

스마트 단말에서의 이벤트 기반 스트리밍 콘텐츠 재생 시점 탐지 방법

나승재*, 서창욱*, 김민철*, 이형우°, 정치윤**, 한승완**

*°한신대학교 컴퓨터공학부

**한국전자통신연구원 SW·콘텐츠연구소 사이버보안기반연구부 휴먼인식기술연구실

e-mail: asd268@hs.ac.kr*, scw0531@naver.com*, glfrptnjdnl1@naver.com*, hwlee@hs.ac.kr°,

iamready@etri.re.kr**, hansw@etri.re.kr**

Streaming Contents Execution Point Analysis using Activated Events on Smart Devices

Seung Jae Na*, Chang Wook Seo*, Min Cheol Kim*, Hyung-Woo Lee°,

Chi Yoon Jeong**, Seungwan Han**

*°School. of Computer Engineering, Hanshin University

**Human Identification Research Section, Cyber Security Core Technology Research Department,

SW·Content Research Laboratory, ETRI

● 요약 ●

최근 청소년들의 스마트 단말 사용이 급격히 증가하면서, 스마트 단말을 통하여 유해 콘텐츠에 접근하는 비중이 점점 더 늘어나고 있다. 이에 본 논문에서는 스마트 단말에서 스트리밍 콘텐츠가 실행되는 메커니즘을 분석하고, 이를 기반으로 실행 이벤트의 연관성을 분석하여 스트리밍 콘텐츠가 재생되는 시점을 자동으로 탐지할 수 있는 방법을 제안하였다. 제안된 방법을 사용하면 스마트 단말에서 스트리밍 유해 콘텐츠를 효과적으로 차단 할 수 있을 것으로 기대된다.

키워드: 스마트 단말(Smart Device), 스트리밍 콘텐츠(Streaming Contents), 재생시점 탐지(Execution Detection)

1. 서론

3)국내 스마트 단말 사용자는 3,000만명을 넘은 것으로 나타나고 있으며 지속적으로 증가하고 있다. 스마트 단말내 운영체제로는 안드로이드가 55% 정도에 달하는 등 많은 사용자가 안드로이드 기반 스마트폰을 사용하고 있다. 이와 같이 최근 급증하고 있는 안드로이드 기반 스마트폰은 기존 PC에서 이용 가능한 모든 서비스를 손쉽게 제공한다는 장점이 있다. 최근 통계에 따르면 방학 때 청소년의 음란동영상 이용간수가 학기 중에 비해 73% 이상 높아지는 것으로 나타났다. 이는 청소년들이 대부분 스마트폰을 보유하게 되면서 유해 콘텐츠에 접속하기가 더욱 쉬워졌기 때문이다. 스마트폰 이용이 보편화되면서 성인물을 접하게 될 기회가 훨씬 늘어난 셈이다. 이에 스마트폰 성인물 등을 차단하는 방법에 대한 연구가 필요하다. 성인물을 반복적으로 탐닉하게 되면 중독에 이르게 되고 성인물의 본질인 선정성과 폭력성이 공격적인 행동을 유발할 수 있다. 심한 경우 자기 통제가 힘들어져 모방성 범죄나 다른 일탈행위로 이어질 수 있기 때문에 청소년을 대상으로 한 스마트폰 성인물 등 유해 음란물 차단 방법이 필요하다

시점이다[1].

구글 마켓 등을 통해 누구나 손쉽게 안드로이드 기반 스마트 단말에 탑재 가능한 스트리밍 미디어 플레이어 앱을 다운로드할 수 있다. 스마트폰을 소유한 사용자는 연령대와 상관없이 MX Player, VLC Stream Player 및 Live Stream Player 등을 본인의 단말내에 설치한 후에 스트리밍 형태로 시간과 장소에 상관없이 손쉽게 동영상 정보 등을 접근할 수 있다. 물론 단말 내에 앱 허용 등급을 설정하여 콘텐츠의 수위에 따라 단말로 다운로드 가능한 애플리케이션을 제한할 수 있다. 하지만 이와 같은 콘텐츠 필터링 기능은 단말 사용자에게 의해 임의적으로 변경 및 기능 재설정이 가능하기 때문에 근본적으로 유해 콘텐츠에 대한 접근을 제한하기에는 많은 문제점이 있다고 볼 수 있다[2]. 따라서 스마트 단말에서 콘텐츠가 실행되는 시점을 직접 판별 또는 모니터링 한 후에 해당 스트리밍 콘텐츠의 유해성 여부를 판별하는 방법이 가장 핵심적인 기술이라고 할 수 있다.

