Expert System for Security Evaluation

Ashish Kumar
axk5561@rit.edu

Divesh Soni
dms6244@rit.edu

Dhruv Gala
dmg7937@rit.edu

1. What security evaluation aspects have you identified in your research?

We have designed an Android application called Confine Me, which is a hierarchical expert system to acknowledge the users regarding the security loopholes in the devices they use for storage and transmission of sensitive data.

Our project being an Android application, the primary security aspect that can harm the user's privacy is the leakage of their confidential data, the one related to system's security at its lowest level of abstraction. Android platform has a Java stack built on top of a Linux kernel. There is inefficient mechanisms in the existing structure of the Android Framework to monitor and if necessary, restrict invocation of the kernel.

Android Security Framework (ASF) claims to resolve more security issues than it actually does. ASF fails to block the applications from invoking the kernel calls and accessing cache data. To resolve this issue of regulating control over the calls invoked by applications that would involve other application's data or sensitive device data, we use an empirical approach as suggested in [1], by checking on certain system settings and toggling them, if necessary.

Impact on the internal state of the system can not only render the risk of compromising the data directly stored or transmitted by the device, but also it can continue leaking other valuable internal device information like the user's location information. This kind of data leak is extremely difficult to detect and even more complex to resolve once known.

The other aspect of security evaluation for a handheld device would be the device settings pertaining to security. Fortunately, this aspect of the system can be controlled by the user, to a certain extent, by regulating the corresponding device settings that the user has enabled. For instance, rooting the device exposes the user to disparate data exposure vulnerabilities. Also, the installed applications on the device are taken into account when evaluating the device security based on various factors like review, publisher, etc. that are associated to each application installed.

Out of these two security aspects, we have given more credence to ensuring the confidentiality of the user data by means of securing the internal structure of the system calls made at the application layer. Thus, settings pertaining to the invocation of application layer calls are analyzed and a specific measure is pre-assigned to each rule that deals with one of such security setting. Our expert system gives 60% weightage to this internal security evaluation and 40% to the evaluation of various system settings pertaining to the security of the Android device.

The expert system that we designed which goes by the name of Confine Me, is hierarchically designed to give appropriate amount of significance to each security aspect with respect to Android platform. Based on those settings that can be tweaked by users to facilitate more secure device functionality, recommendations are made to the users to enhance their device's security.

2. How do you address these aspects and problems in your project?

In order to evaluate the security of the device at the kernel level, it is necessary to first know from which layers in the Android System Architecture can the kernel-level calls be made. When made from the application layer, the calls will have the ability to retrieve critical data associated with other applications like passwords or usage information. Such calls can even leak local device information like sensor data which can give the user's current location.

Rules are designed to evaluate the security of the kernel-level data and each rule is given a different weightage which is determined by Fuzzy Logic. For instance, when the Android device operates in developer mode, it is more vulnerable since the application layer has an increased control over the kernel and the data from the browsing cache. Thus, checking whether the device is operated in developer's mode gives a good estimate of the device security at that point. Also, if the device is operated in developer's mode, the user is suggested by our expert system to toggle it off unless necessary to protect exposure of user-critical data.

In a similar way, various other settings are checked by our expert system called Confine Me, and their corresponding values are recorded for the security evaluation. Fuzzy logic is then used to arrive at the valuation for each setting, and then a final device security measure can be obtained by integration of the results of all such rules categorized under the system security domain.

Another aspect of uncovering the superficial security vulnerabilities in the device is by considering the basic user settings pertaining to the security of the device. This is done by evaluating the values set by user and then based on the significance of each setting, independent weightage is given to compute the risk factor in terms of system security settings.

The user is not only given the feedback about the extent to which the device is secured, but also a list of recommendations which could compensate for the existing security loopholes in the Android device under use. These recommendations involve modifying the appropriate security settings, like disabling the "Show lock pattern when entering", etc.

3. What novel and interesting ideas in computer security evaluation have you identified, addressed, and/or solved in your project?

The security in smartphones is extremely undermined, especially on a platform like Android which is the most widely deployed of them all. Security mechanisms enforced are not remotely comparable to the ones in the desktop counterparts.

The fact that the Android Security Framework (ASF) has major security loopholes leaves a lot of room for improvement for the third-party security service providers. But using anti-virus software's for the smartphone not very popular among the users, who are barely aware of the threats the existing system Android architecture poses. Moreover, even the commercial anti-virus and malware detection software's fail to leave out the detection of unauthorized access of the application and cache data by malicious applications. [1]

The most significant violation of confidentiality would be leakage of application data like passwords, bank account details, credit/debit card information, etc. All this is possible since the calls at application layer are forwarded to the kernel with no restrictions or any means of authorization at all.

