Chapter 8 : Conclusion

Embedded Systems and Mobile Communication have revolutionized the electronics and IT industry. The scope for putting on embedded systems is ever widening, encompassing nearly all activity on the life today. Impact of Technology is everywhere. Every portion of the society - educated and uneducated, urban and rural, both new and experienced, is using technology. The mobile phone has become probably the most inseparable devices in everybody’s lives. They feature numerous features that are vitally important to us. Today, with the oncoming of Smartphone’s, the applications have likewise be a little more intelligent.

Mobile applications are emerging inside software market. Beside phone functionality, mobile devices get more and even more processing power, memory, sensors, and better displays conducive to an environment fit for additional complex software being developed. The online world is employed worldwide for a wide variety of distributed applications. Unfortunately, several of these applications are sometimes hindered by two phenomena that cannot be abstracted away within a distributed setting: network delay and disconnection. The growing interest in mobile working has given rise to concerns in regards to the risk of mobile viruses and also other attacks on smartphones.

As networks grow in bandwidth and smartphones become main stream, applications have reached focus just as one essential growth area. Vendors are racing to reduce the entry barriers for application developers, this is clear from their effort to simplify their SDK and development environment and roll application stores. Then again, developers are presented with either platforms, each using own positives and negatives and monetization models.

Developers often encounter challenges when coming up with and implementing their applications. Challenges include managing a number of device targets in a platform, and then an absence of consistent standards, managing special API versions, as well as resource constraints of mobile devices. Vendors have achieved considerable success by developing their own operating systems, delivering distinct devices and interfaces, leaving the issues to porting applications across these platforms. Android(70%) and Apple(21%) that share over 90%of market share are garnering by
far the most attention one of several developer community and enjoys positive network effects, while their competitors like Windows 8 and Blackberry are simply just catching up.

8.1 Unprecedented Growth in Mobiles

In just over a decade, we've gone from a device that made and received calls to tiny computers in pockets. There are more than 6 billion cellular subscriptions in the world, and over 1250 cellphones are reported lost or stolen in India on a daily basis. When one thinks of this early stage in mobile application development and security, it is critical to know these products are not in the users' custody constantly, which enhances the odds of them being lost or stolen. This allows an untrusted party to gain access to an organization's data. Also, mobile operating systems are relatively immature and contain well-documented vulnerabilities.

So, exactly how should the developers provide mobile applications to the users that fulfill their requirement for immediate access, but additionally give to them the assurance that their details are safe?

Mobile applications have threats, vulnerabilities and risks a lot like people that are caused from typical web and client/server applications. However, as a result of inherent nature of devices being small, smart and portable, mobile applications demand additional focus in protecting data from potential attackers. Here are four key areas to target when contemplating mobile application development and security.

8.2 Comprehending the Distinctive Threats

Mobile devices pose unique threats. Think for just a minute about walking in to a restaurant and grabbing a pizza or dosa and getting a seat beside an energy supply. Besides the user, is someone with knowledge about every gadget known to man. The user gets up to go to washroom, leaving all the gadgets on table, able to be plucked by an unsavory character. Does the user know the implications for the applications that are residing on the mobile devices? It'd take less than a minute for the thief to gain access to the unit. It's highly likely, or else certain; there is certainly data around the device that could compromise the victim's identity. As somebody who frequents
coffee houses in many areas, people are seen to leave their devices unattended practically daily. The programmers have to account for this risk, which leads to the next point.

8.2.1 Implement Strong Application Security at the Outset

Activities including secure architecture and design reviews, threat modeling, secure code review and penetration testing have their importance in providing great assurance to the applications in having a great security posture. In so many cases, mobile applications are just rushed to the market. It should be understood from experience that it's less pricey and faster to leverage a well-defined security activity, which might be created to prevent vulnerabilities from the start.

8.2.2 Sensitive Data should not be Stored on Devices

Always store sensitive data on server-side systems and not on the mobile device. The likelihood of this data being lost or stolen is incredibly high, plus it doesn't need to be stored on the device without careful consideration and putting the correct controls into place. The reason for this being that mobile devices accompany us on our excursions to operate, home, the airport, cafes, restaurants and everywhere else the user goes. Devices are frequently lost or stolen, which may result in loss in sensitive data or else properly secured.

8.2.3 Understand Mobile Operating Systems

There are various built-in "features" of mobile operating systems which could impact on what the programmer ought to develop applications. As an illustration, in Apple's iOS, a snapshot is taken with the current screen when software is in background. This snapshot is stored for the file system. Imagine in the event the screen shot contained sensitive data.

