편집위원 : 이임영 (순천향대학교)

SIOT: Secure Gateway for Internet of Things

김연근, 고영환, 김민수, 장진수, 배상욱, 노주환, 강병훈, 박경수, 김용대, 신승원
한국과학기술원

요 약

오늘날 사물 인터넷(Internet of Things)은 급속도로 발전하며 인간 생활 곳곳에 다양한 형태로 존재하고 있다. 이들은 대부분 개인 정보 등의 민감한 데이터들을 다루기 때문에 사물 인터넷 환경에 대한 강력한 보안을 필요로 하지만, 현재 사물 인터넷 환경은 비정상적 접근을 허용하거나 업데이트를 통한 펌웨어 변조 등의 많은 보안 취약점들을 가지고 있다. 본 논문에서는 현재 사물 인터넷 환경이 가지는 보안 문제점들을 살펴보고, 이들을 해결하기 위해 딥 러닝 기반의 이상 탐지, 로그 위/변조 탐지, 기기 무결성 검증 등의 다양한 보안 기법들이 집약된 보안 게이트웨이인 SIOT를 제안한다. SIOT는 저성능의 사물 인터넷 기기들이 충분한 보안 기능들을 탑재할 수 없음에 주목하여 다수의 보안 기능들을 효율적으로 통합하여 제공하는 새로운 사물 인터넷 보안 프레임워크로써, 지속되는 사물 인터넷 보안 연구에 큰 기여를 할 것으로 기대한다.

Ⅰ. 서 론

최근 많은 화두가 되고 있는 사물 인터넷 (Internet of Things; IoT)은 기존의 독립적이고 독자적인 센서(Things라고 불리우는 디바이스들) 중심 기술이 아닌 다양한 센서 및 기기들이 네트워크에 연결되어 정보를 공유하고 서로 연계하면서 새로운 서비스 등을 구현하거나 운영할 수 있는 방식을 말한다. 최근 IoT에 대한 관심이 급증하면서, 헬스케어나 스마트홈에 활용 가능한 IoT기기 혹은 서비스들이 꾸준히 소개되고 있으며, 동시에 실제 시장에서도 새로운 비즈니스 기회로 각광받고 있다.

이런 IoT 환경이 새로운 기회를 제공하고 있지만, 반대로 이로 인해 발생하는 새로운 보안에 대한 문제 역시 대두되고 있다. 일반적으로 IoT 기기들은 개인 정보와 사생활 정보와 같은 민감한 데이터들을 주로 다루고 있기 때문에 매우 강력한 보안을 필요로 하지만, 현재 실제로 출시되고 있는 IoT 기기들과 해당 기기들이 설비된 IoT 환경은 이러한 보안 문제에 대한 고려가 매우 부족한 실정이다. 따라서, 앞으로 더 많은 IoT 기기들이 설치되고 IoT 시장이 더욱 커질 것으로 전망되는 가운데 [1][2], 안전한 IoT 환경을 구축하기 위한 기술에 대한 연구의 중요성이 더욱 커지고 있다.

본 논문에서는 IoT 환경에 존재하는 보안 취약점들을 바탕으로, 각 취약점을 해결하기 위한 방어 기법들이 집약된 보안 게이트웨이인 SIOT를 제안한다. 현재 제공되는 대부분의 IoT 서비스 들이 각각의 센서들을 관리하는 게이트웨이를 기반으로 하는 경우가 많다. 이 경우, 게이트웨이에서 모든 센서들의 정보를 확인하고 센서들 간의 통신을 모니터링하는 것이 가능하다. 따라서, 본 논문에서는 이러한 게이트웨이 구조를 확장하여 IoT 환경을 위한 새로운 보안 서비스를 제공하고자 한다. 먼저 2장에서 현재 IoT 환경이 가지는 보안 문제점들을 소개하고, 3장에서는 SIOT의 구조 및 집약된 방어 기법들에 대한 자세한 내용을 기술한다. 이후 4장에서 각 방어 기법에 대한 관련 연구들을 소개한다.

Ⅱ. IoT 환경의 보안 문제점

본 논문에서는 유수의 정보 통신 기술 및 컴퓨터 소프트웨어 업체 네 곳에서 IoT 보안에 관하여 공개한 보고서들을 기반으로 IoT 환경에서 생길 수 있는 다양한 보안 취약점을 종합하였고, 이들을 총 6가지 종류로 구분하고자 한다 [3]–[6].

1. 통신 취약점

Symantec 사의 보고서에 따르면, 해당 업체에서 분석한 IoT 장비 중 약 20%는 암호화 되지 않은 통신을 사용하고 있을 만큼 암호화 되지 않은 통신을 사용하지 않는 IoT 장비가 다수 존재한다. 또한, IoT 장비가 암호화 통신을 사용하더라도 취약한 암호 기법, 잘못된 암호화 구현 등 보안 문제가 있는 경우 통신의 기

밀성과 무결성을 보장할 수 없다. 통신의 기밀성에 문제가 있는 경우 통신 채널을 통해 전달되는 개인 정보, 로그인 자격 증명 등을 공격자가 도청을 통해 탈취할 수 있고, 무결성에 문제가 있는 경우 허가되지 않은 사용자가 IoT 장비를 조작하는 명령을 통신 채널을 통해 전달 할 수 있는 문제가 발생 할 수 있다.