This means that without the user being acknowledged about this inter-application transfer, malicious codes have the potential to sniff the data that exists across the device. This vulnerability doesn't just restrict itself at gaining access to other application's local data but can also access the browsing cache data. This cache can include something as sensitive as cookies which has the power to grant access to the appropriate web portals the user accesses through the smartphone browser on a frequent basis. Access to user's log trace over the web merely can give out valuable information and harm the user in unforeseen ways.

This can be stopped by monitoring each call to the kernel by identifying the process id which generated the call, and restricting the access to the kernel if the call involves retrieval of data not related to the application itself. This distinction can be made from the set of predefined permissions that the application obtains grants for, when it is being installed.

Additionally, the same application invoking kernel too frequently can exhaust valuable system resources. This is well known as Denial of Service (DOS) attack in the hacking community. Preventing this pattern of DOS attacks can be possible by restricting the calls from same applications by tracing their PIDs. For the Android platform, the application layer calls are identified by PIDs greater than 10000. Thus, a predefined number of calls with such PIDs can be allowed over a given period of time, while restricting any excess.

These two security evaluation of preventing the application data from being maliciously accessed by other applications and prevention of DOS attacks are the two novel ideas deployed by our expert system.

4. What problems have you identified but not solved? Why?

Android ROM is the basic OS firmware layer of the phone. This is the base for all phone operations [3]. There are basically two types of Android ROMs viz. STOCK ROM, which comes preinstalled with phone and CUSTOM ROM, which are aftermarket versions that could be installed on a rooted phone. There are various advantages of installing a Custom ROM on your device. Android with its new market share has found ways to incorporate itself with many corporate giants. With the growth in demands there is no way that the sales of android are going down. A lot of research work is going on to find a way to incorporate custom ROM's into the corporate world.

In android devices, the partition that is most important is the system partition, as it holds all the important system files. As per the general policy they are all marked as RO i.e. read only. However, for certain Custom ROM's it is observed that it marks these system files as RW i.e. Read and Write. For even certain ROM, they go even further and mark these critical files as RWX i.e. read write execute permission for all users. This will allow a user to update or modify the contents of the system files such as system/app or system/bin, which are the most crucial partition files on the android device.

In our expert system we have designed the mechanism of determining if the device under inspection is rooted with a Custom ROM or not. An extension to finding the actual Custom ROM which is currently installed on the device would help to further get information about the security permission rights that are set for all the users of the said device. According to the kind of Custom ROM installed on the android device we could draw conclusions regarding the security level for that device on the basis of different parameters. However implementing the mechanism to find the kind of Custom ROM that is currently installed is a tedious method and is generally out of the scope of the project. Moreover, a lack of data about all the Custom ROM's that are available in the market is the primary reason that propelled us to exclude this metric.

Another aspect of evaluating the security of the android handheld system is to scan the media files. Media files are the most of the files that are downloaded from the internet and/or shared amongst the android users. Scanning these multimedia files for potential threats to determine the security also plays an important role here. But scanning all these multimedia files is way out of the scope of this project. There are a lot of applications in the market to scan the system for potential malware or viruses. An expert system could just test if any such application is installed in the android handheld system, but then different antivirus application have different efficiencies. So it is very tedious to a lot a certain weightage to the overall security of the system for this scenario.

5. Do you think your design approach represents the best way to solve or address these problems? Why yes or no?

We have designed our expert system hierarchically, which I believe, is the best possible way to model disparate sets of security evaluation criteria. It provides a clear distinction between rules of each category and facilitates any addition of new security evaluation category as a whole.

The Android application we built is designed hierarchically, which allows separation of distinct security aspects of the Android platform to be done independently. Thus, evaluation of similar set of rules and assigning significance to each rule that is deployed by our system is easier when compared to the rules of the same category, instead of packing all the rules together in a single bundle for the sake of ease of design.

Our expert system enables potential addendums of any new aspects of security evaluation of the Android device on which the application Confine Me runs on. This feature of scalability makes our expert system a highly efficient and scalable application which can undergo constant improvement over time. Here, scalability is achieved in terms of the security evaluation criteria undertaken by the expert system.

Moreover, if the user wishes to view detailed categorical security evaluation of the device, we have provided the feature to present a detailed view to the user. This enables the user to clearly determine whether the device vulnerabilities are major threat or the system settings are overlooked to render the device exposed to security threats.