Mobile device usage will continue to evolve and grow. Businesses are even starting growth of mobile intranet applications. Between Google and Apple's respective application stores, there are all-around 1.5 million mobile applications’ serving over
25 billion downloads. Value of securing mobile applications will still only intensify because proliferation of mobile devices and applications overtake business. The programmers should take time and energy to create secure applications planning ahead foreseeing the possible attacks.

Think the programmer developed a secure mobile app? Reconsider that thought. Many mobile app developers have a naive notion of app security that leads them into believing their apps are secure when they are not. Some developers authenticate users and encrypt passwords and feel that they're all set, but there might always be security holes so wide the programmer could sail a ship through them. The effects of releasing an insecure app range from financial loss, reputation tarnish, lawsuits, and Twitter shame.

When coming up with mobile apps and mobile backend services, the programmer should look at the six security properties of confidentiality, integrity, availability; authentication, authorization, and nonrepudiation (see Figure below). Simply considering how each security property pertains to the app won't make it more secure. The programmer needs to perform threat modeling on the design and discover methods to secure the app dependent on distinctive technology and employ cases. Don't forget that the mobile backend services must be secure too.

8.2.4 Six Security Properties

<table>
<thead>
<tr>
<th>Security Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Information is only available to the people intended to use or see it.</td>
</tr>
<tr>
<td>Integrity</td>
<td>Information is changed only in appropriate ways by the people authorized to change it.</td>
</tr>
<tr>
<td>Availability</td>
<td>Apps and services are ready when needed and perform acceptably.</td>
</tr>
<tr>
<td>Authentication</td>
<td>A person’s identity is determined before access is granted if anonymous people are not allowed.</td>
</tr>
<tr>
<td>Authorization</td>
<td>People are allowed or denied access to the app or app resources.</td>
</tr>
<tr>
<td>Nonrepudiation</td>
<td>A person cannot perform an action and then later deny performing the action.</td>
</tr>
</tbody>
</table>

Source: Mike Guertler, Principal Analyst, Forrester Research
The phenomenal worldwide development of smartphone sales is not due to the chance to make flawless phone calls. No, most of the people get a smartphone since they would like to access the World Wide Web and use applications on the run. Application developers have note: Mobile devices are the next big platform for serious business and consumer apps. Smartphones and tablets at the moment are where the users will find apps for mobile payments, banking, healthcare, customer care, product inventory, police officers and even more. The rise trajectory for mobile applications is steep. According to good mobile marketing firm Mobil Thinking, possibly a billion people worldwide will access mobile financial services by 2015. PayPal alone expects to see $7 billion in mobile payment volume in 2012.

But perhaps the user has stopped to consider how secure these mobile apps are? When the user use the mobile wallet, how is it possible the charge card information could possibly be exposed to cyber thieves? When a user sees a doctor and he pull out the medical records using a tablet, is the user confident that his personal data is determined to remain private? People assume the applications as well as the sensitive data they choose feel safe, but often that assumption is wrong.

Mobile platforms possess specific protection conditions which are not available on other platforms. Should the developer try to deal with a smartphone like, say, a PC or perhaps a World Wide Web browser, he will probably overlook essential matters. An example is, mobile devices possess a specific variety of memory to preserve living within the particular device. Because of this, mobile phones sustain information so long as they can consider that the memory incorporates a limited read/write capability. When developer models software that writes information on the cell phone, both deliberately or being a by-product associated with action, the particular information is almost always recoverable. A programmer really should have an alternative way of contemplating where data gets written, and the way to eradicate it and completely clean out.

Developers shouldn't depend upon the security which could possibly be part of a good device. For instance, pass codes along with the default encryption through an iPhone or iPad could possibly be easily circumvented. In the event the application employs very sensitive data, programmers should want to do more to guard it.

With most certainty, one can express, unquestionably, that mobile device has an insecure application. Latest applications are increasingly being downloaded at
spectacular pace - quicker and quite a few with no warning ever sold. During Christmas time, 2012, 1.76 billion apps were downloaded on the particular Apple App Store along with the Google Play Store and 50 million new iOS and Android mobile devices were activated.

8.3 Need to Determine What the User does not know

Several organizations possess the actual foresightedness to supply constant training for their development staff members so they really may dependably produce better, secure code. With the highest assurance, one can express, unequivocally, that many mobile devices include vulnerable software. Most recent applications are increasingly being downloaded from internet at high speeds - quicker than ever imagined and quite a few are being put in markets without any warning actually. Usually, it is estimated that it costs $1,000 to discover a vulnerability and $4,000 to mend it. It's far more cost-effective to instruct developers develop security in first. Certain classes associated with vulnerabilities, like SQL Injection, were first discovered over a decade ago, yet stay pervasive. With better education and training, programmers are able to eradicate certain vulnerabilities from other organizations rather than possess similar issues appear repeatedly.