2. 권한 관련 취약점

IoT 환경은 기기 및 사용자들을 관리자로부터 분리하여 기능을 수행하는 데 필요한 리소스에 최소한의 접근 권한만 부여하여 관리해야 한다. 하지만 불필요한 리소스에 접근 권한 부여, 권한 관리 시스템의 취약점, 운영체제 상의 취약점을 통해 최상위 권한을 획득할 수 있는 소위 "루팅" 등의 문제로 인하여 공격자가 관리자 권한을 획득하는 문제가 발생할 수 있다. 공격자가 관리자 권한을 가지고 있다면 IoT 장비의 모든 자원에 대해 접근이 가능하기 때문에 악의적 구성 설정 변경, 암호 변경, 임의 SSL 인증서 설치 등 기밀성, 무결성, 가용성에 심각한 문제를 일으킬 수 있다.

3. 소프트웨어 취약점

IoT를 구성하는 장비들의 운영체제 및 어플리케이션 등 소프트웨어에서 코드 인젝션, 버퍼 오버플로우 등 다양한 자체 취약점들이 필연적으로 생길 수 밖에 없다. 이러한 취약점들은 패치가 되기 전까지 공격자들이 먼저 발견하여 IoT를 공격하기 위한 수단으로 사용될 수 있다.

4. 업데이트 및 패치 취약점

IoT 기기의 소프트웨어에 존재하는 제로데이 취약점을 제거하거나 기능 개선을 하기 위해서는 소프트웨어 업데이트 및 패치는 IoT 환경에서 반드시 필요한 기능이다. 그러나 업데이트 및 패치 파일의 전자 서명 검증을 하지 않는 등 해당 기능을 수행하는 시스템의 보안이 취약한 경우 IoT 장치에 대해 공격자의 악성 코드가 포함된 소프트웨어를 다수의 IoT 기기에 한꺼번에 업데이트 및 패치하는 방식으로 공격이 가능하다.

5. 센서 취약점

일반적인 IoT 토폴로지의 말단에는 다수의 센서를 장착한 IoT 장비들이 배치되어 다수의 센서를 이용한 데이터 수집을 통해 주변 환경에서 발생하는 사건이나 변화를 감지하고, 그 상황에 맞게 사용자는 서비스를 제공받는다. 이는 IoT 환경의 가장 큰 특징이라고 할 수 있지만, 반면에 공격자가 정규 채널 및 부 채널을 통해 센싱 결과가 조작하여 센서의 무결성에 문제를 일으키고 실제 주변 상황을 파악하지 못하게 만들어 잘못된 서비스를 제공하는 문제를 발생시킬 수 있다.

6. 사용자 인터페이스 관련 취약점

IoT 장비들은 개발, 관리 등의 목적으로 소프트웨어 및 하드웨어 인터페이스를 포함한다. 소프트웨어 인터페이스는 주로 웹 페이지 형태로 제공되어 있어 크로스 사이트 스크립팅, SQL 인젝션 등의 웹 취약점 등이 존재할 수 있어 이를 통해 공격자는 허가되지 않은 자원에 접근이 가능할 수 있다. 또한 UART, JTAG 등 IoT 장비 하드웨어에 남아있는 인터페이스를 통해 장치 메모리를 읽거나 펌웨어를 추출하여 공격을 수행하는 데 필요한 취약점을 찾는데 사용하거나 직접 펌웨어에 악성코드를 덮어 쓰는 방식으로 공격을 수행할 수 있다.

Ⅲ. SIOT 디자인

IoT 환경에 대한 연구 및 조사를 통해 구분한 6가지 취약점들을 기반으로, 본 연구에서는 안전한 IoT 환경을 구축하기 위해 다양한 보안 기법들이 집약된 IoT 보안 게이트웨이인 SIOT를 제안한다.

1. 디자인 원칙 및 구조도

현재 IoT 환경에 존재하는 보안 취약점들이 다양한 만큼, 안전한 IoT 환경을 구축하기 위해서는 다양한 보안 기법들이 제공되어야 한다. 하지만 대부분의 IoT 기기들은 1) 저성능 및 저전력이기 때문에 2) 그리고 보안을 전문으로 하지 않는 중소기업에서 설계를 하기 때문에, 각각의 기기에서 필요한 모든 보안 기능들을 정확하게 구현하고 제공하는 것은 불가능하다. 또한 IoT 기기들의 개발 환경 또한 제조사에 따라 상이하기 때문에 필요한 보안 기능들을 매번 새로 구현해야하는 어려움이 존재한다. 따라서 본 연구는 IoT 기기들에 다양한 보안 요구 조건을 구현하는 대신에 전체 IoT 인프라가 안전하게 구현하는 것을 목표로 한다. 이를 위하여 각각의 IoT 기기의 보안 요구 조건을 최소화하는 대신 풍부한 컴퓨팅 자원을 가지는 게이트웨이에 필요한 보안 기능들을 집약 시키며, 각 게이트웨이의 보안 기능과 IoT 기기들의 보안 기능들을 효율적으로 결합하여 전체적으로 안전한 IoT 인프라를 구현하고자 한다.

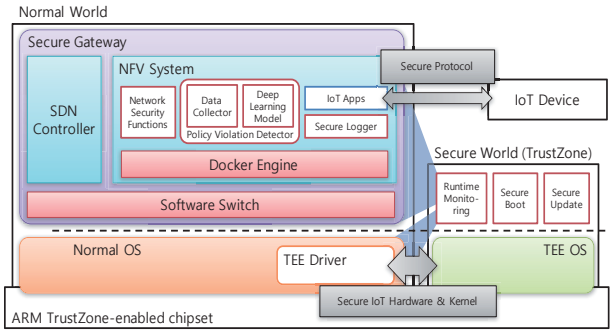


그림 1. SIOT 구조도

〈그림 1〉은 본 연구에서 제안하는 SIOT의 구조도를 나타낸다. 기기들의 출입이 자유롭고 지속적으로 변하는 IoT 환경에서 필요한 보안 기능들을 올바르게 제공하기 위해서는 동적인 네트워크 제어 기술과 유연한 보안 기능 관리 기술이 필요하다. 이를 실현하기 위해, SIOT는 (1) 제어 평면과 데이터 평면을 분리하여 논리적 중앙 집중화된 서버인 SDN 컨트롤러를 통해 동적인 네트워크 제어를 제공하는 소프트웨어 정의 네트워킹(Software-defined Networking;SDN) 기술 [7]과, (2) 하드웨어 형태의 네트워크 장비들을 가상화하여 소프트웨어 형태로 제공하는 네트워크 기능 가상화(Network Function Virtualization; NFV) 기술[8]을 기반으로 한다.

