2023년 한국정보보호학회 동계학술대회

CISC-W'23

Conference on Information Security and Cryptography-Winter 2023

2023년 12월 2일 (토) | 이화여자대학교

주최 한국정보보호학회 Korea Institute of Information Security & Cryptology

주관 이화여자대학교 EWHA WOMANS UNIVERSITY

후원 국가정보원 NATIONAL INTELLIGENCE SERVICE

과학기술정보통신부

KISA 한국인터넷진흥원

ETRI 한국전자통신연구원 Electronics and Telecommunications Research Institute

NSR 국가보안기술연구소 National Security Research Institute

SK broadband

한국정보보호학회 Korea Institute of Information Security & Cryptology

Quick-Filter: 대규모 언어 모델을 활용한 악성 APK 필터링 프레임워크

노주영*, 박지훈*, 최기상**, 최상훈[1], 박기웅†

세종대학교 정보보호학과 (대학원생, 학부생**, 연구원[1], 교수†)

Quick-Filter: A Framework for Filtering Malicious APKs using Large-scale Language Models

Joo-Young Roh*, Ji-Hoon Park*, Ki-Sang Choi*,
Sang-Hoon Choi[1], Ki-Woong Park†
Dept. of Computer and Information Security, Sejong University
(Graduate Student*, Undergraduate Student**, Researcher[1], Professor†)

요 약

안드로이드의 개방성과 다양성은 공격자들에게 목표가 되었으며, 여전히 많은 악성 APK들이 발견되고 있다. 우리가 제안하는 Quick-Filter 프레임워크는 APK로부터 매니페스트를 추출한다. 이후 LLM(Large Language Model)에 질의를 통해 애플리케이션을 빠르게 분석, 분류하는 새로운 방법론이다. Quick-Filter는 실험을 통해 효과적으로 악의적인 APK를 분류할 수 있음을 보여준다. 하지만, 우리가 사용하는 LLM은 악성코드 분석을 목적으로 설계되지 않았으므로 분석가들은 Quick-Filter의 결과에 맹목적인 신뢰를 할 수 없다는 한계가 존재한다. 그럼에도, 부분적인 가능성과 빠른 판단과 분류, 더하여 기존 모델에서 사전 훈련되지 않은 악성 APK에 대해 분류가 가능하므로 전통적인 안티바이러스 솔루션과 정적 분석 도구의 한계를 해결할 것으로 기대된다.

I. 서론

스마트 환경의 중추적인 역할을 하는 스마트폰에서 주로 사용되는 안드로이드 운영체제는 세계 스마트폰 시장의 대부분을 차지하고 있다 [1]. 그러나 안드로이드 운영체제의 특징인 개방성과 다양성은 보안 측면에서 약점이 될 수 있다. 모바일 애플리케이션의 급증은 보안 위협의 증가와 동반되어, 악성 소프트웨어 탐지 작업을 그 어느 때보다 중요하게 만들었다 [2]. 안드로이드 애플리케이션의 설치 매체인 APK 파일은 종종 사용자와 시스템에 심각한 위험을 초래할 수 있는 악성코드를 숨기고 있을 수 있다. 전통적인 바이러스 백신 솔루션과 정적 분석 도구는 종종 정교하고 지속해서 변화하는 멀웨어 전술을 탐지하는 데 한계가 있다 [3].

이러한 문제에 대응하기 위해 AI 기반의 멀웨어 분류 및 분석 연구가 주목받고 있다. 특히, 기존 모델보다 코드 작성과 디버깅 지원 기능이 향상된 모델들이 등장하면서 코드를 분석하는 정적 분석에 AI를 활용하여 분류하는 기법들이 연구되고 있다. 대규모의 학습데이터가 내재 된 AI 모델을 활용한다면, APK 파일 권한 정보를 빠르게 분석하여 악의적인 행위 여부를 빠르게 파악할 수 있다 [4].