While mobile development is not a completely different from other database integration efforts, it isn't treated precisely the same. Mobile applications present both new and existing threats. Comprehending the key variations in operating systems and Application Programming Interfaces (APIs) is vital in creating secure mobile applications. There are many great mobile software security courses accessible for ones organization's consideration. Auction web sites 1,000 vulnerabilities offered to, developers cannot often be likely to produce defensible applications without correct training.

8.4 Force Alter in Application Security

In most organizations with application security programs one can see ample recommendations, programming standards and the like, but a majority of these are specific to web applications or client/server applications. These practices have to be
updated and inclined to mobile technologies. Is the best organization using iOS, Android, Blackberry, Windows Phone 8, etc.? There are various techniques to build and enforce mobile application security program. These various steps are discussed below:

### 8.4.1 Develop Mobile Security Standards

And Apply Them! All organizations incorporate some way associated with standards and guidelines for developers to stick to when coming up with applications. However, their details are oftentimes not focused entirely on security, plus many instances there is absolutely no reference to mobile applications. There are lot of differences between Android and iOS when coming up with sure auto-complete is powered down or password fields are appropriately protected - much like we're going to stress about inside a browser. The programmer still has to make sure we've solid security standards and guidelines such as the technologies which can be used. Do the standards and guidelines reference security or mobile security?

### 8.4.2 Conduct Design Assessments with Threat Modeling

In the main, applications are becoming to become a growing volume of complex, using cutting-edge technologies and even more backend resources. With the actual mobile channel, another level complexity is put into our infrastructure that's highly intricate primarily. Occasionally, applications add new, use or enhance current infrastructures to support the latest mobile channel. Adding a mobile channel uses a thorough design and architecture review through a focus on threat modeling. One must understand the newest threat-scape with mobile applications plus the potential risks towards the business. Thorough design and architecture reviews that are included with the actual threat modeling technique help uncover the risks before a credit application goes live, to ensure remediation activities might be accordingly.
8.4.3 Conduct a Manual Substantiation

After design/architecture reviews have been performed with threat modeling, the programmer should conduct some level of manual authentication. The actual scope and also a more impressive range rigor will be based how much risk presented from the device. The actual application's size and complexity will determine the particular multiple stages of authentication through iterative code reviews and penetration testing. Organizations must engage mobile authentication experts to work alongside internal teams. Companies will need to have a close look towards constructing a solid testing group from inside.

8.4.4 Thorough Dynamic and Static Authentication

Dynamic and static authentication techniques continue to be in their infancy and, so, minimal are available with the entire dynamic authentication associated with mobile applications. However, that will not signify payday cash security activities don't match the secure mobile development process. Once these technologies are mainstream and efficient, the programmer should definitely evaluate our mobile code during development using static approaches to make sure bad APIs aren't abused which other security controls are coded appropriately. Dynamic and static analysis for mobile continues to improve and fit nicely into our security activities. Some applications require higher degrees of rigor and, so, the company should perform every one of the aforementioned activities. Several cases, danger level may warrant only manual or dynamic authentication. Either way, mobile applications will continue with their popularity. More push will come down in the sections to come up with increasingly apps to enhance employee efficiency and client needs. However, it is necessary that many organization developing mobile apps contain a well-defined and stable mobile software security process. Users are unaware that these applications that they can download may be malicious and offer to get more entry to company assets than needed. What exactly can lessen the risks presented by mobile devices and their apps, with the direct access which is provided to proprietary corporate data and networks are listed here. These actionable solutions can be used to mitigate risks:
8.4.5 School & Notify

Typically, people realize that the user will find applications for everything, nonetheless they don't quite equate these small, smart devices as being computers with huge processing power. Moreover, folks are unaware how the applications that they download could possibly be malicious and gives for additional use of company assets than needed. These malicious activities come about unbeknown on the unwitting user. User education is key and must cover:

- Keep in mind the cause with the download (mainstream App Store - or not, it doesn't matter);
- Be cognizant on the permissions that applications are requesting;
- Understand some built-in functionality that could put the company data and network in danger;
- Much more doubt, rule against each other and do not install the application.

8.4.6 Device a vigorous BYOD/MDM Resolution

Bring-your-own-tool and mobile device management are widely-used terms when discussing wanting to protect the enterprise on the mobile device explosion and inherent risks. When organizations want employees in order to bring their mobile devices or use company-owned devices, they need to ensure that they could lessen the data footprint these devices and mitigate risks. The best way is always to select an MDM vendor to create internal app stores where users can download only company-approved applications. Another method should be to require users to utilize a VPN to get in touch to internal resources.

