.
Taxonomy of General Security Protection Methods

In-app VPN Support

With modern enterprise mobility comes a new functional requirement that arose directly from the day-to-day experiences of many business employees. The desired capability, known as in-app virtual private network (VPN), addresses the familiar clumsiness of having to establish a separate VPN session to access a protected application. Usually this separate VPN is created from outside a perimeter, through a gateway, into a firewall-enforced local area network.

Business users generally dislike the need to repeatedly create such a VPN session each time access is needed for apps such as viewing paycheck stubs, helping customers, updating financial records, and so on. A typical use-case involves an employee wanting to read a paycheck stub, and then having to authenticate a separate VPN session before being able to click on the app. Such a redundant process can contribute to user abandonment and low adoption rates. A better and faster experience would involve just clicking on an icon to view the paycheck.

In-app virtual private network (VPN), addresses the familiar clumsiness of having to establish a separate VPN session to access a protected application. Thus was born the idea that remote access to the app and the associated VPN-type controls could be combined into a seamless process. Such in-app VPN capability, if implemented properly, allows users to click once for secure access to an enterprise app using a mobile device resident on an untrusted network. This is also the concept that underlies the notion of device-to-cloud architecture that is so prevalent in enterprise hybrid cloud set-ups.

While in-app VPN is clearly designed to ease the burden of mobile end-users who need to access and use business apps from a variety of different network origins, the implications are more complex for security administrators. Since the control obligation is transferred from the perimeter edge and the VPN tunnel to the application execution environment, security obligations are transferred accordingly.

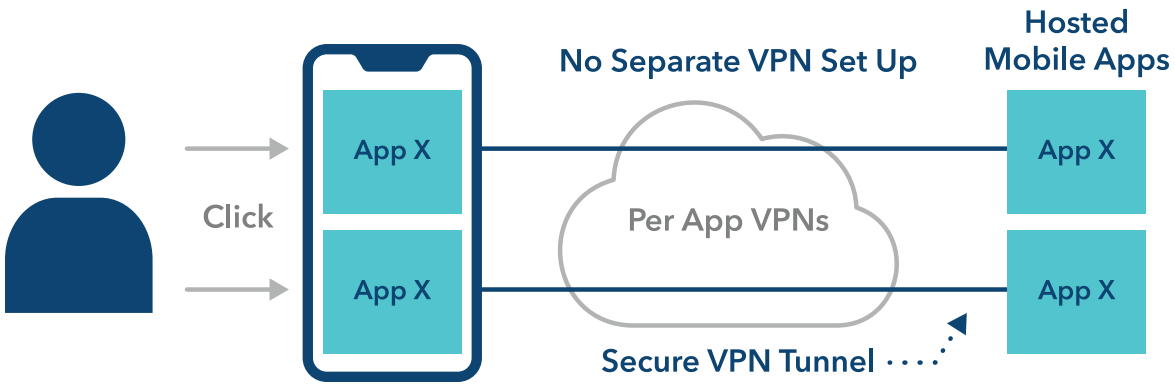


Figure 3-2.
Per-App VPN Approach

Mobile app security policies

The first step in establishing security controls adjacent to an enterprise app involves defining appropriate mobile app security policies. These would be developed using a combination of general policies that apply to all enterprise apps and more specific policies that apply only to the app being protected. Such fine-tuned tailoring is an important consideration in the development of modern micro-segmentation solutions.

Perhaps more important, however, is that by embedding security policy implementation into the invocation, access, use, and retirement of a user session with an enterprise mobile app, the user no longer must support policy controls such as two-factor authentication in the context of a VPN. Rather, these policies can become more intimate and adjacent to the app. This could allow, for example, less critical apps to require only one-factor authentication.