본 논문에서는 APK의 매니페스트 파일을 LLM(Large Language Model)을 통해 분석하는 새로운 접근법인 Quick-Filter를 소개한다. Quick-Filter는 여러 방면에서 수집한 APK에서 매니페스트 파일을 추출하고, 이를 LLM의 프롬프트를 통해 질의응답 하는 형식으로 애플리케이션의 안전성을 빠르게 분류할 수 있다. 우

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)
본 연구는 과학기술정보통신부 및 정보통신기획평가원의 정보통신방송기술국제공동연구사업(Project No. RS-2022-00165794, 40%), 국방ICT융합사업(Project No. Project No.2022-0-00701, 10%), 실감콘텐츠핵심기술개발사업(Project No. RS-2023-00228996, 10%),대학ICT연구센터 육성지원사업(Project No. 2023-2021-0-01816,10%) 및 한국연구재단 개인기초연구과제(Project No.RS-2023-00208460, 30%)의 지원을 받아 수행된 연구임.

리가 제안하는 Quick-Filter는 안드로이드 악성 코드 분석가들이 분석 대상의 범위를 축소하는 데 효과적으로 활용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 매니페스트와 이를 이용하여 분석한 사례들과 프레임워크에서 사용될 LLM에 대하여 서술한다. 3장에서는 우리가 제안하는 프레임워크에 대하여 서술하고, 4장에서는 프레임워크의 실용성을 평가한다. 5장에서는 결론 및 향후 연구계획에 대하여 설명한다.

II.관련 연구

2.1.1 매니페스트

매니페스트 파일은 소프트웨어의 다양한 파일들을 정의하고 그들 사이의 관계를 설정하는 메타데이터 파일이다. 이 파일은 소프트웨어 이름, 버전 번호, 라이선스 정보, 파일 목록, 구성 데이터, 그리고 의존성 등의 중요 정보를 포함한다. 안드로이드에서는 AndroidManifest.xml파일이 사용되며, 여기에는 애플리케이션의 구성 요소, 권한 등이 정의된다. 이처럼 매니페스트 파일은 각기 다른 환경과 용도에 맞게 소프트웨어 패키지의 중심적인 정보를 관리하는 핵심적인 역할을 담당한다.

2.1.2 매니페스트를 이용한 악성 APK 분석

최근에는 AI를 활용하여 매니페스트 파일 내부에서 추출할 수 있는 각종 요소를 바탕으로 악성과 정상을 분류하는 연구 사례가 진행됐다. 매니페스트을 통해 얻을 수 있는 권한, Intent 요소 그리고 버전 정보 등을 추출하고 이외의 인증서 정보, 파일 생성 시간 그리고 파일 리스트와 피처 정보를 학습하고 정상과 악성을 분류한다. 피처 간의 조합 그리고 피처와 머신러닝 알고리즘의 조합을 통해 최적의 결과를 나타내는 연구가 진행됐다 [5, 6, 7]. 또한 매니페스트와 다른 요소 간의 결합을 통해 최적의 결과를 도출하는 방법이 논의되었다. 이중 매니페스트 요소와 CFG를 결합한 Droid-MCFG라는 악성 코드 탐지 방안이 있다. CFG에 안드로이드 API 호출 순서를 나열하고 Word2vec 기반의 전이 학습으로 피처를 추출한 이후 TCN(Temporal Convolutional Network) 모델을 활용한다. 이는 96.24%의 정확도를 달성했으며 이외에도 다양한 조합 관련 연구가 지속적으로 진행되고 있다 [8].

2.2.1 LLM(Large Language Model)

1) PaLM2

Google은 Bard의 학습 모델인 PaLM2을 공개하였다. PaLM2의 특징은 기존 학습 모델에 비해 우수한 다국어 성능, 낮은 운영 비용, 그리고 빠른 응답 시간이다. 또한, 사용 모델에 대한 미세 조정이 가능하다.