SIOT가 제공하는 보안 기법들은 그 목적에 따라 크게 3가지로 분류된다. 먼저 네트워크상에서의 공격 및 이상 접근들을 탐지하고 차단하기 위한 보안 기법으로써, SIOT는 가상화된 네트워크 보안 기능들과 딥 러닝 기반의 학습 모델을 제공한다.

또한 네트워크에 연결된 IoT 기기들의 이상을 탐지하고 전송되는 데이터의 위/변조를 탐지하기 위한 보안 기법으로써 IoT 기기 로그 수집 및 분석 기능을 제공한다. 뿐만 아니라, 보안 기능들이 집약된 SIOT의 무결성을 보장하기 위해 안전 부팅 및 원격 기기 검증 기법을 제공한다. 분류된 각 보안 기법에 대한 자세한 설명은 아래와 같다.

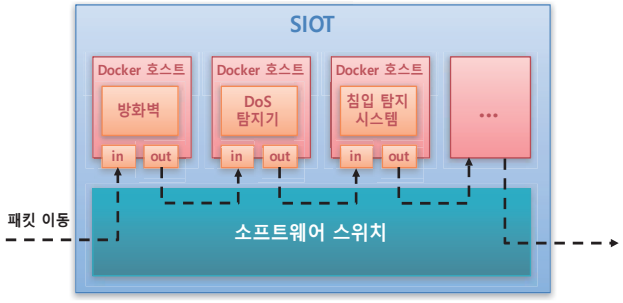


그림 2. NFV 환경 및 가상화된 네트워크 보안 기능

2. 네트워크 공격 및 이상 접근 차단

IoT 기기들은 네트워크를 통해 다른 기기 또는 시스템과 통신하며 다양한 서비스를 제공한다. 하지만 이러한 구조는 네트워크를 통한 다양한 공격을 가능하게 했으며, 이는 IoT 환경에서 큰 보안 위협으로 작용한다. 예를 들어, 공격자는 네트워크 스캔을 통해 알려진 취약점을 가진 IoT 기기를 찾고, 해당 취약점을 공격하는 패킷을 전송함으로써 사용자의 IoT 기기들을 탈취 할 수 있다. 이렇게 탈취된 기기들은 분산 서비스 거부 공격(Distributed Denial of Service; DDoS)등의 또 다른 공격에 활용될 수 있다[9]. 네트워크를 통해 이루어지는 IoT 기기들에 대한 공격들을 탐지하고 방어하기 위해서는 다양한 네트워크 보안 기능들을 필요로 하지만, 필요한 모든 보안 기능들을 저성능 IoT 기기에 구현하는 것은 불가능하며, 현존하는 하드웨어 형태의 네트워크 보안 장비들을 IoT 환경에 설치하는 것 또한 비용상의 문제를 초래한다.

본 연구에서는 네트워크 보안 기능들을 IoT 기기 내에 구현하는 대신, 보다 더 풍부한 컴퓨팅 자원을 가지는 게이트웨이에 구현함으로써 IoT 네트워크에 대한 공격들을 탐지하고 방어한다. 특히 탄력적이고 유동적인 보안 기능 개발 환경을 제공하기 위하여 NFV 기술을 사용하여 소프트웨어 형태의 가상화된 네트워크 보안 기능들이 게이트웨이 내에서 동작하도록 한다.

〈그림 2〉는 게이트웨이 내부에 설치된 NFV 환경 및 가상화된 네트워크 보안 기능들을 나타낸다. NFV 환경은 호스트 단위로 관리되며, 각 호스트는 독립적으로 실행되는 하나의 네트워크 보안 기능을 가진다. 이는 네트워크 상황에 따라 동적으로 NFV 호스트를 올리고 내림으로써 유동적인 네트워크 보안 기능 관리를 가능하게 한다. 또한, 다양한 네트워크 보안 기능들을 동시에 제공하기 위해서, 네트워크 플로우가 다수의 NFV 호스트를 순차적으로 지나도록 스위치를 설정함으로써 네트워크 보안 기능의 체이닝을 지원한다. 이후 네트워크 공격이 탐지될 시, 해당 공격에 대한 정보는 SDN 컨트롤러에 전달되고, 이 내용을 바탕으로 SDN 컨트롤러는 탐지된 공격을 막는 플로우 룰을 생성하여 설치한다.

본 연구에서는 설명한 NFV 환경 및 네트워크 보안 기능들을 구현하기 위해 컨테이너 어플리케이션 중 하나인 Docker[10]를 사용한다. Docker 엔진을 통해 생성된 NFV 호스트들은 2개의 분리된 네트워크 인터페이스를 가지며, 이는 네트워크 보안 기능들의 체이닝을 가능하게 한다. 또한 NFV 호스트와 SDN 컨트롤러간의 통신을 위해 도메인 소켓 기반의 채널을 구현하여 각 보안 기능에 의해 탐지된 공격에 대한 정보를 전달하도록 한다. 앞서 기술한 보안 기법을 통해 알려진 공격들에 대한 방어

를 실현할 수 있으나, 알려지지 않은 공격이나 물리적인 공격을 동반하는 공격 기법에 대한 탐지 및 차단이 어렵다는 단점이 있다. 예를 들어, 공격자가 하나의 IoT 기기를 탈취하여 도어락의 문을 여는 등의 악의적인 행위를 하는 것은 방화벽이나 침입 탐지 시스템 등의 보안 기능들에 의해 탐지되기 어렵다. 이를 해결하기 위해, 본 연구에서는 추가적으로 딥 러닝 기반의 학습 모델을 통해 네트워크로부터 수집된 데이터들을 분석함으로써 비정상적인 접근 및 정보를 탐지한다. 본 기법은 앞서 언급한 문제를 효과적으로 해결할 뿐 아니라 새로운 센서나 IoT 기기가 네트워크에 연결되더라도 탐지 기법에 대한 별도의 수정 없이 동작할 수 있다는 장점 또한 가진다.

