

IoT 공통 보안 원칙 v1.0

(IoT Common Security Principle v1.0)





IoT Common Security Principle v1.0

IoT 공통 보안 원칙

v1.0

목 차 contents

1. 개요	04
2. IoT 공통 보안 7대 원칙	07
3. 공통 보안 7원칙의 세부 설명.....	10
• 참고문헌.....	23

01. 개요



01

개요



최근 사람과 사람, 사람과 사물의 연결에서 생활 속 모든 것들을(Daily life objects) 상호 연결시키려는 사물인터넷(Internet of Things: IoT) 기술이 新 성장 동력의 핵심으로 주목 받고 있다. IoT 기술은 다양한 물리 공간의 사물들과 가상 공간의 프로세스 및 데이터 콘텐츠들이 인터넷으로 상호 연결되어 초연결사회(Hyper-connected society)가 구축되고, 사용자 중심의 지능형 서비스를 제공하기 위해 거대한 정보가 ‘생성-수집-공유-활용’되는 광범위한 기술이다.

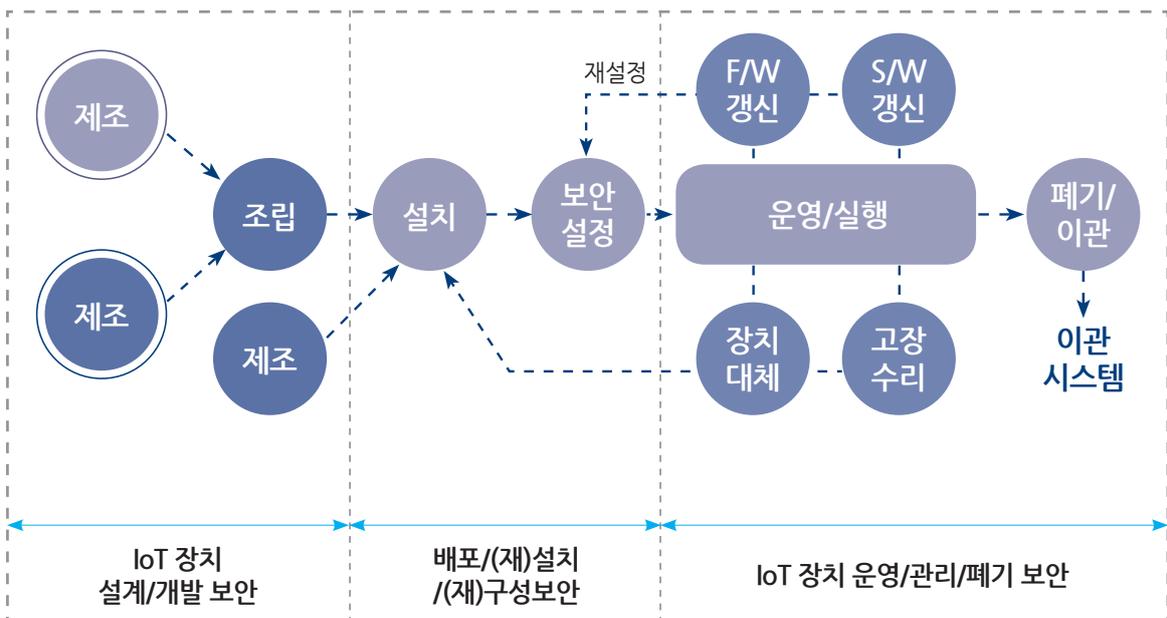
2009년에서 2011년 사이에 한국은 전 세계에서 가장 빠른 스마트기기 보급률 증가 추세를 보이며 IT 유관 시장의 가장 강력한 성장 동인 역할을 수행했지만 2013년 이후 증가 추세가 완만해지면서 새로운 성장 동력이 필요한 시점이다. 다양한 분야에서 혼재하는 스마트화는 스마트폰과 스마트패드 등 개인용 소형 장치부터 스마트카, 스마트홈, 스마트 빌딩으로 발전중이고, 더 나아가 스마트시티와 스마트국가를 목표로 발전해가고 있다. 사물인터넷 기술(IoT 기술)은 이러한 스마트 커뮤니티의 핵심 기술로 부각되고 있고 작은 장치와 가상의 콘텐츠까지 연결하려는 초연결 인터넷 개념으로 자리 잡고 있다.

IoT 기술의 활성화 및 신규 서비스 창출을 위해 보안은 반드시 제공해야 하는 핵심 기술이다. 인터넷에 연결된 장치 수의 증가는 공격할 수 있는 대상의 증가와 위협 요소의 확장을 의미한다. 특히 의료 서비스나 산업 시설 제어 서비스에 적용되는 사물인터넷 장치와 통신 기술에서 보안 기술은 필수적이다. 이러한 서비스가 침해되었을 경우 단순한 경제

적 피해를 넘어서 인명 피해가 유발될 수 있기 때문이다. 또한 주변의 일상 사물들이 연결된다는 것은 개인 정보 유출이나 프라이버시 침해가 우려되는 범위의 증가를 의미하고, 그 침해 정도도 현재와 비교할 수 없을 정도로 증폭될 것은 자명하다.

