안드로이드 커널에서 시스템 콜 후킹을 통한 개인정보 유출 차단 기법

김성수○ 홍충선*
경희대학교 컴퓨터공학과
{ mjs0514, cshong }@khu.ac.kr

Prevention of personal Information leakage through system call hooking in Android Kernel

Sung Soo Kim○ Choong Seon Hong*
Department of Computer Engineering, KyungHee University

요 약

스마트 폰 사용자의 기하급수적으로 증가함에 따라 안드로이드 플랫폼이 가지는 취약점을 이용한 악성 애플리케이션들 또한 급격하게 증가하고 있다. 대부분의 악성 앱들은 사용자의 소중한 개인정보 유출을 목적으로 하기 때문에 사전에 탐지하고 차단하기 위해 많은 연구들이 진행되어 왔다. 하지만 기존 기법들의 특성상 많은 한계점을 지니고 있다. 그 중에서 커널 레벨의 시스템 콜 후킹과 white-list 기반의 접근정책을 통해 인가되지 않은 개인정보의 접근과 인터넷 연결을 실시간으로 탐지하고 차단하는 기존 연구가 존재한다. 본 논문에서는 이 기존 연구의 white-list 기반의 접근정책의 문제점을 제시하고 그 문제를 해결하기 위해 SVM(Support Vector Machine) 분류 알고리즘을 적용시키는 방법을 통해 해결하는 방안을 제안하였다.

1. 서 론

스마트 폰이 나오기 시작한 이래로 스마트 폰의 사용자 수는 기하급수적으로 늘어나기 시작했다. 2014년 1월 21일 SA에서 세계 88개국을 대상으로 운영체제(OS)별 스마트 폰 사용자 수와 점유율을 조사했는데, 그 결과에 의하면 대한민국의 안드로이드 스마트 폰 사용자 점유율이 93.4%로 세계에서 가장 높았으며 조사 대상국 중 안드로이드의 비중이 90%를 넘는 유일한 나라였다.[1] 또한 안드로이드는 악성 앱이 퍼지기 쉬운 오픈 형 마켓구조를 띄기 때문에 안드로이드를 타겟으로 하는 악성코드의 수가 점점 증가하고 있다. ASEC이 집계한 바에 따르면, 모바일 악성코드 중 V3 모바일에 진단이 추가된 악성 앱만 2013년 한 해 동안 총 132만 6139건이 악성 앱으로 진단됐다.[2] 이런 악성 앱을 이용한 대표적인 공격방법으로 SMS를 이용한 스미싱 공격, 악성코드를 삽입해서 리패키징한 앱을 통한 개인정보 유출, 좀비 스마트 폰을 통한 DDoS 공격 등이 존재한다. 이와 같은 공격들을 탐지하고 차단하기 위한 선행 연구들이 존재한다.

본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음 (NIPA-2014-(H0301-14-1003)).
*Dr. CS Hong is the corresponding author.

기존 연구들로 애플리케이션 레벨에서의 정적분석과 동적분석 그리고 커널 레벨에서의 동적 분석이 있다. 그 밖에 제안된 다양한 선행 연구들 중에서 사용자의 스마트 폰에서 실시간으로 개인정보 유출을 탐지하고 차단하는 방법을 제시한 연구가 있다.[3] 제안하는 방법으로는 LKM(Loadable Kernel Module)[4]으로 구현된 시스템 콜 후킹 모듈을 커널에 삽입하여, 악성 앱으로 의심되는 사용자 애플리케이션이 개인정보에 접근하거나 화이트 리스트에 존재하지 않는 개인정보로 유출을 시도할 경우 사용자에게 알리고 허용하거나 차단하는 방법이다.

본 논문은 이 연구의 화이트 리스트를 관리하는 문제점을 제시한 후 그 문제점을 해결하기 위해 SVM(Support Vector Machine)을 기반으로 한 방법을 제안한다.

2. 관련 연구

2.1 안드로이드 상의 개인정보 유출 탐지 및 차단[3]

그림 1은 개인정보 유출을 탐지하고 차단하기 위해 제안한 Alarm app이 개인정보 유출을 탐지했을 경우 접근 제어까지의 동작 흐름을 보여주고 있다. 만약 User app이 시스템 콜을 통해 접근하는 파일의 경로가 개인정보

에 해당하거나 인터넷 연결에 관련된다면 Alarm app이 커널에 삽입된 LKM 후킹 모듈을 통해서 그 시스템 콜의 파라미터 값을 읽어 온다.