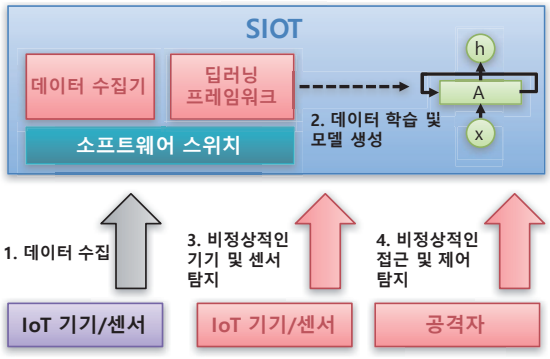


그림 3. 딥 러닝 기반의 학습 모델 동작 방식

〈그림 3〉은 딥 러닝 기반의 학습 모델의 동작 방식을 나타낸다. 먼저 데이터 수집기를 통해 IoT 환경에 설치된 다양한 센서들과 IoT 기기들로부터 지속적으로 정보를 수집한다. 이 정보는 온도나 움직임 유무와 같은 센서들의 센서 값과 문 잠김 또는 불 켜짐과 같은 IoT 기기들의 상태 정보를 포함한다. 수집된 데이터들은 각 데이터들이 수집된 시간과 함께 저장되며, 이후 딥 러닝 프레임워크에 전달되어 학습을 통해 모델을 생성하거나, 생성된 모델에 입력 값으로 전달되어 비정상적인 네트워크 상태를 탐지한다. 또한 비정상적인 기기 접근 및 제어를 탐지하기 위해, 요청된 제어에 의해 변경되는 기기의 상태 정보들을 계산하고 학습 모델에 입력함으로써 해당 요청이 정상적인지 비정상적인지를 판단한다. 만약 비정상적인 네트워크 상태나 제어 시도가 탐지되면 해당 정보를 SDN 컨트롤러에 전달하여 탐지된 공격을 막도록 한다.

본 연구에서는 잘 알려진 딥 러닝 프레임워크 중 하나인 Tensorflow[11]를 사용하여 딥 러닝 기반 학습 모델을 구현한다. 또한 센서 값 및 기기 상태 정보를 단편적인 데이터로 사용하지 않고 이전 데이터와의 연관 관계 및 변화 정도를 함께 관찰하기 위해, 이전 결과값을 함께 사용하는 Recurrent Neural Network(RNN) 모델 중 그라디언트 소실(vanishing gradient)을 해결한 Long Short Term Memory(LSTM) 모델을 사용한다. 모델을 학습하기 위해 각 센서 및 기기로부터 수집된 데이터를 시간 값을 기준으로 통합하여 하나의 데이터셋으로 가공하고, 시간 값은 비트 배열(bit array)로 치환하여 입력한다.

3. IoT 기기 공격 탐지 및 위/변조 탐지 로그

저가/저용량의 컴퓨팅 자원을 사용하는 IoT 기기는 제한된 보안 기능으로 인해 시스템이 작동하는 도중 공격자가 하드웨어/소프트웨어 루팅 또는 업데이트 및 패치 취약점, 권한 관련 취약점 등을 통한 악의적인 행위를 몰래 일으킬 수 있다. 이러한 현상을 탐지 및 분석/처리할 수 있으려면 IoT 기기에서의 모든 수행 기록을 로그로 남길 수 있어야 한다. 하지만 프로그램 설치/삭제, 작업 데이터, 결과, 전송 메시지 등을 안전하게 로그로 남기기 위해서는 몇 가지의 문제점을 해결해야 한다. 첫째, 로깅을 해줄 모듈과 로그를 저장할 디스크 자원이 제한되어 다른 기기 또는 게이트웨이로의 주기적 오프로딩이 필요하다. 둘째, 로그 생성에 필요한 컴퓨팅 자원을 최소화하는 고효율 프로세싱을 기법이 필요하다. 마지막으로 셋째, 공격자가 IoT 기기를 해킹 후 공격 정보가 담긴 로그를 위/변조 또는 삭제를 할 수 있기 때문에 이를 탐지할 수 있어야 한다.

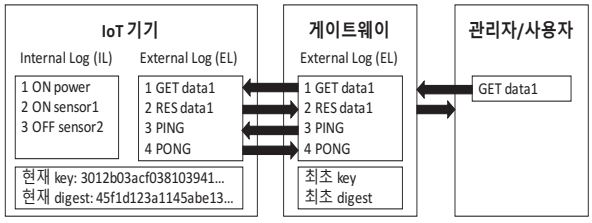


그림 4. 위변조 로그 탐지 시스템 구조

본 연구에서는 위 세 가지 문제점을 해결하기 위해 IoT 기기 공격 탐지 및 위/변조 탐지 로그 기법을 제시한다. 위/변조 로그 탐지 시스템의 전체적인 구조는 〈그림 4〉와 같다. IoT 기기와 통신하거나 제어를 하는 관리자는 모든 메시지를 게이트웨이를 통해 주고 받는다. 이 통신 메시지는 주기적으로 IoT 기기의 상태를 확인하는 PING/PONG 메시지와 함께 모두 게이트웨이에 external log(EL)로 저장된다. 반면 IoT 기기에서는 EL에 더하여 IoT 기기에서 자가 생성된 정보(예: 센서 ON/OFF) 로그로 internal log(IL)를 저장한다. 이 때 IoT 기기는 제한된 스토리지 사용량을 최소화하기 위해 주기적으로 또는

로그 검증 요청 시 게이트웨이로 IL과 EL을 전송하며, 게이트웨이는 EL이 위/변조되었는지 확인하기 위해 자신의 EL과 비교한다.

