

REES: Malicious Software Detection Framework for MeeGo-In Vehicle Infotainment

Ho-Yeon Kim*, Young-Hyun Choi*, Tai-Myoung Chung**

*Department of Electrical and Computer Engineering Sungkyunkwan University Suwon, Korea**

*School of Information Communication Engineering Sungkyunkwan University Suwon, Korea***

Suwon, Gyeonggi-do, Republic of Korea

{hykim, yhchoi}@imtl.skku.ac.kr*, tmchung@ece.skku.ac.kr**

Abstract—According to growth in mobile handsets, mobile malware also has been increasing. The target of malware is servers, personal computers, tablets, and smartphone, not excepting the IVI (In-Vehicle Infotainment). In this paper, we analyse MeeGo that is one of the mobile operating system in IVI and propose the REES (Restricted Execution Environment System) that it provides mobility while program is being analysed. To reduce the run time overhead, REES analyse malware on cloud computing.

Keywords— IVI, mobile malware, malicious software

I. INTRODUCTION

The use of mobile devices has rapidly grown in recent years due to their convenience. There has been a dramatic increase in the use of smart phones and tablet PCs. According to Gartner's report, the sales of smartphone were 115million in the third quarter of 2011[1]. This statistic has increased by 41 percent compared to last year. Due to the popularity of smartphone, the target of malware has changed to the mobile handsets from the personal computers. McAfee in "McAfee Threats Report: Third Quarter 2011" reported that the number of mobile malware has been exceeded by 1,200[2]. This statistics is low regarding the total number of malwares-70 million-, however, it is relatively high considering the short period of time mobile devices have becomes so popular.

Vehicles can have many mobile in-car services by vehicle-IT convergence. Therefore, IVI (In-Vehicle-Infotainment) is no exception, because it is kind of mobile platforms[3].

IVI provides a mobile service via mobile network such as sending SMS (short message service), sending MMS (multimedia message service), GPS based service, and running mobile application. A mobile malware which is aimed at IVI can execute the whole mobile malicious behaviour also it can confuse the driver to get into an accident. Let's assume that a particular application measures the speed of a vehicle based on the GPS data and the audio volume is controlled automatically by the speed of the vehicle. This application presents a driving pleasure to the driver. However, malware can be wrongfully used; it can manipulate the volume of the music suddenly, and the driver gets disturbed. Therefore, the malware which is installed in the vehicle is more dangerous than existing malwares.

In this paper, we analyse an access control of MeeGo that is a mobile operating system for IVI and propose the Restricted Execution Environment System (REES) to protect a mobile handsets. The REES is a malware detection system which analyses programs and provides the mobility at the same time.

The rest of the paper is organized as follows. Section II presents an overview of IVI and the MeeGo. Section III analyses the security mechanism of the MeeGo to graft onto the REES. In Section IV, we introduce our framework and conclude this paper in Section V.

II. IN-VEHICLE-INFOTAINMENT

A. IVI(In-Vehicle-Infotainment)

The newly released vehicles are loaded with an IVI to use the entertainment system such as web surfing, sending SMS/MMS, watching DVD, connecting to mobile devices, and use of navigation. The IVI uses the cellular network, such as 3G and 4G, instead of V2I (Vehicle-to-Infrastructure). The utilization of IVI is shown in Figure 1.

