International Journal of Engineering and Techniques - Volume 3 Issue 1, Jan – Feb 2017

RESEARCH ARTICLE OPEN ACCESS

Rethinking Permission Enforcement Mechanism on Mobile System

Ms.CHITRA.S[1], Ms.ARUNADEVI.R[2]

[1,2](Department of CS & IT, Dhanalakshmi Srinivasan College of Arts & Science for Women, Perambalur-621 212.)

Abstract:

To protect sensitive resources from unauthorized use, modern mobile systems, such a Android and iOS, design a permission-based access control model. However, current model could not enforce fine-grained control over the dynamic permission use contexts, causing two severe security problems. First, any code package in an application could use the granted permissions, inducing attackers to embed malicious payloads into benign apps. Second, the permissions granted to a benign application may be utilized by an attacker through vulnerable application interactions. Although ad hoc solutions have been proposed, none could systematically solve these two issues within a unified framework. The first such framework to provide context-sensitive permission enforcement that regular's permission use policies according to system-wide application contexts, which cover both intra-application context and inter-application context. We build a prototype system on Android , named FineDroid, to track such context during the applicaton execution. To flexibly regulate the context-sensitive permission rules, FineDroid features a policy framework that could express generic application contexts. We demonstrate the benefits of FineDroid by instantiating several security extensions based on the policy framework, for three potential users: end users, administrators and developers. Furthermore, FineDroid is showed to introduce a minor overhead

Keywords: **Sensitive resources, permission, framework, intra-application context, inter-application context, FineDroid.**

I. INTRODUCTION

A mobile phone is a small, resource-restricted and highly security-sensitive device whose fundamental goal is to enable secure services for its users. These devices have been deployed in a large number of heterogeneous industries and used by a large user base. Rethinking permission granting in modern operating systems. Android become very popular because of its various advantages and capacities. There are many ways to get information or data of the use from a service on their mobile phone. The modern mobile systems such as Android, iOS design a permission-based access control model to protect sensitive resources from unauthorized use. In this model, the accesses to protected resources without the accesses to protected resources without permission enforcement system. Since Android has been expanding its market share rapidly as the most popular mobile platform. Ideally, the Android permission model should prevent malicious applications from abusing sensitive resources. However, the current permission model could not enforce a fine-grained control over permission use contexts (in this paper, when we say context we mean the application execution context). As a result, malicious entities could easily abuse permissions, leading to the explosion of Android malware these years and the numerous reported application vulnerabilities.

Since Android has been expanding its market share rapidly as the most popular mobile platform this paper mainly focuses on the permission model of Android. Currently, the coarse-grained permission enforcement mechanism is limited in the following two aspects.

- ***Intra-application Context Insensitive:***

  Current permission model treats each application as a separate principal and permissions are granted at the granularity of application, thus all the code packages in the application could access the protected resources with the same granted permissions. In fact, not all the code packages in a single application come from a same origin.

- ***Inter-application Context Insensitive:***

  Application interaction is a common characteristic of mobile applications. However, this new characteristic is transparent to the current coarse-grained permission enforcement mechanism, exposing a new attack surface, i.e., the permissions granted to a vulnerable application may be abused by an attacker application via inter-application communication.

Given these problems, plenty of extensions have been proposed to refine the Android permission model. Context-aware permission models are proposed to support different permission policies according to external contexts, such as location, time of the day. However, these works still could not address the two limitations described above. There are also some work dedicated to reduce the risk of inter-application communication or to isolate untrusted components inside an application. However, none could achieve unified and flexible control according to the system- wide application context. In this paper, we seek to fill the gap by bringing context-sensitive permission enforcement. We design a prototype, called FineDroid to provide fine-grained permission control over the application context. For example, if app' A is allowed to use SEND_SMS permission in the context C, when app A requests SEND_SMS permission in another context C', it would be treated as a different request of SEND_SMS permission. In FineDroid, we consider both the intra-application context which represents the internal execution context of an application, and the inter-application context which reflects the IPC context of interacted applications. It is non-trivial to track such context in Android.

II. SYSTEM ARCHITECTURE

FineDroid designs several techniques to automatically track such contexts along with the application execution. To ease the administration of permission control policies, FineDroid also features a policy framework which is general enough to express the rules for handling permission requests in a context-sensitive manner.

To demonstrate the benefits of FineDroid, we create two security extensions for administrators and developers. First, since permission leak vulnerability is very common and dangerous, we show how administrators could benefit from our system in transparently fixing these vulnerabilities without modifying vulnerable applications. Second, we provide application developers with the ability of restricting untrusted third-party SDK by declaring fine-grained permission specifications in the manifest file.

