International Journal of Smart Home
Vol. 9, No. 10, (2015), pp. 117-124
http://dx.doi.org/10.14257/ijsh.2015.9.10.13

Layered Execution Event Detection System of Harmful Streaming Contents Activated from Android-based Smart Devices

Han Seong Lee and Hyung-Woo Lee

Division of Computer Engineering, Hanshin University, Yangsan-dong, Osan, Gyyeonggi, 447-791, Rep. of Korea
jkhanseong@naver.com, hwlee@hs.ac.kr

Abstract

The recent increase in the proliferation of Android-based smart devices has resulted in easy access to harmful content, such as pornographic videos, becoming a major problem. Thus, the need has also arisen for software that can automatically block pornographic videos harmful to teenagers. In this paper, we analyze the mechanism by which streaming content is executed in Android-based smart devices and, consequently, propose a mechanism that automatically detects when harmful content such as a pornographic video is being viewed on a smart device. On the basis of the results of analysis of the mechanism by which harmful streaming content such as a pornographic video is executed on Android-based smart device, we developed and implemented a mechanism that extracts and analyzes the characteristics of internal events that occur at the time a harmful streaming content is executed. The mechanism developed is able to differentiate such content from safe streaming content.

Keywords: *Smart Devices, Streaming Contents, Event Detection, Execution Event, Android*

1. Introduction

The use of smartphones is rapidly increasing because they enable users to easily access desired information at any time and from anywhere. Over 30 million people are using smartphones in Korea, and the number continues to rise. The Android operating system is used in 55% of all smart devices, which means that many people use Android-based smartphones. Regardless of age and gender, most people use smart devices in Korea, and the number of teenage smartphone users has increased sharply as well. A smart device is beneficial because users can use the services that were originally only available on a PC-based desktop environment through a convenient interface that is similar to that of a PC, irrespective of time and location. While smartphone users can access useful information such as weather and traffic, teenagers have also been accessing harmful sites and viewing pornography and other harmful content in increasing numbers. The use of smartphones by teenagers is more common now, with more than 70% of teenagers possessing smartphones where they can easily access harmful content such as pornography [1]. According to recent statistics, the frequency with which teenagers view pornographic videos is 73% higher during a school break than during school period. The prevalent use of smartphones provides teenagers with more opportunities for viewing adult content. Therefore, there is a need to block adult content on smartphones because, if a teenager repeatedly indulges in adult content, s/he may become addicted and show aggressive behaviors because of the sexuality and violence intrinsic in such adult content. Furthermore, teenagers may lose self-control, leading to copycat crimes and other deviant behaviors. Therefore, effective methods are needed in blocking the execution of harmful content such as pornography from being executed on the smartphones of teenagers.

ISSN: 1975-4094 IJSH

In this paper, we report on software that we have developed for Android-based smart devices – which accounts for 55% of the Korean smart devices market – that can detect execution events in streaming videos and extract the characteristics of harmful content such as pornography at the time of execution. This software extracts the characteristics of major events that occur when a video is streamed on Android-based smart devices based on analysis of running video contents via a streaming service on the smart device. With this function, we propose a fundamental method of blocking harmful content such as pornography on Android-based smart devices.

2. Harmful Streaming Contents on Android-based Smart Devices

Through open markets such as Google Play, anyone can easily download a streaming media player app for an Android-based smart device. Smartphone users of any age can easily access videos through streaming apps such as Hubi, DirectTV, and MX Player as shown in Figure 1. They can also download media players such as MX Player, VLC Stream Player, and Live Stream Player on their smartphones and then easily access streaming videos regardless of time and location. The rating allowable for installing an app on a smart device can be set and the installation of applications containing certain content controlled according to their ratings. However, content filtering is fundamentally limited in restricting access to harmful content because device users can arbitrarily change or reset the settings [2]. Therefore, determining the harmfulness of streaming content after directly distinguishing or monitoring the content at the time of execution on the smart device can be considered the core technology needed.

