ETRI

전자통신동향분석 제23권 제1호 2008년 2월

임베디드 운영체제 보안 기술 동향

The Trends of Embedded Operating System Security Technology

임베디드 S/W 기술 동향 특집

정영준 (Y.J. Jung) 임베디드OS연구팀 선임연구원
임동혁 (D.H. Lim) 임베디드OS연구팀 연구원
서영빈 (Y.B. Seo) 임베디드OS연구팀 연구원
김재명 (J.M. Kim) 임베디드OS연구팀 팀장

목 차

임베디드 시스템이라 함은 우리가 생활하는 주변에서 흔히 접할 수 있고, 얘기만 들어도 쉽게 이해할 수 있는 PDA, 스마트폰, MP3 플레이어, 셋톱박스 등의 정보가전 기기들을 비롯하여 로봇, 텔레매틱스, 공장자동화, 군사기기와 센서노드에 이르는 디지털 기기들에 컴퓨터가 내장되어 들어가 있는 시스템을 의미한다. 이러한 임베디드 시스템은 도래하고 있는 디지털 컨버전스 시대에서 모바일 및 정보가전기기로 그 사용성이 날로 확대되어 가고 있으며, 현 시점에서도 각 개인이 이미 최소한 하나씩의 임베디드 기기를 휴대하고 있을 정도로 임베디드 기기에 대한 의존도는 상당히 높다고 할 수 있다. 이렇듯 임베디드 기기에 대한 높은 의존성으로 인해 향후 지금보다 더 많은 개인 정보를 임베디드 기기에 저장하게 될 것으로 예상되는데, 이에 따라 갈수록 사용이 많아질 임베디드 기기 내의 개인 정보의 보안성에 대한 관심이 증폭되고 있으며, 본 고에서는 이와 관련된 최근의 임베디드 보안 기술에 대해 설명하도록 한다.

Ⅰ. 개요

임베디드 시스템이라 함은 특정한 목적을 수행하기 위해 설계된 시스템이며, 임베디드 소프트웨어 기술이라 함은 이러한 임베디드 시스템에 내장되어 있는 소프트웨어 기술을 말한다. 데스크톱에 익숙해 있는 일반 사용자들에게는 약간 낯설 수 있으나, 실제로 임베디드 시스템은 엘리베이터, TV, MP3 플레이어, 셋톱박스, 디지털 카메라, PDA, 휴대폰, 자동차 엔진 제어, 의료기기 등 일상 생활과 매우 밀접한 관계를 갖고 있으며, 그 응용의 범위가 매우 넓다. 그래서 각 응용 임베디드 소프트웨어 시장은 응용 시장 요구가 매우 빠르게 변화하고 있다.

이러한 임베디드 시스템은 최근 응용 시장의 요구가 매우 빠르게 변화하고 그에 따른 여러 가지 서비스와 관련 기반 기술들이 개발되어 현재는 데스크톱 PC에서의 인터넷 뱅킹, 홈쇼핑 등의 응용이 임베디드 시스템에서 가능해지고 있으며, 이에 따른 개인의 중요한 정보를 임베디드 시스템에서 담고 있음에 따라 임베디드 시스템에서 사용하는 임베디드 운영체제의 보안 기술의 필요성은 시간이 갈수록 중요해질 것으로 예상된다.

그런데, 임베디드 운영체제를 위한 보안 기술은 기존 데스크톱 PC나 엔터프라이즈급 서버의 보안 기술과는 달리 제한된 성능 및 가용 자원 때문에 일반 컴퓨팅 시스템에서 사용하는 보안 기술을 그대로 사용하기 어려워 각 임베디드 응용의 분야에 적합한 보안 기술을 개발해야 하며, 임베디드 시스템의 특성상 다양한 규모와 수준의 임베디드 시스템을 위한 보안 기술들을 재구성하여 특정 임베디드 시스템의 목적에 맞추어 구축할 수 있는 재구성 및 설정 기능도 요구되고 있는 실정이다. 그러나, 이렇게 중요성이 커져가고 있는 임베디드 운영체제를 위한 보안 기술은 국내외적으로도 특별히 관심을 가지고 개발된 사례를 찾아보기가 힘든 상태이나, 중요성이 부각되고 있는 만큼 그 기술적 동향을 알아보기 위해 다음과 같은 측면에서 임베디드 운영체제의 보안 기술 동향에 대해 설명을 할 것이다.

- 임베디드 운영체제 보안 커널 기술
- 임베디드 운영체제 보안 서비스 미들웨어 기술
- 임베디드 운영체제 보안 설정 도구 기술

● 용 어 해 설 ●

임베디드 운영체제: PDA, 스마트폰, MP3 플레이어, 셋톱박스 등의 정보가전기기들을 비롯한 로봇, 텔레매틱스, 공장자동화, 군사기기와 센서노드에 이르는 디지털 기기들에 컴퓨터가 내장되어 들어가 있는 것을 임베디드 시스템이라고 한다. 이러한 임베디드 시스템을 위해서는 기존의 데스크톱 및 엔터프라이즈 서버급 컴퓨터와는 다른 자원적 제한에 안정적으로 구동될 수 있는 운영체제가 필요한데 이러한 것을 임베디드 운영체제라 한다.

Ⅱ. 임베디드 운영체제 보안 기술의 시장 동향

전세계 전문 시장 조사 기관의 조사를 찾아보아도 아쉽게도 임베디드 운영체제의 보안 기술이라는 제목으로는 시장 보고서를 찾기가 어렵다. 단지, 아래에서 설명하는 몇 가지 데이터를 통해 임베디드 운영체제 보안 기술과 관련된 시장의 중요성이 날로 커지고 있음을 추측할 수 있다.