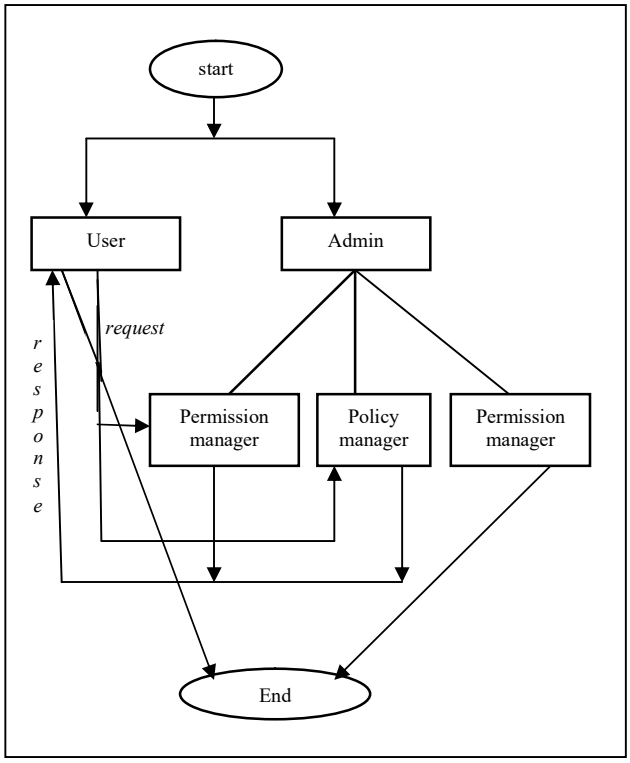

Fig.1 system architecture

A. *USER:*

A user is the client i.e. the person who gives the input data .

B. ADMIN

The admin is the person who manages all the user accounts as well as the information of the places is also updated by him.

C. PERMISSION MANAGER:

To handle a permission request, the Policy Manager module examines all the policies in the system.

D. POLICY MANAGER:

Permission Manager first queries Policy Manager to select a policy which best matches the current application context.

E. PERMISSION RECORD:

permission requests are handled by querying the Permission Record to grant all the permissions declared in the application manifest file.

In this paper, we make the following contributions:

- ✓ *We propose context-sensitive permission enforcement to deal with severe security problems of mobile systems. Considering the characteristics of mobile applications, it is important and necessary to take the application context into account when regulating permission requests.*
- ✓ *We design a novel context tracking technique to track intra-application context and inter-application context during the application execution.*
- ✓ *We design a new policy framework to flexibly and generally regulate permission requests with respect to the fine-grained application context.*
- ✓ *We demonstrate two security extensions based on the context-sensitive permission enforcement system, by just writing policies and sometimes a small number of auxiliary code.*
- ✓ *We evaluate the security benefits gained by the two security extensions and report the performance overhead.*

III. PROBLEM DESCRIPTION

3.1 EXISTING SYSTEM:

In existing permission model, each application is treated as a separate principal, while the above analysis indicates that the application code may be contributed by multiple parties (app developers, SDK providers, repackaged payload, etc). Current permission enforcement mechanism is limited in simply applying a single security policy for the entire app. To identify the leaked permissions or capabilities, they have developed a tool called Woodpecker.

Woodpecker is used to examine how the Android-essential permission-based security model is enforced on current leading Android-based smart phones.

3.1.1 DISADVANTAGE

- ✓ *Malicious entities could easily abuse permissions.*
- ✓ *Leading to the explosion of Android malware.*
- ✓ *A user may wish to grant the SEND_SMS permission only when she clicks the "send" button but in any other contexts of the application.*

3.2 PROPOSED SYSTEM:

We propose context-sensitive permission enforcement to deal with severe security problems of mobile systems. Considering the characteristics of mobile applications, it is important and necessary to take the application context into account when regulating permission requests. We design a novel context tracking technique to track intra-application context and inter-application context during the application execution. Design a novel context tracking technique to track intra application context and inter-application context during the application execution. Design a new policy framework to flexibly and generally regulate permission requests with respect to the fine-grained application context. Context-sensitive permission enforcement to deal with severe security problems of mobile systems.

3.2.1 ADVANTAGES:

- *We design a new policy framework to flexibly and generally regulate permission requests with respect to the fine-grained application context.*
- *We demonstrate three security extensions based on the context-sensitive permission enforcement system, by just writing policies and sometimes a small number of* auxiliary Code.
- *Minor overhead*
- *Seamless context tracking*
- *Security benefits*

IV. METHODOLOGY

1. *Inter-application context insensitive*
2. *Permission manager*
3. *Policy manager*
4. *Permission record*

MODULE DESCRIPTION:

4.1 INTER-APPLICATION CONTEXT INSENSITIVE:

Although application interaction is a common characteristic of mobile applications, it is transparent to the current coarse-grained permission enforcement mechanism, exposing a new attack surface, i.e., the permissions granted to a vulnerable application may be abused by an attacker application via inter-application communication.