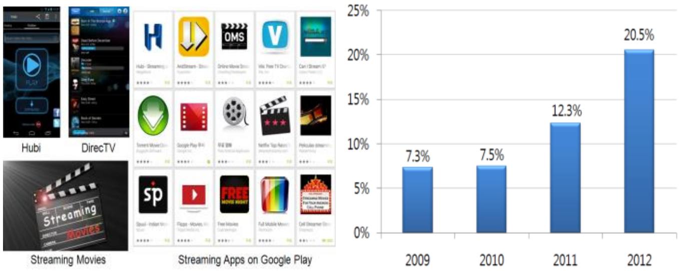


Figure 1. Streaming Apps for Android Devices and Harmful Media Contents Access Rate by Teenagers

According to the Ministry of Gender Equality and Family in Korea, the number of teenagers accessing harmful media through smartphones continues to increase every year as in the above Figure 1. Of the mobile apps that the teenagers use, approximately 6% are adult apps, and their access rate to illegal overseas pornographic sites is at 5%. The main media through which the teenagers access adult content on their smart devices are P2Ps, file sharing sites, and Internet cafes. Easy access to harmful content has resulted in numerous problems, such as deterioration of family relationships and verbal abuse. When a teenager has an existing P2P account, signing up for a mobile P2P account on the same site on a smart device is very easy and simple, and s/he can easily access adult content without any additional authentication process. Moreover, most smart devices with teenage users have no harmful content-blocking service installed such as Smart Sheriff, which is freeware [2].

A harmful content-blocking method can also be set when setting up an Android-based smart device [3]. For the iOS and Android platforms, the user can control whether or not a service is provided based on its adult content rating during the web connection stage or via app installation settings. However, users can easily bypass such methods using an online SSL web proxy server [4]. Accordingly, hardware-based approaches that depend on the existing setting process have the fundamental problem of being unable to block access to harmful content on smart devices. Therefore, the most desirable approach is to determine the harmfulness of the streaming content after directly distinguishing or monitoring it at the time of its execution on the smart device.

3. Analysis of Streaming Contents Execution Architecture on Android-based Smart Devices

3.1. Analysis of Android Streaming Service Architecture

The basic architecture for providing a streaming service on an Android-based smart device includes binding Inter-Process Communication (IPC) with AudioFlinger on MediaPlayer and calling a Kernel Driver using a Media Framework as illustrated in Figure 2. In the figure, MediaPlayer.java is called by the media player app and then MediaPlayer.cpp is called to be bound with MediaPlayerService, which is contained in the media framework in the Android platform.

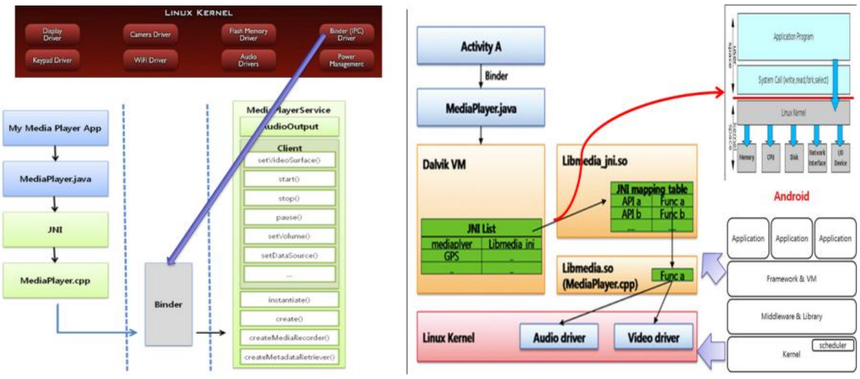


Figure 2. Internal Architecture of the Android Streaming Service

When a media service is running, a Java-based API call executes a binding process through IPC with the MediaPlayerService that is included in the media framework of the Android platform. The details of the internal architecture of the media framework in the Android platform are as follows: The streaming service is executed using the Android media framework and, as shown in Figure 2, to play media, the NuPlayer Driver contained in MediaPlayer Factory is called by MediaPlayerService in order to stream data.

There are four types of media player engines (see Table 1 below), one of which can be used for streaming services. NuPlayer is the core module that mainly executes HTTP Live Streaming (HLS) and RTSP streaming services. Hence, to discover when normal streaming video or harmful content such as pornographic streaming video services is actually being executed on an Android-based smart device, the events that call NuPlayer have to be analyzed.