Typical general security policies that will exist for all mobile app usage in the enterprise include the following:

- **Authentication—**The proof factors, environmental context, and adaptive conditions required to validate the reported identity of anyone desiring access to a mobile app must be carefully defined for each organization. Increasingly, this implies the use of at least two strong proof factors.
- **Auditing—**Monitoring activity associated with the access and usage of a mobile app is an important consideration in the protection of any resources managed by that app. Privacy monitoring is a common subset, where the emphasis is on whether user information is transferred to the app developer or purveyor.
- **Access Control—**The rules on which access to the app is established will typically be driven either by the organizational position of some individual or by the contextual need based on the project or group to which an individual belongs. These general rules are often derived from non-mobile app access policies.

These general rules are typically complemented by more specific security policy directives associated with given mobile apps. For instance, if a business app is put in place to support some function, then the decision to allow—or not allow—collection of information such as location will depend on the business context. Logistics-based organizations, for example, might need location information, whereas most other groups would not.

Mobile app containers

A common functional solution to supporting specific security policies for mobile apps involves the creation of a local container into which the apps are hosted. The container provides a virtual execution environment in which the mobile app can reside, and within which—in theory—the security team can enforce desired policies. When done wrong, however, app containers might restrict the natural function of an app and lead to user abandonment.

The overall motivation for any type of containment is to create constructs known as micro-segments. The architectural notion is that by minimizing the size and scope of a given workload, app, virtual machine, or other computing entity, its protection can be shrink-wrapped to the local need, rather than for an entire enterprise. Code injection and other security methods also contribute to this more localized security protection for apps.

Most security experts today agree that containers and their supporting micro-segmentation architectures are effective security approaches to reduce the risk of modern threats. These same experts would also, however, uniformly agree that orchestration of containers and micro-segments is easier said than done. Many research and development efforts in cloud security today focus on this challenge of distributed orchestration [1].

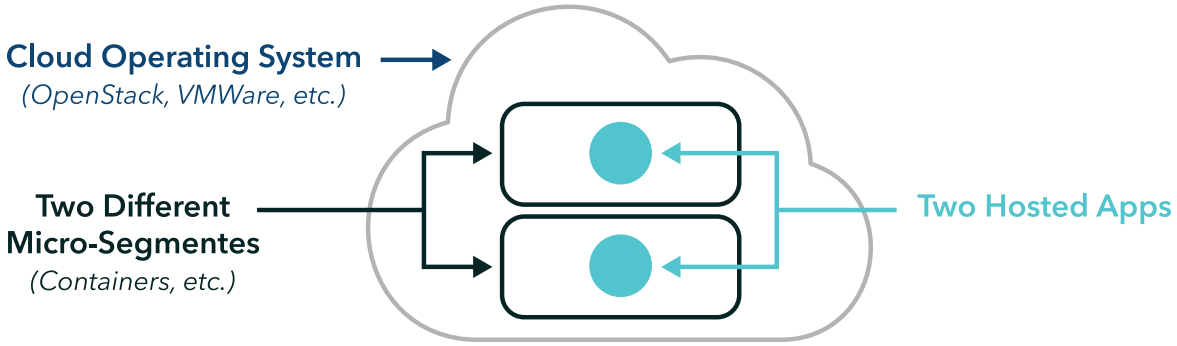


Figure 3-3.
Containerized Micro-Segmented App Hosting

Run-time application self protection

A popular technique for protecting any application involves the use of advanced algorithms to support desired security policies. These algorithms would typically be designed to operate in real-time alongside the application being protected, which implies that they would focus on behavioral aspects of the application. In most cases, the behavioral observation involves only input/output characteristics, but some methods might include internal workings.

An additional characteristic of this method is autonomy, generally referred to as self-protection. The goal is for the run-time security to possess the ability to contextually adapt to any changes in behavior with respect to expected profile, as well as shifts in the threat or consequence environment. The industry refers to this combined security capability for application as run-time application self-protection or RASP.