이에 본 연구에서는 우선 안드로이드 기반 스마트 단말에서의 스트리밍 서비스를 이용한 동영상 콘텐츠 서비스의 실행 구조에 대한 근본적인 분석 과정과 함께 동영상 실행시 발생하는 주요 특징을 도출하였으며, 이를 기반으로 안드로이드 기반 스마트 단말에서의 음란 동영상과 같은 유해 콘텐츠를 차단할 수 있는 방법에 대해 제시하고자 한다.

3) 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신방송 연구개발 사업의 일환으로 수행하였음 [13-912-06-006, 스마트 단말용 스트리밍 유해 콘텐츠 차단 기술 개발]

II. 스마트 단말내 스트리밍 콘텐츠 실행 구조 분석

1. 안드로이드 스트리밍 서비스 호출 구조

안드로이드 기반 스마트 단말에서 스트리밍 서비스가 작동되는 구조를 분석하면 다음과 같다.

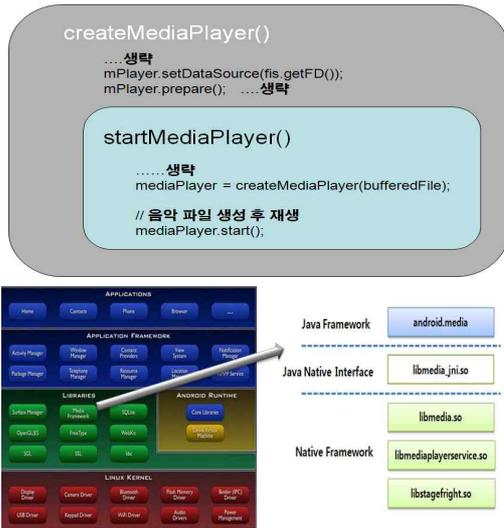


그림 1. 안드로이드 스트리밍 서비스 호출 구조
Fig. 1. Android Streaming Service Call Architecture

안드로이드 단말에서 멀티미디어 관련 어플리케이션에 대한 서비스가 구동되면 이를 미디어 서버 프로세스에서 받아 필요한 서비스를 제공하는 방식으로 실행된다. 아래 그림과 같이 자바 기반 API 호출에 의해 미디어 프레임워크내 IPC(Inter-Process Communication)로 변환될 수 해당 요청이 미디어 서버 프로세스내의 미디어 플레이어 서비스에 의해 처리되는 구조이다.

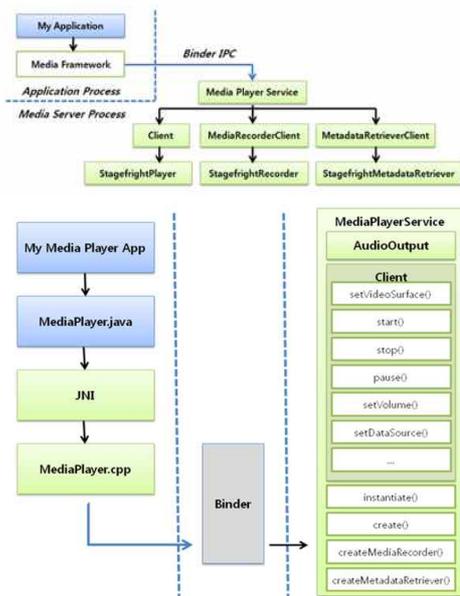


그림 2. 스트리밍 서비스 내부 바인딩 과정
Fig. 2. Streaming Service Internal Binding Process

미디어 플레이어 서비스 구조를 분석하면 그림과 같이 미디어 플레이어 앱에서 MediaPlayer.java 코드를 호출하고 이는 MediaPlayer.cpp 코드를 다시 호출하여 안드로이드 플랫폼 내부 미디어 프레임워크에 포함되어 있는 MediaPlayerService와의 바인딩 과정을 수행하게 된다.