위 방법은 적은 디스크와 네트워크 자원만으로 EL의 변조를 탐지할 수 있지만 IL의 변조/삭제 유무를 알 수 없다는 근본적 문제가 있다. 따라서 IL의 변조가 있을 때 게이트웨이가 이를 탐지할 수 있게 하기 위한 방법으로 체인 해시 기법을 활용한다. 체인 해시는 로그 무결성을 증명하는 데 자주 쓰이는 기법[12][13]으로, 기본 원리는 메시지 간의 의존성을 주기 위해 새 메시지 생성 시 기존 digest와 함께 해시를 하여 하나의 새로운 digest만을 가지고 기존 digest를 삭제하는 방식이다. 본 연구에서는 이 기법을 응용하여 〈그림 5〉와 같은 해시 체인을 통한 위/변조 탐지 로그를 설계한다. (1) 해시 체인을 시작할 때 정해진 메시지와(예: dummy log) 최초 key를 SHA1 해싱을 하여 최초 digest(Hash0)를 만든다. 이 때 digest 생성에 사용된 key값은 자체적으로 MD5 해싱을 하여 새로운 값(Key1)으로 변경한다. (2) 다음 새로운 로그 생성 시 새 key로 digest를 생성 후 기존 digest와 다시 한번 해싱을 하여 최종 digest(Hash1)를 만든다. 이 방법은 체인 형식으로 계속 이어진다.

해시 체인 기법을 사용하면 다음과 같이 위/변조 로그를 탐지할 수 있다. 우선 IoT 기기와 게이트웨이는 각각 최초 key와 digest를 공유한다. 게이트웨이는 EL 생성 시 로그를 저장하고 digest를 생성하지는 않는다. 반면 IoT 기기는 IL이나 EL이 생성될 때마다 로그를 저장한 뒤 해시 체인 기법을 사용하여 digest와 key를 업데이트 한다. 이후 일정 시간이 지나거나 특별한 이벤트(예: 공격 의심 징후)가 생기면 게이트웨이는 IoT 기기에게 로그 요청 메시지를 보내어 로그 인증을 실시한다. 요청을 받은 IoT 기기는 IL과 EL에 더불어 최종 digest를 게이트웨이로 전송한다. 게이트웨이는 받은 로그의 순서에 따라 해시 체인을 만들고 최종적으로 게이트웨이가 만든 digest가 IoT 기기에서 보낸 digest와 같은지 비교를 한다. 이 때 만약 digest가 다르다면 로그가 변조되었다는 것을 관리자/사용자에게 알림으로써 공격을 탐지한다.

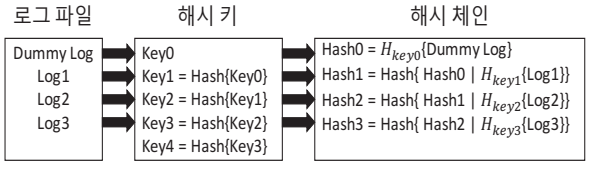


그림 5. 해시 체인을 통한 위변조 탐지 로그 설계

4. SIOT 무결성 보장 및 원격 검증

게이트웨이에서 동작하는 프로그램이 제작자 혹은 사용자가 의도한 대로 올바르게 동작하기 위해서 는 반드시 펌웨어의 무결성을 보장해야 한다. 이를 위해 SIOT는 ARM Trusted Firmware[14]를 보드에 이식하여 구현하였다. 또한 기기 인증과 같은 보안 소프트웨어들을 안전하게 동작시키기 위해 펌웨어에서 부팅 시 ARM TrustZone[15] 기능을 활성화 시켰다. TrustZone은 Secure World와 Normal World라는 보안, 비보안 방식을 지원하며, TZASC(TrustZone Address Space Controller)를 통해 DRAM을 보안 메모리와 비 보안 메모리로 구분할 수 있다.

가. 안전 부팅

– ARM Trusted Firmware 설명 및 구조

ARM Trusted Firmware는 기능에 따라 5단계의 부트로더로 구성이 된다. 각 부트로더는 하드웨어 초기화 및 기기 설정을 하는 코드들로 구성이 되어 있으며, 부트로더1은 롬(ROM)에 존재하는 코드로 부팅 시 가장 먼저 실행되는 코드이며, 부트로더2는 이후 사용될 부트로더 이미지들을 설정된 메모리 주소에 적재하고, 다음 실행될 부트로더 순서를 결정한다. 부트로더3-1과 부트로더 3-2는 Secure World를 위한 예외 벡터와 처리 코드들을 설정하며, 부트로더 3-3은 Normal World를 위한 설정을 담당한다.

그림 6. 해시 검증 과정

– 안전 부팅 디자인

각 부트로더의 무결성을 보장하면서 안전한 부팅을 지원하기 위하여 각 부트로더들은 다음 부트로더를 저장장치에서 메모리로 적재할 때 해당 부트로더의 해시 값을 검증한다. 만약 부트로더가 변조되어 기존에 저장되어 있던 정상 해시 값과 일치하지 않으면 더 이상 부팅 과정이 진행되지 않는다.

