Computer Security SNU
Fall 2015

Project 4. Information Flow Tracking in Android

Due Data: 11:59am, Dec. 9, 2015

이 프로젝트에서는 TaintDroid 를 사용하여 모빌 앱을 분석하는 기본과제와 여러 주제 가운데에서 하나를 택하여 수행하는 선택 과제로 구성되어있다.

TaintDroid 는 Android 에서 dynamic taint tracking 을 빠르게 수행할 수 있도록 만든 시스템으로 모빌 보안쪽으로 가장 큰 영향을 준 시스템 중의 하나이다. 자세한 내용은 다음 논문 [1]을 참조한다.

TaintDroid 설치 방법은 http://appanalysis.org/download.html 에 자세히 나와 있다. TaintDroid 는 Ubuntu 에서 build 해야 하기 때문에 Ubuntu machine 이나 Ubuntu virtual machine 을 사용하여야 한다.

이 과제의 deliverable 은 본인이 개발을 한 github source repository 와 report (10page 이내)이다. Github source repository 에는 본인이 새롭게 개발한 코드를 어떻게 수행해야 하는지에 대한 README 가 있어야 한다.

기본 과제

개인 정보를 사용하면서 인터넷을 사용하는 세개~ 다섯개의 앱 (예: 카카오톡, 페이스북, 네이버, 모빌 금융, 게임 앱 등 본인이 분석하고 싶고 인기도가 높은 앱)을 Google Play 에서 선택하여 개인 정보가 어떻게 사용되는지 분석한다. 어떤 개인 정보가 어떤 상황에서 어떤 형태로 유출되는지 분석하고 해당 개인 정보가 앱과 시스템의 어떤 수행 경로를 통해서 외부로 유출되는지 설명할 수 있어야 한다.

앱을 분석하기 위해 TaintDroid 를 에뮬레이터를 사용하여 수행하거나 아니면 실제 디바이스에서 수행한다. TaintDroid 웹 사이트에서 지원되는 공식 버전은 Android 4.3 에 기반했기 때문에 해당 버전에서 수행되는 앱을 선택해야 한다. 에뮬레이터를 사용하여 분석하는 경우에는 다음에 주의해야 한다. 앱에 따라에뮬레이터에서는 잘 동작하지 않는 경우도 있다. 예로 수행 환경이에뮬레이터인 경우를 검사해서 앱을 종료하는 경우도 있다. 앱을 잘 선택해서에뮬레이터에서 잘 동작하도록 하던지 아니면 시스템을 고쳐 앱이에뮬레이터환경이라도 잘 수행되도록 한다.

앱을 분석하기 위해서는 입력 이벤트를 주어야 한다. Monkey 와 같은 fuzzing tool 이 있지만 우리가 원하는 것은 시나리오를 정해 놓고 해당 시나리오를 수행하는 것이다. 이를 위해 Robotium [2] 을 사용한다.

선택 과제 (다음 내용 중 하나를 반드시 선택해서 수행해야 한다.)

1. 앱에서 많이 사용되는 library 몇 개 (예: caching library, security library)를 선택하여 library 에서 taint tracking 이 어떻게 되는지 조사한다. 만약에 library 에서 tracking 이 잘 안 되면 (overtainting 또는 undertainting)이 발생하면 그 원인을 파악해서 해결책을 제시한다.

2. TaintDroid 는 native library 안에서 tracking 은 지원하지 않는다. Native library 안에서의 tracking 을 구현해서 native library 를 사용하는 app 에서 native library 를 포함한 tracking 을 보여준다.
3. 앱 분석을 수행할때 TaintDroid 에서 외부로 taint 된 개인 정보 데이터를 보내는데 잡히지 않는 경우가 있으면 해당하는 부분에서 표시를 하도록 시스템을 수정하고 앱을 분석하여 보여준다.
4. TaintDroid 는 성능을 위해 정확도를 trade 하는 경우가 있다. 이런 경우에 taint 된 데이터가 유출되었다고 검출되었을때 실제로 그런지 (true positive) 아닌지 (false positive)인지 자동으로 측정하는 시스템을 설계하고 그 시스템이 합리성을 보여준다. 원하면 false negative 도 고려해서 시스템을 설계해도 된다.
5. 몇 개가 아니라 100 개 이상의 앱을 선택하여 분석한 후에 앱들의 특성에 대해 더 자세한 분석을 한다. 이 분석을 가능하게 하기 위해 최대한 자동으로 앱을 분석할 수 있는 시스템을 개발한다.
6. Monkey 나 Robotium 같은 input generation tool 은 제한된 수행 경로만 할 수 있다. Symbolic execution 을 구현하여 체계적으로 많은 수행 경로를 체크하도록 시스템을 설계한다. Robotium 같은 tool 과 비교하여 Symbolic execution 이 효과적인지를 파악하기 위해 code coverage 를 계산해야 한다.'
7. TaintDroid 에서 coarse-grained taint tracking 으로 구현된 부분을 전부 fine-grained taint tracking 으로 구현하여 앱 분석에서의 차이점을 기술한다.
7. 현재 TaintDroid 는 Dalvik VM 에서 taint tracking 을 구현하였다. Android 5.0 부터는 ART 를 수행 환경으로 사용하고 있다. ART 환경에서 taint tracking 을 구현한다. 이 주제는 매우 난이도가 높음을 미리 경고한다.
8. 더 깊게 보고 싶은 주제를 직접 제안하여 승인을 받으면 수행 가능하다. 이경우 제안 주제는 위에 나온 주제와 비교해서 난이도가 비슷해야 한다.

References

[1] W. Enck et al., TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, ACM Transactions on Computer Systems (TOCS), June 2014.
[2] Robotium. https://github.com/robotiumtech/robotium