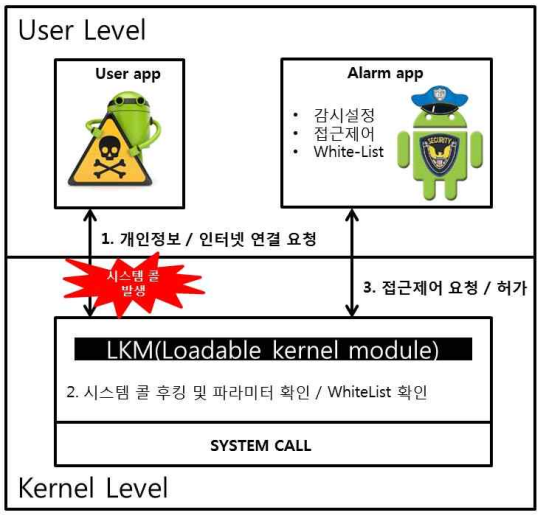

그림 1. 스마트 폰에서 동작 흐름

표 1은 파일의 접근하거나 인터넷 연결을 위해서 호출되어야 하는 시스템 콜들이다. 이 시스템 콜들의 파라미터 값에서 접근하는 파일의 경로와 연결 IP주소가 Alarm app의 화이트리스트에 존재하지 않는 경우 사용자에게 접근의 허가나 차단을 요청하는 기법이다.

표 1. 시스템 콜의 파라미터 값

시스템 콜	함수 선언부
open()	int open (const char ***pathname**, int flags)
read()	ssize_t read (int **fd**, void *buf, size_t count)
write()	ssize_t write (int **fd**, const void *buf, size_t count)
connect()	int connect (int sockfd, const struct sockaddr ***addr**, socklen_t addrlen)
sendto()	ssize_t sendto (int sockfd, const void *buf, size_t len, int flags, const struct sockaddr ***dest_addr**, socklen_t addrlen)

2.2 문제점

위의 제안 기법은 시그니처 기반의 탐지가 어려운 개인정보 유출 행위를 탐지할 수 있다는 것과 사용자의 스마트 폰에서 개인정보의 접근과 인터넷 연결을 실시간으로 탐지하고 차단 할 수 있는 것은 매우 큰 장점이다. 하지만 실시간으로 차단하기 위한 방법으로 white-list라는 접근 정책을 사용 했는데, 정작 제일 중요한 이 리스트의 구성을 일반 사용자가 결정하게 했다는 것이 문제라고 할 수 있다. 일반 사용자가 보안에 관심이 많더라도 앱의 접근경로와 정보를 유출시키는 IP의 옳고 그름을 판단하기란 쉽지 않은 일이다.

3. 제안 사항

본 논문에서는 위에서 제시한 문제점을 해결하기 위해서 SVM(Support Vector Machine) 분류 알고리즘을 사용하고자 한다. 먼저 그림 2의 전체 동작 흐름을 설명하고 SVM 분류 알고리즘에 대해서 설명한다.

3.1 동작 순서

그림 2는 제안한 Alarm App의 동작 순서를 나타내고 있다. 우선 감시하고자 하는 User App을 선택한다. 그 다음 Alarm App은 기존에 학습된 데이터를 통해 User App를 분류한다. 그리고 기존에 데이터와 함께 다시 학습하도록 한다. 동시에 시스템 콜 후킹 모듈은 지속적으로 시스템 콜을 모니터링 한다. 3번째로 분류가 모두 끝나면 그 수치 값을 가지고서 후킹 모듈의 값을 기다린다. 마지막으로 모니터링이 끝나면 개인정보 유출 및 파일 접근의 발생여부와 함께 분류 값을 종합적으로 판단할 수 있는 최종 값을 만들어 낸다. 그 값을 보고 사용자가 허용할지 차단할지를 결정하도록 한다.

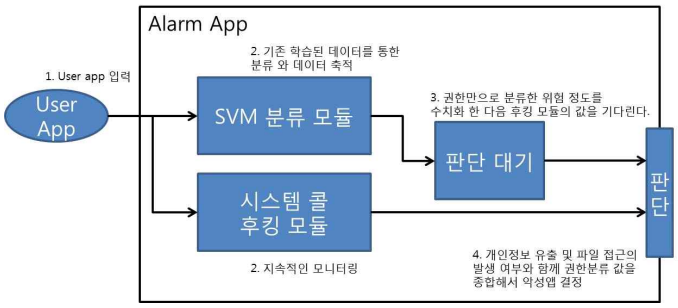

그림 2. 전체 동작 흐름

3.2 SVM 분류 알고리즘

그림 2에서 볼 수 있듯이 분류별 App들의 권한을 표본별로 데이터화 시킨다. 예를 들면 '핸드폰 분실대비 & 위치추적을 위한 여기요' 라는 앱의 설치 시 요구하는 권한은 네트워크 통신, 전화통화, 저장용량, 시스템 도구 등등이 존재한다. 이런 권한들을 표 2에서[5] 분류

한대로 권한을 Mapping하면 {1, 2, 12, 15}와 같이 된다. 이러한 표본 데이터들을 Normal과 Abnormal로 구분 짓는다. 구분 짓는 기준은 악성 앱에서 자주 사용되는 권한들의 빈도수에 가중치를 두어서 계산한다. 그 다음 그림 3과 같이 분류된 데이터들의 20%를 임의로 추출해서 SVM에게 학습시킨다. 그리고 학습시킨 데이터를 통해 학습한 내용을 전체 데이터와 비교하여 전체데이터의 분류율을 계산한다.

