로드 가능한 정책을 사용하는 SELinux의 정책 재구성을 통한 성능 향상

Performance improvements through policy reorganization in SELinux with loadable policy

고 재 용, 조 경 연, 이 상 길,이 철 훈
충남대학교 컴퓨터 공학과

Jae-yong Ko, Kyung-yeon Cho, Sang-gil Lee, Cheol-hoon Lee
Chungnam National Univ.

요약

Priority-TE란 SELinux 에서 사용하는 타입에 우선순위를 부여하여 성능상의 이점을 확보하기 위한 방식이다. 기존의 Priority-TE 관련 연구는 SELinux 의 모놀리식 정책 방식에 Priority-TE 기반의 정책을 적용하였다. 모놀리식 정책 방식은 사용자가 시스템 부팅 후에도 추가적인 정책의 생성 및 수정, 삭제를 할 수 없기 때문에 사용함에 불편함이 있을 수 있다. 본 논문에서는 로드 가능한 정책방식이 적용된 SELinux 상에서 Priority-TE 기반 정책을 도입하여, 시스템 부팅 후에도 원활히 타입의 우선순위를 변경하여 시스템의 성능을 개선하는 기법을 제안한다.

I. 서론

리눅스의 가장 대표적인 보안 커널인 SELinux는 리눅스 2.6버전 이후로 공식적으로 포함되어 범용적으로 사용될 수 있으며, TE(Type Enforcement)나 세밀한(fine-grained) 접근제어 기법을 통해 보안을 수행한다[1].

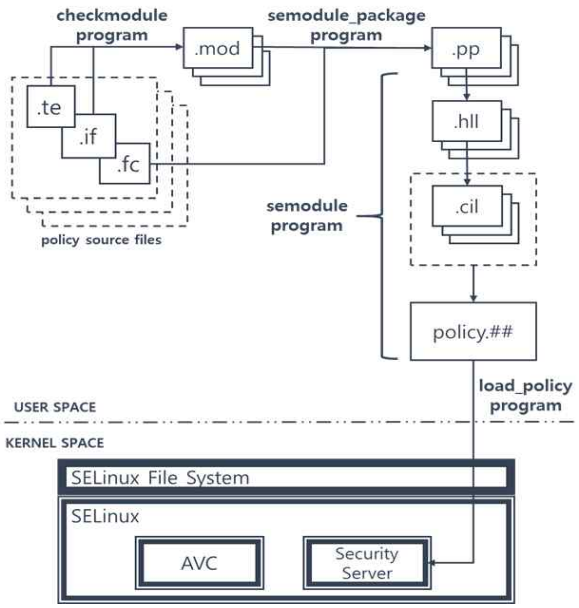
특정 시스템에 SELinux를 적용할 때, 성능상의 오버헤드를 고려할 수 밖에 없게 되는데, 이러한 오버헤드는 SELinux가 가진 구조와 특성에서 비롯된다. SELinux 와 관련한 대부분의 오버헤드는 AVC(Access Vector Cache)에서 캐시미스가 발생했을 때, 정책 데이터베이스에서 접근질의 비교를 하면서 생긴다[2][3].

이에 대한 대안으로 Priority-TE를 적용한 정책을 사용할 경우, 타입에 우선순위를 부여한 정책을 사용하여 캐시미스가 발생했을 경우에도 최선의 성능을 보장할 수 있다. 하지만 기존의 연구는 모놀리식 정책기반에서 연구되었기 때문에 부팅 후에는 정책을 수정, 생성, 삭제 등의 행위를 할 수 없어 사용자가 사용하는데 불편하다는 단점이 있다.

본 연구에서는 부팅 후에도 정책 수정이 가능한 환경에서 Priority-TE를 적용한 정책을 사용함으로써, 시스템 부팅 후에도 동적으로 타입의 우선순위를 변경하여 성능을 개선할 수 있도록 하고자 한다.