IoT 기반 융합 서비스가 활성화 될수록 기하급수로 증가하게 될 IoT 연결 장치 (connected devices)들은 작게는 데이터를 수집하는 센서와 간단한 제어가 가능한 액츄에이터를 포함하여, 복수개의 센서와 액츄에이터를 갖는 이중 복합 시스템들까지 다양해진다. 따라서 기존 시스템 중심으로 설계된 인터넷 보안 기술로 안전과 프라이버시 보호를 수행하기에는 무리가 따른다. 따라서 IoT 장치 및 서비스의 ‘설계-개발’ 단계부터 보안과 프라이버시 보호 체계를 고려해야 한다. 또한, IoT 장치를 ‘배포, 설치’하는 단계에서도 사전에 잠재적 보안 위협을 차단할 수 있도록 해야 하며, 실사용이 이루어지는 ‘설정-운영-실행-폐기’ 단계에서는 이 전 단계를 모두 고려하여 전주기적 침해 요소의 분석 및 대응 방안을 마련해야 한다. 즉, 보안의 잠재적 위협요소와 취약점을 전주기 단계에서 점검할 수 있는 기본적인 공통 보안 요구사항과 사용 주체별로 고려해야 하는 최소한의 보안 점검 항목이 필요하다. 다음 그림은 IoT 장치 및 서비스의 전주기 보안 고려사항을 나타낸다.

IoT 장치의 전주기 단계별 보안 고려사항



※ F/W: Firmware, S/W : Software

02. IoT 공통 보안 7대 원칙



- IoT 장치의 설계/개발 단계의 보안 요구 사항
- IoT 장치 배포/설치(재설치)/구성(재구성) 단계의 보안 요구 사항
- IoT 장치 및 서비스 운영/관리/폐기 단계의 보안 요구 사항

02

IoT 공통 보안 7대 원칙



사물인터넷은 이종 장치들과 유·무선 네트워크 기술, 그리고 지능화 플랫폼을 기반으로 개발되어질 것이다. 따라서 IoT 서비스를 구성하고 있는 IoT 장치의 설계부터 폐기까지 전주기에서 보안 위협과 취약성을 점검해야 하며, IoT 서비스 역시 설계에서 운영까지, 모든 단계 별 보안 요구사항을 점검하여 보안을 내재화해야 한다. IoT 공통 보안 7대 원칙은 IoT 장치 및 서비스의 제공자(개발자)와 사용자가 IoT 장치의 전주기 세부 단계에서 고려해야 하는 공통의 보안 요구 사항이다.

① 정보보호와 프라이버시 강화를 고려한 IoT 제품·서비스 설계

② 안전한 소프트웨어 및 하드웨어 개발 기술 적용 및 검증

③ 안전한 초기 보안 설정 방안 제공

④ 보안 프로토콜 준수 및 안전한 파라미터 설정

⑤ IoT 제품·서비스의 취약점 보안패치 및 업데이트 지속 이행

⑥ 안전한 운영·관리를 위한 정보보호 및 프라이버시 관리체계 마련

⑦ IoT 침해사고 대응체계 및 책임추적성 확보 방안 마련

IoT 장치의 설계/개발 단계의 보안 요구 사항

- (1) 정보보호와 프라이버시 강화를 고려한 IoT 제품·서비스 설계
 - “Security by Design” 및 “Privacy by Design” 기본 원칙 준수
- (2) 안전한 소프트웨어 및 하드웨어 개발 기술 적용 및 검증
 - 시큐어 코딩, 소프트웨어, 어플리케이션 보안성 검증 및 시큐어 하드웨어 장치 활용

IoT 장치 배포/설치(재설치)/구성(재구성) 단계의 보안 요구 사항

- (3) 안전한 초기 보안 설정 방안 제공
 - “Secure by Default” 기본 원칙 준수
- (4) 보안 프로토콜 준수 및 안전한 파라미터 설정
 - 통신 및 플랫폼에서 검증된 보안 프로토콜 사용 (암호/인증/인가 기술)

IoT 장치 및 서비스 운영/관리/폐기 단계의 보안 요구 사항

- (5) IoT 제품·서비스의 취약점 보안패치 및 업데이트 지속 이행
 - S/W와 H/W의 보안 취약점에 대해 모니터링하고 업데이트 지속 수행
- (6) 안전한 운영·관리를 위한 정보보호 및 프라이버시 관리체계 마련
 - 사용자 정보 취득·사용·폐기의 전주기 정보의 보호 및 프라이버시 관리
- (7) IoT 침해사고 대응체계 및 책임추적성 확보 방안 마련
 - 보안 사고에 대비한 침입탐지와 사고 시 분석 및 책임추적성 확보

03. 공통 보안 7원칙의 세부 설명



- 정보보호와 프라이버시 강화를 고려한 IoT 제품 · 서비스 설계
- 안전한 소프트웨어 및 하드웨어 개발 기술 적용 및 검증
- 안전한 초기 보안 설정 방안 제공
- 보안 프로토콜 준수 및 안전한 파라미터 설정
- IoT 제품 · 서비스의 취약점 보안패치 및 업데이트 지속 이행
- 안전한 운영·관리를 위한 정보보호 및 프라이버시 관리체계 마련
- IoT 침해사고 대응체계 및 책임추적성 확보 방안 마련