2. 안드로이드 미디어 플레이어 호출 구조

안드로이드 플랫폼 내부 미디어 프레임워크 구조를 보다 자세히 살펴보면 다음과 같다. 안드로이드 미디어 프레임워크를 이용하여 스트리밍 서비스가 구동되며 다음 그림과 같이 미디어 플레이어를 위해 MediaPlayerService 내부에서 MediaPlayer Factory에 포함된 NuPlayer Driver를 호출하여 스트리밍 데이터에 대한 처리 과정이 수행된다.

스트리밍 서비스와 관련된 재생엔진에는 아래 표와 같이 네가지 종류 중 하나를 이용할 수 있으며, HLS와 RTSP 스트리밍 서비스의 재생을 담당하는 것으로는 핵심적인 모듈이 NuPlayer라는 것을 알 수 있었다. 따라서 안드로이드 단말에서 정상적인 형태의 동영상 스트리밍 서비스 또는 음란 동영상과 같은 유해 콘텐츠가 실제로 구동되는 시점을 확인하기 위해서는 NuPlayer가 호출되는 이벤트를 중심으로 분석하는 것이 필요하다는 것을 확인할 수 있다.

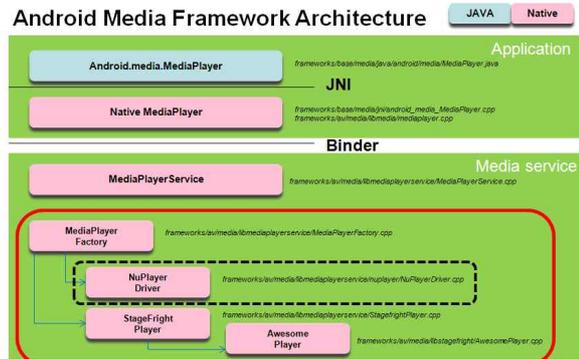


그림 3. 안드로이드 미디어 프레임워크 구조
Fig. 3. Android Media Framework Architecture

안드로이드 단말에서 멀티미디어 관련 미디어 플레이어가 호출되는 구조를 자세히 살펴보면 다음과 같다. 앞에서 제시한 바와 같이 MediaPlayer Factory 하부 모듈 중에서 NuPlayer가 호출되어 동영상에 대한 디코딩 과정이 수행되고 이를 통해 원격 미디어 서버로부터 수신한 스트리밍 정보를 각 사용자의 단말에 출력하게 된다[3].

재생 엔진	설명
STAGEFRIGHT_PLAYER	안드로이드 젤리빈 버전의 기본 재생 엔진이다. MIDI를 제외한 모든 로컬 미디어 파일과 HTTP PD방법을 사용한 원격지 미디어 재생을 담당한다.
NU_PLAYER	HLS(http://*.m3u8) 및 RTSP 스트리밍(rtsp://)의 재생을 담당하는 재생엔진이다. HLS: HTTP Live Streaming RTSP: Real Time Streaming Protocol
SONIVOX_PLAYER	MIDI파일의 재생을 담당하는 재생 엔진이다. midi: Musical Instrument Digital Interface
PV_PLAYER	진저브레드 버전 이전에 사용되던 OpenCore 플레이 어 엔진을 말하는 것으로 현재 쓰이지 않는다.

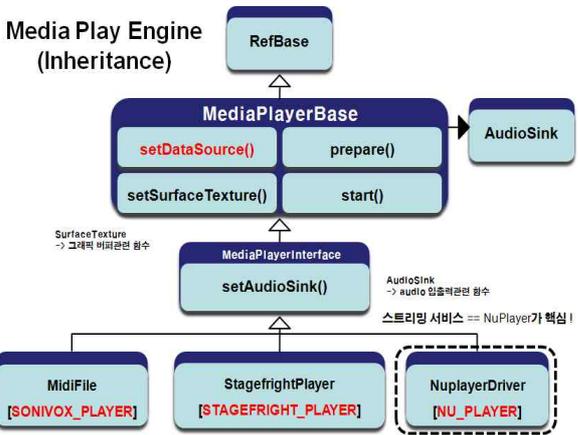


그림 4. 안드로이드 미디어 플레이어 호출 구조
Fig. 4. Android Media Player Call Sequence

3. 안드로이드 스트리밍 작동 구조

안드로이드 기반 스마트 단말에서 스트리밍 콘텐츠가 실행되는 과정은 단말 내부에 저장된 동영상 스트리밍 정보를 실행하는 방식과 원격 URL 접속을 통해 스트리밍 서비스를 제공하는 두 가지 방식으로 나눌 수 있다. 이 중에서 대부분의 유해 콘텐츠인 경우 URL 접속을 통해 원격 서버로부터 동영상 정보를 전송 받아 실행하는 구조이다. 따라서 아래 그림의 오른쪽 부분과 같이 HLS(HTTP Live Streaming) 또는 RTSP Streaming인 경우 NuPlayer를 구동하게 된다. 기타 일반적인 동영상인 경우에는 SonivoxPlayer 또는 StageFrightPlayer를 통해 구동된다.

