

Not So Great Expectations

Why Application Markets Haven't Failed Security

Application markets have rapidly become a widely popular mechanism for expanding the features and utility of mobile devices such as cell phones. The cottage industries that sprung up around these markets serve millions of

Application Markets

Publishing an application to a market begins with a fee-based developer registration. This serves at least two purposes: it provides a semi-verifiable developer name for market listings, and it creates a barrier to entry (albeit small). In the latter case, the developer fee (US\$99 yearly for Apple and US\$25 one-time for Google) forces a malicious developer to at least go to the effort of acquiring a fake credit card. When the developer is ready to publish an application, he or she uploads the binary package using the market's Web interface. At this time, the developer also sets a price and assigns a category (entertainment, financial, and so on) to present the application in the market interface.

Apple and Google have a dichotomy in vetting processes for applications uploaded to their markets. Whereas the details of Apple's application review are tightly veiled, its response to the US Federal Communication Commission's July 2009 inquiry sheds some light:

Submitted applications undergo a rigorous review process that tests for vulnerabilities such as software bugs, instability on the iPhone platform, and the use of unauthorized protocols. Applications are also reviewed to try to prevent privacy issues, safeguard children from exposure to inappropriate content, and avoid applications

PATRICK
MCDANIEL AND
WILLIAM ENCK
*Pennsylvania
State
University*

applications daily to a ready user audience. Markets entice developers by placing low economic and technical barriers to entry, thereby fostering fast-paced innovation. They streamline purchase and installation to serve even the most casual users with ease. Simply put, markets make producing and consuming applications easy.

Markets also present obvious security concerns—users are trained to download applications with impunity from a huge number of developers about which they know little. Moreover, these applications often request nearly unfettered access to the data and device interfaces (for example, texting, voice-dialing, or GPS location), which seems to invite malicious applications and questionable functionality. Not surprisingly, such fears have been substantiated. A recent discovery of numerous applications sharing GPS locations and other personal information with online advertisers is just one example of dubious features found in market applications. The public reaction to these stories is often the same: users and pundits decry markets

for their failure to properly vet the applications or developers. This underscores the widely held expectation that security is the market's responsibility.

We argue that application markets have not failed security; rather, the failure is our expectation that they will do so. Markets don't claim to provide security. In fact, making software secure is fundamentally orthogonal to the distribution method (market or otherwise). Users expect security because they fundamentally don't understand how markets work and how security is achieved. We discuss in depth why this is so and ponder where it leaves us in the future, positing several areas where security enhancements can be incorporated. We begin here by discussing how markets work. For the purpose of discussion, we restrict ourselves to the two dominant markets, Apple's App Store for the iPhone and other iOS devices and Google's Android Market for Android phones and devices. However, most observations we make throughout apply equally to any application market, regardless of the vendor.

that degrade the core experience of the iPhone.¹

Apple's response also indicates that at least two reviewers examine each application and that 95 percent of applications are approved within 14 days. In contrast, Google's Android Market has no formal review process. Instead, it relies on Android's security permission system and user judgment. However, Google will promptly take down applications that don't comply with its content policy (see www.android.com/market/terms/developer-content-policy.html), which covers intellectual property, offensive material, and privacy violations.

Neither vetting process provides much security. Both Apple and Google's terms of service essentially say, "Thou shall not submit malware," and both have reactive security procedures—that is, when an insecure application slips into the market (and it has for both), it's quickly removed on discovery to minimize impact. In fact, both Apple and Google have a "kill switch" to remotely remove applications downloaded from their markets. Google recently publicly acknowledged using the kill switch on an application in its market.

App Market Limits

Often, when someone discovers a security problem in a market-delivered application, the public raises the same question: Why didn't the market check to make sure the application was secure, or ensure that it at least wasn't malware? The answer is unsatisfying—because it couldn't. The reasons the market can't test for security or malicious behavior are as complex and multifaceted as security itself.

Accept that security is contextual and individualized. Each user has a set of expectations about what an application should and

could do. Should the application be able to send an SMS message to people in the address book? For some, the answer is yes; for others, it's no. The definition of security (and, implicitly, the behaviors to look for during analysis) isn't fixed enough to articulate in any coherent way. Thus, markets don't even have an identifiable point of departure to begin application analysis.

