Hewlett Packard Enterprise

Business white paper

Designing a defense for mobile applications

Examining an ecosystem of risk

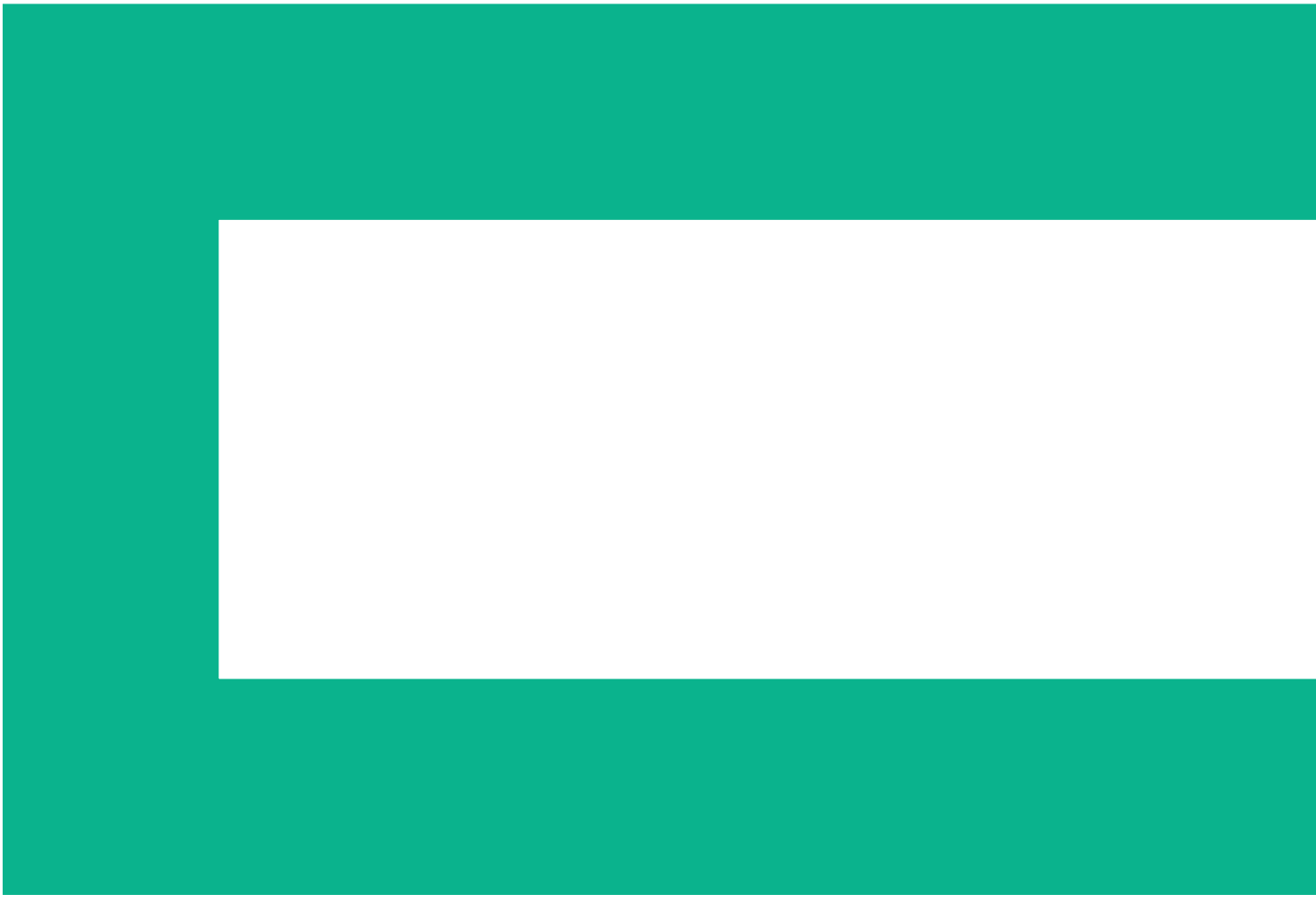


Table of contents

It wasn't long ago that being on the cutting edge of business meant having a website where customers could purchase your products, but that quickly changed. Soon, it wasn't enough just to have an e-commerce presence; you had to offer an interactive and engaging customer experience (see Web 2.0).

Now the standard has shifted once again; in order to truly compete in the modern economy, your business needs to have a mobile storefront on smart phones and tablets. Without this mobile presence, you'll lose business to competitors who have them.

With this new challenge comes high risk as well as great reward. With mobility, the client side of your applications is more important than ever. Instead of accessing your website from the safe confines of work or home, your customers can now perform sensitive transactions anywhere. As a result, mobile applications, and the devices and data they interact with, can now be more easily attacked.

Attackers will probe all components of your application for vulnerabilities, and the only question is whether you will find them first. This paper highlights the various considerations for defending mobile applications—from the mobile application architecture itself to the myriad testing technologies needed to properly assess mobile applications risk.

