IoT 디바이스 보안 플랫폼 기반의 보안 관제 시스템의 설계 및 구현

정준영○ 김병선 조진성

경희대학교 컴퓨터공학과

jjy920517@khu.ac.kr, ykbs0903@khu.ac.kr, chojs@khu.ac.kr

Design and Implementation of Security Control Center based on IoT Device Security Platform

Junyoung Jung○ Byoungseon Kim Jinsung Cho

Department of Computer Science and Engineering

KyungHee University

요 약

최근 디바이스 보안 문제와 관련된 각종 IoT 보안 사고가 점차적으로 증가하고 있다. 이러한 보안 이슈를 해결하기 위해서는 IoT 환경 내 디바이스 보안 측면을 고려한 IoT 디바이스가 필수적으로 사용되어야 하고, 각종 사이버 위협에 적절히 대응할 수 있는 보안 관제 서비스가 운영되어야 한다. 본 논문에서는 선행 연구인 IoT 디바이스 보안 플랫폼의 각종 보안 기능을 기반으로 보안 관제 수행에 필요한 기능을 정의하였고, 정의된 기능을 바탕으로 보안 관제 시스템을 설계 및 구현하였다.

1. 서 론

최근 IoT(Internet of Things) 디바이스가 증가함에 따라 IoT 디바이스 보안 문제가 큰 이슈가 되고 있다[1]. 각종 IoT 디바이스가 상호 연결되는 IoT 환경의 보안 위협은 사용자의 신체적, 물질적 피해를 발생시킬 수 있기 때문에 IoT 디바이스 보안 이슈는 반드시 해결되어야 할 문제이다.

하지만 대다수 기업들은 실제로 IoT 디바이스에 어떤 보안 취약점이 존재하고 어떤 보안 기술이 필요한지 인식하지 못하여 디바이스가 요구하는 보안 수준을 고려하지 않고 제작되는 경우가 대부분이다. 또한, 이러한 디바이스와 연동된 보안 관제 시스템은 디바이스의 각종 보안 기능의 부재로 인한 각종 사이버 공격 정보에 대한 수집, 분석, 대응을 수행하는데 많은 어려움이 존재한다. 따라서, 보안 관제 서비스가 사용자/관리자의 신뢰를 절대적으로 요구하는 분야인 만큼 디바이스 보안 측면을 고려한 IoT 디바이스를 통한 보안 관제 요구사항 분석 및 이에 기반한 시스템 설계가 선행되어야 한다[2].

한편 본 연구진은 기존 연구로써 고성능 IoT 디바이스를 위한 디바이스 보안 플랫폼을 개발하였다. 본 디바이스 보안 플랫폼은 TPM을 기반으로 시스템 내 다양한 측면에서의 보안 기능을 제공한다.

본 논문에서는 앞서 언급한 IoT 디바이스 보안 플랫폼을 기반으로 보안 관제 시스템 설계를 위한 기능적 요구사항을 분석한다. 분석 결과를 바탕으로 보안 관제 기능을 정의하고, 보안 관제 시스템을 설계 및 구현한다.

논문의 목차는 다음과 같다. 2장에서는 IoT 디바이스 보안 관제 시스템 설계를 위한 요구사항 분석 및 관제 기능을 정의한다. 정의된 기능을 바탕으로 3장에서는 SCC 시스템 아키텍처를 설계 및 구현하고, 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

기존 연구로써 개발된 IoT 디바이스 보안 플랫폼(Secure Pi)은 oneM2M 보안 요구사항을 만족함과 동시에 하드웨어 보안 칩인 TPM을 기반으로 아래와 같은 시스템 보안 기술을 제공한다[3][4].

- Secure Key Storage & Management: 암호화 키를 안전하게 보관 및 관리하기 위한 기술
- Secure Boot: 펌웨어 무결성을 보장하기 위한 기술
- Secure F/W Update: 안전한 펌웨어 업데이트 지원 기술
- Remote Attestation: 디바이스의 신뢰 상태 증명 기술
- Secure Communication: 디바이스 간 안전한 통신
- Mandatory Access Control: 보안 레벨 기반의 접근 제어
- Filesystem Integrity: 파일 시스템 무결성 보장 기술
- Filesystem Encryption: 파일 시스템 기밀성 보장 기술

3. 제안하는 시스템

앞서 언급한 Secure Pi를 기반으로 본 장에서는 IoT 보안

관제 시스템 설계를 위한 요구사항 분석 및 기능을 정의한다. 또한, 정의된 기능을 바탕으로 설계된 보안 관제 시스템(SCC; Security Control Center)에 대해 소개하도록 한다.

