Software Security Goes Mobile

Abstract

As mobile devices continue to take a more mainstream role in our everyday life, the risk associated with vulnerabilities in the software that runs them is rapidly changing. This paper will explore the role that mobile devices are starting to take in our lives, discuss who is enabling that change, and then examine the new risk landscape that is created.

Introduction

Over the last few years the use of mobile devices with advanced computational powers and Internet capabilities have rapidly moved into mainstream use. Adoption of these devices is changing user behavior and the ecosystem that has developed has complicated the accountability model for secure development.

Adoption Rates

Migrating from the early days of dedicated enterprise use (with devices such as RIM's BlackBerry) smart phones have seen widespread adoption among the greater population. According to the "comScore 2010 Mobile Year in Review"[1] report 27% of U.S. mobile phone users own smart phones, up 10.2% from the previous year. A similar percentage and trend is reported for Europe and these numbers are only going to continue to grow. According to Morgan Stanley Research the number of smart phones shipped is estimated to outnumber 'feature phones' for the first time in 2011.[2] This high rate of market adoption is combined with a changing usage pattern among consumers.

Changing Behavior

Increasingly, smart phones are used as companion devices while consumers engage in other activities. The "Mobile Movement Study"[3] by Google shows that 89% of smart phone users use the devices for activities other than making a phone call on a daily basis. This type of activity includes browsing and searching the Internet, watching videos, or using a purpose-built application. Increasingly studies are showing that use behavior on mobile devices is starting to mirror traditional Internet usage on desktop computers.[4]

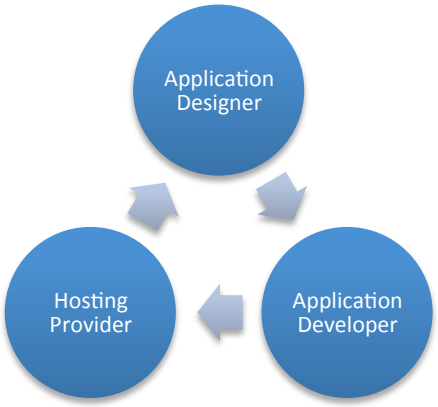
Corporations have not overlooked these trends; increasingly purpose-built mobile applications are being used to extend a traditional web presence. These applications not only provide traditional functionality such as e-commerce, banking, and email but increasingly the unique capabilities of mobile devices –such as location, and imaging technology– are also being incorporated allowing for offerings that were previously infeasible.

---

[1]http://www.comscore.com/Press_Events/Presentations_Whitepapers/2011/2010_Mobile_Year_in_Revie
[2]http://www.morganstanley.com/institutional/techresearch/pdfs/MS_Internet_Trends_060710.pdf
[3] http://www.thinkwithgoogle.com/insights/library/studies/the-mobile-movement/
[4]http://www.morganstanley.com/institutional/techresearch/pdfs/MS_Internet_Trends_060710.pdf

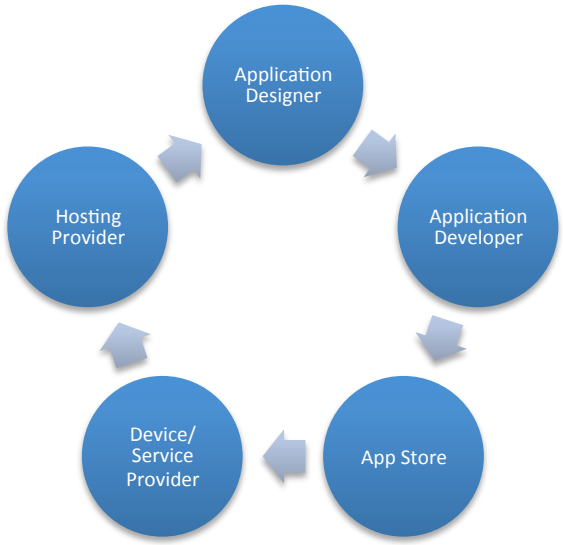
Evolving Model of Accountability

The emerging ecosystem of mobile devices is a very different world than that of traditional desktop computing and web-based applications and services. In the mobile ecosystem, there are far more stakeholders involved in the procurement, provisioning, and use of devices than we have previously seen.