2) GPT

OpenAI가 개발한 GPT는 텍스트뿐만 아니라 이미지 처리 능력도 제공하며, 시험 문제, 학술 논문 등의 시각 데이터를 이해하는 능력을 보여준다. 텍스트 처리에서도 GPT는 다른 현대적인 모델들인 PaLM과 LLaMA보다 더 뛰어난 성능을 보여준다.

3) CLOVA

CLOVA는 생산성 향상을 위한 핵심 AI 도구로, 개인과 기업의 정보 탐색, 지식 습득, 반복 업무 처리 등 다양한 분야에서 효율성을 극대화한다. 또한, HyperCLOVA X와 CLOVA Studio를 통해 기업들은 자체 데이터를 활용하여 특화된 LLM을 구축할 수 있다.

2.2.2 상용화된 LLM을 이용한 악성 APK 분석 방법의 장점

상용화된 LLM을 이용한 멀웨어 분석의 장점은 대규모의 매니페스트 정보가 미리 학습되어 있다는 것이다. 따라서, 비정상적인 매니페스트로 구성된 APK 분류가 가능하다. 이러한 장점은 자동화된 작업으로 악성코드 분석가의 부담을 덜고 분석에 있어 비용이 절감되는 효과를 가진다.

III. Quick-Filter 프레임워크

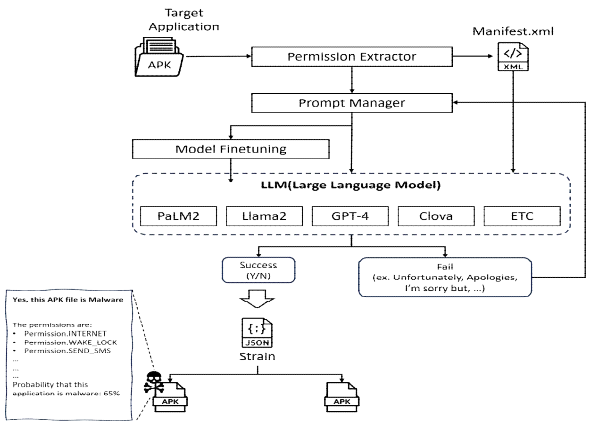

그림 1. Quick-Filter 프레임워크 구조

표 1. Quick-Filter를 활용한 악성 APK 분류 실험

APP& Malware	SHA256	LLM Models			
		PaLM2	CLOVA X	GPT 3.5	GPT 4.0
IRTA (Malware)	ffd29c57df2fb49013dfad0cf4182921e6ec36df41d69a4cb7838b09a4ba779b	O	X	△	△
Xloader (Malware)	ada2808ef254c39e70f74c93c7fd3b7f458ea439beebfc1520650fd0e3e34990	O	X	△	△
Wroba (Malware)	83ba2b1c0352ea9988edeb608abf2c037b1f30482bbc05c3ae79265bab7a44c9	O	X	△	△
TeaBot (Malware)	a34c1e334e9d76e97b8e8ac6b88bbc45cbed7ab7fc3a62e2f348c940136778af	O	X	△	O
SpyNOTE (Malware)	eec5096dfca6824317863f9225c29f6c4b3442c48fefa62dc382e3569bca5a60	O	X	△	O
v2rayNG (Normal)	1ab89f7fbc73c7118fc414c8e2a92425d7e6b5d8a353431c0b63df987dc1e70a	X	X	X	X
Psiphon (Normal)	fc77d4a5bae04a621fc32aab7fbf52d01774cb5a7dde9e4795cb787f56127396	X	X	X	X
Mitt3 (Normal)	1fbe532b179199bcc251ed0cc71e59e781ac3f1cdb8da3f6a87e1faa061ff367	X	X	X	X
Marketagent (Normal)	f1fc57728f37ea2cd209ffafd16feb8c274fadd8d4e7a16a103b7197e458bd55	X	X	X	X