3.1 시스템 요구사항 분석 및 기능 정의

Secure Pi의 기본적 보안 기능을 기반으로 보안 관제 수행에 필요한 기능적 요구사항은 아래와 같다.

・민감 데이터 가용성 보장: Secure Pi는 TSS(TCG Software Stack)를 통해 TPM에 각종 암호화 키 데이터를 저장, 관리한다. 이러한 키 데이터가 사용될 수 없을 경우, 다른 보안 요소기술들의 정상적인 동작을 보장 할 수 없으며, 이는 TSS Daemon을 통해 알 수 있다. 따라서, Secure Pi의 TSS Daemon을 주기적으로 모니터링 하여 Secure Key Storage & Management 확인을 위한 기능이 필요하다.

・펌웨어 무결성 보장(Secure Boot): Secure Pi는 TPM을 사용하여, 각 부트 단계마다 미리 생성한 서명과 각 부트 과정에서 생성한 서명의 일치 여부를 판단하여 펌웨어 교체 공격을 막을 수 있다. 따라서, 보안 관제 시스템은 Secure Boot 확인을 위한 기능이 필요하다.

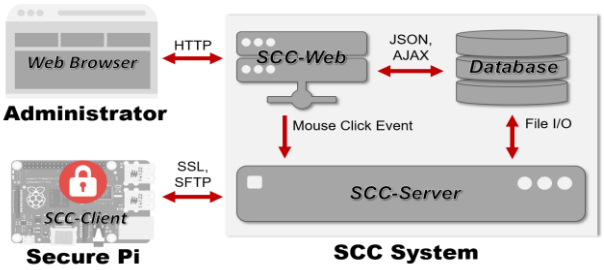
・안전한 펌웨어 업데이트 보장: Secure Pi는 TPM을 사용하여 이전 버전의 펌웨어 설치를 방지하는 Secure Firmware Update를 제공한다. 그러므로 IoT 보안 관제 시스템은 Secure Firmware Update 확인을 위한 기능이 필요하며, Update 서버의 역할을 해야한다.

・펌웨어 무결성 보장(Remote Attestation): IoT 디바이스는 펌웨어 교체 공격을 방지할 수 있어야 한다. Remote Attestation은 다른 기기와 상호 인증하여 펌웨어 무결성을 보장하는 기술이다. IoT 보안 관제 시스템은 Remote Attestation 확인을 위한 기능이 필요하며, Attestation 서버의 역할을 해야한다.

・파일 시스템 내 파일의 무결성 보장: 파일 시스템 내 파일의 무결성을 보장하기 위해 IMA(Integrity Measurement Architecture)와 EVM(Extended Verification Module)을 사용한다. IMA 해쉬 결과는 공격자도 생성할 수 있는 문제점이 있기 때문에 EVM을 함께 활용하여 IMA의 무결성을 강화하는 것이다. 하지만 암호화 키가 공격자에게 노출되었을 경우, 공격자는 EVM을 생성할 수 있다. 따라서 Secure Pi는 TPM 내 암호화 키를 사용하여 IMA/EVM의 문제점을 해결했다. 그러므로 IoT 보안 관제 시스템은 Filesystem Integrity 확인을 위한 기능이 필요하다.

・파일 시스템 내 파일의 기밀성 보장: 파일 시스템 내 파일 기밀성을 제공하기 위해 eCryptFS라는 소프트웨어를 사용한다. 그러나 공격자에게 eCryptFS가 사용하는 FEKEK (File Encryption Key Encryption Key)가 노출되면 암호화 파일을 복호화 할 수 있다. 따라서 Secure Pi는 TPM을 통해 FEKEK 유출을 방지하며, IoT 보안 관제 시스템은 Filesystem Encryption 확인을 위한 기능이 필요하다.