03

공통 보안 7원칙의 세부 설명



●
●

1 정보보호와 프라이버시 강화를 고려한 IoT 제품·서비스 설계

- “Security by Design” 및 “Privacy by Design” 기본 원칙 준수



IoT 제조사와 서비스 제공자는 안전한 IoT 제품 개발 및 서비스 이용환경을 조성하기 위해 해당 서비스에 대해 정확히 이해하고, IoT 제품과 서비스의 설계 단계에서부터 제품 및 서비스가 적용될 환경을 기반으로 보안 취약점을 사전에 분석하여 이를 보완하고 강화할 수 있는 기술을 적용해야 한다. 사전 분석된 위험분석정보를 기반으로, 민감한 정보에 대해 기밀성과 무결성을 제공해주고, 정보 접근 권한을 관리하는 방안 등을 포함시켜야 한다.

'Security by Design'은 IoT 제품 및 서비스의 설계 단계부터 보안을 내재화하고, 지속적인 대응을 수행하여 서비스 사용자 및 사업자의 자원 및 정보를 보호한다는 개념으로 다음을 포함한다.

- IoT 장치가 갖는 저전력/저성능 특성을 고려하여 기밀성, 무결성/인증, 가용성 등 정보 및 기기의 오용을 최소화 하면서 경량화 할 수 있는 방안을 고려한다.
- IoT 서비스에서는 IoT 장치 및 정보에 대하여 서비스 운용환경에 맞는 장치의 접근권한관리, 종단간 통신보안, 무결성/인증 제공 등의 방안을 제공한다.
- 소프트웨어 보안 기술과 하드웨어 보안 기술의 적용을 적극 검토하고, 안전성이 검증된 표준 보안 기술을 활용한다.

'Privacy by Design'은 IoT 제품 및 서비스의 설계 단계에서부터 프라이버시 침해 위협요소를 분석하여 지속적으로 점검하고 침해가 발생하기 전에 선제적인 대응을 한다는 프라이버시 보호 개념이다⁽¹⁾. 프라이버시 강화는 IoT 서비스 제공에 필요한 최소한의 정보만을 취득하고, 사용자가 동의한 기간과 서비스 범위 내에서만 정보를 사용하여 개인의 민감한 정보를 보호하는 방안으로 다음의 고려사항들을 포함한다.

- IoT 장치와 IoT 서비스 운영 정책에 사용자의 프라이버시 보호 방법론을 기본으로 적용한다.
- IoT 장치가 수집하는 프라이버시 정보에 대하여 암호화 전송, 익명 저장 및 무결성/

인증 방안 등을 포함한다.

- IoT 서비스는 수집된 프라이버시 정보에 대한 비식별화, 접근관리/인증, 기밀성, 안전한 저장 등에 대한 방안을 포함한다.
- IoT 서비스 제공자는 사용자에게 프라이버시 정보의 사용 범위 및 기간 등을 포함한 운영 정책을 가시화하여 투명성을 최대한 보장한다.

마지막으로, 특정 지점에서 발생한 보안 침해사고가 서비스 연속성 유지에 필요한 핵심영역까지 영향을 미치지 않도록 IoT 서비스를 물리적·논리적으로 분리된 구조로 설계하고, 개발용으로 사용된 IoT 제품의 물리적 포트 및 불필요한 데몬들을 제거할 것을 권고한다.

2 안전한 소프트웨어 및 하드웨어 개발 기술 적용 및 검증

- 시큐어 코딩, 소프트웨어, 어플리케이션 및 소프트웨어 보안성 검증 및 시큐어 하드웨어 장치 활용



시큐어 코딩 적용

IoT 서비스 환경(IoT 장치, 플랫폼 등)은 다양한 운영체제와 애플리케이션으로 구성되어 있다. 보안지식이 없는 개발자가 구현한 프로그램에는 다양한 보안취약점들이 발생할 수 있으며, 이는 IoT 기기와 서비스에 심각한 오동작, 결함을 야기할 수 있고 잠재되어있는 위험 요소는 공격자의 주요 대상이 된다. 보안 취약점은 소프트웨어 개발 생명 주기의 어느 단계에서나 나타날 수 있다.⁽²⁾

개발자와 공격자의 접근 방법에는 차이가 있다. 개발자는 의도에 초점을 맞춰 접근하지만 공격자는 애플리케이션이 할 수 있는 것과 명확하게 거부되지 않은 행위는 허용된다는 원칙에 따라 비정상 동작을 유도해 보는 것에 초점을 맞춰 접근한다.

안전하지 않은 프로그램은 공격자가 서버 또는 사용자의 컴퓨터를 제어하도록 접근을 허용하며 공격자는 응용 프로그램 또는 서버의 보안 취약점을 찾아 공격을 시도한다. 공격이 성공했을 때 취약점 및 소프트웨어 등에 따라서 데이터베이스, 사용자 시스템, 또는 소프트웨어 및 관련 정보, 관련 서버의 운영체제 등에 영향을 미친다.