8.4.7 Bar Jailbreaking and Rooting

Most mobile operating systems run inside a state where their apps have their very own "sandbox" and have to request entry to shared resources or simply have entry to their sandbox of internet data. Sandboxing helps the user to prevent applications from accessing other applications' data. The act of jailbreaking and rooting devices destroys for good business of a sandbox and allows applications to run as root. This gives these
devices to silently monitor everything, including application data and traffic. Deploying MDM root or jailbreak detection services, along with education, will help to mitigate the chance of jailbroken devices in the organization's apps, data and networks.

### 8.4.8 Outline & Impose a Data Sensitivity Strategy

The largest risk to mobile security could be the disclosure of sensitive data. Organizations must have policies about data's sensitivity, so when and where users can see and/or manipulate that data. Some organizations don't let any data which has a sensitivity amount of "internal" to go out of their internal network. This may include transmission through e-mail, USB drives, laptops, etc. Typically, the user can find multiple amounts of data classifications with specific policies for each. The policies really should be used on mobile devices. Highly sensitive data shouldn't be stored on mobile devices.

### 8.4.9 Restrict Internal Access

Mobile devices communicate via cellular networks, Wi-Fi networks, Bluetooth, near field communications (NFC), and rf identification (RFID), to name a few. Several devices can connect backward and forward, providing having access to internal resources. As an illustration, most laptops be capable of connect with devices and transmit files, or use shared networks, through Bluetooth or USB. Mobile devices can connect straight to our internal Wi-Fi networks, which provide usage of internal resources. All strategies to mobile device communication ought to be carefully examined. Internal access ought to be restricted having a "deny all first" strategy to mitigate the threat of malicious software connecting to and tampering with internal resources.

As our mobile domain continues to evolve, along with the dissemination of mobile devices into corporate environments increases, the amount of threats and attacks through mobile applications will continue to surge. The threats are not going to be targeted at the mobile device operating systems but also mobile applications, their
data, and connectivity. Think carefully about allowing mobile devices make use of business resources, just as we have always with laptops and desktops.

### 8.4.10 Protect Communication among Mobile and Server

Another essential aspect of building mobile apps is always to secure communication between the tool and server. Most applications hook up to information being stored about the server using standard web protocol (HTTP) and pass data between the two using standard data formats like XML or JSON. Using public Wi-Fi or cellular signal from around the globe and connecting to data for the server enables hackers to intercept and view information being transferred on the wire using sniffing tools and man-in-the-middle attacks. Because World-wide-web will be the # 1 method to obtain information for most of us these days, and also the Web uses the HTTP protocol to communicate between the web browser and server, the programmer can actually realize why appreciable link can be simply intercepted and hacked.

One of many most effective ways to secure communication would be to merely employ HTTPs rather than HTTP protocol when building business applications and accessing data for the server. Using SSL link to the web server automatically ensures that the result has been encrypted with a digital certificate which can be setup and configured about the web server. Typically, digital certificates provide a minimum 128-bit encryption the whole way approximately 4096-bit encryption which ensures the information being transmitted in the wire is safe and not on the market to hackers.

### 8.4.11 Malicious Functionality

The group of Malicious Functionality is a set of unwanted and dangerous mobile code behaviors which are stealthily placed in a Trojan app that this user is tricked into installing. The person thinks they may be installing a game or utility and instead get hidden spyware, phishing UI, or unauthorized premium dialing.

- Activity monitoring and data retrieval
- Unauthorized dialing, SMS, and payments
- Unauthorized network connectivity (exfiltration or command & control)
- UI Impersonation
8.4.12 Vulnerabilities

The category of mobile security vulnerabilities are errors in design or implementation that expose the mobile device data to interception and retrieval by attackers. Mobile code security vulnerabilities also can expose the mobile device or perhaps the cloud applications used through the device to unauthorized access.

- Sensitive data leakage (inadvertent or side channel)
- Unsafe sensitive data storage
- Unsafe sensitive data transmission
- Hardcoded password/keys

8.5 Categories of Risks in Mobile Applications

Users are becoming more dependent upon our texting and tablets. The mobile phone accompanies users virtually everywhere, including at the office. Operate, executives soon recognized the potential for increased productivity from mobility and started allowing personal devices to touch base to corporate networks, including e-mail. This paradigm shift in the way in which people live and work spawned a flurry of bring-your-own-device and mobile device management design and implementation activities among many organizations.

But in the rush permitting personal devices to be used for work, we in application security neglected to examine thoroughly the modern risks external applications may introduce to the organizations.

We have become better, and portions of computer are consumed by mobile devices and whatever we can perform with him or her. Each of us consume a numerous efficiencies, we must consider what having real-time access to everything the world thinks we start to use method for our businesses. Kinds of risks do they bring and what should one instead concern with?

