안드로이드 접근 제어 정책 연구 동향 분석

엄서정 [1], 조영필 [2]
[1] 한양대학교 컴퓨터소프트웨어학과 (미래자동차-SW 융합전공) 석박통합과정
[2] 한양대학교 컴퓨터소프트웨어학과 교수

tjwjdus@hanyang.ac.kr, ypcho@hanyang.ac.kr

Research Trends in Access Control Policy of Android

Seo-Jung Urm[1], Yeong-Pil Cho[2]
[1]Dept. of Computer and Software (Automotive-Computer Convergence), Han-Yang University
[2]Dept. of Computer Science, Han-Yang University

요 약

안드로이드는 오픈소스로 많은 제조 업체들이 자신들의 서비스에 맞게 커스터마이징 하기도 한다. 다만, 접근 제어 정책들을 올바르지 못하게 수정하거나 추가함에 있어서 발생하는 취약점을 통해 공격자가 해당 접근 제어 메커니즘을 우회하여 보안적 위험 문제가 발생할 수 있다. 이에 따라 안드로이드의 다양한 접근 제어 정책들을 살펴보고, 제조 업체가 추가하는 접근 제어 정책을 검사하거나 발생하는 취약점에 따라 우회할 수 있는 방법들을 찾아내는 최근 연구들의 동향을 분석하고자 하며 앞으로의 관련 연구의 지속적인 필요성을 제시한다.

1. 서론

안드로이드는 구글에서 개발한 운영체제로, 세계 모바일 운영체제 점유율 70%로 많은 모바일 기기 운영체제로 채택되고 있다[1]. 안드로이드는 AOSP (Android Open Source Project)로 많은 제조 업체들이 자신들의 서비스에 맞게 커스터마이징 하기도 한다. 안드로이드에는 다양하고 복잡한 접근 제어 정책이 시행되고 있어, 제조 업체들이 자신들의 서비스에 맞게 기능들을 도입하면서 해당 서비스에 적절한 접근 제어 정책을 추가하지 못하거나, 기존 접근 제어의 무결성을 위배하는 정책을 추가함으로써 보안적으로 취약점이 발생하기도 한다. 이러한 취약점을 통해 공격자가 보안 정책을 우회함으로써 민감한 데이터 유출 등 보안적 위험이 초래된다.

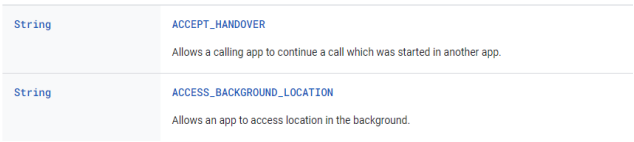
안드로이드 접근 제어 정책에는 DAC(Discretionary Access Control), MAC(Mandatory Access Control)등이 있고, Android Permission, Capabilities 등의 권한을 검사하는 메커니즘도 있다. 최근 연구에서는 이러한 접근 제어 메커니즘을 분석하여 공격자가 우회할 수 있는 취약점을 발견하거나 올바른 접근 제어 메커니즘을 검사하는 연구가 진행되고 있으며, 끊임없는 보안적 위험 문제로 제기되고 있다.

이에 본 논문은 안드로이드의 접근 제어 메커니즘을 살펴보고, 관련 연구의 동향을 분석함으로써 앞으로의 관련 연구의 지속적인 필요성을 제시한다.

2. 안드로이드 권한 모델

2.1. Android Permission

안드로이드에는 권한을 나타내는 종류가 두가지 있다. 그 중 Permission의 경우 미들웨어에서 적용되는 권한 모델로 (그림1)[2]와 같이 앱의 Manifest.xml 파일에 정의되어 있다. Permission의 경우 해당 주체가 특정한 기능을 수행할 때, 기능에 대한 권한을 나타내는 것으로 안드로이드 서비스 프레임워크에서 서비스에 대한 기능이 호출되고 실행될 때 해당 기능에 대한 Permission 검사가 이루어지게 된다.



