임베디드 리눅스 시스템에 SELinux 적용 방법 연구

Study on Method that Applies SELinux on Embedded Linux System

조 경 연, 고 재 용, 이 상 길, 이 철 훈
충남대학교

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

요약

최근 IoT 디바이스가 증가함에 따라 IoT 환경에서 사용되고 있는 임베디드 시스템에 대한 보안의 중요성이 대두되고 있다. SELinux는 Linux 의 보안성을 강화시키는 보안 강화 커널로, 시스템의 취약점을 이용한 공격을 방지할 수 있는 Linux 시스템 보안의 핵심적인 요소이다. 본 논문에서는 임베디드 리눅스 빌드 시스템중 가장 활발히 진행되고 있는 Yocto 프로젝트를 활용하여 임베디드 리눅스 시스템에 SELinux를 적용하는 방법을 연구하고자 한다.

I. 서론

최근 IoT(Internet of Things) 환경에서 사용되는 디바이스가 증가함에 따라 임베디드 시스템의 보안 위협에 대한 우려 또한 증가하고 있다. IoT 환경의 임베디드 시스템은 네트워크에 연결되어 있어 공격자가 취약점을 악용하여 시스템을 파괴하거나 정보를 도용하여 다른 시스템을 공격할 수 있는 환경에 노출되어 있다. 이와 같은 공격을 방지하기 위해 PC 에서 사용되는 보안 패치나 백신 소프트웨어를 임베디드 시스템에 적용시키기에는 리소스가 제한되어 있다는 한계가 있다.

공격을 막기 위한 다른 방법으로 임베디드 시스템에 접근제어(Access Control)를 적용하여 시스템을 보호할 수 있다. 접근제어는 보안 정책을 기준으로 하여 정책 내에 허용되지 않는 접근을 제어한다. Linux 커널에는 MAC(Mandatory Access Control) 접근제어를 구현한 SELinux(Security Enhanced Linux)를 통해 보안을 강화시킬 수 있다[1][2]. 기본적으로 커널 컴파일 시에 SELinux에 대한 기능이 비활성 되어 있어 이를 시스템에 적용하기 위해서는 별도의 설정이 필요하다. 본 논문에서는 임베디드 리눅스 시스템에 SELinux를 적용하기 위해 Yocto 프로젝트를 활용하는 방법을 제시한다.

Ⅱ. 관련 연구

1. SELinux

SELinux는 오픈소스인 Linux 커널의 보안 강화를 위해 미 국방성 주도로 진행된 프로젝트로, Linux 커널 2.6 버전부터 공식적으로 통합되었다. SELinux는 리소스에 대한 소유자와 그룹, 기타 메커니즘을 이용하여 읽기, 쓰기, 실행 권한을 제어하는 기존의 Linux와는 달리 시스템 내의 각 프로세스와 리소스에 보안 속성이 부여된다. 보안 정책에 정의된 접근 허용 여부에 따라 프로세스의 리소스에 대한 접근이 결정된다[3].

2. Linux 기반 임베디드 기기에 SELinux 적용

Linux를 사용하는 Nokia 770 Internet Tablet(N770)에 SELinux를 적용한 연구이다. N770에서 xattr을 지원하도록 파일 시스템 수정 및 커널 설정을 변경하고, N770에 맞게 SELinux 정책 크기 축소 및 정책 규칙 수정을 하였다. 또한 SELinux 라이브러리 및 응용 프로그램을 크로스 컴파일한 후 기기에 포팅 하여 임베디드 기기에서 SELinux를 활성화 하는 방법을 제시하였다[3]. 그러나 제시한 방법이 N770이라는 특정 하드웨어 아키텍처에 한정되어 있다는 한계가 있다.

Ⅲ. 본론

1. Yocto 프로젝트

Yocto 프로젝트는 하드웨어 아키텍처에 관계없이 임베디드 제품을 위한 커스텀 Linux 시스템을 개발하는데 도움이 되는 템플릿, 방법 및 툴을 제공하는 오픈 소스 프로젝트이다. 즉, Yocto 프로젝트는 임베디드 리눅스 빌드 시스템이며 루트파일 시스템 이미지, 툴체인, 커널 이미지, 부트로더 이미지를 빌드하는데 사용될 수 있다. Yocto 프로젝트 하위에는 Poky, BitBake, OpenEmbedded-Core (OE-Core)와 같은 프로젝트로 구

성된다[4].

1.1 Poky

Poky는 Yocto 프로젝트의 레퍼런스 시스템이며, BitBake와 OpenEmbedded-Core뿐만 아니라 커스텀 배포판을 빌드 할 수 있는 일련의 메타데이터가 포함되어 있다. Poky의 주요 목표는 임베디드 개발자가 필요한 모든 기능을 제공하는 것이다[4].

