보안운영체제의 검증을 위한 정형기법 및 도구에 관한 연구

조지호[0*], 이동익*, 김형천**, 강정민**, 이진석**
*광주과학기술원 정보통신공학과
**국가보안기술연구소
*{jhcho[0], dilee }@kjist.ac.kr
**{khche, jmkang, jinslee }@kjist.ac.kr

A Study on Formal Methods and Tools for Verification of Secure OS

Cho, Ji-Ho[0*], Lee, Dong-Ik*, Kim, Hyung-Chun**, Kang, Jung-Min**, and Lee, Jin-Seok**
*Dept. of Info. & Comm., Kwang-ju Institute of Science and Technology
**National Security Research Institute

요 약

본 논문에서는 소프트웨어 공학에서 널리 사용되고 있는 정형기법을 보안 운영체제의 검증에 활용하기 위해 각 검증기법의 특성과 도구에 대해서 설명한다. 정리증명의 대표적인 도구인 PVS와 모델체킹의 대표적 도구인 SMV, 특수한 목적을 위해 개발된 NPA등에 대해서 알아보고, 각 방법을 비교, 분석하여 보안운영체제의 검증에 적합한 도구를 찾아본다.

1. 서 론

최근 공격형태가 다양화되고 지능화됨에 따라 기존 보안 시스템의 한계가 드러나게 되었고, 이에 대한 해결책으로 보안운영체제에 관한 연구가 활발히 이루어지고 있다. 보안운영체제는 접근통제를 비롯하여 식별 및 인증, 암호화, 침입 탐지 등의 보안 서비스를 강화하여 구현된다.

그러나 이러한 보안운영체제의 안전성을 입증할 수 있는 근거를 제시하기 어렵다. 보안운영체제의 안전성에 대한 가장 직접적인 평가방법은 운영체제 내의 모든 가능한 연산을 대상으로 실행결과의 안전여부를 검사하는 것이지만, 이는 현실적으로 불가능하다. 그러나 정형기법을 이용할 경우 운영체제 동작 논리상의 안전성 보장 여부를 이론적으로 증명해 볼 수 있다.

정형기법은 정형명세(Formal Specification)와 정형검증(Formal Verification)으로 나뉜다. 정형명세는 수학적 개념을 기초로 한 의미(Semantics)와 제한된 구문(Syntax)으로 구성되어있는 언어를 이용하여 작성된다. 이러한 정형명세의 대표적인 언어에는 Z, VDM, CSP, Petri Net등이 있다[1,2].

정형검증의 대표적인 도구로는 HDM, Gypsy Verification Environment, PVS, SMV, Spin 등이 있다.

본 논문에서는 보안운영체제 검증에 적합한 정형기법에 대하여 조사해보고자 한다. 운영체제는 매우 규모가 큰 소프트웨어 시스템이므로 한번에 검증하는 것보다는 작은 부분으로 나누어 모듈별로 검증하는 것이 합리적이다. 각각의 특성에 맞는 검증기법의 선택을 위해 현재 소프트웨어 공학분야에서 널리 이용되고 있는 정형기법들을 비교, 분석했다. 2장에서는 정형검증에서 많이 사용되고 있는 도구들에 대해서 알아보겠다. 3장에서는 각 검증기법을 비교, 분석하여 보안 운영체제의 검증에 적합한 검증기법에 관하여 논한다. 마지막으로 4장에서 결론을 맺는다.

2. 정형검증 기술

정형검증은 모델체킹(Model-Checking)과 정리증명(Theorem Proving)으로 분류할 수 있다. 정리증명은 공리(axioms)와 시스템의 정확성(correctness)을 증명하기 위한 증명규칙을 사용한다. 초기에는 이러한 증명이 수동으로 이루어 졌으나, 현재 증명의 한 단계에서 다음 단계로 가기 위한 여러 방법들을 제안하는 체계적인 도구들이 개발되어 이용되고 있다[3].

모델체킹은 상태기계로 표현된 시스템의 명세가 특정한 속성을 만족하는지 검증한다. 이 기술의 장점은 정리증명이 사용자의 수학적 지식에 의존하는 것에 비해 모든 검증이 자동으로 이루어진다는 점에 있다. 모델체킹 기술은 시스템의 모든 가능한 상태를 체크하여 특정 속성이 만족되는지를 검증한다. 속성은 주로 시제논리를 이용하여 작성된다. 모델체킹은 시스템의 모델링, 명세, 검증의 세 단계로 구성된다[3].