6. Why do you think your tool is the best to design hierarchical evaluation system?

We used Android Studio for designing our hierarchical system. Some of its features are discussed below -

1. Real time coding and bug fixing: Android Studio uses the IntelliJ editor which is a high quality product for code editing. It allows for smart editing and has advance features like real time rendering and code refactoring, so that developers can keep an eye on their code in real time. Due to this feature, the bug fixing becomes easier as the bad line of codes become easily identifiable. Then this static code analysis feature uncovers more subtle errors in the code even before the code is run.

2. Rich Layout editor - In Android studio, the UI features can simply be dragged and dropped into the emulator making it easier for the developer to create best UI environments according to the app's necessity. Layouts can be previewed and all the screen specification like buttons and Data can easily be verified.

3. Template based wizard: The latest version Android studio comes with a template wizard which has improved functionality and helps the developer to choose android version number and name and different available API's. This wizard has templates for almost all the common elements that are used by the developers and familiarizing oneself with the wizard makes Android studio fun to use.

4. Improved developers console - The developer console for the android studio is optimized for smart features and is already integrated with Google Analytics. It has inbuilt graph generation option and due to Google has access to various services like translation or ad monitoring. This allows the developers to keep an eye on app performance by using a single screen. This helps save time and prevent more overhead.

5. Access to Google services - Some of Google cloud features are directly available to the android studio through the IDE. There are also various plugins available which can be used in an app directly.

Android studio has native IDE so it is highly compatible with the Google products and ecosystem. This is one of its most important features and what made it more popular than other IDE's in the market.

7. How did you design your hierarchy?

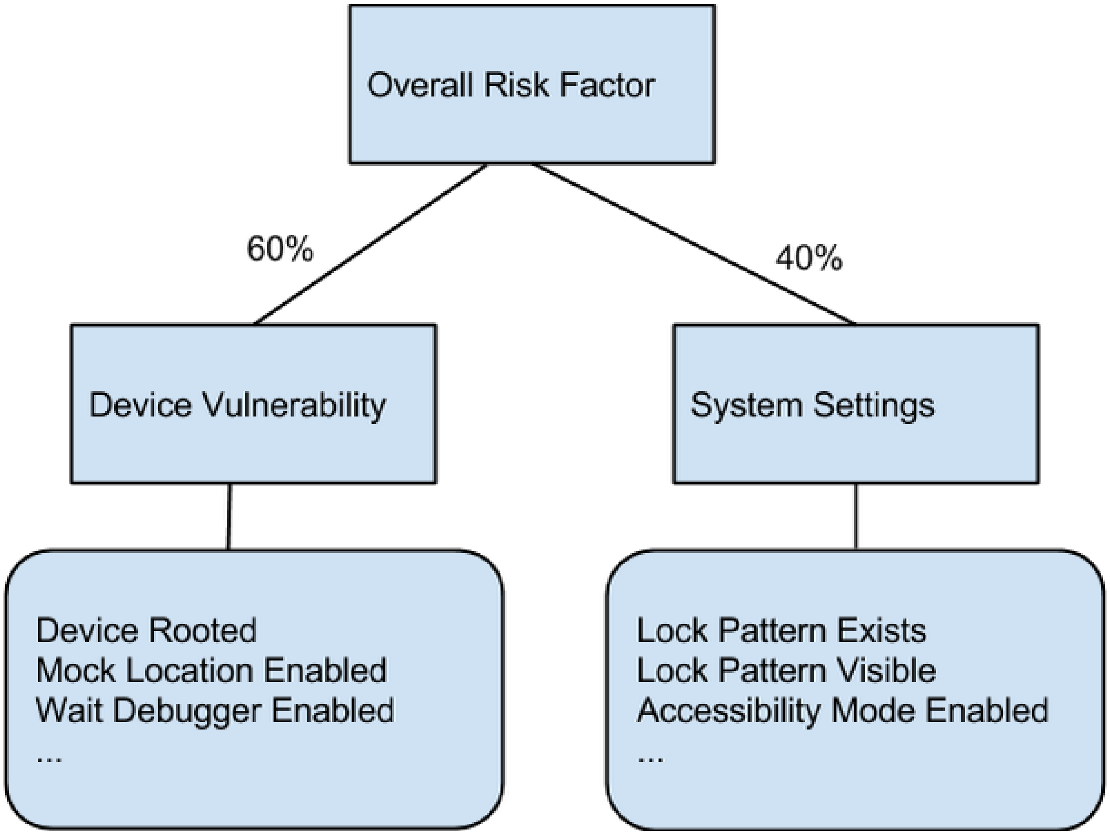

Fig: Hierarchical Design of Confine Me

Our expert system for Android device security evaluation is based on Fuzzy systems. We created a knowledge base that consists of factors that are related to phone model, lock patterns for mobile devices, voice activated passwords, USB debugging and some other factors. These all factors are then assigned different scores according to their severity level.

