Xamarin.Android and the Perils of Platform Independence

PreEmptive Solutions

PreEmptive Solutions

Xamarin.Android and the Perils of Platform Independence

Who Should Read This Benchmark?

Xamarin development organizations with regulatory, compliance or other app and data security obligations.

A recently completed app security and risk management study that represented over 350 development organizations strongly suggested that Xamarin.Android developers are not adhering to the common practices and norms established by the vast majority of Android Java developers. Rather, their obfuscation, anti-tamper and anti-monitoring strategies are virtually indistinguishable from C# developers targeting desktop – *rather than mobile* – devices.

This anomaly – two native Android developer communities whose only obvious difference is their development tool of choice adopting markedly different security practices – raises a number of potentially material security and risk management questions.

1. **What is the underlying root cause?** Are equivalent Xamarin.Android controls unavailable or too complex or expensive to implement? Coming from a .NET tradition, do Xamarin.Android developers lack relevant platform-specific expertise to effectively assess their exposure?
2. **Are Xamarin.Android apps potentially less secure than Android Java apps?** If one assumes that Android app hardening practices have evolved in direct response to Android app threats and vulnerabilities, how significant are these gaps in app security practices?
3. **Will regulators, courts, or public markets punish Xamarin.Android app developers?** Recent history has shown that organizations that fail to take "reasonable" and/or

The Study

In November 2018, 350+ development organizations representing 55+ industries in 100+ countries shared their organizations' risk management priorities and mitigation strategies.

This benchmark contrasts the behaviors of

- 88 respondents developing Android apps using C# and Xamarin.Android,
- 112 respondents developing Android apps using Android Java, and
- 233 respondents developing .NET desktop apps using C#.

Using this benchmark

Organizations developing with Xamarin can compare their relevant compliance program elements, structure and processes to the practices reported here.

"An organization's failure to incorporate and follow application industry practice or the standard called for by any applicable governmental regulation weighs against a finding of an effective compliance and ethics program."

-U.S. Federal Sentencing Guidelines

For a more complete benchmark analysis, additional factors such as industry focus, organizational size, and jurisdiction should also be considered.

"recommended" security precautions should expect regulatory, civil, and even criminal penalties. Will Xamarin developers that forego security practices broadly embraced by the majority of Android developers be penalized in this way?

4. **How can Xamarin.Android developers assess for themselves and, as needed, mitigate these potential risks?**

Taken as a group, Xamarin.Android development teams behave almost identically to .NET Desktop development teams rather than, as one might expect, Java Android development teams.

Context: Android Application Security

The Android ecosystem is fast moving, open, and loosely managed. This "perfect storm" has given rise to impressive innovations and inventions – and, perhaps not surprisingly, an equal measure of vulnerabilities and bad actors ready to exploit the confusion that often precedes opportunity. Rooting Android devices (grabbing admin privileges), app reverse engineering and tampering and hijacking consumer transactions continues to be among the most lucrative and appealing of hacker targets.

This is why the organizations, regulations and frameworks like the PCI Council, NIST, OWASP and (indirectly) HIPAA, GDPR and others are calling out Android-specific vulnerabilities and recommending appropriate precautions and controls.

Controls are typically organized into those that "prevent" incidents, "detect" when preventative controls fail and incidents occur, and "defend" or "respond" to those occurrences. An application incident or exploit can be generic such as "reverse

Evidence

Figure 1a graphs the percentage of developers who indicated that they have implemented controls to:

- "Prevent" unauthorized debugging or monitoring,
- "Detect" when the above mentioned preventative controls fail, and to
- "Respond" or "defend" in real-time when incidents occur.

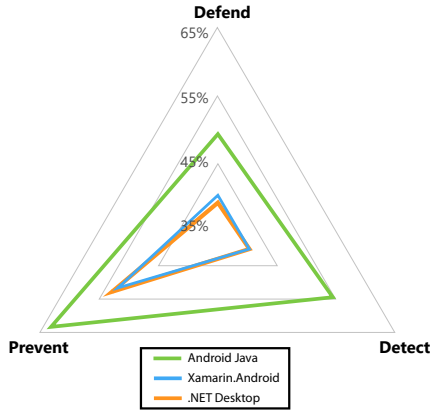

Figure 1a: Unauthorized monitoring or debugging controls by development platform

Figure 1b highlights the different practices and patterns between Xamarin Android and Android Java development groups.