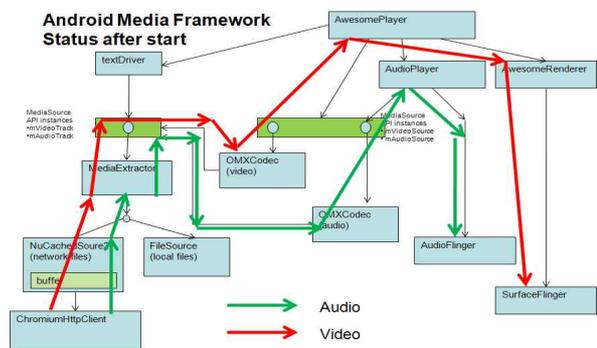
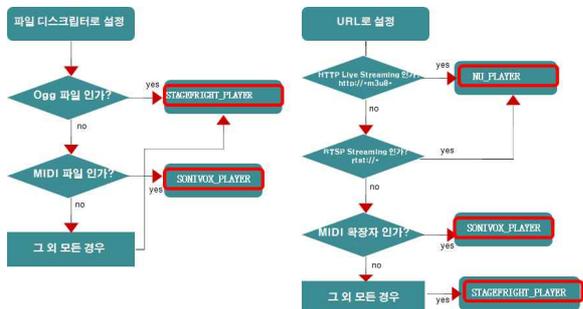


그림 5. 미디어 프레임워크 기반 스트리밍 실행 단계
Fig. 5. Streaming Execution Process based on Android Media Framework

위 그림은 안드로이드 미디어 프레임워크를 기반으로 동영상 스트리밍 데이터가 실행되는 구조를 제시하고 있다. 제시한 바와 같이 NuPlayer를 기반으로 HLS 연결 서비스인 경우 스트리밍 데이터에 대해 전송 받아 이를 안드로이드 단말 내부 미디어 프레임워크를 이용하여 재생하는 과정을 수행한다.

결국 안드로이드 기반 스마트 단말에서 스트리밍 서비스가 실행되는 구조적인 내부 특성을 파악하였기 때문에 이를 토대로 성인물과 같은 유해 콘텐츠가 실행되는 시점을 자동적으로 판별할 수 있는 메커니즘을 다음과 같은 과정을 통해 구축할 수 있다.

III. 스트리밍 콘텐츠 재생 시점 탐지 메커니즘

안드로이드 기반 스마트 단말 내부에서 음란 동영상과 같은 유해 콘텐츠가 실행되는 시점을 탐지하기 위해서는 (1단계) 단말 내부에서 스트리밍 콘텐츠의 실행과 관련되어 발생하는 상위 계층에서의 이벤트 (LogCat Event)와 하위 계층에서 발생하는 이벤트(System Call Event)에 대한 수집 과정을 수행한다.

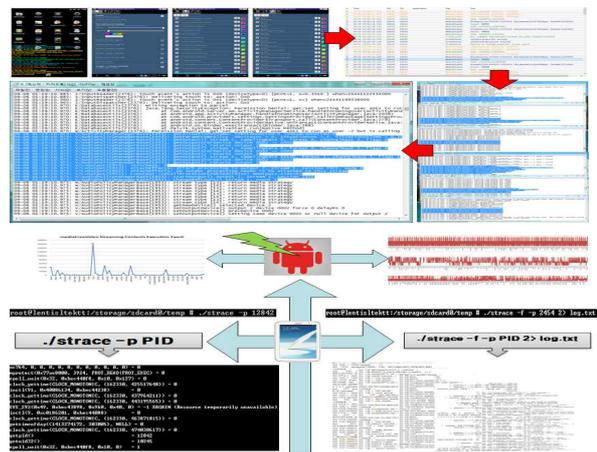


그림 6. 스마트 단말내 스트리밍 이벤트 수집 단계
Fig. 6. Streaming Event Aggregation Step on Smart Device

그리고 (2단계) 이들 상위/하위 계층에서 발생하는 두 가지 종류의 이벤트 집합내 이벤트 정보에 대한 그룹핑 과정을 통해 각 집합내 이벤트 정보에 대한 연관성을 도출한 다음 이를 토대로 (3단계) 스마트 단말에서 스트리밍 서비스가 구동되는 시점의 상위/하위 계층 이벤트를 특징을 도출한다.