Mobile applications have similar threats, vulnerabilities and risks to prospects posed by typical web and client/server applications. Having said that, because users hold the
power and ability to download whatever they wish and manage their devices to their liking, we must take into consideration our prime five risks and ways to mitigate them:

8.5.1 Intrinsic, Blind Faith

App stores come pre-placed on our mobile devices and supply use of a huge amount of mobile applications. We blindly trust that this app stores have performed required groundwork for the apps into their stores. Yet, in reality, app store vendors lack the cycles to ensure that the apps they cook available won't throw open our employees/users to risks that will harm the business.

8.5.2 Functional Perils

Opening, editing, sending, receiving and e-mailing documents; syncing backups; checking directly into my current location; etc. - these are generally a smaller subset of tasks that can be filled on the devices. But what the results are basically open a PDF from my offer e-mail in to a PDF viewer which may have been downloaded previously. Suppose the user then sync that document to the PDF viewer? Now, the potentially sensitive document has been managed by somebody else's application probably insecure application and sync storage, in fact it is completely outside my control. How about only sign in to my current location via Facebook or Foursquare?

8.5.3 Malware

Malware has forever been a problem within the IT world, and it’s also the same within the mobile sphere. Malware can wreak havoc by stealing sensitive data, monitoring traffic, connecting to internal networks and infecting internal machines. That is certainly just for starters. Malware will continue to evolve in apps from app stores, and attackers continue to refine their methods to malfeasance.
8.5.4 Root Requests

Rooting and jailbreaking are commonplace. Users or attackers run exploits up against the mobile operating system to produce them with unfettered usage of the file system and permit these to function as the "root" user with the operating system. Quite a few users appreciate the freedoms that having root access offers them. Root access also comes with a gateway with other app stores, for example Cydia, or the ability to download applications from untrusted sources. The applications running as root deliver functional and malware risks for the business. Sometimes, the functional/malware line starts to get fuzzy with all the root applications because, typically, the applications provide more functionality compared to typical non-root applications provide.

8.5.5 Inappropriate Applications

Clearly, don't assume all applications work at work, and I'll hire the imagination to classify which ones would be classified as Unsafe For Work.

8.6 Mobile Security Challenges

The volume of mobile applications went from zero to a 1.5 million within a nothing but four years, as well as keeps growing in quantum leaps. Because mobile app world is constantly on the evolve, so will the risks.

Mobile device manufacturers and operating system vendors out there today are increasingly being asked to supply additional security capabilities included in the operating system and software to adapt to government regulations, identify malicious activity, detect viruses and spyware and also secure application data. When building applications for mobile devices and storing data on the device the programmer will need to consider and analyze the next key aspects:

- Could be the data being stored encrypted and compressed?
- Is there a lifespan of knowledge stored on the device?
- How sensitive is the information being stored? If very sensitive, should it be stored at all?
• Has application entry to the information been secured in some way e.g. login user ID and password?
• In the event the device is lost or stolen, can entry to the appliance be immediately revoked and terminated?
• If the device is lost or stolen, can the unit be wiped remotely?
• Would be the data readily available for offline use? If yes, could be the data erased and cleared up with the application?

While using constant demand on business plus it to supply “more with less,” organizations can deliver real business value by using a Mobile Application Framework and mitigate risk, thus providing businesses with real financial savings.

When looking for Mobile Application Frameworks for building native applications for Android and Apple devices, the programmer will need to evaluate resistant to the following criteria:

• Is the mobile application framework from the reputable vendor?
• Gets the vendor held its place in business for a long time and does the vendor have an established past record?
• Does the mobile application framework leverage existing developer skill sets?
• Does the mobile application framework deliver the inspiration needed for mobile applications i.e. security model, navigation, graphical user interface elements, examples?
• Does the mobile application support both Android and Apple operating systems?

One of the challenges when building mobile applications is to make certain users are just given use of information that they're authorized to look at which sensitive information is never stored around the device itself. One way to secure application use of business data on corporate servers would be to build a solid framework that may handle every of knowledge security and access for the server.

Below are outlined common techniques that can ensure secure application access to data over a server:

• Role-based security to manipulate user access and visibility to business data. This will allow the programmer to easily manage and administer access and in addition turn “off” use of home the server if needed.
Will not store passwords or PINs for the tool and always perform all application security checks on the server.

- Encrypt all sensitive information on the server and only send the specified level of information on the mobile application.

- Log all application activity about the server from all devices and restrict having access to applications and data in line with the unique device identifier.

- Prompt for extra PIN number to reach critical paths from the server and re-validate and verify the login home elevators subsequent requests following initial login for the application continues to be verified.

- Implement a Firewall and DMZ made up of and exposes the business’s external facing servers towards the outside world. This supplies an extra layer of security with an organization’s network, applications and data.