Table 1. Android Media Framework Architecture

Player engines	Explanation
STAGEFIGHT_PLAYER	an Android Jelly Bean version of basic player engine. It plays all local media files ept MIDI and remote media based on the HTTP PD method.
NU_PLAYER	a player engine that streams HLS (http://*.m3u8) and RTSP streaming (rtsp://).
SONIVOX_PLAYER	a player engine that plays an MIDI file.
PV_PLAYER	an OpenCore player engine that had been used before the Gingerbread versions, but not d anymore.

The sequence through which a multimedia media player is called on an Android device is explained in detail as follows. As stated above, a call is made to NuPlayer from submodules of MediaPlayer Factory, followed by video decoding. Then, the streaming information received from the remote media server is sent to each user's device [3].

3.2. Analysis of Android Streaming Service Execution

Execution of the streaming service based on the media framework begins with a call to NuPlayer Driver in MediaPlayer Factory. Then, NuPlayer, which plays HLS and RTSP streaming services, is activated [5]. Execution of streaming content on an Android smart device is divided into two phases: execution of streaming video saved on the smart device and acquisition of streaming video by remotely connecting to its source URL. Harmful content is primarily available through the latter phase, i.e., the video is transmitted from a remote server by connecting to the URL. NuPlayer is used to play HLS and RTSP streaming. SonivoxPlayer and StageFrightPlayer are used to play other video files.

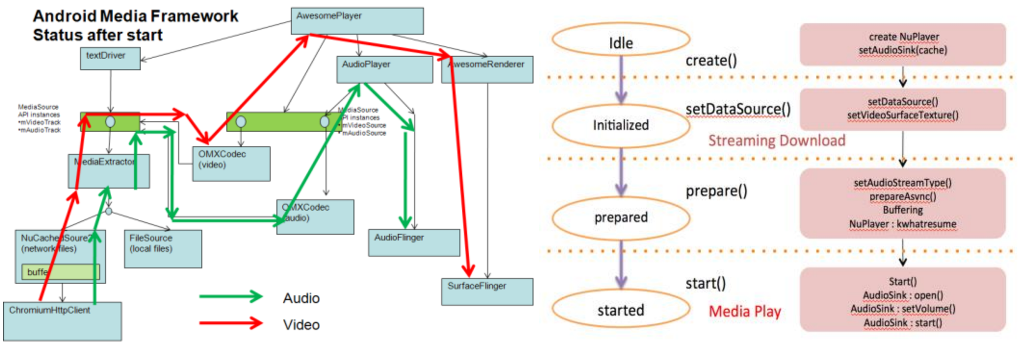


Figure 3. Streaming Execution Process based on Android Media Framework

Figure 3 illustrates the process by which streaming video data are executed on the Android media framework. As shown in the figure, for an HLS connection service based on NuPlayer, streaming data is received and played using the media framework on the Android device.

A streaming service on an Android-based smart device is, in most cases, accessed via a remote URL connection. Thus, a user is provided with a video streaming service on a device through a sequential process comprising four components: a kernel, framework, application, and network, (as depicted in Figure 4). To confirm the time of execution of a streaming service on an Android-based smart device, upper-layer and lower-layer system events that are called through the kernel and interface need to be analyzed. Upon achieving this, we can then build a mechanism that can automatically determine when harmful content is being executed.

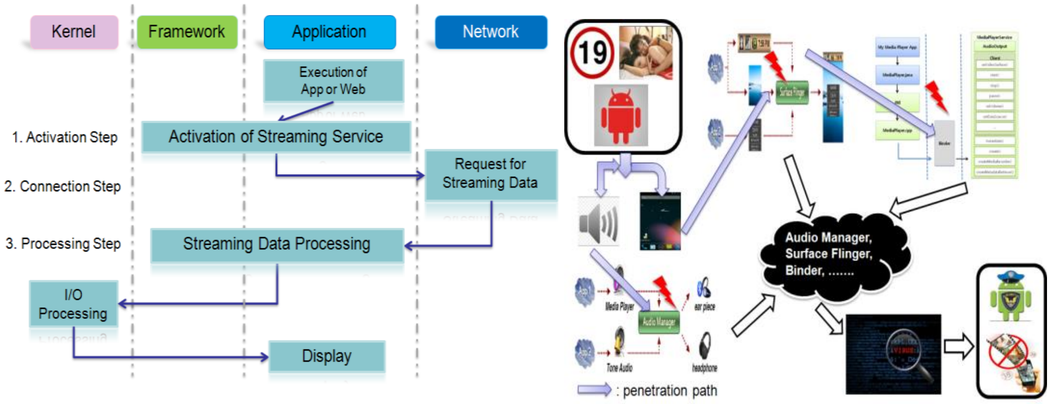


Figure 4. Streaming Execution Process on Android

Based on the internal structural characteristics for executing a streaming service on an Android-based smart device, we can build a mechanism that can automatically determine when harmful content such as pornography is being executed.