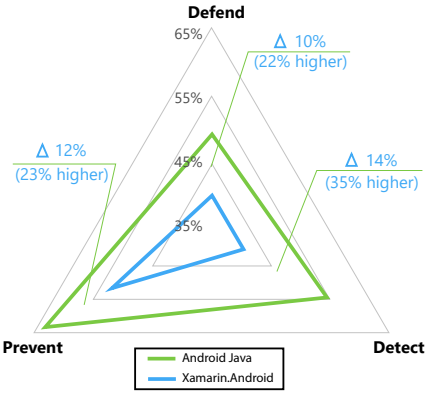

Figure 1b: Difference in adoption of anti-debugger controls.

Android Java adoption of anti-debugger controls is between 23% and 35% higher than Xamarin Android adoption. Is this justified? Dangerous?

engineer" to view source code or platform specific like rooted-device (an Android device state where apps can have administrator privileges allowing unfettered access to memory, storage, and the network).

The larger message here is not Xamarin-specific or Android-specific or even mobile-specific. Application requirements must account for platform-specific vulnerabilities, corresponding mitigation practices, and technology trends. Cross-platform and/or runtime-independent development platforms do not relieve developer organizations of this obligation.

Assess and Mitigate (Xamarin) Platform Risk

There is little controversy surrounding the growing need for application hardening controls such as anti-tamper or and anti-root. These controls are becoming increasingly common. As such, Organizations would be well advised to establish a written policy regarding if, how, and under what conditions these controls should be considered.

Given the speed at which security and compliance topics are evolving, benchmarking development practices against peers, suppliers and customers seems especially prudent – regardless of what technology stack may be in use.

Figure 2 illustrates the representative steps that development organizations can take to assess relative risks and to put into play any implementation changes that may be necessary to bring app and data controls in line with industry and supply-chain norms.

Figure 2 illustrates how developers can ensure that their app and data controls are in line with industry and platform norms.

Assessment and compliance workflow

1. Complete an industry benchmark to provide an objective measure of common and well-understood practices.
2. If there are gaps, identify the root causes, e.g. no prior requirements set, potential controls deemed to be too complex or expensive, etc.
3. If controls are identified that can more effectively mitigate risks, establish appropriate policies supported by tools and resources necessary to comply.
4. Communicate policies and a timeline for adoption.
5. Measure and ensure progress.
6. Incorporate policies into broader compliance programs and ensure policies remain up to date.

Assessment and Compliance Resources

The following resources have proven helpful to PreEmptive Solutions clients and are recommended for development teams looking to kick start this process.

Resources are both general purpose and Xamarin-specific.

Assessment and Compliance Task	Supporting Resource(s)
Task 1	Complete an industry benchmark • Application Security Benchmark
Task 2-3	Identify the root causes underlying any gap in Android security best practices. Establish appropriate policies supported by tools and resources necessary to comply. Links to: • Xamarin Apps Get Second Major Security Boost With Dotfuscator Pro 4.39 • Managing Risk in an Application Economy • Xamarin and Dotfuscator: Do you believe in magic? • An app hardening use case: Filling the PCI prescription for preventing privilege escalation in mobile apps • PreEmptive Protection Dotfuscator Implementation Project Plan for Xamarin • PreEmptive Protection Implementation Project Plan

These resources can help organizations build the case to more effectively set priorities and build more effective application security and risk management controls.

Conclusion (getting started)

Please do not hesitate to contact PreEmptive Solutions at support@preemptive.com (existing clients) or sales@preemptive.com to review your requirements and to map your organization's unique risk profile to scalable, reliable, and up-to-date app hardening technologies.

Worldwide Headquarters
767 Beta Drive, Suite A
Mayfield Village, OH 44143
Phone +1 440.443.7200
solutions@preemptive.com

European Headquarters
140 bis rue de Rennes
75006 Paris, France
Phone +33 01.83.64.34.74
eurosolutions@preemptive.com