보안은 오동작 또는 결함이 나타날 때에 추가할 수 있는 것이 아니다. 따라서 개발자는 장치와 관계없이 소스코드 구현단계부터 내재될 수 있는 보안 취약점을 사전에 예방하기 위해 시큐어 코딩을 적용해야 한다. Java, C/C++로 개발된 소프트웨어는 현재 마련되어있는 시큐어 코딩 가이드를 활용하고^(3, 4), 가이드가 개발되어있지 않은 언어는 국제표준⁽¹⁴⁾에 근거하여 별도의 분석도구 및 방법론을 이용하여 반드시 소스 코드에 대한 보안 품질 검증을 수행해야 한다 (S/W 보안 검증 시 단계별 및 통합적 관리 필요).

● 시큐어 코딩 가이드 예 ●

유 형	내 용
입력 데이터 검증 및 표현	입력 데이터에 대한 검증 누락 또는 부적절한 검증, 데이터의 잘못된 형식지정으로 인해 발생할 수 있는 보안 취약점 예) SQL 삽입, 자원 삽입, 크로스사이트 스크립트, 운영체제 명령어 삽입, LDAP 삽입, 디렉터리 경로 조작 등
보안기능	보안기능(인증, 접근제어, 기밀성, 암호화, 권한관리 등)을 부적절하게 구현할 시 발생할 수 있는 보안약점 예) 부적절한 인가, 중요한 자원에 대한 잘못된 권한설정, 취약한 암호화 알고리즘 사용, 사용자 중요정보 평문 저장(또는 전송)
시간 및 상태	동시 또는 거의 동시 수행을 지원하는 병렬 시스템, 하나 이상의 프로세스가 동작하는 환경에서 시간 및 상태를 부적절하게 관리하여 발생할 수 있는 보안약점 예) 검사시점과 사용시점, 제어문을 사용하지 않는 재귀함수 등
에러처리	에러를 처리하지 않거나, 불충분하게 처리하여 에러 정보에 중요정보(시스템 등)가 포함될 때 발생할 수 있는 보안약점 예) 오류 메시지를 통한 정보 노출, 오류 상황 대응 부재 등
코드오류	타입변환 오류, 자원(메모리 등)의 부적절한 반환 등과 같이 개발자가 범할 수 있는 코딩오류로 인해 유발되는 보안약점 예) 널(Null) 포인터 역참조, 부적절한 자원 해제, 무한 자원 할당 등

유 형	내 용
캡슐화	중요한 데이터 또는 기능을 불충분하게 캡슐화하였을 때, 인가되지 않는 사용자에게 데이터 누출이 가능해지는 보안약점 예) 제거되지 않고 남은 디버그 코드, 시스템 데이터 정보노출
API 오용	의도된 사용에 반하는 방법으로 API를 사용하거나, 보안에 취약한 API를 사용하여 발생할 수 있는 보안약점 예) DNS lookup에 의존한 보안결정

소프트웨어 보안성 검증

IoT 제품 · 서비스 개발 시, 제품 및 서비스의 생산성을 높이고 품질을 향상시키기 위해 다양한 S/W를 활용할 경우, 현재까지 알려진 보안 취약점에 대한 보안성 검증을 수행하고 보안패치를 반드시 적용해야 한다. 알려진 보안 취약점에 대한 보안성을 검증하기 위해 아래의 가이드라인 절차와 같이 수행하며, 참조사이트를 통해 알려진 취약점을 검색 · 대응한다.(5, 6, 13)

● 소프트웨어 보안성 검증의 단계별 내용 ●

유 형	내 용
의존S/W 열거	사용한 오픈소스 S/W를 포함하여 의존성을 가지는 S/W들을 확인하고 열거해야 함 예) 오픈소스 프레임워크인 Alljoyn을 사용한다면, Alljoyn과 OpenSSL 등과 같이 Alljoyn을 사용하기 위해서 필요한 추가적인 S/W 및 library 등을 리스트로 열거해야 함
취약점 검색	열거된 의존 S/W들에 대한 취약점을 검색해야 함 예) 의존S/W 열거단계에서 열거된 모든 S/W 및 library에 대한 취약점을 CVE, CWE, OWASP 등을 통해서 검색 수행
취약점 /대응방법 열거	S/W 별로 알려진 취약점을 열거 예) 열거된 S/W에 대한 CVE, CWE, OWASP 등에서 검색된 취약점 및 대응 방법을 항목별로 리스트에 열거함
대응방법 반영	알려진 취약점에 대한 대응절차에 따라 오픈소스 S/W에 반영하여 보완해야 함