다음은 안전성 검증을 위해 활용가능한 대표적인 도구들이다.

2.1. PVS(A Prototype Verification System)

PVS는 기계적으로 체크된 명세와 사용자가 읽을 수

있는 증명을 제공하지만, 완벽한 개발 방법론을 제공하지는 않는다. PVS는 중요한 시스템을 위해 좋은 명세를 만들고, 고유한 속성을 증명하는데 초점을 맞추고 있다.

PVS는 PVS명세 언어(The PVS Specification Language)와 PVS 증명검사기(The PVS Proof Checker)로 구성되어 있다[4].

PVS 언어는 고차논리(higher-order logic)를 기반으로 대상을 모델링하며, 모듈화를 지원한다. 작성된 논리는 증명검사기를 통해 검증되는데, 증명검사기는 결론에서 시작하여 추론규칙을 통해 점진적으로 증명한다.

PVS는 단지 컴퓨터 보안 분야에서만 아니라 많은 분야에서 다양하게 사용되고 있다. NASA 센터에서 우주선 프로젝트와 항공전자공학을 위한 요구사항 분석에 사용되었고, 결함-허용 알고리즘과 분산 알고리즘을 성공적으로 분석하였다. 그 외 다른 애플리케이션의 검증에서 활발히 사용되고 있다[1].

2.2. SMV(The Symbolic Model Verifier)

SMV는 상태기계로 표현된 시스템이 주어진 속성을 만족하는지를 체크하는 도구이다. 이때 대상은 상태기계로 모델링하고, 속성은 시제논리를 이용하여 표현한다. SMV에서 유한상태기계 M과 시제논리식 Φ가 입력되면, M |= Φ(M satisfies Φ)를 판정하여 결과로 '참'을 출력하거나 '거짓'일 경우에는 반례를 함께 출력한다. 이때 모든 상태공간을 철저하게 탐색한다[2,3,5].

SMV는 순서논리 회로설계 검증[12] 및 보안 프로토콜[13], 유한 상태 실시간 시스템[14], 병행 시스템[15] 등의 검증에 사용되고 있다. Cadence SMV와 NuSMV가 가장 많이 사용되고 있다[1].

2.3. Naval Research Lab. Protocol Analyzer(NPA)

NRL 프로토콜 분석기는 Prolog로 작성되어 있고, 암호 프로토콜, 인증 프로토콜, 키 분배 프로토콜 등을 위한 특수목적 검증 시스템이다. NPA는 Dolev 와 Yao[16]의 term-rewriting 모델을 기초로 하고 있는데, 이 모델에서 공격자는 모든 메시지 트래픽을 읽을 수 있고, 메시지를 변경하고 훼손시킬 수 있을 뿐만 아니라, 합법적인 사용자가 할 수 있는 모든 연산들을 수행할 수 있다고 가정하고 있다. 단, 암호 키나 암호문 등의 특정한 단어를 알지는 못한다고 가정한다. 공격자의 목표는 특정단어를 알아내는 것이다[1,7].

사용자는 안전하지 않은 상태를 기술하고 그 상태가 도달가능 하지 않다는 것을 증명한다. 증명은 도달 가능하지 않은 상태에서 후방 전수조사(backward exhaustive search)를 사용한다. NPA는 상태기계모델의 증명뿐만 아니라, 결점을 찾고, 잠재적 공격을 식별하는데 사용될 수 있다.

NPA Temporal Requirements Language(NPATRL) 은 NPA의 기본적인 언어이다. 이것은 키 분배나 키 협의와 같은 일반적인 요구사항을 표현한다. CAPSL(Common Authentication Protocol Specification Language)은 암호 인증이나 키 분배 프로토콜을 위한 상위 수준 언어이다.

NPA는 프로토콜을 검증하는데 널리 사용되고 있다. 인터넷 키 교환 프로토콜[17] 과 Needham-Schroeder 공개키 프로토콜[18] 검증에 사용되었다[1,7].