The primary security advantages of RASP are that—when the method works—the defining characteristics of run-time observation with autonomous self-protection can be convenient and powerful. The challenge, however, is that RASP solutions can be quite difficult to integrate into a range of different application environments. Teams using RASP often report considerable difficulties getting the application to work with the run-time controls.

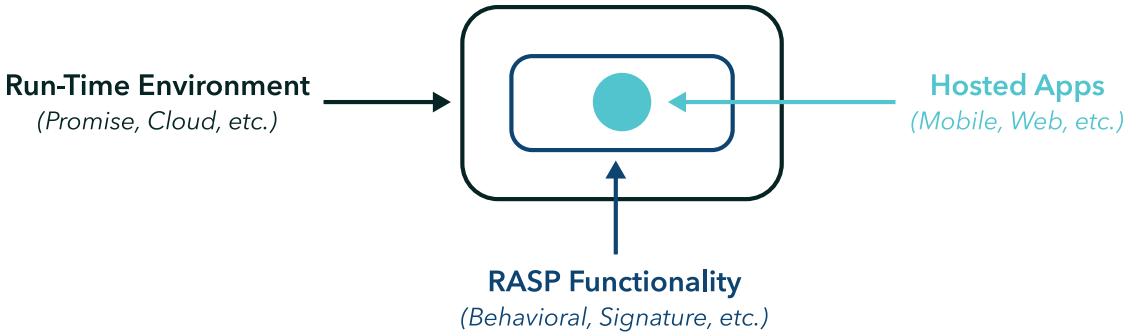


Figure 3-4.
RASP Approach

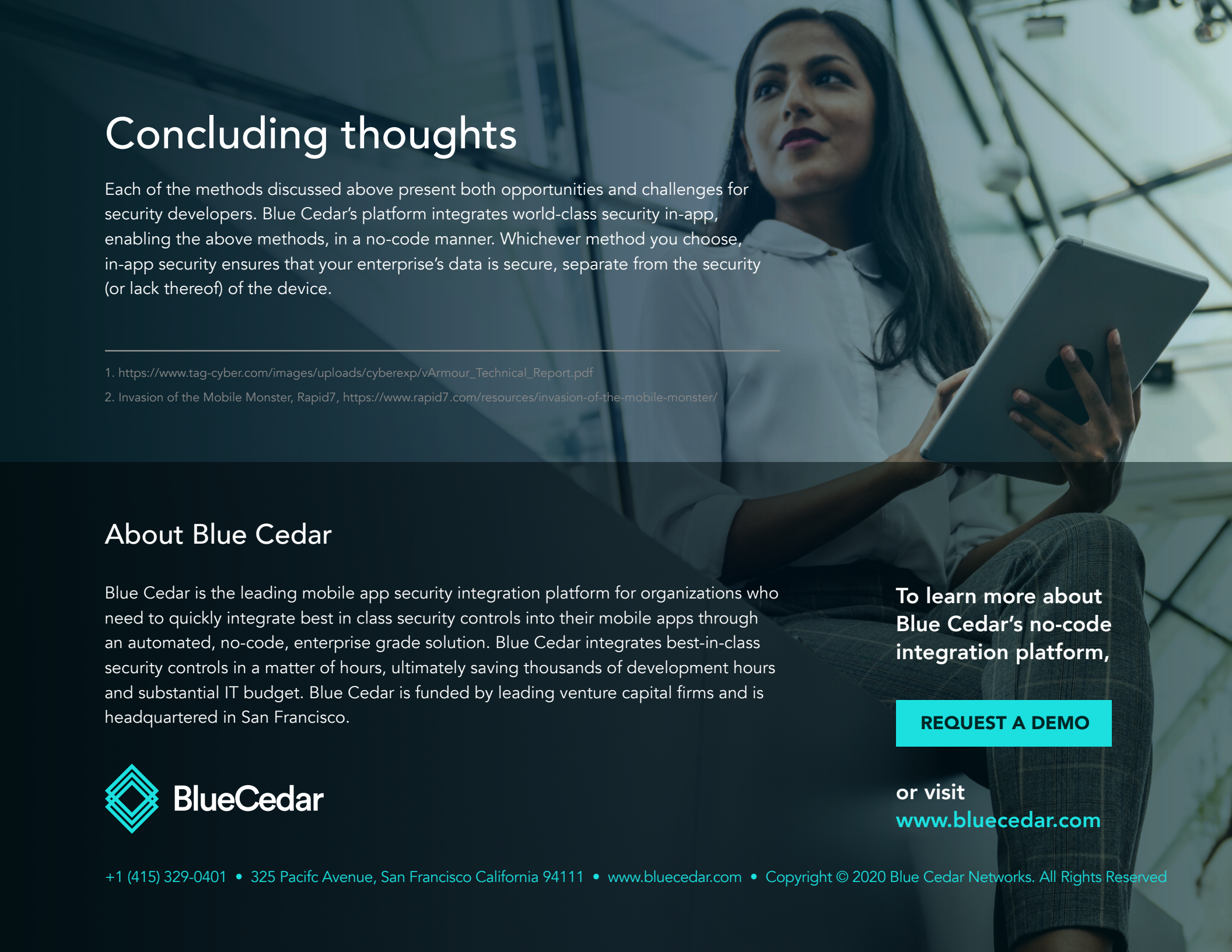
Code injection concept

A creative alternative to micro-containerized security in the run-time environment involves a technique known as code injection. Based on novel programming methods, the technique combines the best elements of the options discussed above. That is, it includes many of the run-time autonomous features of RASP, as well as the shrink-wrapped customization of micro-segmentation. But because it focuses on the software, it optimizes security to the specific app.

The advantages of code injection methods to enhance the cybersecurity of mobile apps are numerous and attractive to enterprise use-cases. By significantly increasing the ability for security teams to reduce the risk of mobile apps, security teams can address threat issues in the context of mobile-resident data being accessed by employees, partners, contractors, customers, and even Internet of Things (IoT) devices.

The inconvenience of trying to make mobile apps work in containers, which often implies changes to the code, often leads to insecure deployments. This is particularly troublesome when bring-your-own-device (BYOD) initiatives are in place. And by some recent estimates, BYOD usage already accounts for 80% of the 1.62 billion mobile devices currently being used in the modern workplace [2].