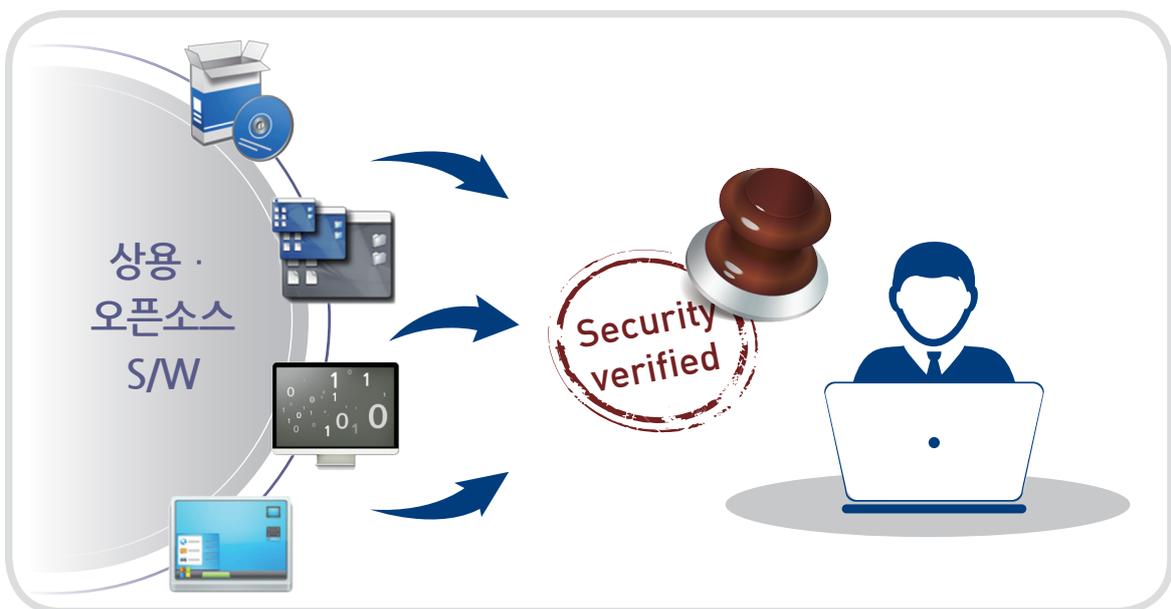
※ S/W : 소프트웨어

시큐어 하드웨어 장치 활용

IoT 장치는 응용 서비스 종류에 따라 다양한 수준의 보안 강도를 필요로 한다. IoT 장치는 공격자에게 쉽게 노출될 수 있는 환경에 주로 설치되기 때문에 부채널 공격이나 펌웨어 코드 추출, 키 값 추출 등 다양한 하드웨어 보안 취약성을 갖는다. 이 때문에 하드웨어 보안을 강화하기 위해 펌웨어/코드 암호화, 실행코드 영역제어, 역공학 방지 기법 등 다양한 하드웨어 보안 기법이 존재하며 이를 IoT 장치의 응용 환경에 따라 적절히 적용할 필요가 있다.

소프트웨어 보안 기술과 하드웨어 보안 기술 융합

소프트웨어 보안 기술과 하드웨어 보안 기술이 융합되는 경우, 소프트웨어 보안 기술과 하드웨어 보안 기술 간에 반드시 신뢰하는 접근 방법 (단방향 및 양방향 인증) 기반의 안전한 보안 채널을 구성하여 전송 데이터에 대한 기밀성과 무결성 기능을 제공해야 한다.



3 안전한 초기 보안 설정 방안 제공

• “Secure by Default“ 기본 원칙 준수

IoT 장치 설치자나 서비스 관리자는 초기 설치 단계와 고장 수리 후 재설치 단계에서 보안 프로토콜들에 기본으로 설정되는 파라미터 값이 가장 안전한 설정이 될 수 있도록 “Secure by Default” 기본 원칙을 준수해야 한다.

IoT 환경에 적용되는 경량화 장치들은 적용 서비스의 요구 기능에 최적화되도록 설계된다. 즉, 대부분의 경량화 장치들은 사용자 입출력 인터페이스(예, 디스플레이 장치나 입력 키패드 등)가 부재하거나 제한적이다. 따라서 장치의 제조사가 장치 제작 시 보안 파라미터를 선 탑재하거나, 설치자가 서비스 도메인에 설정된 장치를 설치한 뒤 서비스 관리자에게 전달해주는 방식이 일반적인 접근 방식이다. 따라서 서비스 관리자나 사용자에게는 적용 서비스의 보안 특성에 따라 제조 시 기본으로 설정된 파라미터들을 설정 및 재설정할 수 있는 방안이 제공되어야 한다.

- 제조사와 설치자가 IoT 장치의 초기 설정을 수행할 때, 보안 모듈과 파라미터는 안전하게 설정되어야 함 (예, 국내외를 사업 대상으로 하는 장치나 서비스의 경우 국제표준 권고 기준인 AES-128 이상의 보안 강도 준수)
- 서비스에서 강력한 암호와 무결성을 요구하는 경우 옵션 중 강한 암호를 기본으로 설정 (예, AE(Authenticated Encryption) 암호 모드 적용)
- 제조 시 기본으로 설정되어진 계정 이름과 패스워드를 설치 시 변경
- 응용 프로그램이 특정 기간이 지나면 암호 키와 인증 패스워드의 만료를 권고할 수 있는 옵션을 활성화하여 설정
- 장치 간, 장치와 인터넷 간에 암호화 통신을 사용하도록 기본 설정
- 다중 요소 인증이 옵션으로 제공될 경우 필요 시 활성화하여 설정
- 다중 사용자로 구성되는 서비스 환경에서는 최소한의 권한으로 초기 설정

4 보안 프로토콜 준수 및 안전한 파라미터 설정

• 통신 및 플랫폼에서 검증된 보안 프로토콜 사용 (암호/인증/인가 기술)

사물인터넷이 주목받으면서 다양한 국내외 표준 기구 및 사설 표준 기구에서 보안 기술들이 논의되고 있다^(10, 11, 12). IoT 제품 개발자와 서비스 제공자는 데이터 통신 및 개방형 플랫폼에 안전성을 보장하는 보안 프로토콜을 적용해야 하고, 보안 서비스(암호/인증/인가) 제공 시 안전한 파라미터들이 설정될 수 있도록 해야 한다.

