2024년 한국정보보호학회 하계학술대회

CISC-S'24

Conference on Information Security and Cryptography Summer 2024

일자 2024년 6월 20일(목) ~ 21일(금)

장소 세종시 정부세종컨벤션센터

주최 한국정보보호학회 Korea Institute of Information Security & Cryptology

주관 고려대학교 세종캠퍼스 KOREA UNIVERSITY SEJONG CAMPUS

후원 국가정보원 NATIONAL INTELLIGENCE SERVICE, 과학기술정보통신부, 행정안전부, 세종특별자치시, KISA 한국인터넷진흥원, ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute, NSR 국가보안기술연구소 National Security Research Institute, KISTI 한국과학기술정보연구원 Korea Institute of Science and Technology Information www.kisti.re.kr

한국정보보호학회
Korea Institute of Information Security & Cryptology

안드로이드 악성코드 패밀리 분류를 위한 시스템 콜 패턴 분석 연구

최재민*, 최상훈[1], 박기웅†

세종대학교 정보보호학과 (대학원생, 연구교수[1], 교수†)

Resarch on System Call Pattern Analysis for Android Malware Family Classification

Jae-Min Choi*, Sang-Hoon Choi[1], Ki-Woong Park†

*Dept. of Computer and Information Security, Sejong University

(Graduate student*, Research Professor[1], Professor†)

요 약

스마트폰이 대중적으로 보급됨에 따라 모바일 악성코드 시장이 증가하고 있으며, 난독화, 암호화, 안티 디버깅 기술을 이용하여 점점 더 고도화되고 분석이 어려운 악성 APK가 배포되고 있다. 이에 따라 시스템 콜을 활용한 악성 APK 탐지 및 분류가 연구되고 있다. 하지만, 기존의 연구들은 시스템 콜 횟수 데이터를 활용하여 악성코드를 탐지하기 때문에 호출 횟수가 비슷한 악성코드에 대해 오탐율이 높다. 추가로, 악성코드의 행동 패턴을 놓칠 수 있다는 한계점이 있다. 본 논문에서는 APK 파일로부터 시스템 콜 순서 정보를 악성코드 탐지 및 분류에 활용하는 방안을 제안한다. 우리가 제안하는 시스템 콜 순서 정보를 통한 분류 방법은 정확한 순서 정보를 기반으로 행동 패턴을 분석하여, 유사한 호출 빈도를 가지는 악성 APK를 보다 정확하게 분류할 수 있음을 보여준다. 향후 연구에서 함수 순서 정보 자동 추출 도구와 데이터 전처리 도구 연구개발을 통해 머신러닝 기반 악성코드 탐지 및 분류를 위한 데이터 셋 추출 연구를 수행하고자 한다.

I. 서론

스마트폰이 대중적으로 보급됨에 따라 모바일 시장이 지속적으로 성장하고 있다[1]. 모바일 시장이 증가함에 따라 모바일 악성코드 시장 또한 증가하고 있다. 특히 시장 점유율의 대부분을 차지하고 있는 안드로이드 운영체제는 특유의 개방성으로 인해 많은 악성 APK가 등장하고 있다[2]. 이러한 다양한 종류의 악성 APK가 등장함에 따라, 악성 APK를 탐지하고 분류하는 중요성이 점점 더 커지고 있다. 이에 따라, 악성 APK 또한 난독화 및 암호화 그리고 안티 디버깅을 활용하는 방식으로 진화하고 있다. 이로인해 전통적인 정적, 동적 분석 도구를 활용하지 못하는 한계점이 발생하였고, 이를 보완한 연구들이 있다[3, 4].

기존의 연구들에서는 시스템 콜 횟수를 기반으로 한 악성 APK 탐지 및 분류를 진행하여 악성 행위자들의 탐지 및 분석 회피 행위를 무시할 수 있다는 장점이 있다. 하지만, 정확한 순서 정보를 제외한 호출 횟수를 기반으로 하기 때문에 악성코드의 행동 패턴을 놓칠 수 있고, 서로 다른 악성코드가 유사한 호출 빈도를 가질 수 있는 가능성이 존재한다. 또한, 복잡한 행동 패턴의 악성코드의 경우 분류에 있어 제한적인 한계점이 존재한다. 따라서 우리는 안드로이드 환경에서 발생하는 시스템 콜 순서 정보를 기반으로, 기존의 접근법보다 정확하게 악

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)

본 논문은 과학기술정보통신부의 재원으로 정보통신기획평가원(IITP)의 정보통신방송기술 국제공동연구(Project No. RS-2022-00165794, 40%), 실감콘텐츠핵심기술개발(ProjectNo. RS-2023-00228996, 10%), 정보통신방송혁신인재양성사업(Project No. 2021-0-01816, 10%) 및 한국연구재단(NRF) 중견후속연구사업(Project No. RS-2023-00208460, 40%)의 지원을 받아 수행된 연구임.