우리가 제안하는 Quick-Filter는 [그림 1]과 같다. Quick-Filter는 상용화된 LLM 모델을 이용한 악성코드의 1차적 분류를 목적으로 한다. 우리가 설계한 악성코드 분류 프레임워크의 동작 원리는 다음과 같다. Quick-Filter에 APK 파일을 전달한다. 이후 Permission Extractor 모듈에 의하여 APK에서 manifest.xml 파일을 추출한다. Prompt Manager는 질의를 생성하고, 이때 LLM 모델에서 생기는 Hallucination 현상을 방지하기 위하여 n개 이상의 Prompt를 생성하게 된다. 동시에 Model Fine tuning 모듈에서 Prompt에 대하여 질의할 모델을 선정 및 정확한 답을 얻기 위한 파라미터 변경을 수행한다. 추출한 xml 기반으로 선정한 모델에 질의하였을 때, 생성된 결과값이 악성코드인지 판단하는 대답을 생성하게 된다면 성공 사례로 생각하여 json 형태로 요약 및 추출한다.

Quick-Filter를 통해 추출되는 결괏값은 [그림 2]와 같다. 우리가 제안하는 프레임워크는 모델 사용 정책이나, 개인정보 보호 정책에 따른 적절하지 않은 결과를 생성하게 된 경우 분류 실패로 판단한다. 분류에 실패하였을 때 Prompt Manager에 의해 Prompt 재가공을 통해 정책 등을 우회하여 정상적인 결괏값을 출력할 수 있도록 유도한다. 우리가 제안하는 프레임워크는 라이브러리 형태로 모델을 관리 할 수 있기 때문에 관리의 유연성과 확장성을 지원한다.

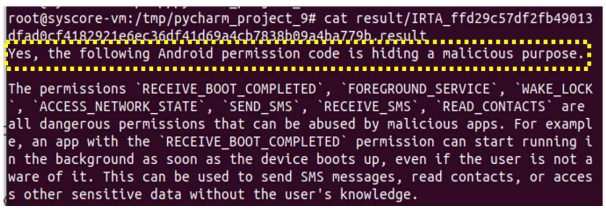

그림 2. Quick-Filter에 판단된 악성 행위 판단 결과 예시

IV. 실험 및 평가

우리는 Quick-Filter의 악성코드 1차 분류에 대한 활용 가능성 검증을 위하여 실험을 진행하였다. 질의를 위해 사용될 대표적인 상용화된 LLM 모델 4가지를 선정하였다. 우리는 악성코드 APK 5개와 플레이스토어에서 10만 회 이상 다운로드된 일반 애플리케이션을 4개를 선정하여 평가를 수행하였다. 이와 같은 실험을 통해 LLM 모델의 활용 가능성에 대한 가설을 검증하고, 악의적인 APK를 가장 효과적으로 탐지하는 LLM 모델에 관한 결과를 도출하였다.

실험 결과는 <표 1>과 같다. 실험에서 우리는 악성 APK의 매니페스트에 포함된 권한 정보를 질의했을 때 악성코드로 분류한다면 'O'라고 표기하였다. 반면에, LLM모델에게 동일한 질의하였을 때 난독화된 권한 및 패키지를 탐

지하지 못하거나, 개발사의 정책이나 개인정보 보호 법에 따른 질의를 회피한다면 'X'로 표기하였다. 추가로, 악성 행위를 판별하는 과정 중에 난독화된 권한 및 패키지를 탐지했지만, 악성 APK로 분류하지 못하고 '악의적인 의도가 숨겨져 있을 수 있다' 라고 답할 때 '△'로 표시하였다.

PaLM2의 경우 모든 악성 APK의 매니페스트를 해석하여 악성 APK를 판단하는 것을 확인할 수 있었다. 또한 악의적인 코드라고 판단하였을 때 매니페스트를 분석하여 왜 악성 APK라고 분류했는지에 대한 근거와 설명을 제시해주었다. CLOVA X의 경우 악성 APK의 분류 모델로는 적합하지 않은 것을 볼 수 있었다. 일반적으로 AI 언어 모델로 법적, 윤리적, 사회 규범에 관한 응답을 하고, 답변을 회피하였으며, 이를 우회하여 질의한 경우에도 XML로부터 난독화된 권한 및 패키지를 탐지하지 못하였다.