For a traditional web site, one could imagine a complicated scenario with three core stakeholders: the application development group, the hosting provider, and the actual company that needs the website. While in some cases these three stakeholders are the same, it is easy to find situations where they are three separate entities. In this situation the accountability for the security of the website is difficult enough. Is it the responsibility of the company designing and outsourcing the development? The responsibility of the development group? Or the responsibility of the hosting provider? Most large organizations today are just starting to come to terms with how this responsibility is shared in this type of model.

However, the mobile eco-system is significantly more complicated. Imagine the scenario of an application introduced for Apple iOS, which drives the iPhone. First, a company designs and creates a specification for the application. As this is a new technology there likely is not an in-house development group capable of creating the application so an outsourcer is engaged. Upon completion of development, the application is then published to the Apple-run App Store. A consumer can then log into the Apple AppStore through their device – provided by a wireless carrier, such as AT&T or Verizon—and can download the app. The application likely then communicates with a hosted service to provide data for the user's interaction. In this scenario we now have five different stakeholders: the development group, the company that designed the application, the application provider (the App Store), the device/service provider, and the hosting provider of the backend services the application accesses.



Accountability for security has become significantly more complicated, and the expectation of accountability changes depending on the perspective. It is reasonable to expect that consumers may hold the wireless provider, the app store, and the sponsoring company accountable for the security of their device and the secure use of the application. However, with so many parties involved a clear model of accountability has yet to emerge.

Mobile Landscape

From a purely technical point of view, it is important to clarify exactly what is encompassed by a discussion of 'mobile' security. Smart phones and the applications that make them 'smart' have a number of components. Some of these components are similar to elements we have seen in the past and some of them are completely unique. In our discussion we are going to include the four major elements that comprise a 'mobile' system: a client device, the operating system that runs on the device, the constant data connection, and the backend server providing data to the purpose-built applications on the device.

Familiar Pieces

The most reassuring aspect of mobile software security are the components borrowed from traditional websites. Most mobile applications employ a traditional client-server model. A relatively small amount of business logic is executed on the client side device, while the majority of work is done on a backend server. In addition the majority of long-term data persistence occurs on the server. For most applications, the implications of this on security are quite significant. As an attacker could easily bypass the client side business logic it must be assumed that it cannot be relied upon for any security controls or secure data persistence. This means that the majority of the attack surface of an application is still the backend server. Focusing on this traditional component of the mobile application will provide the most benefit for securing the application. A benefit of this client-server model is the ability for re-use of the backend server's API. A large number of mobile applications are simply native clients leveraging an existing backend server.

Additionally, the backend server could be, and often is, the same server providing the functionality to traditional browser-based applications. As a result the technology used to create these backend servers is the same as the technology that was used to build our existing web applications. This again provides a benefit when it comes to securing our applications; as we are leveraging the same backend business logic, any security efforts made previously will certainly help in securing the new mobile client.

New Elements

One of the most strikingly new aspects of mobile applications is the custom built operating system on which mobile applications execute. This is especially interesting from a security point of view as it is a completely modern operating system, designed

with the security lessons of the past in mind. These operating systems have built-in security features that have not been previously available. However, even while addressing some of the security shortcomings of the past, new concerns are introduced.

New Security Features

The operating system that runs on these mobile devices provides some very attractive features from a security point of view. The introduction of built-in data encryption, software attribution, and required privilege declaration provides the framework for addressing some traditional security concerns.

The new generation of mobile operating systems has provided easy facilities for encryption of data stored on the local devices. Some operating systems even automatically encrypt all data persisted (such as iOS). This functionality is intended to protect data at rest, while the phone is unlocked and this data will be available. But its existence does alleviate concerns related to data loss as the result of lost devices.

The mechanisms used to publish and install applications enable the operating system to provide an attribution model. This allows the operating system or app store to have insight into who installed an application and who originally published the application. By providing a chokepoint for application installation (some operating systems provide a more significant chokepoint than others) this allows for easy redaction of identified malicious applications.

Another feature of mobile operating systems is to require an installed application to explicitly declare the privileged functions it will use. This allows for identification of the scope of actions the application will take on a users behalf. At face value this is an excellent mechanism for identifying applications with intentions other than their claims.

New Functionality

Apart from explicitly security focused features, mobile operating systems also provide facilities to enhance the user experience on the devices.