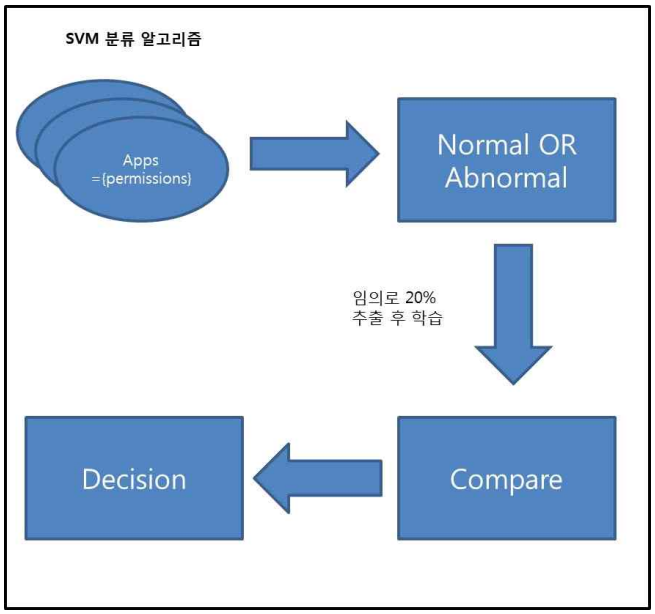

그림 3. SVM 분류 순서도

3.3 성능 평가

SVM을 이용하기 위해 먼저 학습되어질 데이터가 필요한데 성능 평가를 위해서 기존 구글 마켓에 있는 카테고리 중 임의로 5개의 카테고리를 선정했다. 그 다음 카테고리별로 유료, 무료 앱을 각각 10개씩 고른 다음 그 앱들의 권한 데이터를 수집 후 수치화 했다. 이 때 표 2를 바탕으로 수치화한 데이터를 패턴 화 했으며, 전체 패턴 중에서 20%를 임의로 추출하여 학습 데이터로 선정했다. 선정된 학습데이터로 SVM의 기계학습을 진행하고 학습이 완료된 SVM에 전체 패턴을 입력하여 분류 율을 평가하도록 했다.

성능 분석은 SVM의 가우시안 커널 함수를 사용했으며 분류 결과를 얻기 위해 1000 에폭(Epoche)단위로 진행했다.

4. 결 론

본 논문에서 제안하는 기존 연구의 white-list 기반 정책의 문제점을 해결하기 위해 SVM 분류를 통해 애플리케이션의 권한을 통한 분류와 실시간으로 모니터링할 수 있는 방법을 제안하고자 했다. 하지만 성능평가를 위해 많은 애플리케이션 중 대표적인 몇 가지의 애플리케이션을 표본 데이터로 사용했기 때문에 정확도면에서 미흡한 점이 존재한다. 그렇기 때문에 향후 연구로 좀더 정밀한 결과를 나타낼 수 있도록 데이터의 양을 늘려서 성능 분석을 하도록 해야 한다.

표 2. 악성 앱에서 많이 사용되는 퍼미션 순서. [5]

No.	권한
1	Internet
2	Read_Phone_State
3	Write_External_Storage
4	Access_Wifi_State
5	Read_SMS
6	Receive_Boot_Completed
7	Write_SMS
8	Send_SMS
9	Vibrate
10	Access_Coarse_Location

표 3. 가우시안 커널을 이용한 eLBP 분류 결과

Epoche	총패턴수	성공율	실패율
1000	100	97%	3%
	100	89%	11%
	100	93%	7%

5. 참고문헌

[1] 권영전, “한국은 세계 1위 안드로이드 공화국 93.4%가 사용”, 연합뉴스, 2014년 1월 21일 http://news.naver.com/main/read.nhn?mode=LSD&mid=shm&sid1=105&oid=001&aid=0006711113

[2] 2013년 모바일 악성코드 동향, 안랩, 2014년 2월 6일 웹사이트 : http://www.ahnlab.com/kr/site/securitycenter/asec/asecView.do?groupCode=VNI001&webNewsInfoUnionVo.seq=22137

[3] 최영석, 김성훈, 이동훈, “개인정보 유출 탐지 및 차단에 관한 연구 : 안드로이드 플랫폼 환경”, 정보보호학회 논문지, Vol.23, No.4, 2013, pp757-766

[4] 유동훈, 최재규, 김준연, “안드로이드 커널 감염을 이용한 악성코드 분석 방법 연구”, KISA, (2012)

[5] 전철, 장준혁, 김봉재, 정진만, 조유근, “효율적인 안드로이드 애플리케이션 검수를 위한 견고한 퍼미션 기반 악성 애플리케이션 여과 기법”, 보안공학연구논문지, Vol.10, No.2, 2013