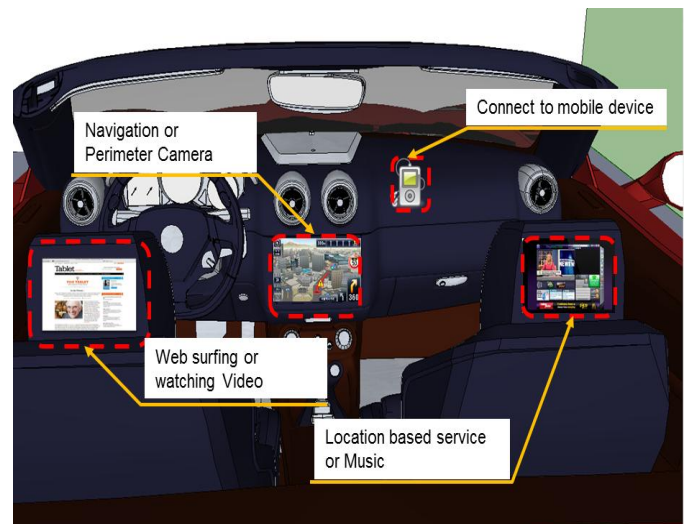


Figure 1. Practical use an IVI

A various mobile operating system can be used for IVI, for instance, Android[4], Tizen[5], Windows Embedded Automotive 7[6] and MeeGo[7]. In this paper, we analyse MeeGo that offers the mobile operating system for IVI adopted by the famous manufacturers.

B. MeeGo

MeeGo is a mobile operating system based on Linux[8]. It has been developing under the leadership of two companies Nokia and Intel. MeeGo was first announced in Feb. 2010 and it is open-source platform that includes core OS, UI libraries and tools. Moreover, MeeGo supports a magnitude of mobile client handsets such as Smartphone, Smart TVs, Tablets, Netbook, and IVI.

One of the most distinctive features of MeeGo is flexibility. MeeGo supports the Atom processor and the ARM processor because it combines the Moblin and Meamo into one. Moblin was developed under the leadership of the Intel and it is suitable Atom processor. Meamo is a Nokia's mobile operating system project. Meamo was designed to fit the ARM processor as opposed to Moblin. For this reason, MeeGo can use these processors and become more applicable to the multiple mobile handsets than any other mobile operating systems.

III. SECURITY MECHANISM OF MEEGO

First of all, to grafting REES onto MeeGo, we need to understand the security mechanism of MeeGo. In this section, we analyse an Access Control (AC) which is one of the security mechanism in the MeeGo. AC can manipulate malware to access to unauthorized resources, still private information, modify setting options, and a hijack administrative privileges.

MeeGo provides a Mobile Simplified Security Framework (MSSF) to protect something from attacker and malicious behaviour[9]. The typical MSSF consists of Integrity Protection, Secure SW Distribution, Crypto Services, TEE

Services, Certificate Manager, and Access Control. Figure 2 shows that the components of the MSSF. The filling in the Figure 2 indicates the access control.

To prevent an unauthorized user or program, MeeGo uses following credentials: Resource Token and Application Identifier instead of typical access control mechanisms such as User ID (UID), Group ID (GID), POSIX capabilities, and Supplementary groups in UNIX operating systems.

Resource token is one of the credentials that indicates a string and is understandable for both users and developers. For instance, resource token is "cellular" and then this application gets a permission for the GSM* call, SMS and GPRS data. Resource token can be divided into two classes: global resource token and package specific resource token. The global resource token is defined as MeeGo and the package specific resource token is defined as the developers. The resource token of the application must locate at Manifest file. Figure 3 and Figure 4 indicate the example of manifest mapping.

```
<aegis>
  <provide>
    <credential name="UserData"/>
  </provide>
</aegis>
```

Figure 2. Server defines resource token "UserData"

```
<aegis>
  <request>
    <credential name="server-pkg::UserData"/>
    <credential name="Cellular"/>
    <for path="/usr/bin/udmanager"/>
  </request>
</aegis>
```

Figure 3. Client defines resource token "UserData" and "Cellular"