4. Execution Event Detection of Harmful Streaming Contents

4.1. Execution Event Activated from Harmful Streaming Contents

To detect when harmful content such as pornographic videos is being executed on an Android-based smart device, upper-layer events (LogCat Event) and lower-layer events (System Call Event) must first be collected (Step 1). Those events occur in relation to the execution of streaming content on a device.

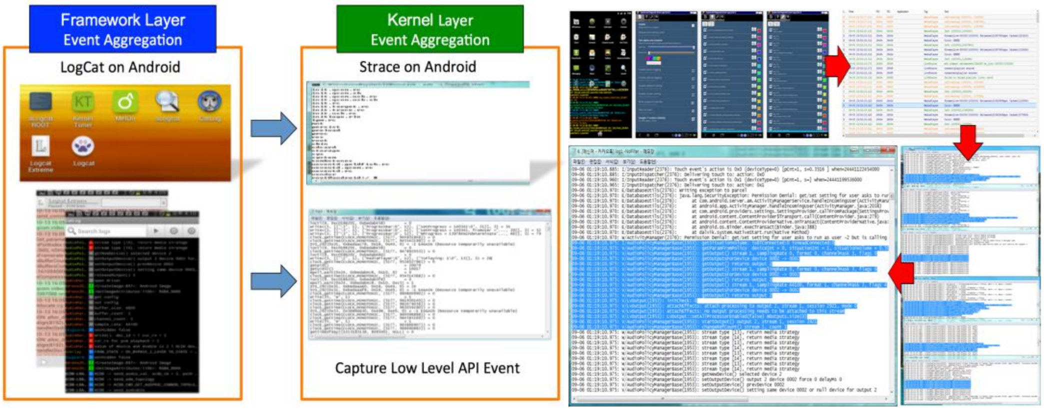


Figure 5. Streaming Event Aggregation Step on Smart Device

In Step 2, the two types of events occurring in the upper- and lower-layers are divided into groups, and interactions among the events in each group are extracted. In Step 3, the characteristics of the upper-/lower-layer events occurring when a streaming service is executed on the smart device are extracted.