- Provide VPN access for really security that may be easily enabled or disabled on the server side.

- Leverage remote monitoring capabilities that provide the chance to remote wipe a computer device when it gets lost or stolen.

- Educate and manage employee behavior and use of mobile applications so that security intact. Supply to them regular updates to make them aware of the security policies.

8.7 Application Security on Android Mobile Devices

Increasing smartphone adoption rates in conjunction with the rapid growth in smartphone application counts are creating a scenario where private and sensitive data is being pushed towards new device perimeter at a growing rate. The smartphone mobile device is quickly becoming ubiquitous. Nevertheless there is much overlap with common operating system models, the mobile device code security model has some distinct points of differentiation.

The mobile code security stack can be cracked into four distinct layers. The best layer of the stack would be the infrastructure layer, followed upward by the hardware, operating system and application layers. These security stack layers each define another part of the security style of a smartphone or mobile device.
The security of a mobile can be divided into layers with each layer being responsible for the security of that defined components certainly nothing more. The top layers in the stack count on all lower layers to make certain their components are appropriately safe. This abstraction based model allows the planning of a particular mobile security mechanism to spotlight one particular specific subject of concern without expending the resources forced to analyze all layers that support its current location within the stack.

### 8.7.1 Infrastructure Layer

The infrastructure layer would be the lowest therefore most supportive layer of the mobile code security stack. This layer may be the foundation that supports all of the other tiers on the model. Almost all of the functional components at this layer are managed by a mobile carrier or infrastructure provider; however integration in to the handset occurs as results are transmitted from this tier upward.

Cellular voice and data carriers operate the infrastructure that carries all data and voice communications from terminus to absolve point. The security of components only at that level typically encompasses the protocols being used because of the carriers and infrastructure providers themselves. Examples of such protocols include code division multiple access protocol (CDMA), global system for mobile communications (GSM), global positions systems (GPS), short messaging systems (SMS), and multimedia messaging systems (MMS). Because of the low foundational nature on this particular security tier, flaws or vulnerabilities discovered when it reaches this tier are often effective across multiple platforms, multiple carriers, and multiple handset set providers.

### 8.7.2 Hardware Layer

Once we go up the stack towards the second tier of the mobile code security stack, were getting into the realm of a physical unit that may be typically beneath direct control over an end user. The hardware layer is identified through the individual end user premise equipment, generally by means of a smartphone or tablet style mobile device. The hardware layer is obtainable to the operating system enabling direct
management of the physical aspects of it. This hardware is generally the “firmware” which is upgraded by the physical manufacturer with the handset and occasionally delivered by proxy through the phone carrier. Security flaws or vulnerabilities discovered only at that layer typically affect all owners who use a particular bit of hardware or individual hardware component. In case a hardware flaw is discovered in a single manufacturer’s device, it's most probably that hardware revisions using that similar design and/or chip will be effected likewise.

### 8.7.3 Operating System Layer

The third tier from the mobile code security stack will be the operating system layer. This layer corresponds towards software running using a device that allows communications between the hardware along with the application tiers. The operating system is periodically updated with feature enhancements, patches, and security fixes that might or may not coincide with patches created to the firmware by the physical handset manufacturer. The operating system provides access to its resources through the publishing of application programming interfaces. These resources are around to be consumed from the application layer since it is the only real layer higher within the stack compared to the operating system itself. Simultaneously, the operating system 'talks' to the hardware/firmware to operate processes and pass data to and from the product.

Operating system flaws really are a quite typical flaw type and currently are generally the mark of for attackers that need a top impact. If the operating flaw is discovered, the entire install base of this particular operating system revision is going to be vulnerable. It truly is when it reaches this layer, and above, where software is the overriding enforcement mechanism for security. Specifically because of the fact that software packages are relied upon, the operating system, along with the application layer above, is easily the most common location where security flaws are discovered.

### 8.7.4 Application Layer

The appliance tier resides over the rest the mobile security stack which is the layer that this end user directly interfaces with. The application form layer is identified by
running processes that utilize application programming interfaces given by the operating system layer as a possible feeder point in to the rest of the stack.

Application layer security flaws generally are a consequence of coding flaws in applications which have been either shipped with or installed onto a mobile device after deployment. These flaws also come in classes which might be exactly like the personal computing area. Buffer overflows, insecure storage of sensitive data, improper cryptographic algorithms, hardcoded passwords, and backdoor applications are merely a sample pair of application layer flaw classes. The consequence of exploitation of application layer security flaws do range from elevated operating system privilege to exhilaration of sensitive data.

When analyzing some device for security implications, the programmer need to think about every one of the layers with the mobile code security stack and determine the effectiveness of the security mechanisms which have been set up. At intervals of layer evaluate which, if any, security mechanisms and mitigations the producer has implemented if those mechanisms are sufficient for that type of data the programmer plan to store and access around the device.