사물인터넷의 경우, 경량 장치들 간 및 경량 장치와 플랫폼 간의 정보 공유 시 적용 환경을 고려한 경량화 보안 프로토콜의 사용이 고려되어야 한다. 이러한 데이터 전송 보안 기술과 더불어 사용자의 인증 및 인증된 사용자의 접근 권한을 안전하게 관리하는 방식에서도 검증된 보안 프로토콜의 적용과 경량화를 고려해야 한다.

• IoT 서비스 제공을 위한 시스템 구성 요소별 보안 •

유 형	내 용
네트워크	사물인터넷 서비스에서 주로 사용되는 통신/네트워크 접속 프로토콜에 적합한 보안 요구사항 만족
사물인터넷 전용 프로토콜	사물인터넷 표준 기구에서 표준화한 데이터 전송 프로토콜에서 권고하는 보안 요구사항 만족
	프로토콜간 연동 시 보안 취약성 해소 필요
사물인터넷 플랫폼	검증된 표준 기구에서 정의하고 있는 사물인터넷 플랫폼에서 요구하는 보안 요구 사항 만족
서비스 모델	서비스별로 다양한 보안 요구사항 및 보안관련 법/규제가 있을 수 있으며, 이를 만족시켜야 함 응용 서비스별 특성을 고려하여 맞춤형 보안 요구사항을 만족해야 함

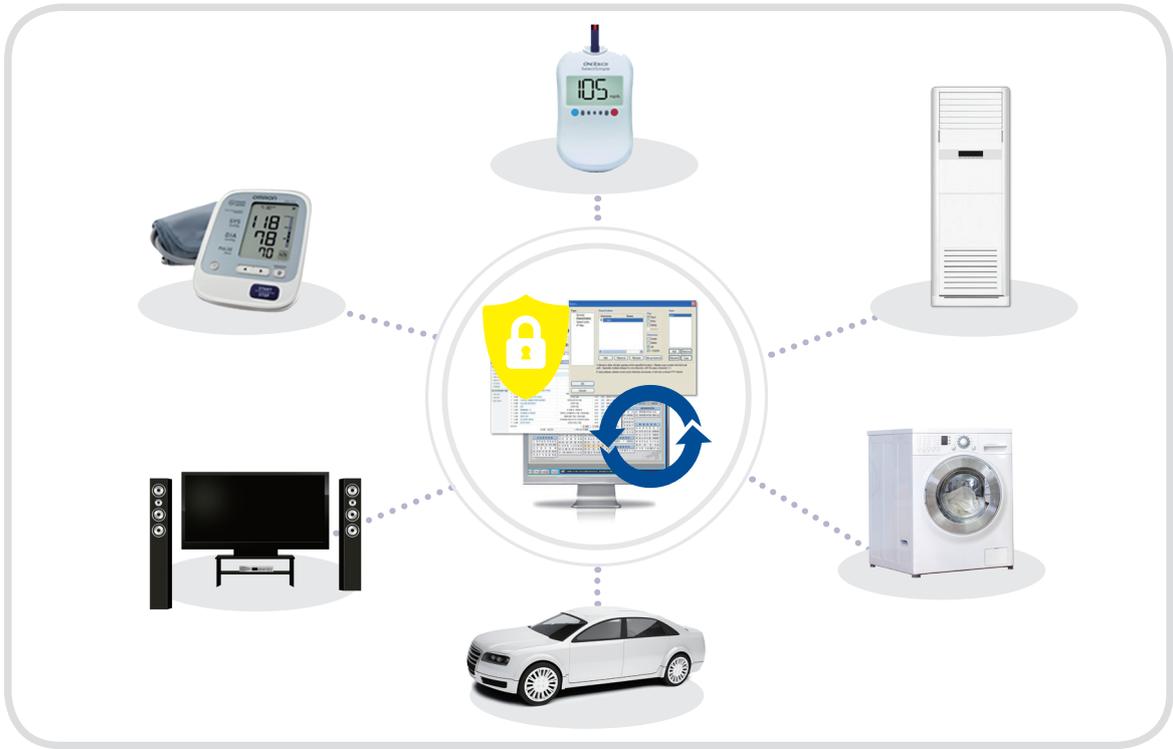
5 IoT 제품·서비스의 취약점 보안패치 및 업데이트 지속 이행

IoT 제품 제조사와 서비스 제공자는 IoT 제품·서비스에서 보안 취약점이 발견되면 이에 대한 분석을 수행하고, 보안 요구사항을 반영한 보안패치를 신속히 배포할 수 있도록 사후 조치 방안을 마련해야 한다. 제품·서비스에 대한 보안취약점 및 보호조치 사항은 홈페이지, SNS 등을 활용하여 사용자에게 공개해야 한다. 아울러, 보안패치 및 업데이트 파일의 배포 과정에서 발생 가능한 위·변조 문제를 사전에 예방할 수 있도록 무결성 검증 기술을 적용해야 한다.