– 안전 부팅 구현

Hikey Board[16]에 안전 부팅을 활성화 하여 ARM Trusted Firmware를 이식하였다. 각 부트로더의 정상 해시 값을 안전하게 보관하기 위하여 X. 509v3인증서 기반으로 구현하였으며, 인증 방식은 2048bit RSA, 해시 알고리즘은 SHA-256를

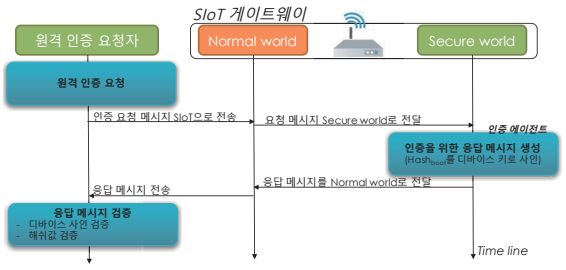


그림 7. 원격 기기 검증 과정

사용하였다. 루트 공개키와 부트로더 1은 모든 인증과정에서 신뢰 근간이 되는 요소들로서 공격자에 의해 변조되지 않기 위해 별도의 하드웨어의 레지스터들과 롬에 저장되어야 한다. 또한 신뢰 인증서는 보안 공개키와 비 보안 공개키를 가지고 있으며 루트 개인키로 서명이 되어있다. 〈그림 6〉과 같이 신뢰 인증서 외에 각 부트로더 단계별로 키 인증서와 해시 인증서가 있으며, 키 인증서에는 해시 인증서를 검증하기 위한 공개키가 포함되어 있으며, 해시 인증서에는 각 단계 부트로더의 해시 값이 저장되어 있다.

이후 부트로더 1이 부트로더2를 검증하기 위하여 루트 공개키를 사용하여 신뢰 인증서를 검증한 후 보안 공개키를 얻는다. 보안 공개키를 사용하여 부트로더 2의 키 인증서를 검증하고 부트로더2의 해시 인증서의 해시 공개키를 얻는다. 이를 활용하여 최종적으로 부트로더2의 해시 인증서에 있는 부트로더2의 정상 해시 값을 얻어 적재하려고 하는 부트로더 파일의 해시 값과 비교한다. Secure World에서 동작하는 부트로더 3-1,3-2 또한 같은 방식으로 검증하며, Normal World에서 동작하는 부트로더 3-3만 비보안 공개키를 사용하여 검증한다. 이는 안전하지 않은 Normal World에서 동작하는 부트로더 3-3은 보드 제조사와 같은 제3자의 개인키로 서명될 수 있기 때문이다.

나. 기기 인증

SIOT를 외부에서 인증하기 위한 기능을 TrustZone을 이용해서 구현하였다. 인증은 기본적으로 PKI를 기반으로 이루어지며, 이때 사용되는 개인키는 게이트웨이 생산 시에 Secure world에서만 접근 가능한 형태로 기기 내에 삽입된다고 가정한다. 즉, Secure world에서 동작하는 인증 서비스만이 개인키를 사용하여 암호화 작업 등을 수행할 수 있다. 구체적인 기기 인증 과정은 〈그림 7〉과 같다.

우선 원격 인증 요청자는 SIOT에 원격 인증 요청 메시지를 전달한다. 이때, 인증 요청 메시지는 재전송 공격을 방지하기 위해 난수 값이나 메시지 생성 시간 값 등을 챌린지로써 포함하게 된다. 인증 요청을 받은 SIOT는 수신한 메시지를 Secure world로 전달한다. Secure world의 인증 에이전트는 수신된 메시지의 챌린지 값을 디바이스의 부팅 이미지 해시 값과 조합 후, 개인키로 서명하여 응답 메시지를 생성한다. 응답 메시지는 원격 인증 요청자에게 전송된다. 원격 인증 요청자는 수신한 응답 메시지를 SIOT의 공개키로 복호화하고, 수신한 챌린지 값과 SIOT 부팅 이미지의 유효성 여부를 검증한다.

Ⅳ. 관련 연구

1. 네트워크 공격 및 이상 접근 차단 연구

최근 IoT 환경에 대한 네트워크 공격을 탐지하고 차단하는 연구가 활발하게 진행되고 있다 [17][18]. 하지만 대부분의 연구들이 직접적인 구현 없이 아키텍처에 대한 디자인만을 언급하고 있다. 또한 산업계에서도 게이트웨이를 활용한 IoT 보안을 실현하고자 하고 있으나 [19][20], 이들 역시 단순한 네트워크 보안 기능만을 제공할 뿐 보안 기능들의 유동적인 관리와 체이닝

표 1. 알려진 IoT 보안 취약점을 해결하기 위해 SIOT에 적용된 보안 기능

공격 및 취약점	SIOT에 적용된 보안 기능
통신 취약점	경량 암호화 기법을 적용한 통신 채널 (ChaCha20[24])
권한 관련 취약점	딥러닝 기반 학습 모델을 통한 비정상 접근 탐지 (Ⅲ-1) 기기 로그 수집 및 분석을 통한 탐지 (Ⅲ-2) 안전 부팅을 통한 기기 무결성 검증 (Ⅲ-3)
소프트웨어 취약점	네트워크 보안 기능을 통한 공격 트래픽 탐지 (Ⅲ-1) 딥러닝 기반 학습 모델을 통한 이상 동작 탐지 (Ⅲ-1) 안전 부팅을 통한 기기 무결성 검증 (Ⅲ-3)
업데이트 및 패치 취약점	기기 로그 수집 및 분석을 통한 탐지 (Ⅲ-2) 안전 부팅을 통한 기기 무결성 검증 (Ⅲ-3)
센서 취약점	딥러닝 기반 학습 모델을 통한 이상 동작 탐지 (Ⅲ-1)
사용자 인터페이스 관련 취약점	네트워크 보안 기능을 통한 공격 트래픽 탐지 (Ⅲ-1)

기법 등이 충분히 고려되지 않았다.