Protecting mobile applications

Mobility by the numbers

How big is the mobile market? Massive. A study by Arc Worldwide recently showed that half of Americans already use a mobile device to shop.[1] According to Smart Insights, mobile device usage tripled for the third year in a row in 2010, with global mobile data traffic forecast to increase 26 percent by 2015.[2] Finally, many experts predict mobile Internet usage will eventually overtake traditional usage in 2015.

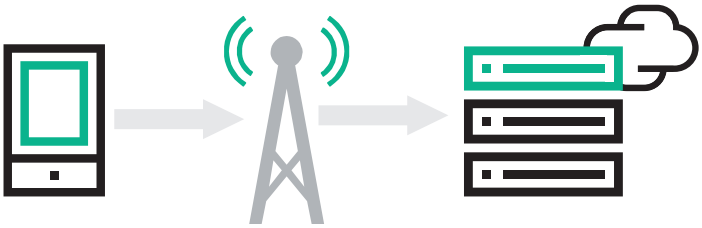
Looking closer, these numbers aren't that surprising. Just recently, we have only been able to use the Internet while at home or at work. Today, in contrast, access is being brought to more and more people, but without the restriction of a physical location. Give everyone a mobile device, and suddenly we can casually browse the web, make purchases, or conduct business wherever we are. That's a lot of mobile traffic; and for attackers looking to take advantage, that's a lot of attack surface.

[1] leolens.leoburnett.com/index.php/2011/03/marketing-to-the-mobile-shopper/

[2] smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-usage-statistics-2010-2015/

Mapping the mobile security ecosystem

The first step is to fully understand what needs to be defended. For example, to secure a castle, one needs to not only understand the exterior walls, but also the internal corridors, passage ways, and even the roads leading to and from the fortress. It is the same with a complex ecosystem like a mobile application. Mobile application security is not just about the application located on the device; it is also about the network traffic going between that application and its back-end server, as well as the server components themselves.



Securing the whole mobile stack: from the mobile device to the network to the server

Many organizations and security companies make the mistake of securing and testing only one or two of these components. However, to properly assess a mobile application, you need to test it in its entirety. This means not only reviewing these components from the outside while the application is running (dynamic testing), but also at the code level (static testing).

Consider the following components of a mobile application and the different testing options:

Client
- Dynamic (client-side fuzzing)
- Static (binary and code review, filesystem/memory analysis)

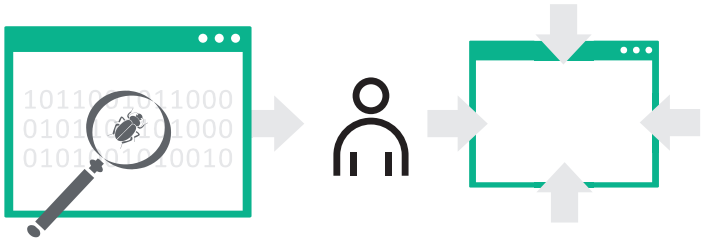
Network
- Dynamic (traffic analysis)

Server
- Dynamic (full dynamic testing of back end)
- Static (full code review of back-end codebase)

Any mobile testing approach that doesn't involve testing the fully running application in its production-like configuration, with all its various components interacting with each other, is an exercise in theory rather than real-world security testing. It is well-known among both attackers and experienced security professionals that applications can be more than the sum of their parts when they are connected together and placed in a running configuration. Attackers count on this type of emergence-like behavior being outside both the understanding and testing purview of the security teams supporting those applications.

Attackers assault your full running application from end to end; as a result, it is critical that you test applications in this fashion also. While essential, dynamic testing is not enough. It's important to combine this external, real-world perspective with an internal view of an application's structure. Many companies make the mistake of seeing these two approaches as independent and separate.

As application testing evolves, it will become increasingly obvious that this is a flawed way of addressing the problem because it compartmentalizes a process that benefits from cooperation. Dynamic findings should point to code, and code findings should reveal dynamic, real-time attack vectors. Testing one without the other should be considered significantly constrained.



Static testing analyzes the source code for vulnerabilities.

Dynamic testing simulates an attack against a running application.

Dynamic testing is conducted against an application that is running. An example is testing a website while it's operational. This type of approach allows testing techniques to interact with the application from an end-to-end perspective.

Static testing is conducted against an application while it is at rest, or not operating. The most common example of this is source code analysis, where an application's code is tested. This can be done whether or not the application is running.

Anatomy of a mobile attack

Device-based attacks

Attacks against devices are the easiest, most intuitive to understand. The most obvious is the stolen laptop scenario, whereby the hardware is stolen, so that an attacker can connect to the system and pull data off. Common vulnerabilities to this kind of attack include unencrypted credentials and cached sensitive data.

Additionally, it is advisable to test the mobile application on how it handles inputs. Mobile applications have significant issues with parsing user-provided input, such as URLs and other content found in text messages, MMS messages, and emails.

Another type of threat against the device itself is the installation of malware on the system that can lead to information leakage and even complete compromise. Attackers commonly install malicious certificates, reconfigure proxy settings, and perform other modifications that allow man-in-the-middle (MiTM) visibility into user transactions. This type of attack is common on the Android platform, where legitimate and benign applications are modified to include a covert channel that can send attackers a steady stream of activity coming from the victim's phone.