첫째로는 Gartner의 2006년 9월 보고서에 따르면 임베디드 소프트웨어 세계시장의 규모가 2006년 1,600억 달러이며 2010년까지 매년 평균 10% 성장이 예상되어 2010년 국내시장 규모는 11조 1천억 원이 예상된다. 이는 임베디드 소프트웨어의 기반 기술인 임베디드 운영체제의 활용도가 그만큼 높아진다고 볼 수 있으며 그에 따라 보안 기술도 필요할 수 있음을 의미한다[1].

둘째로는 IDC의 2006년 자료로 전세계 플랫폼 시장은 1,026억 달러(CAGR 7.8%)로 성장할 것으로 예상되며, 그 중에서도 모바일 PC 플랫폼 시장이 CAGR 14%로 데스크톱 플랫폼 시장의 CAGR 1.5%와 서버 플랫폼 시장의 CAGR 9.3% 보다 월등히 증가할 것으로 예측되어 임베디드 소프트웨어 플랫폼의 중요성은 더욱더 늘어날 것으로 예상하고 있다[2].

셋째로는 전세계 보안 소프트웨어 시장은 2005년 118억 달러, 2010년 205억 달러(CAGR 12%)이고 보안 서비스 시장은 2005년 145억 달러, 2010년 326억 달러(CAGR 18%)이며, 보안장비 시장은 2001년 60억 달러, 2010년에는 136억 달러(CAGR 18%)로 성장할 것으로 예측되어 갈수록 수요가 늘어날 것으로 판단되고 있다[2].

넷째로는 KISA의 국내 정보보호산업 통계조사 보고서에 따르면, 국내의 정보보호업계의 2006년 전체 시장 규모는 7348억 원 규모로, 보안 제품 시장은 6325억 규모로 집계되었으며, 모바일 보안 제품의 경우 36억 원의 규모를 차지하며, 전년 대비 45%의 증가 추세를 보이고 있으며 꾸준한 성장을 거듭하여 2011년에는 90억 원을 돌파할 것으로 예상되고 있다고 한다[3]. 이상에서 보듯이 임베디드 소프트웨어와 그 기반 기술인 임베디드 운영체제의 중요성은 날로 증대되고 있으며 그와 함께 보안 소프트웨어 시장도 성장하여 임베디드 운영체제와 보안 기술이 융합된 임베디드 운영체제 보안 기술 시장의 중요성도 커지고 있다고 볼 수 있다. 그러나, 네번째의 시장 조사 자료 중 비록 모바일 보안 제품에 한정되긴 하지만 아직은 모바일 보안 제품의 시장이 국내에서는 크지 않은 상태이나 조사된 자료에 의하면 성장세는 높을 것으로 예측되고 있다.

여기에서 한 가지 유심히 지켜봐야 할 것은 임베디드 보안 시장 쪽은 보안 기능이 접목된 임베디드 시스템 제품의 형태로 주로 판매되고 있으나, 전세계적으로도 임베디드 운영체제 보안 기술을 솔루션 형태로 판매하거나 기술을 개발하는 곳은 잘 없는 것으로 보인다. 이는 솔루션 형태로 개발되어 다양한 임베디드 시스템에 적용 가능한 임베디드 운영체제의 보안 기술이 필요하다고는 판단되나 업계에서 보기에 지금 당장이 시장의 진입 시기로는 보고 있지 않는 것으로 판단된다. 하지만, 그래도 여전히 임베디드 운영체제의 보안 기술은 빠른 시일 내에 반드시 필요한 기술로 시장에서 각광 받을 수 있을 것이다.

Ⅲ. 임베디드 운영체제 보안 기술의 동향

임베디드 운영체제 보안 기술은 임베디드 시스템에서 사용되는 임베디드 운영체제에 적합하도록 개발됐거나 개발 진행중인 보안 기술들에 대해 알아보도록 할 것이다.

먼저 국외기술 동향을 보면 가장 눈에 띄는 것은 임베디드 운영체제라고 볼 수는 없지만 관련 기술로 SELinux와 SUSE Linux 등 범용 시스템 환경을 위한 보안성이 강화된 리눅스에 대한 연구를 들 수 있으며, 이는 범용성을 위한 리눅스 기반 운영체제에 보안성을 강화한 경우로 볼 수 있다. 이 SELinux는 미국 NSA 주도로 국가정보기반구조 구축과 국방용으로 사용하기 위해 1995년부터 연구를 수행하여 현재의 SELinux를 개발하기에 이르렀다. 다른 한편으로는 전용 RTOS라고 불리는 기존 임베디드 실시간 운영체제들에도 보안 기능을 탑재하는 기술들이 있는데 그것은 Lynux-works사의 LynxOS로 운영체제의 보안을 제공하기 위해 커널과 응용을 분리하여 시스템의 보안 수준을 높였으며, 실제로 미국 항공기 등에 사용되기도 했다. 그 이유 중의 하나는 DO-178B라는 항공기에 사용 가능한 인증을 받았기 때문이기도 할 것이다. 그리고, 텔레매틱스 및 자동차의 제어 소프트웨어로 많이 사용되는 QNX는 시스템 내에 추가적인 암호화 및 복호화 전용 하드웨어를 사용하여 보안 관련 계산 성능을 향상시키기도 했다. 임베디드 운영체제 보안 기술 중 전체적인 솔루션을 개발하기 보다는 각 서브시스템의 보안 기술을 개발한 사례로는 암호화 파일시스템 연구를 했던 Cryptographic File System, Cryptfs, Best-Crypt 등이 있으나 이들이 서버 기반의 시스템을 위해 개발되어 임베디드 시스템에 사용할 경우 암복호화에 상당한 시간이 소요되어 실제로 사용되고 있는 경우는 많지 않다. 또한, 모바일 폰 시장의 점유율 1위를 기록하고 있는 노키아의 Symbian OS에서는 모바일 장치에서의 네트워크 자체의 보안성을 강화하기 위해 Secure Socket 인터페이스를 지원하여

데이터 전송시 암호화를 지원하는 기능을 보유하고 있기도 하다.

