Tiny IML의 보안 정보 흐름 분석

김제민○ 김기태 유원희
인하대학교
jeminya@naver.com, kkt@empas.com, whyoo@inha.ac.kr

Analysis of Secure Information Flow in Tiny IML

Jemin Kim○, Ki-Tae Kim, Weon-Hee Yoo
Inha University

1. 서론

소프트웨어의 보안 문제를 해결하기 위한 방법으로 바이러스 백신, 방화벽, 암호화, 보안 정보 흐름 분석 등이 많이 연구되고 있다.

그 중 본 논문에서는 보안 정보 흐름 분석에 대해 논의한다. 프로그램의 변수 중, 사용자의 개인 정보와 같은 보안이 요구되는 값을 갖고 있는 변수로부터 다른 악의적인 사용자가 쉽게 접근할 수 있는 변수로 정보의 흐름이 존재하는 경우를 정보 누출이라고 한다. 이러한 정보의 누출 여부를 분석하는 것을 보안 정보 흐름 분석이라고 한다.

보안 정보 흐름 분석에 관한 연구 중 고급 언어에서 보안 정보 흐름 분석에 관한 연구는 많이 수행되었다. 하지만 사용자가 실행하게 되는 프로그램은 저급 언어의 형태로 되어있고, 고급 언어로 표현된 프로그램 소스를 얻기 힘들다. 따라서 저급 언어에서의 보안 정보 흐름 분석이 필요하다.

본 논문에서는 Tiny IML(Intel Machine Language)을 정의한다. 또한, Tiny IML 프로그램의 제어 흐름 그래프를 구성하는 방법과 분석에 사용되는 전달 함수를 정의하고, 분석을 위해 구체적 프로그램의 의미를 요약된 프로그램 의미로 바꾸는 방법을 제시한다. 마지막으로, 구성된 제어 흐름 그래프에 전달 함수를 적용함으로써 정보의 누출 여부를 확인한다.

2. 본론

본 논문은 x86 어셈블리 언어의 부분 집합인 Tiny IML(Tiny Intel Machine Language)을 정의한다. 또한, 레지스터 주소 지정, 직접 값 주소 지정, 직접 메모리 주소 지정, 간접 메모리 주소 지정만이 가능하다고 가정하고 프로시저간의 호출은 고려하지 않는다. Tiny IML은 기본적인 명령어인 add, sub, inc, cmp, jmp, je, mov, lea를 제공한다.

배정이 발생할 때, 배정의 왼쪽(Left Hand Side)의 보안 수준이 배정의 오른쪽(Right Hand Side)의 보안 수준보다 낮지 않아야 한다. 그리고 조건절에 높은 보안 수준의 변수가 있고, Then절과 Else절에서 낮은 보안 수준을 갖는 변수로의 배정이 일어날 때 그 변수의 보안 수준을 높이든지, 낮은 보안 수준으로의 배정을 금지해야 한다. 이처럼 낮은 보안 수준의 변수에 높은 보안 수준의 변수의 값이 배정되지 않는 성질을 불간섭이라고 한다. 불간섭을 만족하지 않을 때 정보의 흐름이 안전하지 않다고 한다.

Tiny IML프로그램에서의 정보 누출을 검사하기 위해서는 구체적 도메인, 즉 프로그램의 실제 의미를 요약하여 요약 도메인으로 만들어야 한다. 다시 말해 값을 요약하고 연산을 요약하고 요약 연산이 수행될 때 정보의 누출이 일어나는지 검사하게 된다.

요약 도메인을 정의할 때, 집합 SecurityLevel은 보안 수준을 나타내며, AbstractLoc은 레지스터의 집합 Register와 추상메모리변수의 집합 AbstractMemVar의 합집합으로 메모리나 레지스터를 가리킨다. SecurityLevelMap은 추상 메모리 위치 집합으로부터 보안 수준으로의 사상을 갖는 집합이며, 집합 AbstractMap은 추상 메모리 위치 집합으로부터 구간으로의 사상이다. 요약 상태 집합 AbstractState는

요약 사상 집합 AbstractMap과 보안수준사상집합 SecurityLevelMap으로 이루어지는데 as ∈ AbstractState는 어떤 간선의 꼬리에 있는 명령어가 실행되기 전의 상태를 의미하며 레지스터나 추상메모리변수가 갖는 구간과 보안수준을 의미한다.