3. 보안 운영체제 검증을 위한 검증기법

앞에서 대표적인 검증도구들에 대해서 살펴보았다. PVS는 대표적인 정리증명 기법이고, SMV는 모델체킹의 대표적인 도구이다. NPA는 두 가지 특징이 혼합된(hybrid) 형태의 프로토콜 검증을 위한 도구라 할 수 있다.

보안 운영체제를 한꺼번에 시스템 전체를 검증하기는 매우 어려운 일이다. 따라서 보안 운영체제를 핵심 부분으로 나누어 따로 검증하는 것이 보다 효율적인 방법이다. 앞에서 언급한 것처럼 보안 운영체제에는 인증, 식별, 암호화, 침입탐지, 접근통제 등 여러 보안 기능들이 포함된다.

인증 프로토콜이나 암호 프로토콜 등은 프로토콜 분석기나 모델체킹을 사용하면 효과적으로 검증할 수 있다. 접근통제나 침임 탐지 시스템의 경우는 상태기계로 표현할 수 있기 때문에 모델체킹을 사용하는 것이 더 효과적이다. 또한 프로그램 코드수준의 검증은 모델체킹이나 정리증명을 이용할 수 있다. 보안 정책의 검증은 Z/EVES[11]와 같은 정리증명 도구로 검증하는 것이 효과적이다.

다음 표1은 정리증명 기술과 모델체킹을 비교 분석한 결과이다[8,9,24].

	특징
정리증명	- 높은 수준의 추상화, 강력한 논리 제공 - 상태가 무한한 시스템에 대한 검증 - 복잡한 H/W, S/W 검증에 적합 - 자동화된 정리 증명 도구가 없음 - 식의 유도, 보조정리 등이 사용자에 의해 제시 - 높은 수학적 지식 요구
모델체킹	- 상태가 유한한 시스템에 적합 - 반례 제시로 오류에 대한 직접적인 피드백 제공 - 크고, 복잡한 시스템에 부적합(상태폭발) - 프로토콜 및 하드웨어 검증에 적합

표 1 정리증명과 모델체킹의 비교

다음 그림1은 도구들을 분류한 결과이다. 수직 축은 도구의 사용자가 수학적 지식과 공학적 지식 중 어느 쪽에 더 친숙한지를 나타내고, 수평축은 도구가 범용인지 전용인지를 나타낸다[24].

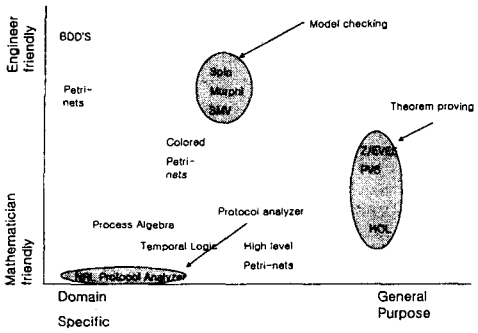

그림 1 검증도구의 분류

표2는 각 정형검증 도구가 실제로 어떻게 사용되었나를 나타낸다.

	정형검증 사례
정리증명	Z를 이용한 BLP 모델의 검증[21] Alloy를 이용한 RBAC의 검증[22] PVS를 이용한 인증 프로토콜의 검증[23]
모델체킹	AutoFocus를 이용한 은행시스템의 보안 프로토콜 검증[19] Murphi를 이용한 Needham-Schroeder 공개키 프로토콜 검증[20] SMV를 이용한 병행시스템의 검증[15]
프로토콜 검증	NPA를 이용한 Needham-Schroeder 공개키 프로토콜 검증[18] NPA를 이용한 인터넷 키교환 프로토콜 검증[17]

표 2 정형검증 사례

4. 결론 및 향후 연구

본 연구에서는 정형검증 기법과 여러 도구들에 비교했다. 정형기법은 명세기법과 검증기법으로 분류되고, 검증기법은 다시 정리증명과 모델체킹으로 분류된다. 정형검증을 활용하면, 보안 시스템이나 프로토콜의 안전성 여부를 논리적이고, 이론적으로 증명할 수 있다.

향후 연구로 광주과학기술원에서 개발한 보안 운영체제를 기능별로 잘 세분화하여 각 부분을 효과적인 검증 도구를 이용하여 안전성을 검증할 계획이다.