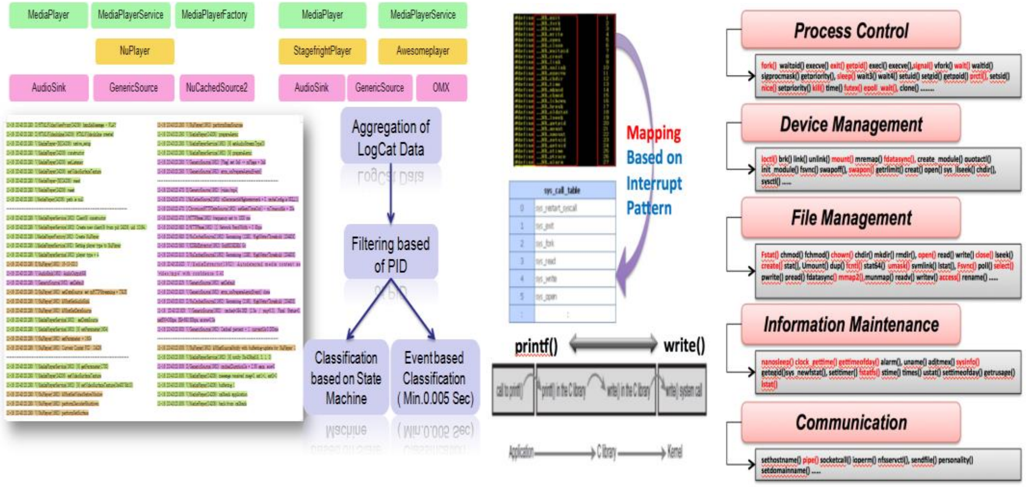

MediaPlayer
MediaPlayerService
MediaPlayerFactory
NuPlayer
AudioSink
GenericSource
NuCachedSource2
MediaPlayer
MediaPlayerService
StagefrightPlayer
Awesomeplayer
AudioSink
GenericSource
OMX
Aggregation of LogCat Data
Filtering based of PID
Classification based on State Machine
Event based Classification (Min.0.005 Sec)
sys_call_table
Mapping Based on Interrupt Pattern
printf()
write()
Application
C library
Kernel
Process Control
fork() waitpid() execve() exit() getpid() execl() execve(),signal() vfork() wait() waitid() sigprocmask() getpriority(), sleep() wait3() wait4() setuid() setgid() getppid() prctl(), setsid() nice() setpriority() kill() time() futex() epoll_wait(), clone() ......
Device Management
ioctl() brk() link() unlink() mount() mremap() fdatasync(), create_module() quotactl() init_module() fsync() swapoff(), swapon() getrlimit() creat() open() sys_llseek() chdir(), sysctl() .....
File Management
Fstat() chmod() fchmod() chown() chdir() mkdir() rmdir(), open() read() write() close() lseek() create() stat(), Umount() dup() fcntl() stat64() umask() symlink() lstat(), Fsync() poll() select() pwrite() pread() fdatasync() mmap2(),munmap() readv() writev() access() rename() ....
Information Maintenance
nanosleep() clock_gettime() gettimeofday() alarm(), uname() adjtimex() sysinfo() getegid()sys_newfstat(), setitimer() fstatfs() stime() times() ustat() settimeofday() getrusage() lstat()
Communication
sethostname() pipe() socketcall() ioperm() nfsservctl(), sendfile() personality() setdomainname() ....

Figure 6. Layer-based Streaming Event Grouping Step on Smart Device

4.2. Execution Event Detection Aggregated from Harmful Streaming Contents

Using the detection database (DB) built through the three-step detection process outlined above, when a streaming service is executed on an Android smart device, unlike with existing techniques [4], the characteristics of the event sets can be extracted in real-time. The system architecture of our proposed detection software and its implementation are illustrated in Figure 7.