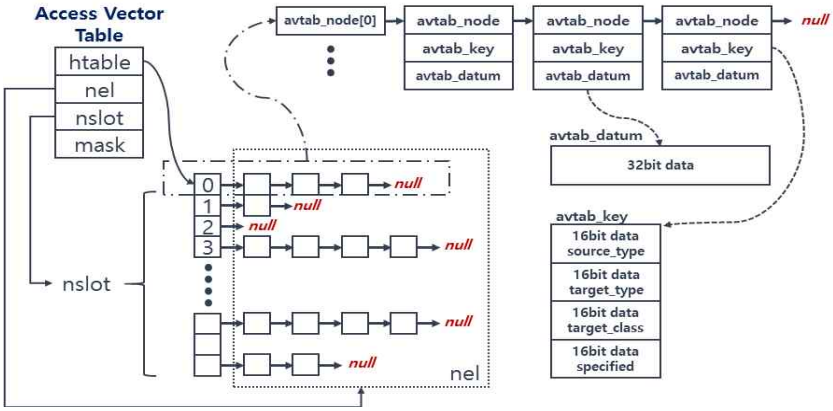
Ⅱ. 본론

1. SELinux 정책생성 과정



▶▶ 그림 1. 로드 가능한 방식이 적용된 SELinux의 정책생성 과정

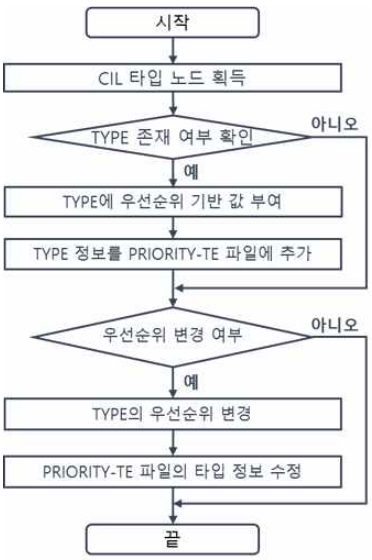
모놀리식 방식과 달리 로드 가능한 방식이 적용된 SELinux에서는 [그림 1]과 같은 방식을 사용하여 정책을 반영한다. [그림 1]에서 정책 소스 파일들에 해당하는 .te(Type Enforcement)파일, .if(Interface)파일, .fc(File



▶▶그림 2. Access Vector Table 구조

Context)파일들이 존재하며 세 가지 파일들은 하나의 정책 모듈을 나타내기 위해 필수적으로 존재한다. 해당 파일들은 Checkmodule, Semodule_package 프로그램에 의해 .pp(Policy Package)로 생성된다[4]. .pp파일은 커널로 적재되기 위해 CIL(Common Intermediate Language)로 컴파일되어 정책 컴포넌트 별로 트리구조로 노드화 된다. 이 후, 모든 정책 정보는 바이너리화 되어 커널 SELinux 보안 서버에 등록된다. SELinux의 접근 질의는 현재 프로세스의 타입과 접근하려는 객체의 타입과 클래스, 그리고 요구되는 권한 순으로 비교하여 그 정보가 모두 일치할 경우 규칙이 보안 서버 내에 존재한다고 판단한다[4].

2. Priority-TE 기법 적용을 위한 설계



▶▶ 그림 3. Priority-TE 적용을 위한 알고리즘

본 연구에서 제안하는 Priority-TE 기법은 규칙을 구성하는 정책 컴포넌트에 우선순위를 부여하는 것이다.

규칙은 정책 컴포넌트의 타입과 클래스, 권한으로 구성되어 있는데 이 중 [그림 2]에 나타난 접근 벡터 테이블(Access Vector Table)의 규칙노드가 선두에 위치하기 위해서는 타입에 해당하는 값이 작아야한다. 이는 우선순위는 값이 작을수록 높은 우선순위라는 것을 나타내며, 이를 통해 접근 질의 과정에서 캐시 미스 시에도 빠른 접근 질의를 수행하게 된다.

본 연구는 Semodule 프로그램의 CIL 컴파일러를 통해 각 정책 컴포넌트가 컴파일 되는 과정에 [그림 3]의 알고리즘을 추가하였다. CIL 컴파일러는 정책 컴포넌트에 바이너리 값을 부여한 뒤, 해당 값을 통해 규칙들을 바이너리 값으로 부여하고 생성하는 과정을 거친다. [그림 3]의 알고리즘은 정책 컴포넌트의 값을 부여하는 과정과 규칙을 생성하는 과정에서 수행된다.

Ⅲ. 결론

SELinux의 접근질의는 모든 시스템 콜에 대하여 이뤄지기 때문에, 오버헤드는 필수불가결한 요소이다. 응답시간이 중요한 프로세스는 SELinux를 적용할 때 이러한 점을 고려할 수 있어야하며, 본 연구에서 제안된 방법을 적용할 경우, 성능 개선 효과와 편리한 정책 개발 환경을 얻을 수 있게 된다.

■ 참 고 문 헌 ■

[1] Frank Mayer, Karl Macmillan, and David Caplan, SELinux by Example, 2006.
[2] http://fedoraproject.org/wiki/SELinux/checkpolicy
[3] A. Kalyanasundaram, B. B. Roy, and S. Rao, “Exploiting Data Parallelism in SELinux Using a Multicore Processor,” in Proceedings of the 47th Annual National Convention of Computer Society of India (CSI), 2012.
[4] Haines, Richardm, The selinux notebook, 2014.