이제 국내기술 동향을 보면 산업계에서는 ㈜시큐브의 모듈 추가 방식의 접근제어 기능을 제공하는 보안 운영체제, ㈜시큐브레인의 불법 침입 방지, 응용 보호, 보안 감사 자료 생성, 네트워크 접근 통제를 지원하는 SecureOS와 서버용 보안 도구를 지원하고 있으며 ㈜티에스온넷에서는 다중등급보안, 서버 해킹방지 및 데스크톱 보안 시스템을 개발했으나 아직까지는 서버급 컴퓨터를 위한 보안 기술이며 임베디드 시스템을 위한 보안 운영체제에 대한 고려는 많이 진행되고 있지 않은 상태이다. 또한, 국내 연구소의 기술 동향을 보면 한국전자통신연구원에서 개발한 SecureOS를 들 수 있는데 리눅스와 Free BSD 기반의 커널 수준의 접근제어, 보안 암호화 파일시스템, 사용자 인증/인가, 감사추적 및 등급별 암호화 통신프로토콜 등 많은 기능이 있으나 이 기술들 또한 서버급 컴퓨터를 위한 기술들이었다.

이제 최근 한국전자통신연구원에서 개발중인 임베디드 운영체제 보안 기술에 대한 동향을 살펴보고자 한다. 최근 개발중인 임베디드 운영체제 보안 기술은 크게 3가지로 나누어서 개발중인데 그 중 중요한 부분에 대해 자세한 사항을 알아보자.

우선 개발중인 임베디드 운영체제 보안 기술의 요구사항은 임베디드 시스템의 제약성(저성능 CPU, 저용량 메모리, 저전력 사용, 무선 통신을 사용하는 제한된 하드웨어 자원, 실시간 제약성과 가격 민감성)을 고려하고 기존 보안 운영체제의 주요한 요구사항(접근제어, 보안 암호화 파일시스템, 보안 통신, 경량 방화벽, 네트워크 침입 탐지/방지, 보안 설정 도구)을 만족시키는 것이다. 이를 위한 임베디드 운영체제 보안 기술의 개략도는 (그림 1)과 같다.

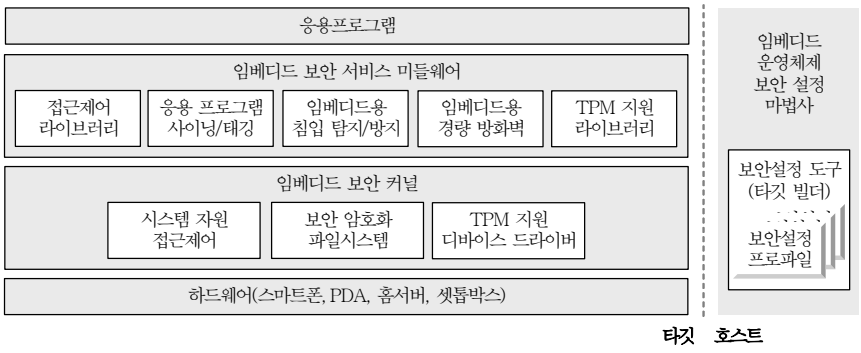
(그림 1)에서 보는 바와 같이 임베디드 운영체제의 특성에 따라 타깃에 적재될 보안 기술과 호스트에서 기 개발된 보안 기술을 적재할 타깃에 맞게 설정을 하는 보안 설정 도구 기술로 나뉘어져 있다.

1. 임베디드 운영체제 보안 커널

임베디드 운영체제 보안 커널의 기능으로는 (그림 1)에서 보는 것처럼 시스템 자원 접근제어, 보안 암호화 파일시스템 및 TPM 지원 디바이스 드라이버 등의 3가지가 있다. 이는 임베디드 보안 운영체제의 상위 계층에 대한 기본적인 서비스를 위한 기능들로 모두 모듈화되어 개발되어 있다. 모듈화되어 개발된 이유는 사용자의 요구에 따라 보안 기능을 사용할 수도 사용하지 않을 수도 있으므로 동적으로 로드 및 언로드 할 수 있는 장점이 있고, 커널 버전이 달라지더라도 모듈 인터페이스가 같은 커널 버전까지는 그대로 사용할 수 있는 장점이 있기 때문이다. 이것은 다양한 임베디드 시스템의 다양한 커널 버전에 쉽게 적응할 수 있어서 임베디드 시스템의 보안 기술을 적용하기에 큰 장점을 준다.