성 코드 패밀리를 분류하는 방법에 대해 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 안드로이드 시스템 콜 기반의 악성코드 탐지 연구를 서술하고, 3장에서는 안드로이드 시스템 콜 추출 방법과 호출 순서에 따른 분류에 관해 서술한다. 4장에서는 호출 순서에 따른 분류 결과에 관해 설명하고, 5장에서는 결론 및 향후 연구를 기술한다.

II. 관련 연구

2.1 안드로이드 시스템 콜

안드로이드 운영체제는 리눅스 커널 위에 안드로이드 Native Demon & Library, HAL(Hardware Abstraction Layer), ART(Android Run Time), System Service, System & Android API가 탑재되어 동작하는 방식이다[5].

안드로이드는 리눅스 커널을 사용하기 때문에 APK 또한 프로세스로 관리된다. 따라서, 안드로이드 APK에 strace 도구를 활용하여 시스템 콜을 추적하면 linux 커널의 시스템 콜을 효율적으로 추적할 수 있다. 이를 통해 APK의 작동 방식에 대하여 분석할 수 있다.

2.1.1 안드로이드 시스템 콜 횟수를 통한 악성코드 탐지 관련 연구

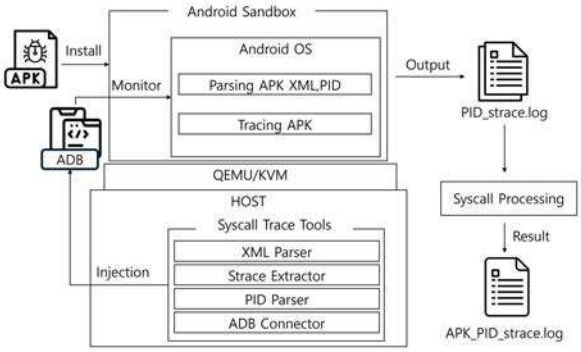
안드로이드 악성 APK 탐지를 위한 연구는 꾸준히 진행되어왔다. Taniya Bhatia는 악성코드 호출 횟수 분석을 통한 악성코드 탐지방안을 제안하였다[6]. 해당 연구는 APK가 실행된 뒤 monkey 도구를 이용하여 1분간 APK가 발생한 시스템 콜 횟수를 추출하여 악성 APK와 정상 APK를 분류할 수 있다. 해당 연구에서는 난독화와 암호화가 적용된 악성 APK를 효과적으로 분류하였다.

M. Jaiswal 외 2인은 악성 게임 APK의 시스템 콜 횟수 분석을 통한 악성코드 탐지방안을 제안하였다[7]. 해당 연구는 정상 APK와 복제된 악성 APK가 발생하는 시스템 콜 횟수를 추출하여 정상 APK와 악성 APK를 분류할 수 있음을 보여주었다. 해당 연구에서는 정적분석 없이 복제 및 재패키지 악성 APK를 효과적으로 분류하였다.

III. 안드로이드 시스템 콜 순서를 통한 분류 프레임워크

본 장에서는 안드로이드 악성 APK 시스템 콜 추출 및 필터, 악성 APK 패밀리별 호출 순서에 대해 분석 및 정의한 결과를 설명한다.

3.1 시스템 콜 추적



(그림 1) 효율적인 APK 시스템 콜 정보 추적을 위한 프레임워크

우리는 APK로부터 시스템 콜을 효율적으로 추적하기 위해 (그림 1) 과 같이 QEMU/KVM 기반의 분석 프레임워크를 제안한다. strace를 사용하여 악성 APK 프로세스를 추적하기 위해서는 PID 획득이 선행되어야 한다. 일반적으로, PID 획득하기 위해 1-10ms 시간이 소요된다. 즉, 10ms 이전에 호출되는 시스템콜들은 추적할 수 없기 때문에 일부 데이터가 유실된다는 한계가 존재한다. 우리는 이를 해결하기위해 안드로이드 System의 Zygote를 추적하여 Zygote가 fork()를 이용해 생성하는 하위 프로세스 중 우리가 원하는 APK의 PID만 추출하였다. 해당 방법을 통해 올바르게 APK의 시작부 시스템 콜부터 추적이 가능하다.

