캡스톤 디자인, 2016/11/01

임베디드 오픈 플랫폼 기반의 네트워크를 통한 펌웨어 무결성 검증 도구

상명대학교 컴퓨터소프트웨어공학과

201321301 문지연

201321312 위사랑

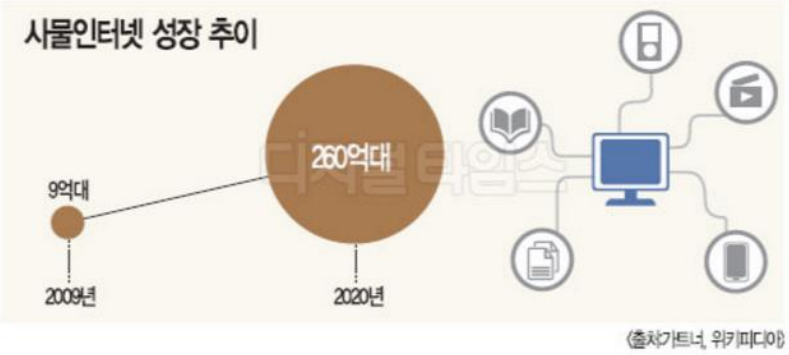
지도교수 이종혁

목 차

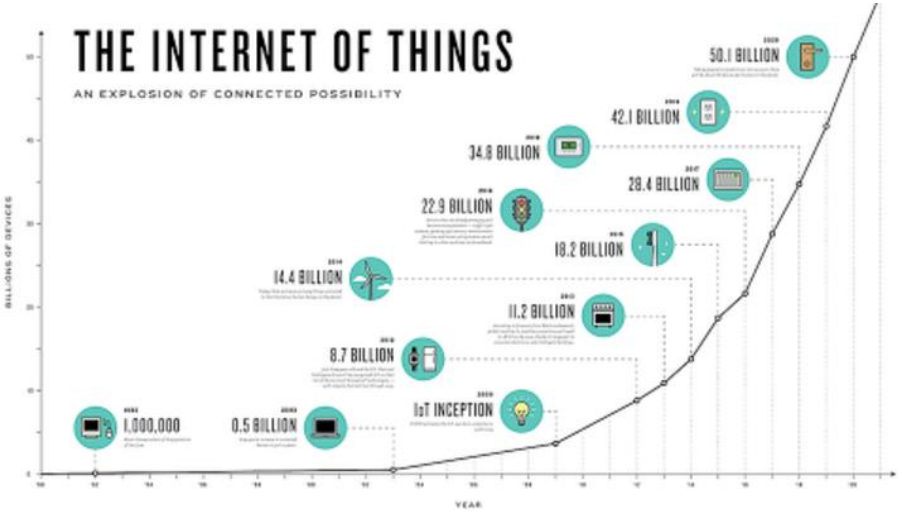
개 요

- 배경
  - 사물인터넷 시대가 도래하면서 운영체제나 응용 레벨에서 탐지가 불가능한 펌웨어 변조의 위험성 증가

→ 펌웨어 무결성 검증 필요



출처:가트너, 위키피디아



개 요

- 배경
  - 펌웨어 해킹 사례
    - 공유기
      - 2012년, IPTIME 유·무선 공유기 펌웨어 해킹 사례 발표[1]
      - 공격자가 사용자의 허가 없이 네트워크 조작 및 악성코드 유포
    - USB
      - 블랙햇 2014 에서 “BadUSB” 라는 이름으로 해킹사례 발표[2]
      - 공격자가 특정 USB장치를 조작하여 해킹용 도구로 만들어 공격