8.8 Counter Measures for Security

As with any new technology, Wireless networks bring benefits but in addition new and increased risks are also coming up each day. Before starting an invisible network, it should be shown to develop policies standards, procedures and guidelines, to perform periodic wireless security assessments and audits to evaluate and track wireless and handheld devices. Wireless security features like encryption and authentication must be enabled. Techniques like trusted computing with signed applications that can get access to critical system functions can provide additional security to mobile devices. As it would be impossible for programs to overwrite operation system files, you may easily reset these devices to its initial (not infected) state.

The best defense against social engineering attacks is usually to offer an informed workforce. Organizations must educate users thereon issues. Establishing a substantial security policy that features specific assistance with passwords, data classification, access control and physical security goes further in combating social engineering. An
implementation of a security architecture that includes the use of firewalls, cryptography, and Intrusion Detection System (IDS) to protect from unwanted intrusion.

### 8.8.1 It’s About Risk

If there's been one primary topic that this research anticipates to have conveyed, it can be that security, application security or else, is basically risk management. With no context of risk, security means nothing. If the programmer hears someone say that their application, system, or building is protected, it is best to immediately think, “Secure against what?” What include the threats how the security highlights of this system made to protect against? What are classified as the vulnerabilities that this system may have? And what can be the consequences if someone of these vulnerabilities were to be exploited? Before the programmer has in mind the solutions to those questions, the programmer can't judge precisely how secure anything is.

The programmer is not, however, at all secure against certain exotic threats, including an attack with a 30-foot tall monster. Consider some of the threats, vulnerabilities, and consequences? If the risk that equates of one's risk analysis is high, you'll want to mitigate it. Should the risk that happens of the risk analysis is low, and then the programmer is OK. But every security feature the programmer deploys really should be appropriate based on the risk.

### 8.8.2 The Principle of Least Privilege

Another primary topic that is expected to be understood is that the Principle of Least Privilege. Basically, entities really should have adequate privileges to complete their job no more. In case the application doesn't have to own Access to the internet, do not request the Internet permission. In order to make the Content Provider accessible to other applications over a device, but only the programmer has to be competent to update the info, make certain the programmer configure separate read and write permissions on it entity. Along much the same lines, store merely the data the programmer needs to do the job. But if the application does not need to understand users’ banking accounts numbers, do not request them. But if the application uses
passwords to encrypt some sensitive data, therefore the programmer will never need to supply that password any place else, store a hashed version and actually, a version containing PBE algorithms discussed earlier which use multiple hash rounds and salt rather than the actual password. Run with just the minimum privileges the programmer needs.

8.8.3 Operate the Permissions System

Android supplies a very robust and extensive permissions system. If the programmer uses any features of the system that are deemed dangerous, the programmer needs to declare that, as well as the user needs to substantiate the access. The programmer may also use permissions to craft a robust access control strategy around each of the pieces of user. Should the application handle sensitive data, employ this feature. Lock down access to any or all of these components, to enable them to supply only since the programmer intends these phones be. This is why the permissions model exists, so utilize it.

8.8.4 Have the Cryptography Right

To be able to protect sensitive data for the level warranted from the risk, the programmer will often need to employ encryption, so be sure the programmer operates the Android-provided libraries and use them correctly. If the programmer wishes to send or receive sensitive data over the network, make sure the programmer has an appropriately configured SSL/TLS system. But, what the programmer may find him doing, ensure the programmer uses precisely what is out there and try to craft the solutions or implementations. Cryptography, plus the protection of sensitive information in general, is an extremely complex topic with plenty of subtle points that could easily reduce its effectiveness by large sums, so be cautious.

8.8.5 Never Trust User Input

In case user accepts input from anywhere outside of the application, whether in the user on the device or from over the network, the programmer is unable to trust if the
programmer truly agrees to this. We walked by using a simple demonstration of SQL injection, and this only agreed to be the start of the many vulnerabilities that the programmer can get as a result of lack of solid input validation. Always check the input to ensure it truly is what the programmer expects so that it is, no longer and no less, when user continues to process it.

8.9 To Summarize

Android includes a large amount of security features constructed into its operating system and help protect the mobile device applications. On Android, by default, no application contains the permissions was required to perform operations that impact other apps or perhaps the device on the whole. This arrangement prevents apps from reading information or data stored by other apps, and keeps them from reading the consumer’s personal data stored around the device.

For just one application to data with every other, it must give explicit permission to the other app to see its data. One example is, suppose an app that a user downloads from the Android Market needs permission to find out her GPS location. In the event the user installs such an app, it prompts her for permission to learn to read her GPS location.