IoT 환경에서의 이상 접근 탐지 및 차단을 위한 연구들 또한 최근 활발하게 진행되고 있다 [21][22]. 하지만 대부분의 연구들은 사용자 또는 관리자가 지정한 몇가지 정책에 한정되어 접근을 제어하기 때문에, 새로운 기기 또는 센서의 등장으로 인해 정의되지 않는 접근에 대한 제어가 불가능하며, 사용자 또는 관리자에게 정책 관리 오버헤드를 준다는 단점 또한 존재한다.

2. IoT 기기 공격 탐지 및 위/변조 탐지 연구

위/변조 메시지를 탐지하기 위한 기법으로 해시 체인이 과거 연구에서 여러 번 제시되었다. Ma와 Tsudik은 믿을 수 있는 3자(third party)나 보안 하드웨어가 없어도 위/변조 로그를 검증할 수 있도록 해주는 Forward-Secure Sequential Aggregate(FssAgg) 인증 기법을 제안한다[12]. FssAgg는 forward signature를 모으는 방법으로 로그 생성시 기존 signature와 함께 새 signature를 만든 뒤 기존 signature를 삭제한다. 이 기법은 본 연구와 비슷하지만 IoT 기기에서만 생성되는 IL을 EL과 연결하여 검증할 수 없다는 단점이 있다. Crosby와 Wallach은 트리 형식의 자료 구조를 사용하여 위/변조 로그 검증 시 필요한 데이터의 양을 줄일 수 있는 방법을 제시한다[13]. 로그를 저장하는 트리의 외부 노드 아이디는 이벤트의 해시로, 내부 노드 아이디는 children 노드 아이디들의 해시를 활용하여 기존 이벤트를 효율적으로 복원할 수 있게 한다. 이 결과8천만개의 이벤트를 저장 시 사용되는 용량을 800MB에서 3KB로 줄였다. 본 연구에서는 IoT 기기가 주기적으로 게이트웨이로 로그를 오프로딩 하지만 위 기법을 적용하면 추가적으로 데이터 사용률을 줄일 수 있을 것으로 기대된다.

3. 기기 무결성 보장 및 원격 검증 연구

ARM TrustZone을 활용하여 모바일 디바이스를 안전하게 사용하고자 하는 연구들이 진행되어 왔다. Asokan 외 연구팀은 모바일 디바이스를 위한 신뢰 컴퓨팅 솔루션들을 연구, 표준, 배포 관점에서 정리하였으며, 특히 하드웨어에서 제공하는 보안 연산 기저를 활용하여 디바이스 인증, 부트 무결성, 신뢰 실행 환경을 설계하는 기본 방안을 기술하였다. Ekberg 외 연구팀은 신뢰 실행 환경에서 지원하는 기능들이 무엇인지와 신뢰 실행 환경이 보장되기 위해 수반되어야 하는 보안요소들에 대하여 논의하였다. 또한, ARM TrustZone 기술이 활성화된 상황에서 모바일 기기의 하드웨어 구조를 기술하였다. 본 논문들을 처음 보안 하드웨어를 설계하는 관점에서 필요한 요소들을 기술하는데 초점을 맞추었고 ARM 기반의 하드웨어는 보드 종류에 따라 다르게 구현될 수 있기 때문에 구현 부분에 대한 서술이 부족한 한계가 있다.

V. 결 론

본 연구에서는 IoT 환경에 존재하는 취약점들에 대한 조사 및 분석을 통해 총 6가지의 IoT 보안 취약점을 정의하였고, 이를 해결하기 위해 다양한 보안 기술들이 집약된 IoT 게이트웨이인 SIOT를 제안한다. 〈표 1〉은 현존하는 IoT 보안 취약점들과 각 취약점을 해결하기 위해 SIOT에 적용된 보안 기능들을 나타낸다. 본 논문에서 통신 취약점을 해결하기 위한 보안 기법에 대한 설명은 제외되었으나, SIOT에 구현된 NFV 기반의 유동적인 네트워크 보안 기능 관리 기술은 ChaCha20 [24]등의 현존하는 경량 암호화 기법을 SIOT에 그대로 적용함으로써 대응할 수 있음을 보여준다. 본 연구팀은 제안된 SIOT가 안전한 IoT 환경을 구축하기에 충분하며, 또한 종합적인 IoT 보안 프레임워크로써 IoT 환경에 대한 보안 연구에 큰 기여를 할 것으로 기대한다.

Acknowledgement

이 논문은 삼성전자 미래기술육성센터의 지원을 받아 수행된 연구임 (과제번호 SRFC-TB1403-01).

참 고 문 헌

[1] M2M.World.News, “IDATE forecasts 80 Billion things connected in 2020”, M2M World News, 2013, http://m2mworldnews.com/2013/09/18/27009-idate-forecasts-80-billion-things-connected-in-2020/

[2] J. Rivera, R. Meulen, “Gartner’s 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relationship Between Humans and Machines”, Gartner Newsroom , 2013, http://www.gartner.com/newsroom/id/2575515

[3] Technical Report, “Security in the Internet of Things”, Wind River Systems (2015)