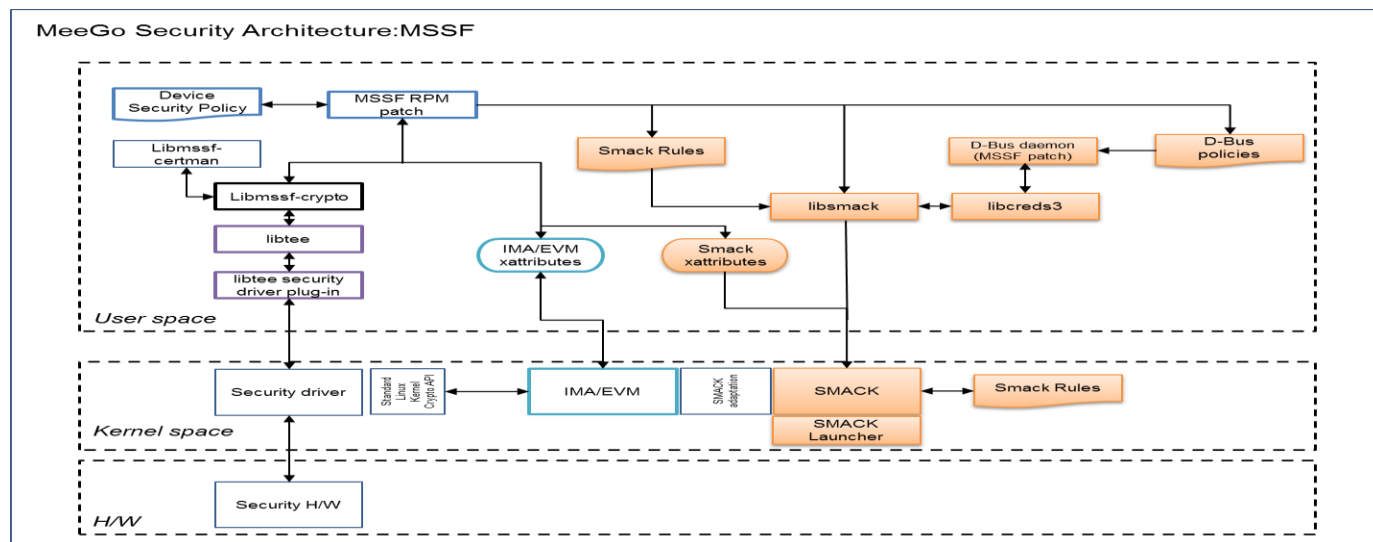


Figure 4. Security Architecture on MeeGo

$AppID = \{SourceID, PackageName, ApplicationSpecificName\}$

Figure 5. Elements of application identifier

Application identifier is one of the credentials. It is generated as a package manager. Figure 4 represents the three elements of application identifier.

An application identifier should not be counterfeit and should be impossible to modify when mobile devices are rebooted and applications are installed.

MeeGo implements the Simplified Mandatory Access Control Kernel (SMACK) using the resource token and application identifier that it is one of the Linux Security Module (LSM). Subject label which is a variable of SMACK is used as an application identifier and Object label is used as a resource token.

The resource token has some problems. When user installs a malicious application, he/she does not take care about the application's permission. In this case, an attacker uses a social engineering. An attacker can repackage a popular application to avoid a doubt and it plays on the user's security weakness. As a result, the user installs applications without any doubt. For this reason, the resource token is not suitable since it is too dependent on the users.

IV. OVERVIEW OF REES

There are two kinds of malware analysis method: static analysis and dynamic analysis[10]. In addition, hybrid analysis also exists that consists of a static analysis and dynamic analysis.

A static analysis detects based on signature. An advantage of the static analysis is a low false positives and it shows a higher speed than other analysis methods. However, the static analysis cannot detect a new generation of malware such as zero-day attack and packed malware.

The dynamic analysis analyses the code during run-time. Thus, dynamic analysis is immune to obfuscation attempts and has no problem with self-modifying programs[11]. Further, the dynamic analysis is not only able to detect an unknown malwares, but also detects potential threats. On the other hand, the dynamic analysis has possibility of having false negatives. Moreover, the dynamic analysis takes a longer time-cost, because it should has to confirm the all the aspects of behaviours. Accordingly, the dynamic analysis requires other technologies. Although several studies have been made on the dynamic analysis methods using cloud computing to reduce the overhead run-time, there is little consideration on how mobility guarantees while program is being analysed[12][13].