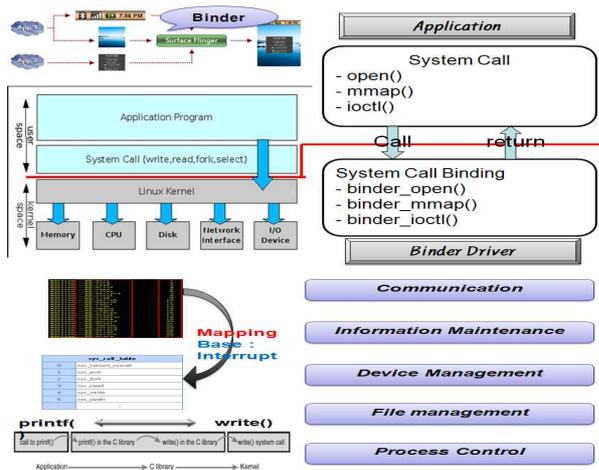


그림 7. 스트리밍 이벤트 그룹핑 수행 단계
Fig. 7. Streaming Event Grouping Step on Smart Device

이와 같은 3단계 탐지 과정을 통해 구축된 판별 DB 정보를 토대로 안드로이드 기반 스마트 단말에서 스트리밍 서비스를 구동하였을 경우에 기존 기법[4]과 달리 실시간으로 발생하는 이벤트 정보 집합의 특징을 도출하는 과정을 아래 구조와 같이 수행할 수 있으며 판별 과정을 위해 구현한 SW는 아래 그림과 같다.

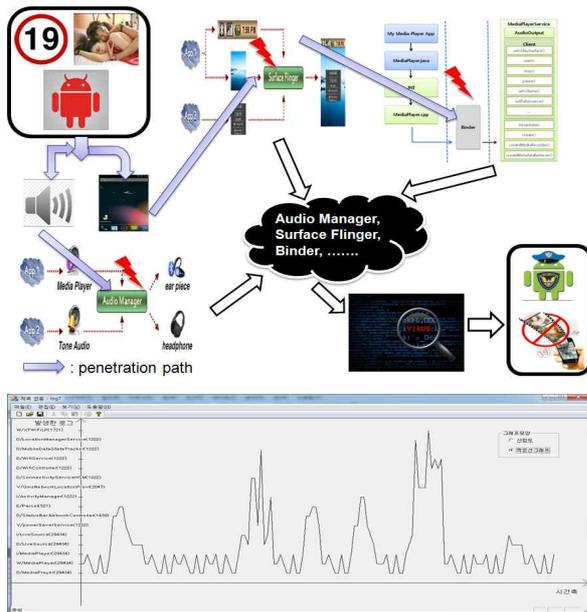


그림 8. 스트리밍 실행시점 판별 구조 및 소프트웨어
Fig. 8. Streaming Execution Detection Structure and SW

IV. 결론

본 논문에서는 안드로이드 기반 스마트 단말에서 스트리밍 콘텐츠가 재생되는 시점을 자동으로 탐지하기 위한 방법을 제안하였다. 제안된 방법은 스마트 단말의 애플리케이션 계층에서 발생하는 이벤트 및 시스템 계층에서 발생하는 이벤트의 연관성을 분석하여 재생 시점을 탐지하며, 이를 소프트웨어로 구현하였다.

참고문헌

[1] <http://www.etnews.com/201308200191>
 [2] <http://jaebok.tistory.com/106>
 [3] The Android Media Framework, Linux Foundation, <http://events.linuxfoundation.org/sites/events/files/slides/Android%20builders%20summit%20-%20The%20Android%20media%20framework%20-%20final.pdf>
 [4] A. Akbulut, F. Patlar, etc., "Agent based pomography filtering system," INISTA 2012, IEEE, pp.1-5, 2012.