While the device provides a built-in data connection, the reliability of this signal is often questionable. In order to address this from a user-experience standpoint the operating systems have made it simple to persist data without user interaction. This functionality is similar to the data persistence features of HTML5, but it departs dramatically from traditional websites. In the past it was difficult to permanently persist data without user interaction. However, with mobile application it is completely transparent to the end user what data is persisted on their behalf on the mobile device. While the benefit of this is a seamless user experience even in areas with an intermittent data connection, the downside is the potential for the application to persist sensitive data without the users knowledge.

The philosophy of the mobile application ecosystem is service oriented. Operating systems have introduced formal mechanisms for inter-application communication. With this model one application can benefit from the functions of another providing a

seamless interaction for the end user. However, we now have an environment where we have introduced a new trust boundary. In traditional websites or desktop applications communication between components requires a strict and proprietary API. There are no formal facilities to 'discover' or leverage other services without explicit action. However, in the mobile world this interaction is brokered by the operating system. Given the fact that the applications installed alongside your own could originate anywhere, we now have a new consideration for secure development.

Risk Landscape

The considerations related to the risk associated with mobile applications are dominated by the use of a client-server model. As always with a client-server model, the majority of security risk exists at the server. One should assume the client-side logic is compromised, as there are no controls that can be put in place to guarantee that it cannot be reverse engineered and modified. If we assume the client is compromised, we must then ensure that all security controls and features exist primarily in the server. Fortunately, there is quite a significant amount of work done so far to date on how to best develop and deploy a secure server and following those practices is paramount. However, even once the server is 'secured' there are significant considerations for the client-side application, especially in the case of mobile devices.

The threat of mobile applications differs from traditional rich user interfaces or even desktop clients. This is in part due to the introduction of the security features discussed above, but also is due to the nature of the device they run on. Mobile applications are empowered to perform a larger range of actions with greater simplicity than we have seen in the past. While for a traditional RIA one might have done a simple security analysis of the client, it is far more important with mobile applications. The range of built in functions that a mobile device can perform on behalf of the application increases the risk. Mobile devices can now perform actions that could directly affect personal safety (sharing location or images) or maliciously incur usage fees (SMS and Phone calls). As a result, greater care must be taken to ensure that the mobile application does not, intentionally or otherwise, allow for malicious use of these actions.

When inspecting the mobile code it is important to enumerate what privileged actions are being performed by the app as well as to determine what type of data is used in the action. Our mentality when inspecting these applications needs to be focused more on validating the behavior of the privileged actions. Is the application required to create a direct connection to a server? What server is it connecting to? What data is being transmitted? How secure is the channel? Can another application request this connection to be made on their behalf? The operating system has provided the correct start for helping us manage the privileged behaviors our applications take. However, the OS simply restricts the privileged action – it does not ensure proper use of that action. Our analysis must provide that additional assurance.

In addition, it is important to remember the new trust boundary when inspecting the code. The operating systems provide a formal mechanism for applications to implicitly

or explicitly execute actions on each other's behalf. This is a new type of consideration when inspecting the applications behavior. While the idea of a service oriented system is not new, the aggregate simplicity of the mobile model is a significant change. It is now far easier for other applications to discover and leverage the services that your application provides. Conversely, it is easy for an application to implicitly rely on services that could be provided by applications other than the ones you assume (such as a 3rd party phone application on an android device). The same lessons of the past hold true, one must authenticate the requester and make conservative assumptions about the integrity of the request. While this type of attitude is reasonable to expect when architecting a large service-oriented system, the adaption of this attitude at a micro-level for mobile application developers cannot be relied upon.

Conclusion

The emergence of mobile devices is not a short-term trend: It is the beginning of a shift in the way that both businesses and consumers interact in a connected world. An increasing number of users are incorporating mobile technology into their daily lives. This increased adoption coincides with a shift in user behavior, as people perform more and more tasks on these devices – including security sensitive tasks. As our world shifts to the mobile device becoming the front-line of user interaction, our concerns as a security community must as well. While the new eco-system of mobile devices provides for some opportunities that have not been possible in the past, they also introduce new risks. This introduction of new risk is complicated by a far more diverse set of stakeholders when it comes to secure development. Not only must we adapt our development and security processes for the new considerations of mobile security, we must also adapt our business processes and relationships to properly distribute the responsibility of securing this new eco-system.