・디바이스 로그인 시도 감지: 리눅스 기반의 COTS IoT 디바이스인 Secure Pi는 로그인 기록을 /var/log/auth.log 파일에 남긴다. 따라서 IoT 보안 관제 시스템은 주기적으로 로그인 기록 확인을 위한 기능이 필요하다.



[그림 1] SCC System의 구조도

・디바이스 허용/거절 패킷 감지: 리눅스 기반의 COTS IoT 디바이스인 Secure Pi는 방화벽으로 Iptables를 사용할 수 있다. Iptables를 사용하면 플랫폼에 들어오는 패킷에 대한 정책을 설정할 수 있고, 허용/거절한 패킷에 대한 로그를 기록할 수 있다. 따라서, IoT 보안 관제 시스템은 주기적으로 허용/거절되는 패킷의 확인을 위한 기능이 필요하다.

3.2 시스템 설계

3.1장에서 정의한 요구사항에 대응하기 위해, Secure Pi의 상태를 모니터링 할 수 있는 보안 관제 시스템 SCC를 개발하였다. SCC는 [그림 1]과 같이 SCC-Client, SCC-Server, SCC-Web, Database로 구성되어 있으며 SCC 관리자는 웹 브라우저를 통해 Secure Pi를 모니터링 한다. SCC-Server, SCC-Web, Database가 있는 SCC System은 Ubuntu PC(16.04 LTS)에서 동작하며, SCC-Client는 Secure Pi에서 동작한다.

SCC-Client는 수집한 Secure Pi의 데이터를 SCC-Server에 안전하게 전달하기 위해 SSL(Secure Sockets Layer)을 사용하였으며, SCC-Server는 SFTP(SSH File Transfer Protocol)를 이용하여 Secure Pi의 펌웨어 업데이트를 진행한다. SCC-Server가 파일 입출력을 통해 Database에 저장한 데이터는 SCC-Web이 JSON(JavaScript Object Notation)과 AJAX(Asynchronous JavaScript and XML)을 사용하여 웹 페이지의 형태로 나타낸다.

[그림 2]는 Secure Pi가 사용하는 보안 요소기술을 모니터링하는 SCC-Client의 기능 구성도다.

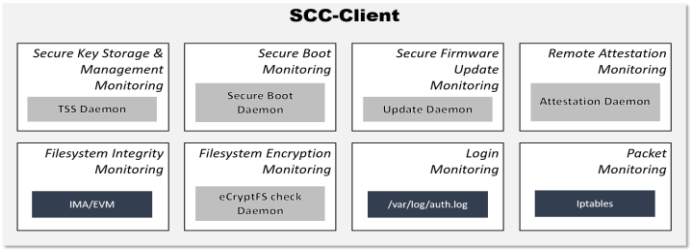
・**Secure Key Storage & Management Monitoring**: TSS Daemon의 정상 작동 유무를 확인하여 전달한다.

・**Secure Boot Monitoring**: Secure Boot Daemon의 정상 작동 유무를 확인하여 전달한다.

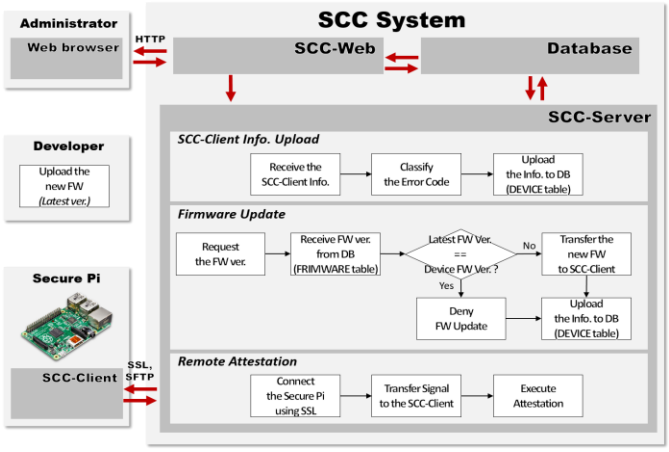
・**Secure Firmware Update Monitoring**: SCC-Server로부터 새로운 펌웨어를 받으면 Secure Firmware Update의 정상작동 유무를 확인하여 전달한다.

・**Remote Attestation Monitoring**: Remote Attestation의 정상 작동 유무를 확인하여 전달한다.