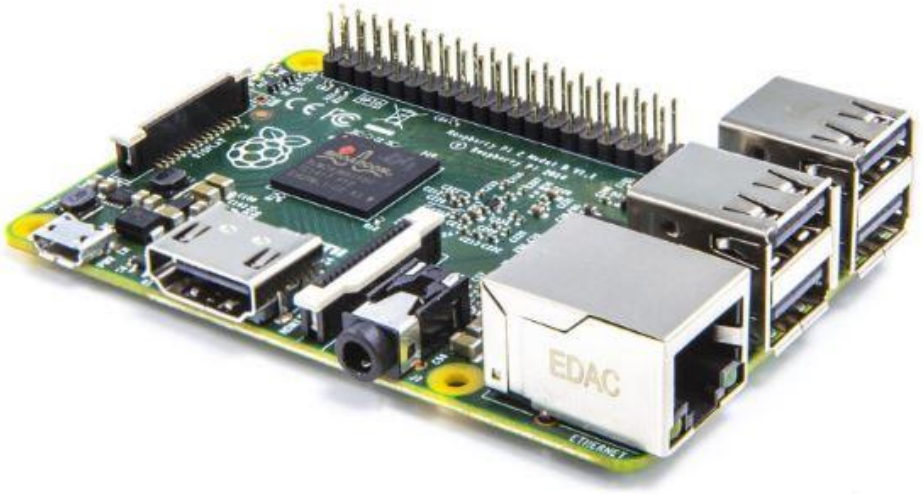
[단독]IP타임 **공유기**에서 치명적 보안 결함 의혹
CNet Korea - 2015. 7. 21.
유무선**공유기** 브랜드 'IP타임' 제품에 외부 침입자가 접근해 **펌웨어**를 바꿔 ... 제품의 **펌웨어** 상태를 주기적으로 확인하고 **해킹** 여부를 확인해 주는 ...

[1] http://returnaddr.org/b0d/view.php?id=mydoc_secudoc&no=6
[2] Karsten Nohl and Jakob Lell, “BadUSB - on accessories that turn evil,” Black Hat USA, Aug. 2014

개 요

- 타겟 디바이스
  - Raspberry Pi 2

    “컴퓨터 교육 및 취미 활동 증진을 위해 만들어진 ARM 리눅스 기반 싱글 보드 컴퓨터”



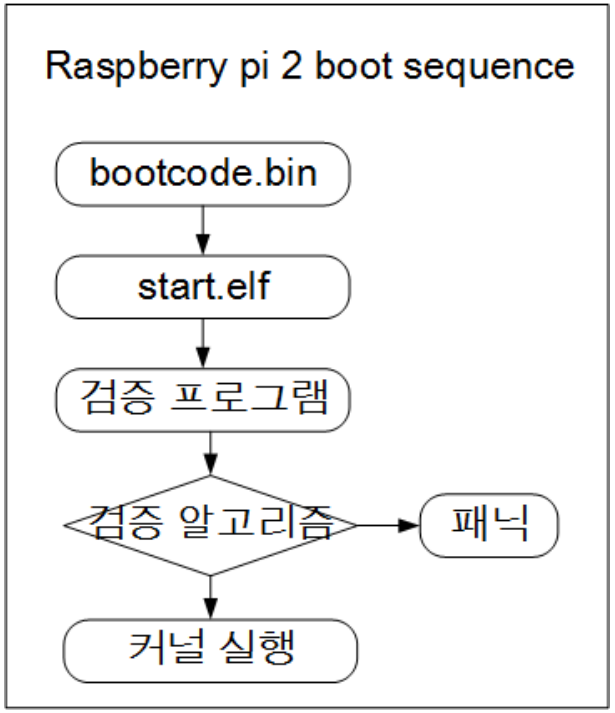
개 요

- 목표
  - 기존의 검증 도구 보완
  - 네트워크를 통한 펌웨어 무결성 검증 도구 개발

기존의 검증 도구

- 운영체제 실행 전에 동작하여, 어플리케이션 레벨에서 탐지가 불가능한 펌웨어의 변조사실을 확인하고 검증

- 부트 프로시저



	기존의 검증 도구
검증 시점	운영체제 부팅 전
검증 도구에 사용하는 알고리즘	**CRC**
검증 대상	펌웨어 파일 • bootcode.bin : 첫번째 부트로더 • start.elf : 두번째 부트로더 • kernel7.img : 리눅스 커널 • kernel.img : 보조 커널
검증 도구 구현	U-boot
순정 파일의 위치	**플래시 메모리 내 존재**