메모리의 주소를 표현하기 위해 구간(interval)을 사용한다. 구간으로써 어떤 프로그램 포인트와 프로그램 포인트 사이에서, 레지스터나 메모리 위치가 가질 수 있는 값이나 주소의 범위를 표현할 수 있다.

보안 정보 흐름 분석은 다음과 같이 수행된다. Tiny IML 프로그램을 입력으로 해서 각각의 노드에 명령어를 갖고 제어의 흐름을 간선으로 표현하는 제어 흐름 그래프를 구성한다. 간선 → 의 왼쪽에 있는 노드를 꼬리, 간선의 오른쪽에 있는 노드를 머리라고 했을 때, 구성된 제어 흐름 그래프의 모든 간선에 대해 전달 함수를 적용하면 간선의 꼬리에 있는 명령어가 실행되기 전의 보안 수준을 이용해 명령어에 따른 정보의 누출을 검사하게 된다.

제어 흐름 그래프를 구성하기 위한 후행자 관계를 이용한다. 또한, 제어 흐름 그래프에서 고급 언어의 if 문과 같은 구조를 인식하여 jmp에 의한 루프나 분기의 범위를 알 수 있어야 되기 때문에 제어 의존 영역을 구성한다. 어떤 분기 명령어에 의해서 영향을 받는 명령어들의 집합을 그 명령어의 제어 의존 영역이라고 한다.

구성된 제어흐름 그래프의 정보와 분기 명령어의 제어 의존 영역 정보를 이용해 보안 정보 흐름 분석을 수행하게 되는데 제어 흐름 그래프의 각 노드의 들어오는 간선과 나가는 간선에 대한 전달 함수를 정의하고 정의된 전달 함수에 의해 분석을 수행하면 정보의 누출 여부를 확인할 수 있다

3. 결론

본 논문에서는 x86 어셈블리어의 부분 집합인 Tiny IML을 정의하여 보안 정보 흐름 분석을 수행하였다. 보안 정보 흐름 분석을 위해 불간섭의 성질을 이용하였다. 또한, 요약 해석을 이용해 제어 흐름 그래프를 구성하고 실제 정보의 누출이 발생하는지를 분석하였다.

그러나 본 논문은 프로시저 안에서의 보안 정보 흐름에 대해 논의하였지만, 실제의 상황에서는 프로시저간의 보안 정보 흐름 분석을 수행할 수 있어야 한다.

또한, 보안 정보 흐름 분석이 자동으로 수행되기 위해서는 사용자가 높은 보안 수준을 직접 할당할 필요가 없어야 하며, 세밀하고 상세한 분석을 위해 정보 흐름 정책을 세분화하고 자동으로 보안 수준을 할당하는 분석 방법이 필요하다.

향후 연구로는 프로시저 간의 보안 정보 흐름을 분석하는 것과 이러한 이론적인 배경으로 실제 분석기를 구현하는 것이 있다. 또한 사용자의 보안 수준 배정없이 자동으로 분석을 수행해야 하며 이를 위해 좀 더 세분화되고 자동화된 보안 정책을 세우는 방법에 대한 연구가 요구된다.

참고문헌

[1] Dorothy E. Denning. Secure information flow in computer systems. PhD thesis, West Lafayette, IN, USA, 1975.

[2] Samir Genaim and Fausto Spoto. Information flow analysis for java bytecode. In *LNCS 3385*, volume 3385, pages 346 - 362. Springer-Verlag, 2005.

[3] Gilles Barthe and Tamara Rezk. Non-interference for a jvm-like language. In *TLDI '05*, pages 103 - 112, New York, NY, USA, 2005.

[4] Patrick Cousot and Radhia Cousot. Abstract interpretationa unied lattice model for static analysis of programs by construction or approximation of fixpoints. *In POPL* , 1977.

[5] Dorothy E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236 - 243, 1976.

[6] Andrew C. Myers. Jflow: practical mostly-static information flow control. In *POPL '99*, pages 228 - 241, New York, NY, USA, 1999.

[7] Gogul Balakrishnan and Thomas W. Reps. Analyzing memory accesses in x86 executables. In *CC '04*, April 2004.