IoT 환경에서는 기존 인터넷 기반 시스템과 달리 사용자 인터페이스(디스플레이나 키보드와 같은 입/출력 장치)가 제한적인 장치들이 많이 사용될 것으로 예측되므로 통신 채널을 통한 S/W 업데이트에서 사용자의 관여를 최소화하는 방법론이 적용될 수 있다. 통신 채널을 활용한 업데이트 S/W의 전송 시 다음의 보안 서비스는 반드시 제공되어야 한다.⁽⁷⁾

● 보안패치 및 업데이트 시 제공해야 하는 보안 서비스 ●

유 형	내 용
주체 간 상호 인증	업데이트 서버와 IoT 장치 사이에 상호 인증 기능을 제공하여 위장 서버나 중간자 공격 등의 취약점에 대응할 수 있도록 함
기밀성	저장 데이터(업데이트 설정 정보 파일: 예, conf, xml, ini 등)와 처리 데이터(주요 파라미터 관련 정보의 임시폴더나 설치 공간) 및 전송 데이터(업데이트 전송 정보)에 대하여 해커의 공격에 대비하여 암호화하여 저장/처리/전송해야 함 IoT 제품·서비스의 보안 패치에 대한 코드 서명(Code Signing) 기법의 적용을 고려해야 함
무결성	저장 데이터(업데이트 정보 파일), 처리 데이터(실행 파일) 및 전송 데이터(업데이트 전송 정보)에 대해 무결성 검사를 수행해야 함



6 안전한 운영·관리를 위한 정보보호 및 프라이버시 관리체계 마련

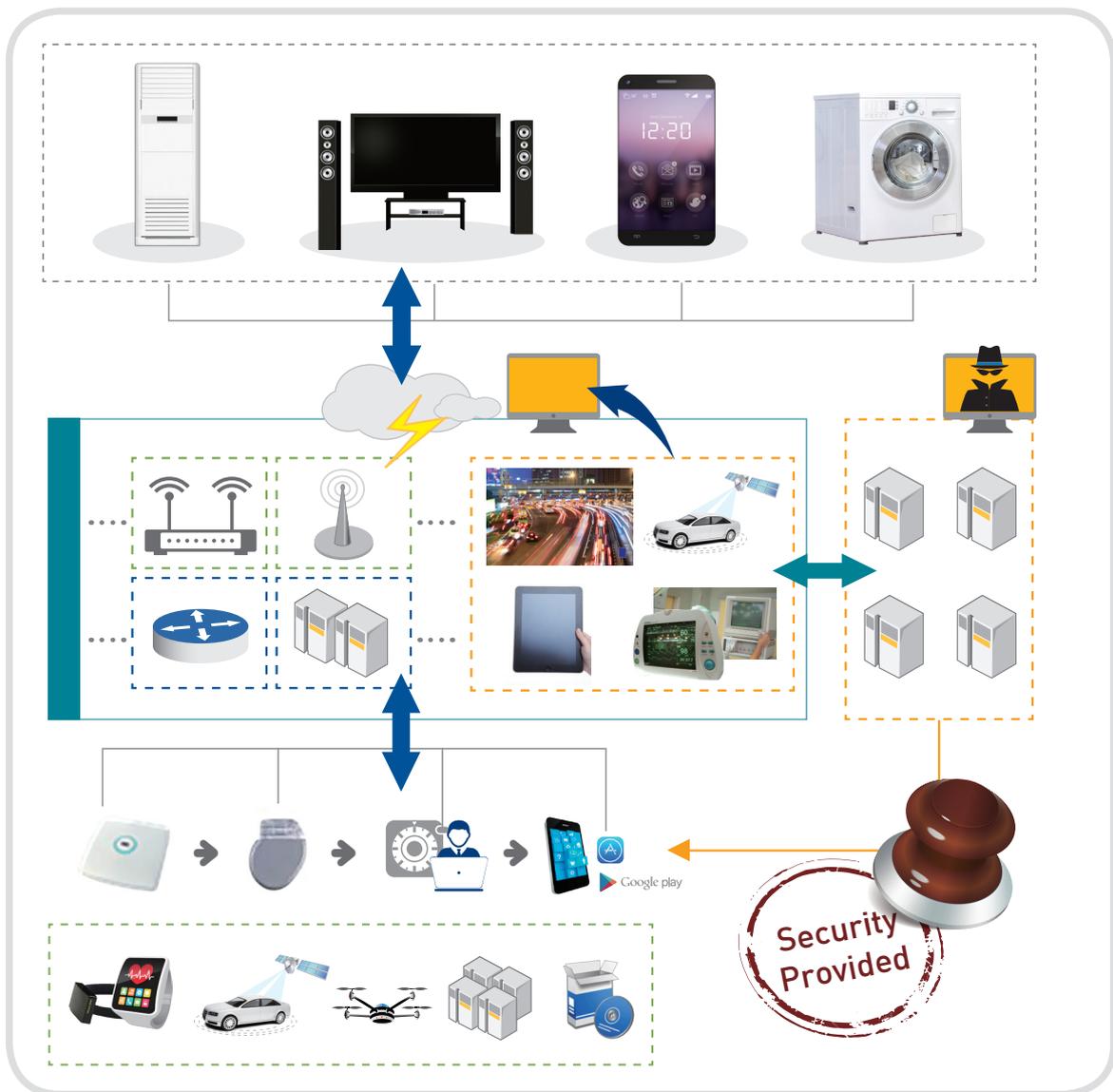
• 사용자 정보 취득·사용·폐기의 전주기 정보의 보호 및 프라이버시 관리