For example - Custom ROM has a weightage of 7 whereas show Password ON has been given a weightage of 1.

A threshold is then assigned for particular score ranges which show the device vulnerability level and changes the color according to the risk factor. The score for the two sections then add up to the final score and then it is compared with the severity level to show that if the device falls in the risk category or not.

8. What limitations does your design have?

Android OS usage over the corporate world is on the rise, there are a number of application currently on the Play store that are available for free to download, which can be possible be a security threat for the system. Our expert system crawls the data regarding a particular app from the play store which relies completely on the user reviews and feedback present on the Play store. However, not all the reviews present on the Play Store are reliable or trustworthy. There is a high possibility of the reviews as well as ratings feedback about an app being incorrect and no capturing the true performance of the said app.

If you observe from a different angle, these ratings and reviews don't provide any clear idea about the security breach that is possible by installing the said app. The reviews and ratings provide the user with the feedback about the functionality of the app. The access permissions that the app requires could account towards the security issue for the android handheld system. The access permissions govern the accessibility of system files and hence determine any malicious possible activity to hinder the security of the system as whole.

Incorporating a method to determine if the said android system contains the required security settings and a anti-virus checking application installed right away could add on to the security checking system. With the increase in the demand for the android system in many of the corporate scenarios there is an increase in the usage of android system for the entire corporate sector. The increase in malware and Trojan files for the many applications over the android system is also on the rise [3]. The main factor to evaluate the security of an android handheld system is likelihood of it to get infected with such malware or Trojan files, which is not included in our hierarchical security evaluation expert system.

9. What are the most important results?

The main advantages of the android system and probably the primary reason why android demand is on the rise is due to the customizability that it offers. But this freedom of customization comes with a price to pay in the form of security issues. More the customizability, greater is the complexity in access issues over the files on the android system. Many of the android users don't know the basic security settings that can be toggled by them to enhance the security of the system. Basic security practices that users tend to neglect is addressed by our system. The expert system detects the security settings and the possible effects of security enhancement possible for the android system.

One of the most productive results from our expert system is securing the device against malicious applications accessing local data associated with other installed applications. This is done by monitoring the system calls made from the application layer with PIDs greater than 10000. Not only blocking unnecessary calls but also tracing the frequency of each of them allows prevention of a single application considerably using up the device resources. Apart from this, the application data for determining the security evaluation of each application gives a good estimate of the threats associated with the applications installed on the device.

Another important aspect associated with our expert system is the computation of the overall risk factor hierarchically, by integrating the outcomes from all the rules enforced.

10. Which results and solutions are novel and could be patented?

The idea presented through our expert system can be used to enhance the security of an android device. It has been shown that by implementing the given set of metrics, our system can evaluate the security features of a mobile device. Our metrics work on the following principles:

1. Whatever is measured can be shown in terms of numbers, percentages or other scale.
2. Metrics would be useful in evaluating the security of the device
3. Related data should be available to support the metrics.

The security scan and vulnerability scan done using the present set of metrics for security evaluation of a mobile device have not been done before on this scale. This might have been done for the desktop environment but for the handheld devices this matrices have been used for the first time.

The part of our expert system which could be considered for being patented is the one where the kernel-level protection is enforced by monitoring the each kernel call. This can be done by checking various system settings like device provisioning, USB debugging, etc. among checking for other important things like verifying whether the device is rooted or not. This kernel-level protection is given much higher significance due to the fact that it can prevent DOS attacks and restrict undesirable intra-application communication.

Though our expert system does provides us with valuable results when using diverse android devices. It is difficult to comment if any of the ideas here can be patented or not. But it is worthwhile to note that during development of this application, we did not came across any similar tool.

References:

[1] Alessandro Armando, Alessio Merlo, Luca Verdarame, *“An empirical evaluation of the Android Security Framework.”*

[2] Andrew Hoffman, Darrel Pollard and Leon Reznik, *“Hierarchical Security Evaluation Framework and its Implementation on Android Smartphones.”*

[3] Anant Shrivastava, *“Security Issues in Android Custom ROMs”*

[4]*http://www.avocarrot.com/blog/5-killer-features-android-studio-know/*