Suppose for a moment that the market could find some acceptable definition for security that satisfied at least some consumers. What then? Identifying which behaviors software can exhibit at runtime is one of the great open challenges in computer security. The software industry has vastly improved testing and software development processes but has no tools to discover what an application will do once it's installed. It's further likely that proving anything nontrivial about how an application will behave at runtime is likely impossible (that is, undecidable).

We might be tempted, as Android has, to simply let users define what rights an application has—and thus limit risk to what each user finds acceptable. Rights assignment is a blunt instrument that makes the user trade off usability with security. The meaning of rights and trade-offs in their assignment are most often alien to users, rendering the rights-acceptance process of no impact for unsophisticated or careless users.

Suppose further that somehow we could identify a means to cer-

tain simple analysis to each application is logistically impossible. Simply put, the volume of applications prevents markets from doing anything more than simple automated tests. Even then, interpreting the results of automated certification requires substantial effort.

More fundamental questions are at issue: Should application markets make security decisions for consumers? Do they possess the right knowledge or incentives to properly weigh utility and risk? Do they even want that responsibility? The answer to all these questions is—in all likelihood—a qualified no. Just as the grocery store can't guarantee the quality and safety of the food it sells, markets can't provide the security consumers desire.

Moving Forward

Despite these limitations, all hope is not lost. Partial certification and monitoring the hygiene of an application market can improve overall security. However, these techniques must be carefully incorporated to provide net value-add.

Automating certification tests can handle the deluge of applications developers submit. Existing tools have been effectively considered at various levels of abstraction, including configuration settings, binaries, and source code. While automated certification can't practically address all security concerns, it raises the bar for what is considered acceptable. However, such market-level certification requires a common

Markets and users must resist embracing the green light fallacy—the desire for vendors to produce a “green light” seal of security that does nothing more than make users feel secure.

tify an application in an effective way. Developers submit thousands of applications to markets each month. Applying anything more

definition for security. Agreeing on this definition is difficult because, in many cases, one person's privacy concern is another's

IEEE  computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field. Visit our website at www.computer.org.

OMBUDSMAN: Email help@computer.org.

Next Board Meeting: 15–16 Nov. 2010, New Brunswick, NJ, USA

EXECUTIVE COMMITTEE

President: James D. Isaak*

President-Elect: Sorel Reisman; **Past President:**

Susan K. (Kathy) Land, CSDP;* **VP, Standards**

Activities: Roger U. Fujii (1st VP);* **Secretary:**

Jeffrey M. Voas (2nd VP);* **VP, Educational**

Activities: Elizabeth L. Burd;* **VP, Member &**

Geographic Activities: Sattupathu V. Sankaran;† **VP,**

Publications: David Alan Grier;* **VP, Professional**

Activities: James W. Moore;* **VP, Technical &**

Conference Activities: John W. Walz;* **Treasurer:**

Frank E. Ferrante;* **2010–2011 IEEE Division V**

Director: Michael R. Williams;† **2009–2010 IEEE**

Division VIII Director: Stephen L. Diamond;† **2010**

IEEE Division VIII Director-Elect: Susan K. (Kathy) Land,

CSDP;* **Computer Editor in Chief:** Carl K. Chang†

*voting member of the Board of Governors †nonvoting member

BOARD OF GOVERNORS

Term Expiring 2010: Piere Bourque; André Ivanov;

Phillip A. Laplante; Itaru Mimura; Jon G. Rokne;

Christina M. Schober; Ann E.K. Sobel

Term Expiring 2011: Elisa Bertino, George V.

Cybenko, Ann DeMarle, David S. Ebert, David A.