4.2 PERMISSION MANAGER:

To handle a permission request, the Policy Manager module examines all the policies in the system, and then Permission Manager could make a permission decision according to the action (e.g. allow or deny) specified in the match policy. Besides, our policy language is extensible for introducing new permission handling actions.

4.3 POLICY MANAGER:

Permission Manager first queries Policy Manager to select a policy which best matches the current application context. If no policy matches, Permission Manager would fall back to the original permission enforcement mode.

To test whether a policy could match a permission request, Policy Manager first checks the requested permission and the requestor application. When both attributes match, Policy Manager further compares the application context. The application context matching is relatively slow, so we use a cache to remember the context matching results. If multiple policies are found to match, Policy Manager would select the one that express the most fine-grained application context. Policy Manager also supports adding and removing policies to/from the system, as well as registering new action types to extend the policy language. The next sect ion will show how these policies can be used to refine current permission model.

4.4 PERMISSION RECORD:

In the original mode, permission requests are handled by querying the Permission Record to grant all the permissions declared in the application manifest file. When a matched policy is selected for the current permission request, Permission Manager just needs to follow the action (e.g. allow or deny) specified in the policy.

V. CONCLUSION

This paper presents FineDroid, which brings context-sensitive permission enforcement to Android. By associating each permission quest with its application context, FineDroid provides a fine-grained permission control. The a pplication context in Fine-Droid covers not only intra-application context , but also inter-application context. To automatically track such application context, FineDroid designs a new seamless context tracking technique. FineDroid also features a policy frame work to flexibly regulate context-sensitive permission rules. This paper further demonstrates the effectiveness of FineDroid by creating two security extensions upon FineDroid for administrators and application developers. The performance evaluation shows that the overhead introduced by FineDroid is minor.

ACKNOWLEDGMENT

The author deeply indebted to honorable Shri A.SRINIVASAN(Founder Chairman), SHRI P.NEELRAJ(Secretary) Dhanalakshmi Srinivasan Group of Institutions, Perambalur for giving me opportunity to work and avail the facilities of the College Campus. The author heartfelt and sincere thanks to Principal Dr.ARUNA DINAKARAN, Vice Principal Prof.S.H.AFROZE, HoD Mrs.V.VANEESWARI, (Dept. of CS & IT) Project Guide Ms.R.ARUNADEVI, (Dept. of CS & IT) of Dhanalakshmi Srinivasan College of Arts & Science for Women,Perambalur. The author also thanks to parents, Family Members, Friends, Relatives for their support, freedom and motivation.

REFERENCES

1. Android Remains the Leader in the Smartphone Operating System Market, Jun. 2015, [online] Available: http://www.idc.com/getdoc.jsp?containerId=prUS24108913.
2. Security Threat Report 2013, Jun. 2015, [online] Available: http://www.sophos.com/en-us/security-news-trends/reports/security-threat-report/android-malware.aspx.
3. Smalley, R. Craig, "Security enhanced (SE) Android: Bringing flexible MAC to Android", *Proc. NDSS*, pp. 20-38, 2013.
4. S. Bugiel, S. Heuser, A.-R. Sadeghi, "Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies", Proc. USENIX Secur., pp. 131-146, 2013.
5. W. Enck, M. Ongtang, P. McDaniel, "Understanding Android security", IEEE Security Privacy, vol. 7, no. 1, pp. 50-57, Jan./Feb. 2009.
6. A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, E. Chin, "Permission re-delegation: Attacks and defenses", Proc. USENIX Secur., pp. 22, 2011.
7. "Sophos security threat report 2013," http://www.sophos.com/en-us/security-news-trends/reports/security-threat-report/android-malware.aspx.
8. M. Grace, Y. Zhou, Z. Wang, and X. Jiang, "Systematic detection of capability leaks in stock android smartphones," in Proc. of NDSS '12.

9. Y. Fratantonio, A. Bianchi, W. Robertson, M. Egele, C.Kruegel, E. Kirda, and G. Vigna, “On the Security and Engineering Implications of Finer-Grained Access Controls for Android Developers and Users,” in Proc. of DIMVA ’15.

10. M. Conti, V. T. N. Nguyen, B. Crispo, "CRePE: Context-related policy enforcement for Android", Proc. ISC, pp. 331-345, 2010.

BIOGRAPHICAL NOTES

Ms.CHITRA.S *is presently pursuing Computer Science M.Sc., Final year the Department of Computer Science From Dhanalakshmi Srinivasan College of Arts and Science for Women, Perambalur, Tamil Nadu ,India.*

Ms.ARUNADEVI.R *- Received M.S.c., M.Phil Degree in Computer Science. She is currently working as Assistant Professor in Department of Computer Science in Dhanalakshmi Srinivasan College of Arts and Science for Women, Perambalur Tamil Nadu, India. She has Published papers in various journals like Mobile Communication in 4G Technology, DNS Support for Load Balancing, Securing Wireless Sensor Networks with Public Key Techniques.*