가. 임베디드 운영체제 접근제어 커널 모듈

시스템 자원 접근제어 커널 모듈은 사용자나 태



(그림 1) 임베디드 운영체제 보안 기술 개략도

스크의 시스템 내 자원에 접근시 접근제어를 통해 해당 자원에 접근이 가능한 사용자나 태스크만 접근이 가능하도록 해준다. 이 접근제어 커널 모듈은 RBAC이라는 접근제어 정책을 채택하고 있는데, 이는 사용자나 태스크가 가질 수 있는 역할에 기반하여 미리 정해진 정책에 따라 접근 가능 여부가 정해지도록 되어 있다. 접근제어 정책은 이외에도 MAC과 DAC이 있는데 MAC은 약어에서 보는 것처럼 강제적인 접근제어 정책을 취하는데, 이는 사용자가 임의로 정보의 등급 등을 바꾸지 못하도록 강제하는 것으로 원래 DARPA에서 군사적 목적으로 개발되어 상업용으로 사용하기에는 다소 융통성이 떨어지는 단점이 있다. DAC은 임의적인 접근제어 정책을 취하는 것으로 이는 사용자의 신분(ID)에 기반하여 접근권한(read, write, execute, create, delete)을 부여하는 것으로 UNIX의 권한(permission) 정책과 같다고 볼 수 있다. 모바일 폰과 같이 단독 사용자(single user) 시스템에서는 MAC과 같은 접근제어 정책을 사용하는 것이 더 어울릴 수 있으나, 임베디드 기기에도 홈서버와 같이 다중 사용자(multiple user) 시스템이 있고 RBAC 접근제어 정책으로도 MAC 접근제어 정책을 에뮬레이션 할 수 있어서 다양한 임베디드 시스템의 공통적인 접근제어 커널 모듈로 활용할 수 있다[4]. 이 접근제어 커널 모듈은 상위 계층의 접근제어 라이브러리를 통해 직접적인 접근이 가능하며 역시 상위 계층의 응용 프로그램 사이닝/태깅 모듈과 연동되어 시스템 내의 무결성을 검증하는 기능으로 활용된다.

나. 임베디드 운영체제 보안 암호화 파일시스템

보안 암호화 파일시스템은 시스템 내 보안성을 위해 저장하고자 하는 데이터를 암호화하여 저장하므로 혹시 있을지 모르는 보안 침해사고 시에도 개인의 중요한 정보를 타인에게는 쓸모가 없도록 하는 기술이다. 기존의 Bestcrypt와 같은 암호화 응용 프로그램들은 사용자가 명시적으로 정보를 암호화 함으로써 귀중한 정보를 보호할 수 있도록 해주지만 이러한 응용 프로그램들은 사용하기가 쉽지 않았으며 번거로운 과정을 거치게 된다. 이러한 단점들을 보완하기 위해 제안된 것이 암호화 기능을 파일시스템에서 지원하여 미리 준비된 메커니즘에 따라 중요한 데이터에 대해 암호화하여 저장할 수 있도록 한다. 이렇게 하는 것은 서버급 보안 암호화 파일시스템과 달리 임베디드 보안 암호화 파일시스템에서는 모바일 기기나 저장장치의 분실 시에도 중요 데이터의 유출을 막아줄 수 있으므로 임베디드 시스템에 상당히 유용하다.

개발된 임베디드 보안 암호화 파일시스템은 다양한 임베디드용 파일시스템에 적용이 가능하나 본 동향분석서에서는 셋톱박스나 홈서버의 저장매체인 하드 디스크를 대상으로 하는 파일 시스템인 EXT3를 기반으로 하는 암호화 기술을 설명한다. 또한, 기존의 서버급 보안 암호화 파일시스템에서는 중요 데이터의 암복호화시 상당히 오랜 시간이 걸려서 실제 임베디드 시스템에서 사용할 시에 문제점으로 지적되는 암복호화 시간을 단축하기 위해 버퍼 캐시 수준에서의 암복호화를 수행하여 복호화 오버헤드(overhead)를 최소화하고 임베디드 보안 암호화 파일시스템의 사용 편의성을 위해 커널 모듈 형태로 개발했다[5].

1) 키 관리 기법

본 절에서는 임베디드 보안 암호화 파일시스템을 개발하는 데 중요한 부분인 키 관리 기법에 대해 논한다.

데이터의 보호가 중요한 만큼 암호화에 사용되는 키를 보호하는 일도 역시 중요하다. 일반적으로 암호화 파일 시스템은 대칭 키 방식의 암호화 알고리듬을 사용한다. 이 경우 암호화 키를 그대로 이용하여 복호화를 수행하므로 암호화 키의 보호가 무엇보다 중요하다. 그래서 암호화 파일 시스템에서는 데이터의 보호를 위하여 키의 보안도 중요한 포인트가 되므로, 임베디드 보안 암호화 파일시스템에서는 키의 보호를 위해 마스터 키 & 파일 키 구조를 사용하였다. 이는 일반적인 암호화 파일시스템에서도 사용하고 있는 방식으로 파일 단위의 암호화를 지원하기

위해 파일마다 서로 다른 키를 사용하여 암복호화를 수행한다.

파일 키는 실제로 디스크에 저장될 때는 마스터 키를 이용하여 암호화된 파일 키의 형태로 저장되며, 암호화된 파일 키는 파일에 첫번째 접근이 일어나는 경우에 마스터 키를 이용하여 복호화 된다. 복호화된 파일 키는 커널의 inode 구조에 저장이 되어 kernel level에서만 접근이 가능하게 되어 보안성을 강화한다. 암호화된 파일 키는 확장 속성(extended attribute)의 형태로 파일과 같이 저장이 된다. 실제 EXT3의 inode 구조에는 128~256bit의 파일 키를 저장할 수 있는 충분한 공간이 있지 못하여 inode 구조에 같이 저장할 수 없으므로 확장 속성을 이용 저장하여 확장성을 높게 했다.

마스터 키는 앞에서 언급한 대로 파일 키의 암호화 키로 활용되며, 또한 암호화 파일 시스템 자체에 대한 접근을 제한하기 위한 패스워드의 역할도 같이 수행한다. 암호화 파일에 대해서는 마스터 키를 입력하지 않으면 읽기/쓰기가 금지된다. 디스크에서는 마스터 키의 MD5 해시 값이 수퍼 블록(super-block)에 저장되며 마스터 키의 입력은 사용자 유틸리티를 통해 커널로 전송된다.