(그림1) Android permission 예시[2]

2.2. Capabilities

Capabilities는 리눅스의 전통 권한 모델 중 하나로 강력한 수퍼 유저 권한을 38개의 권한으로 세분화하여 수퍼 유저 권한과 관련된 특권들을 독립적으로 수행할 수 있도록 한다.

3. 안드로이드 접근 제어 정책

3.1. 임의 접근 제어(DAC)

안드로이드는 리눅스 기반의 모바일 OS로, 내부적으로 Linux의 접근 제어 메커니즘을 활용하고 있다. 리눅스의 접근 제어 모델은 각 주체들에 사용자 ID(UID)와 그룹 ID(GID)를 할당하고, 읽기·쓰기·실행 권한을 부여하여 시스템 자원에 대한 접근을 제어한다. 이때 사용자나 그룹이 자원에 대한 소유자인 경우 다른 주체에 대해 접근 제어 권한을 설정할 수 있다. 그러나 DAC의 경우 런타임 도중 권한이 변경될 수 있으므로 사용자나 그룹 권한이 탈취될 경우, 소유하고 있는 모든 자원에 대한 접근이 가능하기 때문에 보안적 취약점이 있다. 이에 안드로이드에서는 DAC와 더불어 SEAndroid (MAC)와 함께 접근 제어가 시행되고 있다.

3.2. 강제 접근 제어(MAC)

리눅스에서는 SELinux 라고 불리지만, 안드로이드에서는 안드로이드 포맷에 맞게 수정된 SEAndroid 가 적용되고 있다. SEAndroid 는 강제 접근 제어 모델로 미리 정해진 정책 규칙과 보안 level 에 맞게 주체의 접근 권한과 자원에 대한 허용 등급을 매칭하여 접근을 제어한다. 소유자일지라도 해당 정책 규칙에 맞지 않거나 보안 level 이 맞지 않다면 접근할 수 없기 때문에 강력한 보안성을 띈다.

강제 접근 제어 모델에서는 (그림 2)과 같이 Type Enforcement(TE) Rule 라는 규칙을 채택하여 접근 제어를 시행한다. TE Rule 에는 allow 문과 neverallow 문으로 구성되어 있으며, 주체의 type 과 자원 객체에 대한 type 에 따라 권한을 부여하거나 부여하지 않는다.

```
allow        user_t    user_h_t:file    {create read write}
allow        net_d         sysfs:file   {write}
neverallow   user_t        sysfs:file   {write}
```

(그림 2) TE Rule 예시

4. 단일 및 다중 접근 제어 정책 분석 관련 연구

4.1. 단일 접근 제어 정책

최근 연구들 중 단일 접근 제어 정책을 고려한 연구는 SEPAL[3], LeakDetector[4] 등이 있다. SEPAL 의 경우 딥러닝 기반 정책 검사 도구이며, SEAndroid 을 기반으로 TE RULE 을 분석하여 새로운 정책이 기존 정책들의 무결성을 위배하지 않는 지 등을 검사한다. SEPAL 은 NLP 기반 딥러닝 모델을 사용하며 낮은 false positive 로 592,236 개의 커스터마이징 한 정책 rule 에서 7,111 개의 올바르지 않은 rule 들을 발견했다. LeakDetector 의 경우 Android permission 기반으로 이루어지는 접근 제어 메커니즘에서 발생되는 취약점으로, IPC(inter process communication) 메커니즘에서 안드로이드 framework 에서 보내는 intent object 를 통해 민감한 데이터 유출이 이루어질 수 있는 유형을 처음으로 고려한 연구이다. 해당 연구에서는 LeakDetector 로 10 개의 안드로이드 시스템에 적용했으며, 접근 제어를 위반하고 권한 없는 앱들이 민감한 데이터를 탈취할 수 있는 36 개의 케이스들을 발견했다.