[4] Barcena, Mario Ballano, and Candid Wueest. "Insecurity in the Internet of Things." Security Response, Symantec (2015).
[5] Viewpoint Paper, "Securing the Internet of Things – Explore Security and Privacy in an Interconnected World", Hewlett Packard Enterprise (2015)
[6] White Paper, "Bootstrapping Security", Ericsson (2016)
[7] ONF, "Software-Defined Networking: The New Norm for Networks", Open Networking Foundation, 2012
[8] ETSI,"Network Function Virtualisation", Introductory White Paper, 2013
[9] Mirai. https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html
[10] Docker. https://www.docker.com/
[11] Tensorflow. https://www.tensorflow.org/
[12] D. Ma, and G. Tsudik. "A new approach to secure logging", In ACM Transactions on Storage (TOS), 5(1), pp. 1-21, 2009.
[13] S. A. Crosby, and D. S. Wallach. "Efficient Data Structures for Tamper-Evident Logging", In Proceedings of the USENIX Security Symposium, 2009.
[14] ARM Trusted Firmware, https://github.com/ARM-software/arm-trusted-firmware
[15] ARM TrustZone, https://www.arm.com/products/security-on-arm/trustzone
[16] Hikey Board, http://www.96boards.org/product/hikey/
[17] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad. Proposed security model and threat taxonomy for the internet of things (iot). In Recent Trends in Network Security and Applications, pages 420 - 429. Springer, 2010.
[18] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah. A systemic approach for iot security. In Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on, pages 351 - 355. IEEE, 2013.
[19] Cisco Systems Inc. Cisco 910 Industrial Router. http://www.cisco.com/c/en/us/support/routers/910-industrial-router/model.html
[20] J. Maguire. Internet of Things (IoT) Service Delivery using NFV/SDN. Freescale Technology Forum 2014. FTF-NET-F0160.
[21] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In Proceedings of the 14th ACM Workshop on Hot Topics in Networks, HotNets-XIV, 2015
[22] Asokan, N and Ekberg, Jan-Erik and Kostiainen, Kari and Rajan, Anand and Rozas, Carlos V and Sadeghi, Ahmad-Reza and Schulz, Steffen and Wachsmann, Christian. "Mobile Trusted Computing", 2014. Proceedings of the IEEE, pp. 1189-1206.
[23] J.-E. Ekberg, K. Kostiainen, and N. Asokan. "The untapped potential of trusted execution environments on mobile devices", 2014. IEEE Security Privacy Mag., DOI: 10.1109/MSP.2014.38.
[24] ChaCha20, https://tools.ietf.org/html/rfc7539

약 력

김 연 근

2014년 울산과학기술대학교 공학 학사
2016년 한국과학기술원 석사
2016년~현재: 한국과학기술원 박사 과정

고 영 환

2011년 KAIST 전기및전자공학부 공학사
2013년 KAIST 전기및전자공학부 공학석사
2013년~현재 KAIST 전기및전자공학부 공학박사
관심분야: 네트워크 및 분산 시스템, GPU 가속화, 네트워크 보안, 모바일 네트워크

김 민 수

2011년 KAIST 공학사
2013년 KAIST 공학석사
2013년~현재 KAIST 정보보호대학원 박사과정
관심분야: 시스템 보안

장 진 수

2007년 아주대학교 공학사
2014년 KAIST 공학석사
2014년~현재 KAIST 정보보호대학원 박사과정
관심분야: 시스템 보안

배 상 욱

2013년 KAIST 전기및전자공학부 공학사
2015년 KAIST 전기및전자공학부 공학석사
2015년~현재 KAIST 전기및전자공학부 공학박사
관심분야: 네트워크 및 분산 시스템

약 력

노 주 환

2013년 KAIST 전기 및 전자공학과 학사
2015년 KAIST 전기 및 전자공학과 석사
2015년~현재 KAIST 전기 및 전자공학부 박사과정
관심분야: CPS 보안

강 병 훈

1993년 서울대학교 계산통계학 학사
1995년 Univ. of Maryland at College Park, 전산학 석사
2004년 Univ. of California at Berkeley, 전산학 박사
2004년~2010년 Univ. of North Carolina at Charlotte, 조교수
2010년~2013년 George Mason University, 부교수
2013년~현재 KAIST 정보보호대학원 부교수
관심분야: 시스템 보안

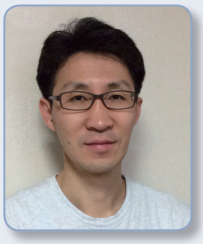
박 경 수

1997년 서울대학교 계산통계학과 전산전공 학사
2004년 미국 프린스턴대학교 전산학 석사
2007년 미국 프린스턴대학교 전산학 박사
2007년~2008년 미국 프린스턴대학교 선임연구원
2007년~2009년 CoBlitz Inc. 최고기술책임자
2009년 미국 프린스턴대학교 조교수
2010년~2012년 KAIST 전기및전자공학부 조교수
2012년~현재 KAIST 전기및전자공학부 부교수
관심분야: 네트워크 및 분산 시스템, 네트워크 미들박스, 네트워크 보안

김 용 대

1998년~2002년 University of Southern California 박사
1993년~1998년 한국전자통신연구원 연구원
2002년~2012년 University of Minnesota 조/부교수
2012년~현재 KAIST 전기 및 전자공학부 / 정보보호대학원 정교수
관심분야: 사이버 물리시스템, 사회관계망, 인터넷 라우팅, 이동통신, P2P, 무선 센서 네트워크, 클라우드 컴퓨팅, 봇넷 등에 대한 보안 기술

약 력



신 승 원

1998년 한국과학기술원 공학 학사
2000년 한국과학기술원 공학 석사
2013년 Texas A & M University 공학 박사
2000년~2002년 티맥스소프트 책임연구원
2002년~2005년 ETRI 연구원
2005년~2006년 MIT 방문 연구원
2006년~2009년 티맥스소프트 수석보 연구원
2011년 SRI International 인턴
2012년 SRI International 인턴
2013년~현재 한국과학기술원 전산학부
정보보호대학원 조교수
관심분야: SDN, NFV, Security, Android, Cloud