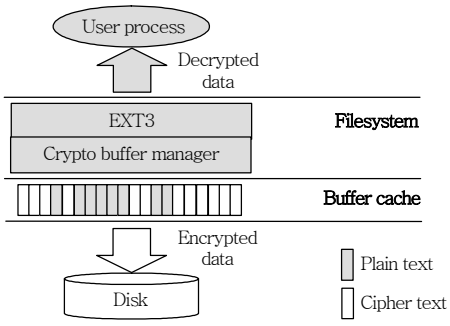
임베디드 보안 암호화 파일시스템에서는 암호화 알고리듬으로 AES를 사용하였고, 암호화 키의 크기를 세 가지로 가변할 수 있도록 하였다. 사용자는 파일에 대해서 보안 등급을 부여할 수 있으며, 이 보안 등급에 따라서 키의 길이가 변화하며, 그에 따라서 오버헤드도 비례하여 증가한다. 이렇게 하는 것은 임베디드 시스템에서 보안성과 실제 오버헤드와의 trade-off 관계에 있으므로, 임베디드 시스템 응용에 따라 사용자가 선택하여 보안성을 높이거나 낮출 수 있도록 했다.

2) 버퍼 캐시수준의 암호화 기법

본 절에서는 임베디드 보안 암호화 파일시스템 중 기존 서버급 보안 암호화 파일시스템과 다른 부분 중의 하나인 버퍼 캐시수준의 암호화를 통한 암복호화시 성능 오버헤드를 최소화하는 기법에 대해 설명한다.

일반적으로 암호화 파일 시스템의 경우, stackable 파일시스템 기법을 사용한다. Zadok이 제안한 이 기법은 기존의 파일시스템을 그대로 이용하고 그 위에 암호화 파일시스템 모듈을 쌓아서 사용하는 방법이다. 기존의 파일시스템 위에 파일을 생성하고 이를 기반으로 암호화 기능을 제공하므로 편리하게 사용할 수 있는 장점이 있지만, 성능에서는 암복호화시 시간이 다소 오래 걸리는 오버헤드를 감수해야 한다. Stackable 파일시스템의 경우 EXT3와 같은 기존 파일시스템의 파일을 그대로 저장구조로 이용하고, 이에 대한 연산(operation) 요청이 있을 때 변환을 하므로 매번 오버헤드가 발생하는 문제점을 그대로 가지고 있다.

이를 효율적으로 수정하고자 하는 것이 버퍼 캐시수준의 암호화이며, 이에 대한 그림은 (그림 2)에서 설명하고 있다. 임베디드 보안 암호화 파일시스템은 버퍼 캐시수준의 암호화를 통하여 캐시된 파일의 연산에 대해 암복호화 오버헤드를 줄이는 방법을 사용했다. Stackable 파일시스템의 경우 버퍼 캐시는 기존의 파일시스템 쪽에서 활용하므로 암호화된 내용이 담겨 있다. 그러므로 stackable 파일시스템의 경우 변환과정이 모든 연산마다 필요하게 된다. 이러한 오버헤드를 피하기 위해 임베디드 보안 암호화 파일시스템은 EXT3를 직접 수정하여 파일시스템에 암호화를 적용하였다. 그리고 암호화 과정에서 decrypt before read, encrypt before write(읽기 전에 암호화한 후 쓰기 전에 암호화 함) 방식을 사용했다.



(그림 2) 버퍼 캐시수준의 암호화 기법 설명

read의 경우는 시스템 콜의 파라미터(parameter)를 통해 사용자가 필요로 하는 버퍼 페이지(buffer page)를 유추할 수 있다. 이 데이터를 이용하여 read 연산의 수행 이전에 decrypt를 수행한다. 페이지에 플래그(flag)를 설정하여 페이지의 암호화 상태를 확인하도록 하여 오버헤드를 줄이고 중복수행을 막아준다.

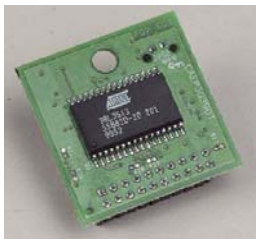
Write의 경우는 파일시스템 자체에서 디스크에 write를 수행하는 경우에 대해서만 고려하면 되므로 비교적 간단하게 처리할 수 있다. 실제 write는 writepage 함수를 호출해서 페이지 단위로 write하게 되는데 이 과정에서 페이지의 상태를 확인하고 페이지를 암호화된 상태로 디스크에 write하면 된다. 파일시스템에서 버퍼 캐시의 암호화를 관리하게 되므로 (그림 2)와 같이 버퍼에는 암호화된 데이터와 복호화된 데이터가 같이 존재하고 동일한 블록에 대한 요청이 여러 번 수행되어도 복호화 과정은 1회만 수행되어 오버헤드는 줄어들게 된다.

다. TPM 지원 디바이스 드라이버

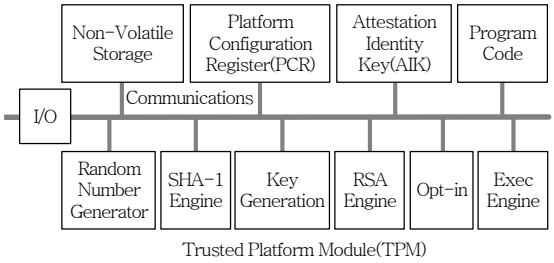
TPM은 기존에 암복호화를 소프트웨어적으로 처리하던 것을 값싼 칩을 이용하여 하드웨어 수준의 신뢰성 높은 보안성을 지원하기 위해 사용한다((그림 3) 참조).

기능으로는 RSA 연산, SHA-1 연산, 난수생성(random number generation), 비휘발성 메모리(NVRAM), 키 생성 및 키 저장 지원기능 등이 있으며 기능이 포함된 TPM 칩의 컴포넌트 구조는 (그림 4)와 같다.