3.2 시스템 콜 필터링

구동중인 APK 프로세스를 대상으로 시스템 콜을 모두 추적하면 노이즈가 많이 발생한다. 예를 들어, clock_gettime() 함수와 같이 System 시간을 기록하는 함수들이 노이즈로 발생한다. 이러한 노이즈를 줄이기 위해 strace 도구의 -e 기능을 이용하여 [표 1]과 같이 추출 함수들을 필터링하였다.

[표 1] 필터링 시스템 콜 함수 목록

종류	시스템 콜 함수
프로세스 관리	clone, execve, fork, getuid, getuid32, geteuid, geteuid32
파일 I/O 및 소켓	accept, bind, connect, mkdir, mkdirat, open, openat, pread64, read, readlinkat, recv, rename, rmdir, send, sendmsg, sendto, setsockopt, socket, stat, unlink, unlinkat, vfork, write, writev

3.3 시스템 콜 데이터 가공

strace를 이용한 시스템 콜을 추적하면 악성 행위가 아닌 기본 안드로이드 시스템 콜이 함께 발생한다. 이때, 해당 악성 APK가 실행하는 순수한 악성 행위의 순서 정보를 판별하기 위하여 아무런 행위도 하지 않는 더미 APK를 생성하여 해당 APK의 시스템 콜을 추적하여 제거하는 과정을 진행했다. 모든 APK는 실행될 때 기본적으로 openat() 함수를 이용하여 “/dev/null”에 접근한다. 이후 JAVA와 Kotlin 코드를 실행하기 위한 프레임워크인 ART에서 .art 파일을 불러오고, dalvik 프레임워크에서 최종적으로 apk를 실행한다. 해당 실험에서는 openat(/dev/null) 함수부터 openat(/base.pak)까지 모든 APK가 동일하게 실행하는 부분으로 처리하여 제거한 뒤 분석을 진행하였다.

IV. 실험 및 분류 결과

4.1 Trojan Malware, Locker Ransomware

[표 2] 각 카테고리별 주요 시스템 콜

Category	시스템 콜 함수
Trojan	openat(), socket(), setsockopt(), writev(), write(), mkdirat(), unlinkat()
Locker Ransomware	clone(). openat(), pread64(), write(), writev()

Trojan Malware의 경우 사용자의 개인 정보 탈취를 목적으로 제작하여 사용자의 파일, SMS, 사진, 동영상 등을 추출하여 공격자의 서버로 전송하는 악성 코드이다. 따라서, 해당 Malware 타입은 사용자의 파일에 접근하여 개인 정보를 추출하고 공격자의 서버로 전송을 하는 과정을 진행한다. Locker Ransomware의 경우 사용자의 화면을 강제로 잠금으로써 사용자의 가용성을 침해 하고 이를 토대로 금전적인 요구를 하는 악성코드이다. 따라서, 해당 Ransomware 타입은 임의의 Ransom note를 생성하거나 화면에 출력하는 과정을 진행한다. 위의 기본적인 행위 정의에 따라 카테고리별 주요 시스템 콜을 [표 2]와 같이 정리하였다.

4.2 악성 APK 시스템 콜 함수 분석

본 논문에서는 총 11개의 샘플[8](2종의 카테고리, 8개의 패밀리)에 대하여 분석을 진행하였다. 각각의 악성 APK 샘플에 대하여 악성 행위를 진행하는 시스템 콜 함수를 [표 3]으로 나타내었다. [표 3]의 시스템 콜 순서 정보 열의 시스템 콜 함수는 openat->op, unlinkat->un, writev->wv, write->wr, mkdirat->mk, clone->c, pread64->pr 약어로 표기하였다.

각 패밀리별 유사도는 패밀리별 주요 함수 시작부부터 wv(DESTROY) 혹은 wv(RESUME), 주요 행위가 종료되는 시점까지의 행위에 대해 유사도를 판별하였다.