4.2. 다중 접근 제어 정책

최근 연구들 중 다중 접근 제어 정책을 고려한 연구는 BigMAC[5], PolyScope[6], IAceFinder[7] 등이 있다. BigMAC 의 경우 안드로이드 펌웨어 이미지에서 DAC, MAC, capability 를 기반으로 정책들을 추출하고, 분석하여 공격 가능한 경로를 검색할 수 있다. 정책 분석에 있어서 98%의 정확도를 보이고 삼성 S8+와 LG G7 펌웨어에서 커널 모니터링 서비스와 IPC 관련 취약점들을 발견했다. PolyScope 의 경우 DAC, MAC, Android permission 을 기반으로 파일 시스템에서 공격할 수 있는 공격 유형을 정의하여 동적 테스팅을 통해 취약점을 발견했다. PolyScope 는 각각 5 개의 구글, OEM 안드로이드에 적용하여 권한 확장으로 공격이 가능한 케이스를 발견했다. IAceFinder 의 경우 Android permission, UID 를 고려하여 안드로이드 서비스의 자바 인터페이스와 네이티브 인터페이스 간의 접근 제어 불일치를 찾아내 취약점을 발견한 연구이다. IAceFinder 는 자바 컨텍스트만 고려한 기존 연구들과 다르게 처음으로 네이티브 컨텍스트 또한 분석한 최초의 연구이며 14 개의 안드로이드 오픈소스 Rom 에 적용하여 23 개의 불일치 케이스를 발견했다.

5. 결론

안드로이드의 권한들과 접근 제어 정책을 살펴보며, 이러한 단일 접근 제어 정책, 다중 접근 제어 정책을 고려하여 안드로이드의 접근 제어 메커니즘을 우회하거나 해당 메커니즘의 취약점을 통하여 발생할 수 있는 보안적 위험을 찾아내고, 검사하는 연구들의 동향을 분석하였다. 앞으로 새로운 안드로이드 버전에 따라 커스터마이징 되는 정책들과 새롭게 발견되는 취

약점들이 늘어가는 상황과 복잡한 접근 제어 메커니즘 속에서 강력하고 확실하게 접근 제어가 이루어질 수 있도록 세밀한 연구가 지속적으로 활발히 이루어져야 할 필요성을 제시한다.

Acknowledgment

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2020-0-01840, 스마트폰의 내부데이터 접근 및 보호 기술 분석)

참고문헌

[1] Mobile Operating System Market Share Worldwide, https://gs.statcounter.com/os-market-share/mobile/worldwide

[2] Android Developer documentation, https://developer.android.com/reference/android/Manifest.permission

[3] Yu D, Yang G, Meng G, Gong X, Zhang X, Xiang X, Wang X, Jiang Y, Chen K, Zou W, Lee W, Shi W “SEPAL: Towards a large-scale analysis of SEAndroid policy customization.” In Proceedings of the 30th The Web Conference, April. 2021.

[4] Hao Zhou, Haoyu Wang, Xiapu Luo, Ting Chen, Yajin Zhou, Ting Wang, “Uncovering Cross-Context Inconsistent Access Control Enforcement in Android”, in Network and Distributed Systems Security Symposium (NDSS), November. 2022.

[5] Grant Hernandez, Dave Jing Tian, Anurag Swarnim Yadav, Byron J Williams, and Kevin RB Butler. “BIGMAC: Fine-Grained Policy Analysis of Android Firmware” in Proceedings of the USENIX Security Symposium, 2020.

[6] Yu-Tsung Lee, William Enck, Haining Chen, Hayawardh Vijayakumar, Ninghui Li, Zhiyun Qian, Daimeng Wang, Giuseppe Petracca and Trent Jaeger “PolyScope: Multi-Policy Access Controlk Analysis to Compute Authorized Attack Operations in Android Systems” in Proc. 30th USENIX Security Symp., Aug. 2021.

[7] Hao Zhou, Haoyu Wang, Xiapu Luo, Ting Chen, Yajin Zhou and Ting Wang “Uncovering Cross-Context Inconsistent Access Control Enforcement in Android” in Network and Distributed Systems Security (NDSS) Symposium, April. 2022.