As a consequence, we propose the REES [14]. The REES use a cloud computing to reduce data analysis time and use the hybrid analysis to decrease a false positive and false negative.

The typical structure of REES is shown in Figure 6. In the REES, an IVI receives an external input then REES constructs a restricted execution environment and upload external input to the cloud system at the same time.

The REES is divided into three domains: IVI, restricted execution, and cloud system domain.

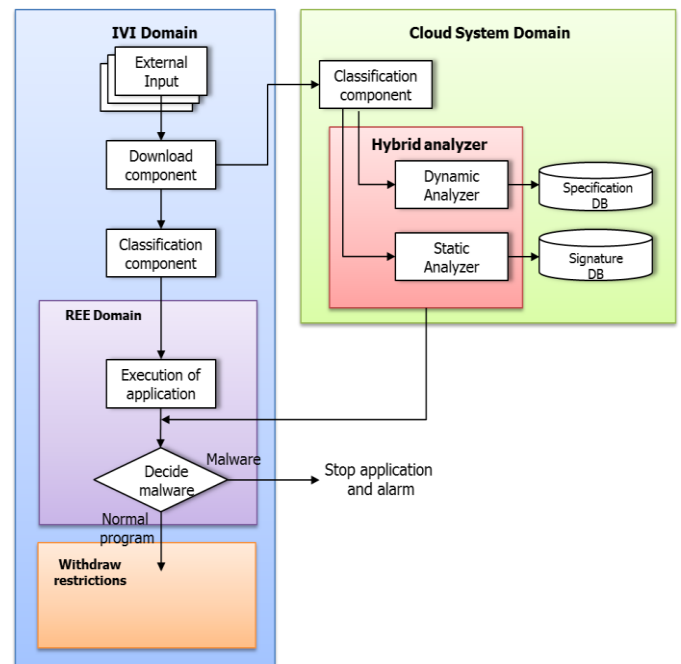


Figure 6. Framework: REES

A. IVI Domain

An IVI domain is divided into two components: download component and classification component.

1) Download component: Download component communicates with the cloud system. To the dynamic analyse, it sends an external input data with the environmental variables of IVI such as the list of installed application, currently stored memory information, list of listening port, and etc. However, sending data caused communication overhead. Thus, the sending data should be reduced.

2) Classification component: Classification components decide the type of data and service. As a result, the classification component constructs a suitable restricted execution environment.

B. Cloud Computing Domain

A cloud computing domain is an environment for conducting malware analysis. In the traditional dynamic analysis, all the real-time variables are required. However, REES does not use all variables, because it analyses the malware on the cloud system instead of the real-machine. If all variables are used, then IVI should be connected to the cloud system at all times. It will increase an unnecessary time-overhead, therefore we ignore some variables.

If we use the cloud system, it can bring several and those are as follows:

1) An improved performance: The cloud system can provide a large-scale distributed computing by using a parallel processing among the thousands of nodes. If we use a hybrid analysis, we are able to detect malware more quickly.

TABLE 1. RESTRICTION EXECUTIONS

	Restrict sharing information between applications	Restrict downloading image	Restrict access URL	Restrict SMS	Restrict file transfer	Restrict system manager access
Execute application through an external storage	Yes	No	Yes	Yes	No	Yes
Download a file though an e-mail	Yes	Yes	Yes	No	No	No
SMS receipt that included the URL	No	No	Yes	No	No	No
Receive MMS	No	Yes	Yes	No	No	No
Using Bluetooth	No	No	No	Yes	Yes	Yes
Access the Internet	No	Yes	No	Yes	No	Yes

- 2) **A free of analysis environment from limited resources:** An IVI should guarantee mobility. Therefore, IVI has been developed with low battery and low performance that there is trade off relation between performance and mobility. A cloud computing can analyse without hardware limitation.