It's important to note that automated tools—even those that scan code—can often miss covert channel code within applications because if malware is expertly written, the functionality will look quite normal to a scanner. This is why both manual and automated review of code, combined with dynamic analysis, is so crucial.

As an example, if an automated scanner alone were used to test a given application containing data smuggling malware, it could very well miss the issue entirely. Alternatively, a manual code review would likely find that a sweep of local content was being performed, followed by a connection outbound to a location other than the primary server destination. This outbound connection could also be detected by running the application in its normal environment—for example, installing and running the application on a test device in a lab setting.

Network-based attacks

One of the most dangerous vectors for mobile applications remains the network component, specifically the capture and/or modification of mobile network traffic by attackers. This is especially dangerous in public places where Wi-Fi is used, since many mobile applications switch to using Wi-Fi when it is available. At that point, widely-available tools like Firesheep can be used to pull sensitive information right out of the air.

Credential-stealing attacks are possible because so few applications properly secure the sensitive data they use. Mistakes in this realm come in many forms, from failing to securely manage TLS/SSL certificates and their associated interactions to failing to use encryption at all (see Firesheep). Then there is the favorite web application vulnerability of the last few years: encrypting only the login and then switching to cleartext. Unfortunately, mobile application developers have to yet learn to avoid this pitfall; mobile applications continue to repeat the same mistakes as web applications.

Analyzing network communication used by an application is done most effectively by combining static and dynamic techniques. The code can transparently reveal the protocols being used to communicate over the network, and dynamically testing the running application in the lab can confirm this behavior.

Server-based attacks

With all this discussion of mobile and its associated attack surfaces, we might forget that mobile applications send their data off to a server somewhere. The vast majority of mobile apps are interacting with websites on the back end, and many of them employ web services. This means old favorite attack methods like SQL Injection, cross-site scripting, and cross-site request forgery step squarely back onto the stage.

That's the interesting thing about the mobile security landscape and why it's so instructive to perceive applications as ecosystems; it's not just about the device getting malware, passwords being sent in cleartext over the network, or someone MiTM-ing sensitive transactions. It's also about the web infrastructure that hosts the application on the server side. This means performing a web application assessment on the server side of the mobile application.

Today, the web assessment of mobile applications may appear uninteresting and obvious because they are so well-understood. Unfortunately, this assumption is incorrect for two reasons.

- First, web vulnerabilities show no signs of slowing down even with dedicated web teams.
- Second, the development of mobile applications is often done separately from an organization's web application development. Separate organizations tend to mean different coding standards and security practices.

Imagine a world in which most organizations already struggle with web application security within their primary, high-profile web properties, due to lack of developer exposure to application security concepts. Then realize that this company's mobile applications are likely being developed by a separate group of developers with even less exposure to, and knowledge of, core application security concepts. This means significant vulnerabilities are still being found on the server side of mobile applications.

A path forward for mobility testing

With mobile computing set to overtake traditional forms as soon as 2015, the mobile space is exciting—and for good reason. But the challenges to security in this new environment are every bit as real as the opportunities. Between the device, the network, and the server components, attackers have plenty of surface areas to choose from, and if you want to deploy mobile applications securely, you need to take a multi-faceted approach to their defense.

Complete testing of mobile applications involves more than just testing a binary, scanning a codebase, or even scanning the back end. Multiple skill sets and perspectives into the application are required, as well as an end-to-end testing philosophy that sees an application as more than a collection of disparate components.

Here are some prescriptive takeaways that can help ensure your mobile applications are being tested properly, regardless of who's performing that testing:

- Ensure that your applications are being tested dynamically in a full, running configuration. Remember that this is how attackers will test an attack on them, so it's how you should test your defense.
- Test all three tiers of the mobile stack, since most mobile applications are multi-tiered. With client, network, and server components, thorough testing should occur at all three of those tiers.
- Use both dynamic and static testing approaches for your mobile applications, as they are complementary rather than mutually exclusive.

- Supplement automated testing with manual testing for both dynamic and static testing. Automation has advantages over human testing, but should be considered a supplement rather than a replacement.
- Be skeptical of any proposed mobile security solution, whether internally or externally provided, that doesn't embrace a comprehensive, multi-faceted approach to testing.

Hewlett Packard Enterprise provides a wide range of application security services, including mobile testing offerings that embody the end-to-end testing philosophy discussed above. For more information on these testing options, or to discuss any existing application security challenges, please contact us at (888) 415-2778.

About HPE Enterprise Security

HPE is a leading provider of security and compliance solutions for the modern enterprise that wants to mitigate risk in their hybrid environment and defend against advanced threats. Based on market leading products from HPE ArcSight, HPE Fortify, and HPE TippingPoint, the HPE Security Intelligence and Risk Management (SIRM) Platform uniquely delivers the advanced correlation, application protection, and network defenses to protect today's hybrid IT infrastructure from sophisticated cyber threats. Learn more at hpenterprisesecurity.com.

Sign up for updates

★ Rate this document

Hewlett Packard Enterprise

4AA4-1802ENN, November 2015, Rev. 1