5. 참고문헌

[1] M. Bishop, "Computer Security," Addison-Wesley, 2002
[2] 한국소프트웨어 진흥원, "소프트웨어 모델링 및 분석 기법," 2003
[3] E. M. Clarke, Jr., O. Grumberg, and D. A. Peled, "Model Checking," The MIT Press 1999
[4] S. Owre, J. Rushby, and N. Shankar, "PVS: A Prototype Verification System," Proceedings of the 11th International Conference on Automated Deduction, pp. 748-752, 1992
[5] J. Burch, E. Clarke, K. McMillan, D. Dill, and L. Hwang, "Symbolic Model Checking for Sequential Circuit Verification," IEEE Transactions on computer Aided Design of Integrated Circuits and Systems, 13(4), pp.401-424, 1994
[6] K. McMillan, "Symbolic Model Checking, An approach to the state explosion problem," Ph.D. thesis, Carnegie Mellon University, 1992
[7] C. Meadows, "The NRL Protocol Analyzer: An Overview," Journal of Logic Programming 26(2), pp. 113-131, 1996
[8] "Formal Methods": http://www.afm.lsbu.ac.uk/
[9] "Formal Methods Education Resources Tool Pages": http://www.cs.indiana.edu/formal-methods-education/Tools/
[10] J. M. Wing, "A Symbiotic Relationship Between Formal Methods and Security," In Workshops on Computer Security, Fault Tolerance, and Software Assurance: From Needs to Solution. ONR and NSF, 1998
[11] M. Saaltink "The Z/EVES User's Guide," TR-97-5493-06, ORA Canada, 1997
[12] J. Burch, E. Clarke, D. Long, K. McMillan, and D. Dill, "Symbolic Model Checking for Sequential Circuit Verification," IEEE Transactions on Computer-Aided Design of Intergrated Circuits and Systems, 13(4), pp.401-424, 1994
[13] E. Clarke, S. Jha, and W. Marrero, "Using State Space Exploration and a Natural Deduction Style Message Derivation Engine to Verifiy Security Protocols," Proceedings of th IFIP Working Conference on Programming Concepts and Methods, pp 87-106, 1998
[14] S. Campos, E. Clarke, and M. Minea, "Symbolic Techniques for Formally Verifying Industrial Systems," Science of Computer Programming 29(1-2), pp.79-98, 1997
[15] E. Clarke, E. Emerson, and A. Sistla, "Axiomatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications," ACM Transactions on Programming Languages and System 1(2), pp.244-263, 1986
[16] D. Dolev and A. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory 29(2), pp.198-208, 1983
[17] C. Meadows, "Analysis of the Internet Key Exchange Protocol Using the NRL Protocol Analyzer," Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp.216-231, 1999
[18] C. Meadows, "Analyzing the Needham-Schroeder Public Key Protocol: A Comparison of Two Approaches," Proceedings of the 4th European Symposium on Research in Computer Security, pp.351-364, 1996
[19] J. Grunbauer, H. Hollmann, J. Jurjens, and G.Wimmel, "Modeling and Verification of Layered Security Protocols: A Bank Application," International Conference of Computer Safety, Reliability and Security (SAFECOMP), 2003
[20] J. Mitchell, M. Mitchell, and U. Stern, "Automated Analysis of Cryptographic Protocol Using Murphi", In Proceedings of the IEEE Conference on Security and Privacy, pp.141-151, 1997
[21] J. McLean, "A Comment on the Basic Security Theorem of Bell and LaPadula," Information Processing Letters, 20, 1985
[22] A. Schaad and J. Moffett, "A Lightweight Approach to Specification and Analysis of Role-based Access Control Extension," To appear at 7th ACM Symposium on Access Control Models and Technologies (SACMAT), 2002
[23] B. Dutertre and S. Schneider, "Using a PVS Embedding of CSP to Verify Authentication Protocols," In Theorem Proving in Higher Order Logics, pp 121-136, 1997
[24] C. Elks, "Issue in Formal Methods for the Analysis and Description of Security Protocols,": http://www.ee.virginia.edu/~rdw/EE68601/tps.pdf