C. Restricted Execution Environment Domain

Restricted execution environment is constructed to guarantee the mobility. In this domain, a restricted execution offers to various environments an according to external input, because the malicious behaviour is classified a type of services.

The Table 1 shows that the elements of restricted executions. The column indicates a service and the row represents elements of restricted execution each service. For instance, when file download via an e-mail, it is restricted sharing information between applications, downloading an image, accessing an URL. The elements are described as following:

- 1) Restrict sharing information between applications: In the past years, a malware is divided into modular to avoid anti-virus software. Therefore, we restrict to sharing information among applications
- 2) Restrict downloading image: Sometimes, malicious software is used image to hide of malicious URL and itself. So, we restrict to prevent infection of malicious software during download an image.
- 3) Restrict access URL: Recently, a malicious software abuses SSM and MMS to access to malicious web site. Also, a malicious software is using shorten URL to avoid user's doubt. Hence, we restrict an access to URL to protect an insert malicious script and download malicious software.
- 4) Restrict SMS: Mobile malicious software can send SMS which higher fee billing. So we restrict sending SMS to protect unreasonable billing.

- 5) Restrict file transfer: A use of unauthorized Wi-Fi or Bluetooth is helpful malicious execution smoothly. At this moment, malicious software can send private information to attacker. For this reason we restrict a file transmission.
- 6) Restrict system manager access: An excessive use of system manager reduces mobility and may be affect other processes. As it turned out, we define a restriction of an access to system manager.

V. CONCLUSIONS

In this paper, we propose the malware detection scheme that it is called REES and we analyse MeeGo that provide mobile operating system for IVI. The REES offers to mobility while program is being analysed and uses cloud computing system to improved performance. In our future works, we plan to an implementation of the REES and upgrade an element of restricted execution to provide for better mobility.

ACKNOWLEDGMENT

This work (Grants No. 000443010111) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2011.

REFERENCES

- [1] Gartner, Available: <http://www.gartner.com>, 2011.
- [2] McAfee Inc., "McAfee Threats Report: Third Quarter 2011," McAfee, 2011.
- [3] ---, "Caution: Malware Ahead-An analysis of emerging risks in automotive system security", McAfee, 2011.
- [4] The Android website, Available: <http://developer.Android.com>, 2011.
- [5] The Tizen website, Available: <http://www.tizen.com/>, 2011.
- [6] Microsoft Inc., "Microsoft Auto 3.1 Platform Overview-Proven technology adapted for the auto industry," Microsoft Auto, Nov. 2008.
- [7] The MeeGo website, Available: <http://www.meeGo.com/>, 2011.
- [8] Ibrahim Haddad, "Introduction to the MeeGo Project," The Linux Foundation, 2011.
- [9] M. D. Ernst, "Static and dynamic analysis: Synergy and duality," ICSE Workshop on Dynamic Analysis, pp. 24-27, 2003.
- [10] Nokia Inc., "Mobile Simplified Security Framework MSSv2," MeeGo conference, Mscow, Russia, pp. 24-27, May. 2011.
- [11] U. Bayer, *et al.*, "Dynamic analysis of malicious code," Journal in Computer Virology, vol. 2, pp. 67-77, 2006.
- [12] B. G. Chun, *et al.*, "Clonecloud: Elastic execution between mobile device and cloud," in Proceedings of the Sixth Conference on Computer Systems, pp. 301-314, 2011.

- [13] L. Martignoni, *et al.*, "A framework for behavior-based malware analysis in the cloud," Information Systems Security, pp. 178-192, 2009.
- [14] H. Y. Kim, *et al.*, "Using a Restricted Execution to Detect a Mobile Malware," in International conference on Ubiquitous Information technologies and Applications, pp. 252-255, 2011.