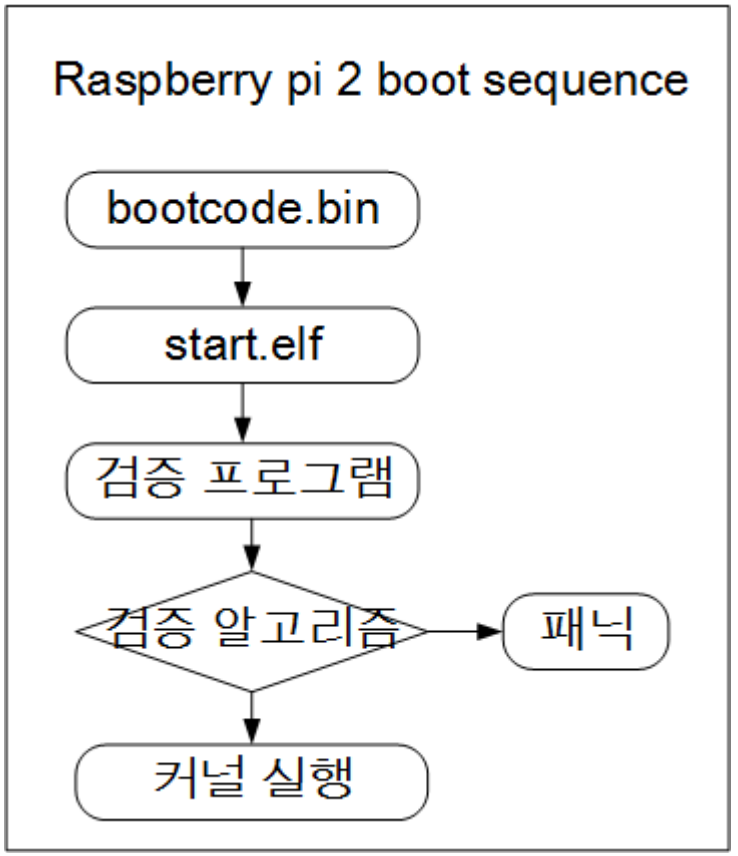
네트워크를 통한 펌웨어 무결성 검증 도구

- 기존의 검증 도구 보완
  - **CRC 알고리즘은 암호학적으로 약함**
    - 서명 알고리즘으로 ECDSA 알고리즘 선정
      - 공개키 서명 알고리즘은 RSA 서명 ,DSA, ECDSA 알고리즘이 존재
      - ECDSA는 안전도는 키 길이의 증가에 따라 거의 지수 함수적으로 증가하므로 키 길이가 제일 작음