Android is based on the Linux Operating System, which has elaborate security mechanisms internal. Each app runs that has a distinct system identity (including its Linux User ID and Group name), which is unique for everyone apps. The Android OS assigns a unique User ID a great app if the app is installed. Linux uses this mechanism to discover apps from the other person and protect the system in the main.

Despite the presence of the security built in the Android OS, users own the final responsibility for protecting their devices. A malicious app from the Android Market could still be written to get permission with a user’s SMS messages, contacts, or GPS location.

The security that is part of the OS protects apps from one another, but isn't going to necessarily shield the user’s data from malicious apps. So this security feature is just one part of the puzzle; this doesn't preclude the necessity for mobile security about the Android device.
Moreover, the people as being the soft spot on the whole network system must not be forgotten. Many attacks or viruses wouldn't be so successful if their human victims would not make them to spread quickly. Additionally, operating systems for mobile devices contain weaknesses. Most of them are generated by weak implementation of some system functions. Most operating systems for mobile devices are new developments and have been created specifically for small mobile devices. The theoretically acquirable security level with these operating systems is extremely high; the security type of the operation systems is designed to give you a robust, protected environment for mobile devices. Nevertheless, most producers of mobile devices often reduce security levels to generate when using the devices easier for users.

Viruses happen to be a rising threat since mobile phones spread all over the world and appeared in such large quantities. It was indicated that a variety of malware (Trojans, viruses and worms) for all those operating systems already exist. With all the first java-malware, called “Red Browser”, the border between different operating systems was broken. This malware can operate with different operating systems. A java worm could infect every mobile phone containing java independent about the operating system. There exists commercial motivation for virus authors too. When a mobile malware sends an SMS or makes calls to premium rate numbers, it relates to expense for the victims and brings big profit for that malware writer because this expensive phone number is owned by him. Certainly, today’s operating systems and mobile devices can be protected against these threats. One can find defenses against all known threats, nevertheless the progress from the malware and attacks could bring new dangerous problems, which really do not yet know. Today’s security systems should foresee these dangers and react to them immediately. The implementation of security functions in the operation systems, like authorization or buffer overflow prevention is essential.

The previous section outlines most of things that were found to be key takeaways from the research. Application security is definitely an extensive topic and this research has served to be the introduction to mobile application security issues. Nonetheless it does contain real, practical, actionable knowledge the programmer can use straight away while developing secure Android applications. It has been a
wonderful experience for the researcher to explore and understand the vulnerabilities of mobile applications and has created a strong wish to discover more. The future scholars and researchers can take what the researcher has leant and put it to use within the development efforts continuing to move forward. Mobile development is where the future lies, for the moment. The programmers just need to ensure that it’s an optimistic, robust, and more secure world.

This project work signifies that the security offered by mobile devices these days is just not enough to halt actual attacks or viruses. Additional applications like firewalls or antivirus programs should be made to avoid harm to the security mechanism of these devices and networks with which they are connected. It really is disappointing to see how the operating system’s developers haven't used the opportunity to create completely secure systems. There are also similar issues with the roll-out of wireless protocols. Looking forward, many future threats cannot be predicted today as Malware is gathering a tempo slowly. There has not been a fantastic worm epidemic for mobile phones. Today the unit and application developers of mobile phones have to be able to prevent a real malware outbreak.

8.11 Future work

Today many antivirus companies offer security solutions for mobile phones. They'll use worries from the customers predicting virus epidemics soon to market antivirus protection. For common consumers antivirus programs and firewalls can be found. To counteract infection, all files are automatically scanned for viruses when they're saved; copied, downloaded, synchronized you aren't modified. For mobile phones with WLAN, a firewall can safeguard the mobile device from intrusion attempts and malware attacks. Businesses can find solutions that provide automated distribution of antivirus updates directly to the mobile phone on the wireless connection. Company IT administrators could see the antivirus service status of each and every company mobile phone.

Mobile operators already provide a selection of their mobile phones with mobile antivirus software. But there are still many phones which come without protection towards consumer. Mobile phone operators could supply any phone with antivirus software; nevertheless they ought to buy every copy with this software on the
antivirus manufacturer this also would enhance the expense of mobile phones. Secondly just one or two individuals are fearful of mobile viruses today and request for protection against threats. If a lot of people would be a little more thinking about helping the security in their mobile phones, then mobile operators offers more security software and security functions on their own mobile phones.

This research focused on importance of mobile application security with special reference to android. The research can further be expanded to other platforms, mainly to iOS, Windows 8 and Blackberry OS.

Also researches in future can try to study more from the end user and company’s point of view, because of the varied embedded systems and architecture of smart phones. It is practically not possible to build antiviruses and threat protection considering that each company’s phones have different boot-loaders and completely different architecture.