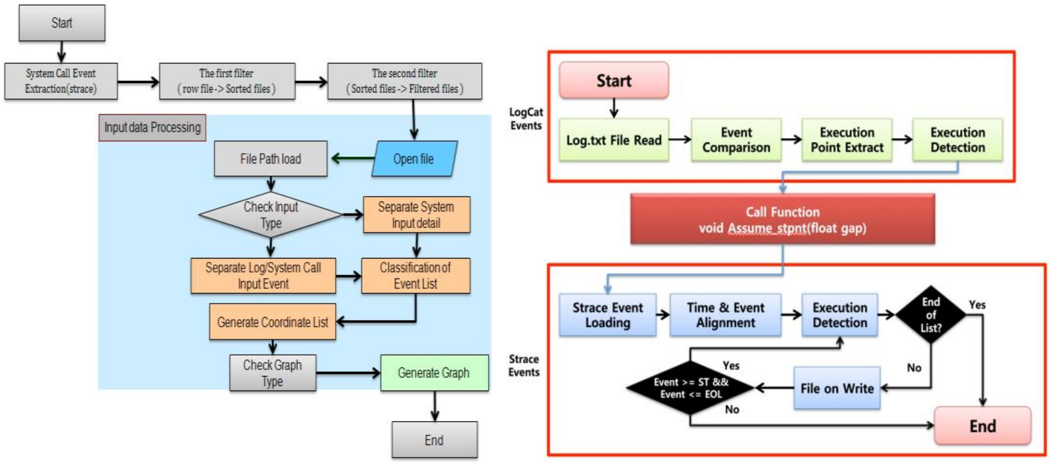
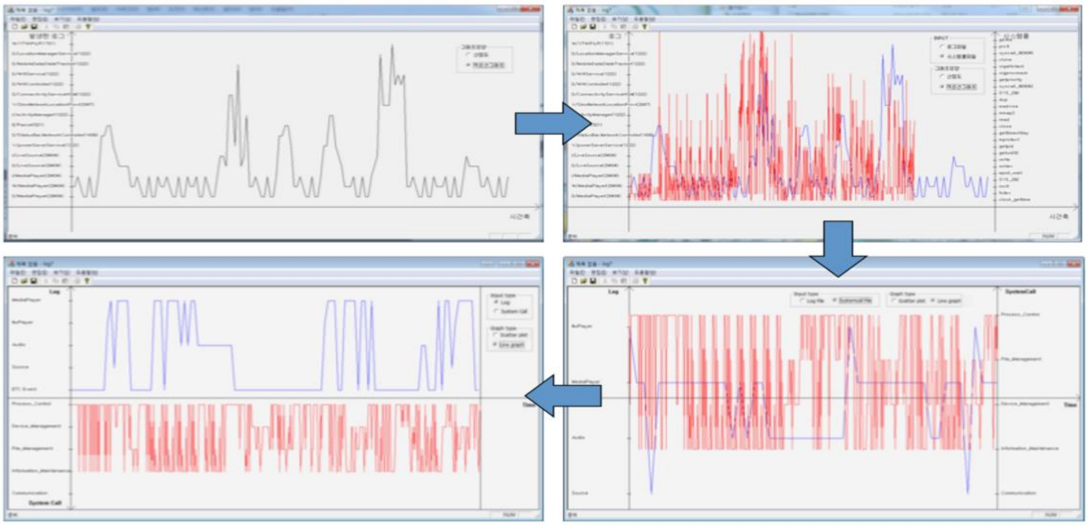
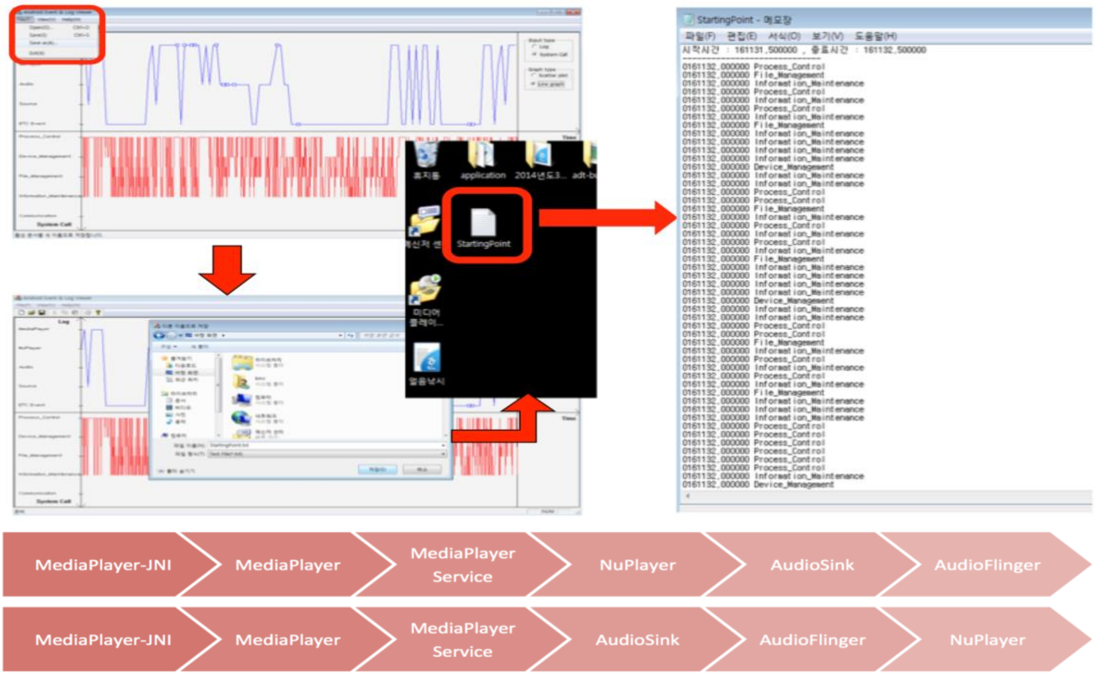


Figure 7. Streaming Execution Detection Algorithm and its Analysis Procedure

By grouping the details in the two types of event sets that occur in the upper-/lower-layers, interactions among the events in each set can be analyzed and the event characteristics corresponding to the execution of a streaming service on a smart device ultimately extracted. The process utilized in the analysis of the execution events is illustrated in Figure 7. Using the detection database (DB) built through the three-step detection process outlined above, when a streaming service is executed on an Android-based smart device, unlike existing techniques [4], the characteristics of the event groups can be extracted in real-time. We implemented detection software with the structure shown in Figure 8.





```
시작시간 : 161131.500000 , 종료시간 : 161132.500000
------------------------------
0161132.000000 Process_Control
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Device_Management
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Device_Management
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 File_Management
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Information_Maintenance
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 Process_Control
0161132.000000 Information_Maintenance
0161132.000000 Device_Management
```