・**Filesystem Integrity Monitoring**: IMA/EVM의 정상 작동



[그림 2] SCC-Client의 기능 구조도



[그림 3] SCC-Server의 순서도

유무를 확인하여 전달한다.

- **Filesystem Encryption**: eCryptFS의 정상 작동 유무를 확인하여 전달한다.
- **Login Monitoring**: 외부 로그인 시도를 확인하여 전달한다.
- **Packet Monitoring**: 허용/거절되는 패킷을 확인하여 전달한다.

[그림 3]은 SCC-Server의 동작 과정을 그린 순서도다. SCC-Server는 다음과 같은 세가지 기능을 수행한다. 첫째, SCC-Client가 8가지 보안 기술의 결과 값을 전달하면 이를 지정된 에러 코드에 맞게 분류하여 Database DEVICE 테이블에 저장한다. 둘째, SCC-Web에서 펌웨어 업데이트를 하기 위한 마우스 클릭 이벤트가 발생하면, Database FIRMWARE 테이블을 확인하고 현재 Secure Pi의 펌웨어 버전과 최신 펌웨어 버전을 비교한다. 만약 두 버전이 일치하지 않으면 SCC-Client에 신호를 보내어 Secure Firmware Update를 실행시키고, FIRMWARE 테이블에 업데이트 동작을 기록한다. 두 버전이 일치할 때에도 FIRMWARE 테이블의 업데이트 동작 요청이 들어왔다는 것을 기록한다. 셋째, Secure Pi가 부팅되어 SSL 연결이 되면, SCC-Client에 신호를 보내어 Attestation을 실행시킨다.

마지막으로 SCC는 웹 브라우저를 통해 Secure Pi를 모니터링 하기 위해 Node.js 기반으로 동작하는 SCC-Web과 MySQL을 DBMS로 사용하여 Secure Pi의 정보를 저장하는 Database가 존재한다.

4. 구 현

[그림 4] SCC의 메인화면

[그림 4]는 3장의 설계를 기반으로 구현한 SCC의 메인화면이다. SCC 관리자는 메인화면에서 이상 디바이스의 모니터링이 가능하다. 디바이스에 대한 자세한 정보를 얻기 위해서는 디바이스 상세 정보 테이블을 참고하면 되고, 각 디바이스의 로그 보기 버튼을 눌러 디바이스의 상태 확인이 가능하다.

3. 결론 및 향후 계획

IoT 디바이스의 보안을 위해 디바이스 수준을 고려한 보안 플랫폼과, 이를 모니터링 하는 보안 관제 시스템이 필요하다. 본 논문에서는 IoT 디바이스 보안 관제 시스템의 핵심 요소기술을 정의하였고, 이전 연구를 통해 개발된 IoT 디바이스 보안 플랫폼을 기반으로, 보안 관제 시스템을 제한하였다. 이를 통해 보안 관제 시스템의 관리자는 IoT 디바이스들을 효율적으로 침해 대응 및 관제 할 수 있으며, 실시간 감시를 통해 다양한 불법적 주체에 대한 접근을 차단할 수 있다. 즉, oneM2M에서 제안하는 IoT 디바이스의 보안 진단 및 컨설팅이 가능하다. 향후 연구로 RTOS/펌웨어 기반의 저사양 IoT 디바이스 플랫폼의 설계를 완료하여 구현 중에 있으며, 이 또한 모니터링 할 수 있도록 고려하여 보안 관제 시스템의 품질을 향상시킬 계획이다.

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터(IITP)에서 지원하는 서울어코드활성화지원사업(2011-0-00883)과 SW중심대학지원사업(2017-0-00093)의 지원으로 수행되었음.

참고 문헌

[1] 정보통신기술진흥센터, “IoT 보안 이슈와 시사점”, 2015 ICT Spot Issue, 2015

[2] J. Pescatore, G. Shpantzer, “Securing the Internet of Things Survey”, SANS Institute, 2014

[3] 허신욱, 김호원, “사물인터넷 보안 요구사항과 oneM2M 표준 보안 기술 분석”, 정보과학회지, 35(1), p16-22, 2017

[4] 김병선, 조진성, “Secure Pi: COTS IoT 디바이스 보안 플랫폼”, 한국정보과학회 2015년 동계학술대회 논문집, p437-439, 2015