Grier, Hironori Kasahara, Steven L. Tanimoto

Term Expiring 2012: Elizabeth L. Burd, Thomas

M. Conte, Frank E. Ferrante, Jean-Luc Gaudiot, Luis

Kun, James W. Moore, John W. Walz

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate**

Executive Director, Director, Governance: Anne

Marie Kelly; **Director, Finance & Accounting:**

John Miller; **Director, Information Technology**

& Services: Ray Kahn; **Director, Membership**

Development: Violet S. Doan; **Director, Products**

& Services: Evan Butterfield; **Director, Sales &**

Marketing: Dick Price

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700,

Washington, D.C. 20036

Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614

Email: hq.ofc@computer.org

Los Alamitos: 10662 Los Vaqueros Circle, Los

Alamitos, CA 90720-1314 • **Phone:** +1 714 821

8380 • **Email:** help@computer.org

Membership & Publication Orders

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 •

Email: help@computer.org

Asia/Pacific: Watanabe Building, 1-4-2 Minami-

Aoyama, Minato-ku, Tokyo 107-0062, Japan •

Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553 •

Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Pedro A. Ray; **President-Elect:** Moshe

Kam; **Past President:** John R. Vig; **Secretary:**

David G. Green; **Treasurer:** Peter W. Staecker;

President, Standards Association Board of

Governors: W. Charlston Adams; **VP, Educational**

Activities: Tariq S. Durrani; **VP, Membership**

& Geographic Activities: Barry L. Shoop; **VP,**

Publication Services & Products: Jon G. Rokne;

VP, Technical Activities: Roger D. Pollard; **IEEE**

Division V Director: Michael R. Williams; **IEEE**

Division VIII Director: Stephen L. Diamond;

President, IEEE-USA: Evelyn H. Hirt

revised 10 Sept. 2010



desired feature. A potential solution is to push the ultimate decision policy to the device, letting users choose from “paranoid” and “normal” policies.

Continually monitoring and evaluating market hygiene benefits all users. Apple and Google have removed dangerous applications from their markets and even remotely from phones. Third-party projects such as the App Genome project (mylookout.com) and Whatapp (whatapp.org) provide expert security ratings and privacy reviews of applications. Such services fill a valuable void. Currently, the process of identifying bad applications is largely ad hoc. Dynamic analysis techniques such as taint tracking can make this process easier and identify more bad applications faster.

Finally, any market-level security fixes must still give users some level of control. User control is Salter and Schroeder’s seminal principle of psychological acceptability applied to phones. Users purchase phones and therefore feel entitled to administrate them. Removing users from the equation only encourages poor security practices such as phone “jail-breaking.” We’ve already seen the iKee.B iPhone botnet exploit a vulnerability present only in jailbroken iPhones.

Markets aren’t in the business of security, nor can they be. We must focus on developing platforms that prevent or detect malicious behavior, educating users, vetting developers, and reducing applications’ ability to abuse users, networks, and data. Of course, we’ve held the same goals for general system security for decades with, at best, mixed results. Recent movements within the software and security communities, such as the App Genome project, offer hope, but much more effort is needed.

Finally, markets and users must resist embracing the green light fallacy—the desire for vendors to produce a “green light” seal of security that does nothing more than make users feel secure. Such security theater works to undermine security not only by propagating falsehoods but also by implicitly removing the consumer’s responsibility to be vigilant. Users must accept that markets can provide little more than as-is guarantees about the applications they support. Only informed and cautious consumers can avoid the pitfalls of bad applications. Thus, only by changing user expectations can we hope to combat malware in the new software world. □

Reference

1. “Apple Answers the FCC’s Questions,” Apple, 2009; www.apple.com/hotnews/apple-answers-fcc-questions/.

Patrick McDaniel is an associate professor in the Department of Computer Science and Engineering at Pennsylvania State University and codirector of the Systems and Internet Infrastructure Security Laboratory. His research interests include network and systems security, telecommunications security, and policy. He’s a member of IEEE, the ACM, and Usenix. Contact him at mcdaniel@cse.psu.edu.

William Enck is a doctoral candidate in the Systems and Internet Infrastructure Security Laboratory in the Department of Computer Science and Engineering at Pennsylvania State University. His research interests include operating systems security, telecommunications security, and systems and network security. He’s a member of IEEE, the ACM, and Usenix. Contact him at enck@cse.psu.edu.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.