IoT 장치를 통해 다량의 개인정보가 수집·저장·전송될 수 있으며, 개인정보가 유출될 경우 심각한 프라이버시 침해 문제가 발생할 수 있다. 따라서 최소한의 개인정보만 수집·활용될 수 있도록 개인정보보호정책을 수립해야 한다. 개인정보보호정책 수립 시에는 빅데이터 분석과정에서 특정 개인을 식별할 수 있는 새로운 개인정보가 생성·유통될 수 있기 때문에 이를 적절히 통제할 수 있는 기술적·관리적 보호조치도 포함되어야 한다.

IoT 제품·서비스의 설계 및 개발이 완료되었다면 설계 시 수립된 보안위험 분석을 기반으로 안전한 운영과 관리를 위한 보안대책과 기술적 방안이 마련되어야 한다. 또한

IoT 서비스의 운영 과정에 대한 안전한 정보보호 및 프라이버시 관리체계와 기술적 방안이 마련되어야 한다.

정보보호 관리체계는 IoT 서비스를 위한 유·무형 자산과 이에 대한 위험 식별, IoT 장치의 비인가 접근 및 도난·분실을 방지하기 위한 물리적 접근통제, 침해사고 발생 시 서비스 연속성이 유지될 수 있도록 백업 및 복구 절차 수립 등을 포함하고 있어야 한다. 아울러 설치·배포된 IoT 장치의 주기적인 보안 업데이트, 패치 적용, 폐기절차 등 사후관리 방안 등이 포함되어야 한다^(2, 8).



7 IoT 침해사고 대응체계 및 책임추적성 확보 방안 마련

• 보안 사고에 대비한 침입탐지와 사고 시 분석 및 책임추적성 확보

사이버상의 해킹 기법은 점점 지능적이고 복합적인 기술로 발전하고 있어 대응체계를 수립하고 사고의 원인과 책임을 분석하는 것이 어려워지고 있다. 주변의 여러 사물들이 인터넷에 접속되고 사용자의 관여가 최소화되면서 기존에는 없었던 다양한 서비스가 제공 되는 IoT 환경에서는 해킹 기법의 복잡도 또한 더욱 커질 것으로 예측된다.

IoT 서비스는 다양한 유형의 IoT 장치, 유·무선 네트워크 장비, 플랫폼 등으로 구성되며, 각 영역에서 발생 가능한 보안 침해사고에 대비하여 침입탐지 및 모니터링이 수행되어야 한다.

아울러, 침해사고 발생 이후 원인분석 및 책임추적성 확보를 위해 로그기록을 주기적으로 안전하게 저장·관리해야 한다.

단, 저전력·경량형 하드웨어 사양 및 운영체제가 탑재된 IoT 장치의 경우, 그 특성상 로그기록의 생성·보관이 어려울 수 있으므로, 이런 경우에는 서비스 운영·관리시스템에서 IoT 장치의 상태정보를 주기적으로 안전하게 기록·저장할 수 있어야 한다⁹⁾.

참고문헌

- (1) Information and Privacy Commissioner, Privacy and Security by Design: An Enterprise Architecture Approach, (캐나다 백서)
- (2) Security Considerations in the IP-based Internet of Things, IETF(Internet Engineering Task Force, <http://www.ietf.org>)
- (3) 시큐어코딩(C, Java) 가이드, (행자부, 2014)
- (4) OWASP 시큐어 코딩 규칙 참고 가이드, (OWASP Korea 챗터, 2011년)
- (5) OWASP Internet of Things Top Ten, (<https://www.owasp.org>)
- (6) 취약점 검색 사이트, (예, <http://securityfocus.com>, <http://cve.mitre.org>, <http://cwe.mitre.org>, <http://www.owasp.org>)
- (7) S/W 업데이트 체계 보안 가이드라인
- (8) 정보보호 관리체계(ISMS) 인증제도 안내서, 개인정보보호관리체계(PIMS) 인증준비 안내서, <http://isms.kisa.or.kr>
- (9) 침해사고 분석 절차 안내서, <http://www.kisa.or.kr>
- (10) CoAP(Constrained Application Protocol), IETF(Internet Engineering Task Force, <http://www.ietf.org>)
- (11) MQTT(Message Queuing Telemetry Transport), OASIS (Organization for the Advancement of Structured Information Standards, <http://oasis-open.org/>)
- (12) oneM2M Specification Release 1 (<http://www.onem2m.org>)
- (13) OpenSSL 공개 보안 라이브러리 (<https://openssl.org>)
- (14) ISO/IEC, ISO/IEC 27034-1 — Application security - Part 1: Guideline for application security



IoT Internet of Trust 보안얼라이언스