MediaPlayer-JNI → MediaPlayer → MediaPlayer Service → NuPlayer → AudioSink → AudioFlinger

MediaPlayer-JNI → MediaPlayer → MediaPlayer Service → AudioSink → AudioFlinger → NuPlayer

Figure 8. Harmful Streaming Content Execution Event Extraction Tool and Detection Results

5. Conclusions

In this paper, we presented a method that automatically detects when harmful content such as pornographic videos are being executed through a streaming service on Android-based smart devices. We implemented the method as a mechanism that detects when the harmful streaming content is being executed based on events in the upper and lower execution layers of Android-based smart devices. We also described our implemented mechanism and software detect when harmful content is being executed. Finally, we confirmed experimentally that the system is able to distinguish between the execution of safe and harmful video content according to their different event patterns.

Acknowledgments

This research was partially supported by Grant of Hanshin University.

References

[1] E. H. Choi, H. S. Hwang and C. S. Kim, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces," International Journal of Information and Communication Engineering, vol. 9, no. 4, **(2011)**, pp. 358-362.

[2] J. G. Proakis, "Digital Communications, 4th ed.", New York, NY, McGraw-Hill, **(1993).**

[3] J. L. Hennessy and D. A. Patterson, "Instruction-level parallelism and its exploitation," in Computer Architecture, A Quantitative Approach", 4th ed., San Francisco, CA: Morgan Kaufmann Pub., **(2007)**, pp. 66-153.

[4] A. Hashmi, H. Berry, O. Temam and M. Lipasti, "Automatic abstraction and fault tolerance in cortical microachitectures", in Proceeding of the 38th Annual International Symposium on Computer Architecture, New York: NY, **(2011)**, pp. 1-10.

[5] B. Alavi, "Distance measurement error modeling for time-of-arrival based indoor geo location", Ph.D. dissertation, Worcester Polytechnic Institute, Worcester: MA, **(2006)**.

[6] Y. Z. Ben, D. K. John and Anthony, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", University of California, Berkeley: CA, Technical Report CSD-01-1141, **(2001).**

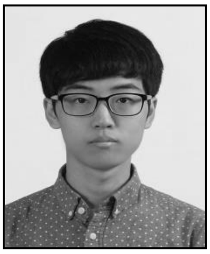
[7] "Malardalen Real-Time Research Center, The worst-case execution time (WCET) analysis project", [Internet], Available: http://www.mrtc.mdh.se/projects/wcet/.

[8] H. Nowakowska, M. Jasinski, P. S. Debicki and J. Mizeraczyk, “Numerical analysis and optimization of power coupling efficiency in waveguide-based microwave plasma source”, IEEE Transactions on Plasma Science [Online], vol. 39, no. 10, **(2011)**, pp. 1935-1942, Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6003795.

[9] L. A. Artem, Y. Aitzhan, “Extracting Heart Rate Variability from a Smartphone Camera”, Journal of information and Communication Convergence Engineering, vol. 11, Issue 3, **(2013)**, pp. 216-222.

[10] H. S. Lee, H. -W. Lee, “Harmful Streaming Contents Event Analysis Mechanism for Detection of Execution on Android-based Smart Devices”, 2015 International Conference on Future Information & Communication Engineering (ICFICE2015), **(2015)**.

Authors



Han Seong Lee, Currently he is an undergraduate student at the Division of Computer Engineering, Hanshin University, Gyyeong-gi Province, Korea. His research interest is in the areas of Android Smart Device, Network and Java Programming for Computer Security & Digital Forensics.

Hyung-Woo Lee, He received B.S, M.S and Ph.D degrees in Computer Science from Korea University, Seoul, Korea, in 1994, 1996 and 1999 respectively. Currently he is a Professor at the Division of Computer Engineering, Hanshin University, Gyyeong-gi Province, Korea. His research interest is in the areas of Network Security, Cryptography and Computer Forensics.