The code injection process is thus an attractive solution to supporting the type of mobile app security initiatives in an enterprise. This process begins with a home-grown or procured mobile app that is either native, web-based, or hybrid. The code injection is then performed to create embedded security. The result is enforcement of security policies for BYOD mobiles, customers, and IoT devices accessing enterprise data either on-premise or in the cloud.



Concluding thoughts

Each of the methods discussed above present both opportunities and challenges for security developers. Blue Cedar's platform integrates world-class security in-app, enabling the above methods, in a no-code manner. Whichever method you choose, in-app security ensures that your enterprise's data is secure, separate from the security (or lack thereof) of the device.

---

1. https://www.tag-cyber.com/images/uploads/cyberexp/vArmour_Technical_Report.pdf
2. Invasion of the Mobile Monster, Rapid7, https://www.rapid7.com/resources/invasion-of-the-mobile-monster/

About Blue Cedar

Blue Cedar is the leading mobile app security integration platform for organizations who need to quickly integrate best in class security controls into their mobile apps through an automated, no-code, enterprise grade solution. Blue Cedar integrates best-in-class security controls in a matter of hours, ultimately saving thousands of development hours and substantial IT budget. Blue Cedar is funded by leading venture capital firms and is headquartered in San Francisco.

BlueCedar

To learn more about Blue Cedar's no-code integration platform,

REQUEST A DEMO

or visit www.bluecedar.com

+1 (415) 329-0401 • 325 Pacifc Avenue, San Francisco California 94111 • www.bluecedar.com •