이 TPM 칩을 위한 디바이스 드라이버를 지원하



(그림 3) TPM 칩의 예



(그림 4) TPM 컴포넌트 구조

여 신뢰성 높은 보안 기능을 지원할 수 있도록 하고 있다.

2. 임베디드 운영체제 보안 서비스 미들웨어

임베디드 운영체제 보안 서비스 미들웨어의 기능으로는 (그림 1)에서 보는 것처럼 접근제어 라이브러리, 응용 프로그램 사이닝/태깅, 임베디드용 네트워크 접근제어 및 TPM 지원 라이브러리 등이 있다. 이는 임베디드 보안 운영체제의 하위 계층인 보안 커널 부분과 연동되어 임베디드 시스템 전체가 보안성을 가지고 안정적으로 시스템을 유지하기 위한 기본적인 기능들이다.

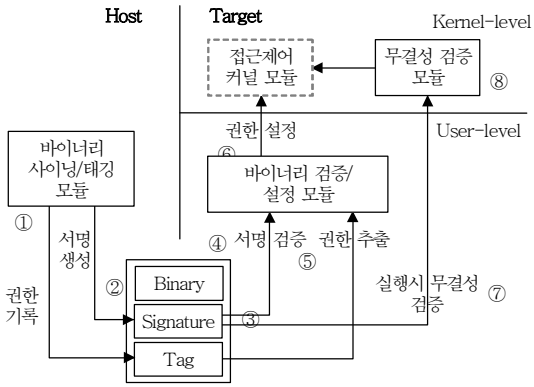
가. 접근제어 라이브러리

1절 가.에서 소개한 접근제어 커널 모듈을 사용하여 시스템의 역할을 등록하고 삭제하는 등의 관리를 할 수 있도록 지원해주는 기능을 제공하고 있으며, 현재는 단순히 접근제어 커널 모듈과의 인터페이스 기능만 제공하고 있다.

나. 응용 프로그램 사이닝/태깅

응용 프로그램 사이닝/태깅은 임베디드 시스템에서 응용 프로그램을 응용 콘텐츠 서버에서 다운로드 받거나 임베디드 시스템에서 구동 시에 위조나 변조에 대응하여 무결성을 검증할 수 있도록 하여 보안성을 극대화할 수 있도록 한다.

응용 프로그램(바이너리) 사이닝/태깅은 (그림 5)와 같이 구성된다. 임베디드 시스템의 특성을 고려하여 먼저 개발 호스트 측의 바이너리 사이닝/태깅



(그림 5) 응용 프로그램 사이닝/태깅 메커니즘

모듈과 실제로 바이너리가 실행되는 타깃 환경의 바이너리 검증/설정 모듈과 무결성 검증 모듈로 구성이 된다. 바이너리 사이닝/태깅 모듈과 바이너리 검증 모듈은 사용자 수준에서 ELF 바이너리를 다루는 프로그램이며, 무결성 검증 모듈은 커널 모듈에 포함되어서 프로그램의 실행 시에 바이너리에 포함된 서명을 검증하여 훼손된 바이너리의 실행을 차단한다[6].

(그림 5)는 응용 프로그램 사이닝/태깅 기법의 구동 메커니즘으로 각각에 대한 설명은 아래에서 하도록 한다.

1) 바이너리 사이닝/태깅 모듈

개발 호스트에서 크로스 컴파일된 응용 바이너리는 바이너리 사이닝/태깅 모듈을 통해서 서명과 태그를 포함한 서명된 바이너리로 변화하게 된다. 먼저 사이닝/태깅 모듈을 통해서 임베디드 보안 운영체제의 접근제어 커널에 제공할 접근제어 정보를 새로운 ELF 섹션(section)으로 바이너리에 기록하게 된다. 이 기록된 정보는 타깃에서 다운로드 받을 때 해석되어 임베디드 보안 운영체제의 접근제어 커널 모듈에 전달된다. 이는 (그림 5)의 ①의 순서에 해당한다.

태깅 작업이 끝나게 되면 서명 작업이 수행된다. 먼저 태깅이 완료된 바이너리의 해시 값을 생성한다. SHA-1 해시 알고리듬을 이용하여 생성된 해시는 이미 개발호스트에서 생성된 개인 키를 이용하여 RSA 전자서명 과정을 수행한다. 이 결과 값은 추가된 ELF 섹션으로 바이너리에 기록된다. 이는 (그림 5)의 ②의 순서에 해당한다.

2) 바이너리 검증/설정 모듈

개발 호스트에서 타깃으로 다운로드된 프로그램은 검증/설정 모듈을 이용하여 설치 과정을 거친다. 설치 과정을 통하여 응용의 역할을 설정하고 다운로드된 프로그램의 안전성을 확인하기 위함이다. 이는 (그림 5)의 ③의 순서에 해당한다.

바이너리 검증/설정 모듈은 전송 받은 프로그램에서 ELF 헤더를 검색하여 추가된 ELF 섹션을 추출해 낸다. 먼저 전자 서명이 포함된 섹션을 제외한 파일 전체에 대한 현재의 SHA-1 해시 값을 구하고, 바이너리에 저장된 전자서명 값을 타깃 시스템에 저장된 공개 키를 이용하여 RSA 전자서명 검증 과정을 거쳐서 앞에서 구한 현재의 SHA-1 해시와 비교한다. 검증과정을 통하여 전송과정에서 있을 수 있는 오류와 바이너리의 변조를 탐지해 낼 수 있게 된다. 이는 (그림 5)의 ④와 ⑤의 순서에 해당한다.