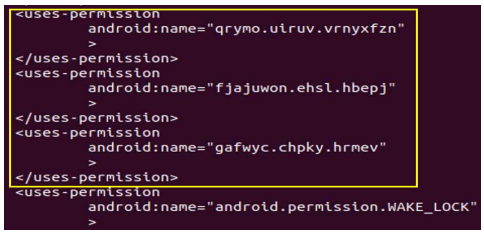

그림 3. Wroba의 난독화된 권한 정보

GPT의 경우 총 두 가지 버전으로 실험을 진행하였다. GPT 3.5 모델 같은 경우 악성 APK의 매니페스트를 분석하였을 때 악성코드가 아니라고 판단하였지만, 난독화된 권한 및 패키지를 탐지하여 APK의 악의적인 용도로 활용될 수 있다는 가능성에 대하여 언급하여 주었다. 추가로 GPT 3.5 모델에서는 IRTA 악성 APK에 대해 난독화된 권한을 찾아내지 못했던 반면, GPT 4.0에서는 난독화된 권한을 찾아냈다. [그림 3]은 Wroba의 시그니처를 가진 악성 APK의 매니페스트 내부에 난독화된 권한의 일부이다.

V. 결론

본 논문에서는 LLM을 사용하여 악성 APK를 분류하는 Quick-Filter를 제안한다. 이 프레임워크는 보안 전문가와 분석가들이 모바일 애플리케이션의 급속한 증가와 함께 증가하는 보안 위협을 자동으로 탐지하는 데 도움이 되도록 설계되었다. 실험을 통해 제안된 Quick-Filter는 악성 APK 분류가 가능하다는 것을 제시하지만 완벽한 해답은 아니다. 상업적으로 사용되는 LLM 모델들은 악성코드 분석을 위한 목적이 아닌 모델의 목적을 위해 훈련되고 설계되기 때문에 맹목적으로 신뢰하기는 어렵다. 향후 연구에서는 제안된 Quick-Filter에서 분석된 아티팩트에 가중치를 두어 악성 APK 분류의 신뢰성을 높이고 상용화된 라이브러리로 발전시키려고 한다.

[참고문헌]

[1] Statcounter: Mobile Operating System Market Share Worldwide from 2022 to 2023

[2] A. Ehsan, C. Catal, and A. Mishra, "Detecting Malware by Analyzing App Permissions on Android Platform: A Systematic Literature Review," Sensores, vol. 22, no. 20, pp. 7928, Oct. 2022.

[3] WF. Elsersy, A. Feizollah, and NB. Anuar, "The rise of obfuscated Android malware and impacts on detection methods," PeerJ Computer Science, vol. 8, Mar. 2022.

[4] B. Souani, et al., "Android Malware Detection Using BERT," Proceedings of the ACNS 2022: Applied Cryptography and Network Security Workshops, LNCS 13285, pp. 575-591, Sep. 2022.

[5] R. Sato, D. Chiba, and S. Goto, "Detecting Android Malware by Analyzing Manifest Files," Proceedings of the Asia-Pacific Advanced Network, pp. 23-31, Dec. 2013.

[6] S. Feldman, D. Stadther, and B Wang, "Manilyzer: automated android malware detection through manifest analysis," Proceedings of the 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems, Oct. 2014.

[7] F. Gagnon, and F. Massicotte, "Revisiting static analysis of android malware," Proceedings of the 10th USENIX Workshop on Cyber Security Experimentation and Test (CSET 17), Aug. 2017.

[8] F. Ullah, et al., "Droid-MCFG: Android malware detection system using manifest and control flow traces with multi-head temporal convolutional network," Physical Communication, vol. 57, Apr. 2023.