1.2 BitBake

BitBake는 파이썬과 쉘 스크립트가 섞여있는 코드를 파싱하는 태스크 스케줄러로, 파싱된 코드는 기본적인 코드 의존성에 따라 태스크를 생성하여 실행한다. 이 과정에서 CPU를 최대로 활용하기 위해 병렬 실행을 지원한다[4][5].

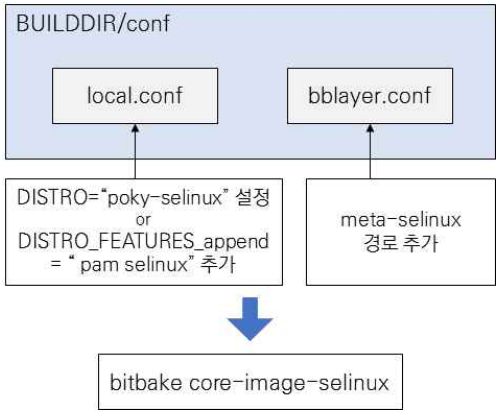
1.3 메타데이터

메타데이터는 BitBake에서 사용되며, 환경설정(.conf), 클래스(.bbclass), 레시피(.bb 및 .bbappend)로 구분된다. 환경설정 파일에는 전역변수가 정의되어 있어 레시피에 영향을 끼친다. 클래스는 전체 시스템에서 사용되며 필요에 따라 레시피들이 상속하고, 레시피는 BitBake가 실행하는 태스크의 논리적 단위이다[5].

2. Yocto 프로젝트를 활용한 SELinux 적용

Yocto 프로젝트는 레이어(layer) 메커니즘을 제공하여 시스템 확장이 쉽고 사용자 맞춤형으로 시스템 이미지를 구성할 수 있다. 레이어의 이름은 meta-로 시작하며, Yocto 프로젝트, OpenEmbedded, 커뮤니티, 제조사에서 수많은 레이어들을 제공하고 있다.

Linux 커널에서 SELinux 기능을 지원하기 위한 레이어로서, meta-selinux가 제공되고 있다. meta-selinux 레이어를 이용하기 위해서는 [그림 1]과 같은 과정이 필요하다. 먼저 Poky 빌드 환경을 구축한 다음, Poky의 빌드 디렉터리에서 환경설정 파일인 local.conf 파일과 bblayers.conf 파일에 SELinux 이미지 빌드에 필요한 내용을 추가한다. local.conf 파일은 모든 빌드 프로세스를 조작할 수 있는 설정 파일로, DISTRO 값을 변경하거나 DISTRO_FEATURE 변수의 내용에 selinux를 추가함으로써 빌드 될 이미지에 SELinux 기능 및 정책을 포함시킨다. bblayers.conf 파일에는 meta-selinux 디렉터리의 경로를 추가하여 레이어에 존재하는 레시피들을 BitBake가 파싱하여 태스크 스케줄링에 사용할 수 있도록 한다. 그런 다음, SELinux 이미지를 빌드하는 명령을 사용한다. 빌드된 이미지를 임베디드 시스템에 포팅하여 SELinux 기능을 사용할 수 있는 환경을 구축한다.



▶▶ 그림 1. Yocto 프로젝트를 활용한 SELinux 이미지 빌드 과정

Ⅳ. 결론

본 논문에서는 Yocto 프로젝트를 활용하여 임베디드 리눅스 시스템에 SELinux를 적용하는 방법을 제시하였다. 임베디드 시스템에서 보안은 중요한 이슈로, SELinux를 임베디드 리눅스에 적용함으로써 시스템의 보안을 강화시킬 수 있다. Yocto 프로젝트는 최근 가장 활발히 진행되고 있는 임베디드 리눅스 빌드 시스템으로, 다양한 하드웨어 아키텍처를 지원하여 확장이 용이하다. 본 논문에서 제시한 방법을 이용하여 빌드된 이미지를 실제 임베디드 보드에 포팅 하는데 활용할 수 있으며, 향후 연구로는 SELinux로 인한 임베디드 리눅스 시스템의 오버헤드 문제를 개선시키는 연구를 진행할 예정이다.

■ 참 고 문 헌 ■

[1] Y. Nakamura, Y. Sameshima and T. Yamauchi, "Reducing Resource Consumption of SELinux for Embedded Systems with Contributions to Open-Source Ecosysmtes", Journal of Information Processing, Vol.23, No.5, pp. 664-672, 2015

[2] L. Fiorin, A. Ferrante, K. Padarnitsas and S. Carrucci, "Hardware-assisted Security Enhanced Linux in Embedded Systems: a Proposal", Proceedings of the 5th Workshop on Embedded Systems Security. ACM, 2010

[3] B. Vogel and B. Steinke, "Using SELinux security enforcement in Linux-based embedded devices", ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[4] https://www.yoctoproject.org/

[5] Otavio, S. and Daiane A., "Development Embedded Linux Using Yocto Project", pp27-acorn+PACKT, 2014.