4.3 분류 결과

샘플별로 분석을 진행한 결과 동일 카테고리 내에서 패밀리별로 시스템 콜 순서 정보가 다른 것을 확인 할 수 있다. 특히, 기존의 변수명을 통한 패밀리 분류시에 혼동을 줄 수 있는 동일 변수명을 사용하는 샘플과 시스템 콜 횟수 기반 분류시에 혼동을 줄 수 있는 시스템 콜 횟수가 유사한 샘플에 대하여 순서 정보를 활용하여 올바르게 분류를 진행하였다. Wipelock과 Octo 패밀리의 경우 각 두개의 샘플이 동일한 순서 정보를 보여주었고, SLocker패밀리의 경우 clone() 함수의 횟수에 차이가 존재하고 이밖의 순서 정보는 동일함을 보여주었다. 특정 패밀리가 없는 샘플의 경우 그 어떤 패밀리와도 유사하지 않음을 보여준다. 결론적으로, 동일 패밀리의 경우 거의 유사하거나 동일하게 순서 정보가 나타나는 것을 볼 수 있다. 다른 샘플의 분류 결과는 [표 3]과 같다.

[표 3] 각 샘플별 System 호출 순서 정보 분석

Category	Family	SHA256	System call Sequence
Trojan	Wipelock	40e1 … cf90	op>un>wv(Create)>wv(Pause)>wv(Destroy)
Trojan	Wipelock	816c … 32c8	op>un>wv(Create)>wv(Pause)>wv(Destroy)
Trojan	NONE	8440 … 4b7a	wv(Create)>wv(Pause)>wv(subCreate)>wv(subPause)>wv(STOP)>wv(RESUME)>op>un
Trojan	Octo	439f … 83d5	mk>mk>wv(Create)>op>w>un>wv(Destroy)
Trojan	Octo	791a … 154e	mk>mk>wv(Create)>op>w>un>wv(Destroy)
Trojan	Coper	5bb7 … 9b95	mk>wv(Create)>op>w>un>op>w>un>wv(Destroy)
Trojan	spynote	0f47 … dd27	mk>op>w>un>op>w>un>wv(Create)>wv(Destroy)>op>w>un
Locker Ransomware	SLocker	2aee … d53b	c#10>wv(Create)>pr64>pr64>wv(Destroy)
Locker Ransomware	SLocker	da86 … 2aa9	c#11>wv(Create)>pr64>pr64>wv(Destroy)
Locker Ransomware	NONE	de2e … 0054	c#11>pr64>pr64>wv(Create)>wv(Destroy)
Locker Ransomware	NONE	4432 … ae20	c#12>pr64>pr64>wv(Create)>wv(handle)>wv(RESUME)

V. 결론 및 향후 연구

본 논문에서는 시스템 콜 순서 정보를 통한 분류를 제안한다. 해당 방법을 통해 11개의 샘플 8개의 패밀리에 대하여 각 패밀리별로 순서 정보를 추출하여 상이함을 증명하였다. 이에 따라 시스템 콜 순서 정보를 통해 안드로이드 악성 APK 패밀리 분류가 가능함을 제시하였다. 향후 연구에서는 분석가의 역량에 따라 분석 결과가 달라지는 부분을 해결하기 위하여 시스템 콜 순서 정보 자동 추출 도구와 데이터 전처리 도구로 발전시켜 분류 모델에 활용하고, 해당 모델을 이용하여 기존 연구의 분류 모델과 비교하여 증명하고자 한다.

[참고문헌]

[1] Business of Apps : Android Statistics in 2024.

[2] Kaspersky : Attacks on mobile devices significantly increase from 2022 to 2023

[3] Sidra Siddiqui and Tamim Ahmed Khan, “An Overview of Techniques for Obfuscated Android Malware Detection”, SN COMPUT. SCI. 5, pp. 328, March, 2024.

[4] A. Bensaoud, J. Kalita and M. Bensaoud, “A survey of malware detection using deep learning”, Machine Learning with Applications, vol. 16, March, 2024.

[5] Android Developers document : Platform architecture.

[6] Taniya Bhatia and Brishabh Kaushal, “Malware detection in android based on dynamic analysis”, 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), pp. 1~6, June, 2017.

[7] M. Jaiswal, Y. Malik and F. Jaafar, "Android gaming malware detection using system call analysis," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1~5, March, 2018.

[8] MalwareBazzar : Malware dataset (n.d.), https://bazaar.abuse.ch/.