검증과정이 완료되면 바이너리에서 추출해낸 접근제어 태그의 정보는 해석되어 접근제어 커널 모듈에서 제공하는 시스템 콜을 통하여 전달된다. 이는 (그림 5)의 ⑥의 순서에 해당한다.

3) 무결성 검증 모듈

무결성 검증 모듈은 앞에서 이야기한 것과 같이 커널 내부에서 실행된 바이너리의 변조 여부를 검증하는 모듈이다. 바이너리가 exec 시스템 콜을 이용하여 수행되면 커널 내의 LSM hook1[1]이 불리게 된다. 이 때 무결성 검증모듈은 전자서명이 포함된 섹션을 제외한 파일 전체에 대한 현재의 SHA-1 해시 값을 구하고, 바이너리에 저장된 전자서명 값을 커널에 로딩된 공개 키를 이용하여 RSA 전자서명 검증 과정을 과정을 거쳐서 앞에서 구한 현재의 SHA-

1) WireX Communications, Linux Security Module, Apr. 2001, http://lsm.immunix.org/

1 해시와 비교하고, 변조된 응용은 실행을 차단하도록 한다.

이러한 검증 과정은 런타임에 수행하기에 비교적 오버헤드가 큰 작업이다. 그러므로 inode 구조에 서명과 검증 결과를 캐싱(caching)하는 방법을 사용하여 그 오버헤드를 줄일 수 있다. 시스템이 부팅된 후에 최초의 바이너리 실행에 대해서는 검증 과정을 수행한다. 그리고 바이너리의 현재의 SHA-1 해시 값과 그 결과는 inode에 캐시된다. 그 다음의 실행 시에는 inode의 변경이 없는 경우 캐시 값을 읽어서 실행/차단 여부를 결정하게 된다. 그리고 inode나 파일의 내용에 변화가 생기는 연산의 수행 시에는 파일의 변경 여부를 기록하고 캐시를 invalid로 변경하여 재실행 시에 서명 검증 과정을 수행하도록 하여 오버헤드를 줄일 수 있다. 이는 (그림 5)의 ⑦과 ⑧의 순서에 해당한다.

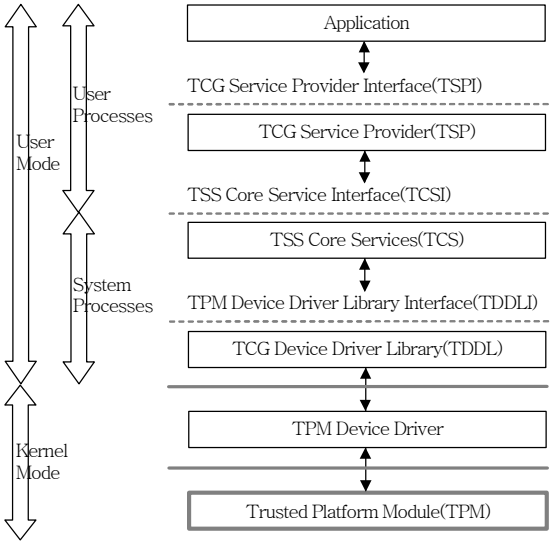
다. 임베디드용 네트워크 접근제어

임베디드용 네트워크 접근제어는 (그림 1)의 임베디드용 경량 방화벽과 임베디드용 침임 탐지/방지 기능을 병합하여 지원하는 기술이다.

기존의 데스크톱 및 서버급 컴퓨터를 위한 방화벽은 다양하고 복잡한 기능을 수행하기 때문에 많은 시스템 자원을 필요로 하여 임베디드 시스템에 사용하기에는 적합하지 않은 면이 있다. 따라서, 임베디드 보안 운영체제의 네트워크 접근제어 기술의 임베디드용 경량 방화벽은 제한된 자원을 사용하는 임베디드 시스템에 적합하도록 최소한의 기능으로 방화벽 서비스를 제공하고 있다. 또한, 임베디드용 침입 탐지/방지 기능은 보안성을 높이기 위해 네트워크를 통한 침입 패턴을 분석한 많은 규칙 정보를 저장하고 이를 이용하여 네트워크를 통한 불법 침입을 탐지하고 방지해내는 구조로 되어 있다. 그러나, 기존의 이러한 구조는 많은 규칙 정보를 저장함으로 인해 자원이 부족한 임베디드 시스템에서는 사용 자체가 어려웠으나, 본 임베디드 보안 운영체제의 침입 탐지/방지 기능은 최적화된 규칙 정보의 조합을 통해 임베디드 시스템에서 사용 가능하도록 하고 있다.

라. TPM 지원 라이브러리

1절 다.의 TPM 지원 디바이스 드라이버에서 설명했듯이 하드웨어 수준의 신뢰성 높은 보안성을 지원하기 위해 TPM 지원 디바이스 드라이버를 활용하기 위한 TPM 라이브러리를 지원한다. 이때 사용하는 TPM 라이브러리는 임베디드 보안 운영체제가 임베디드 리눅스 기반이므로 리눅스 오픈 커뮤니티의 TPM 라이브러리인 TrouSerS를 사용했다. TPM 소프트웨어 스택은 (그림 6)에서 보는 바와 같이 TPM 칩을 제어하기 위한 TPM 디바이스 드라이버와 이를 사용자가 쉽게 사용하도록 하기 위한 TDDL, TPM의 핵심적인 기능을 서비스하기 위한 TCS이며 응용에 대한 서비스를 제공하기 위한 TSP 계층으로 이루어져 있다. 이 TPM 소프트웨어 스펙은 정보보호 분야의 대표적인 표준화 단체 중의 하나인 TCG에서 제정하고 있는 TPM 스펙 1.2의 구조이다[7].



(그림 6) TPM 소프트웨어 스택 구조

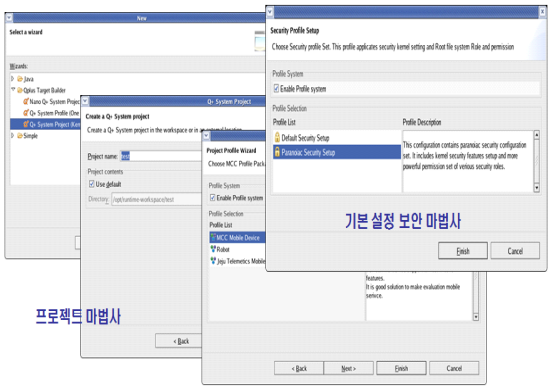
3. 임베디드 운영체제 보안 설정 도구

임베디드 시스템 타깃 설정도구 기술은 임베디드 시스템의 빠르고 쉬운 설정을 위하여 임베디드 운영체제의 개발사들을 비롯한 임베디드 리눅스를 이용

한 오픈 소스 프로젝트들은 설정 도구를 개발하여 제공하고 있다. 본 임베디드 보안 운영체제에서도 다양한 임베디드 시스템을 위한 여러 보안 기술을 각각의 임베디드 응용에 쉽고 빠르게 적용하기 위해 보안 설정 도구를 지원하고 있다. 보안 설정 도구는 아래의 두 가지 기능으로 구성된다.

가. 보안 설정 마법사 지원

임베디드 응용 개발자들의 편의를 위한 기본 보안 설정 마법사를 제공하고 있다. 이는 사용자에게 친화적인 UI를 제공하고 기본적인 보안 설정 단계를 쉽게 제공하고 있어 사용자의 편의를 증진하도록 하고 있다. (그림 7)은 보안 설정 마법사의 스크린 샷이다.



(그림 7) 임베디드용 보안 설정 마법사

나. 보안 프로파일 지원

다양한 임베디드 기기에 적합한 보안 기능을 지원할 수 있도록 보안 설정 프로파일을 지원한다. 현재 지원하고 있는 것은 대표적인 임베디드 응용으로 생각할 수 있는 모바일 단말기기용 보안 프로파일과 정보가전 단말기기용 보안 프로파일의 두 가지를 지원하고 있다.

Ⅳ. 맺음말

기존의 데스크톱 PC와 엔터프라이즈 서버급 보안 기술들은 많이 개발되어 왔으나 아직 임베디드용 보안 기술은 전세계적으로도 초기 단계에 있는 실정이다. 그러나, 앞서 언급했듯이 향후로는 더 많은 임베디드 응용들이 생겨날 수 있으며 누구나 임베디드 기기들을 더 많이 사용하리라는 것은 자명한 사실일 것이다. 이에 따라, 보다 귀중한 정보들을 임베디드 기기들에 저장할 수 밖에 없을 것이며 더불어 임베디드 시스템을 위한 보안 기술의 중요성은 아무리 강조해도 지나치지 않을 것이다. 따라서, 임베디드 시스템에 사용되는 임베디드 운영체제를 위해 임베디드 시스템의 특성이 고려된 보안 기술들의 개발이 시급하다. 그 이유는 국내에서나 국외에서도 그 필요성은 인식하나 아직 실제 개발되어 적용되지 않아서 시급히 개발되어 국내외 시장에 진출한다면 국내 산업계의 임베디드 보안 시장 진출에 큰 도움을 줄 수 있을 것이다. 또, 국제적인 표준화도 아직은 추진할 일이 많이 있으므로 이와 같은 기술 개발 및 표준화에 보다 심혈을 기울여 주요한 기술에 대한 기술 선점을 할 필요성이 있다고 사료되며 국내에서도 많은 기술 개발과 관련 활동이 있기를 기대하는 바이다.

● 용 어 해 설 ●

임베디드 보안 기술: 기존의 데스크톱이나 엔터프라이즈 서버급 컴퓨터에서 지속적으로 보안 기술이 연구 개발되어 왔다. 그러나, 자원적인 제한을 가지고 구동되는 임베디드 시스템을 위한 특화된 보안 기술이 필요하며, 최근 임베디드 기기의 활용성과 그 수요가 폭발적으로 증가함에 따라 개인의 중요한 정보를 많이 담고 있는 임베디드 기기들에 대한 보안 기술의 중요성이 커져가고 있으며 그 수요도 늘어나게 될 것이다.

용 어 정 리

AES	Advanced Encryption Standard
DAC	Discretionary Access Control
ELF	Executable and Linkable Format
EXT3	The Third Extended FileSystem
LSM	Linux Security Module

MAC	Mandatory Access Control
NSA	National Security Agency
RBAC	Role Based Access Control
SELinux	Security Enhanced Linux
TCG	Trusted Computing Group
TPM	Trusted Platform Module

참 고 문 헌

[1] Data Quest, Forecast: Security Software, Worldwide, 2005-2009, Gartner, 2006. 9.

[2] Market Analysis, Korea IP VPN Services and Security Appliance 2006-2010 Forecast and Analysis, IDC, 2006.

[3] 2006년 국내 정보보호산업 통계조사 보고서, KISA, 2006.

[4] 은성경, "Issues in Embedded Operating System Security," EMSOFT, 2006. 10.

[5] 이재흥, 허준영, 박재민, 조유근, 홍지만, "Buffer Cache Level Encryption for Embedded Secure Operating System," EUC, 2007.

[6] 임동혁, 임용관, 정영준, 김재명, "임베디드 보안 운영체제를 위한 바이너리 보안 프레임워크의 설계," NCS, 2006. 12.

[7] TPM Specification version 1.2, Trusted Computing Group, 2006. 3.