- 추가 기능
  - **기기 내부의 순정 값의 변조 위험 가능성이 존재**
    - 네트워크를 통한 검증(TFTP) 절차 구현을 통해 해결

네트워크를 통한 펌웨어 무결성 검증 도구

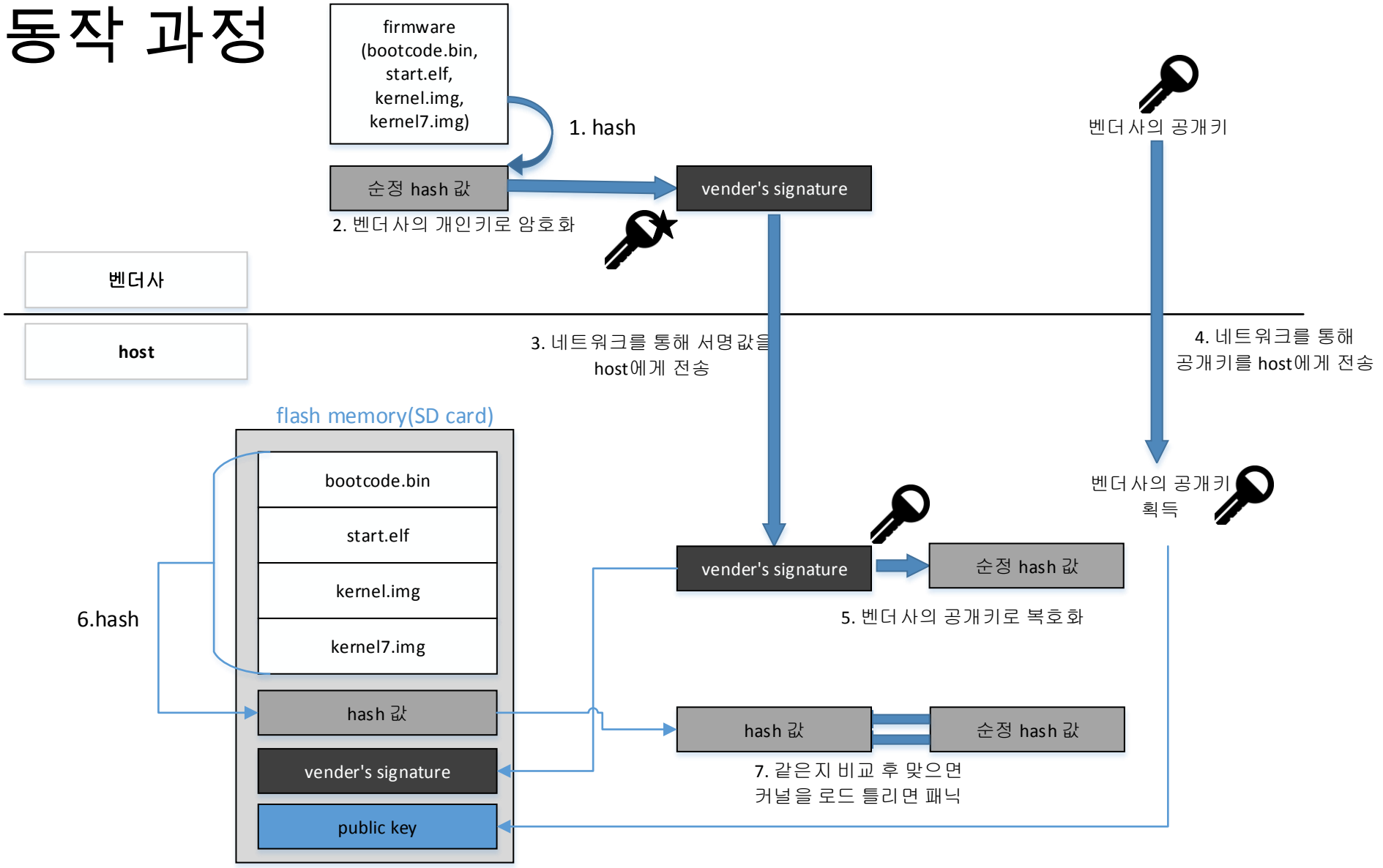
- 운영체제 실행 전에 동작하여, 어플리케이션 레벨에서 탐지가 불가능한 펌웨어의 변조사실을 확인하고 검증
- 부트 프로시저



	제안하는 검증 도구
검증 시점	운영체제 부팅 전
검증 도구에 사용하는 알고리즘	**SHA1, ECDSA**
검증 대상	펌웨어 파일 • bootcode.bin : 첫번째 부트로더 • start.elf : 두번째 부트로더 • kernel7.img : 리눅스 커널 • kernel.img : 보조 커널
검증 도구 구현	U-boot
순정 파일의 위치	**네트워크를 통해 전달받음**

네트워크를 통한 펌웨어 무결성 검증 도구

- 동작 과정



네트워크를 통한 펌웨어 무결성 검증 도구

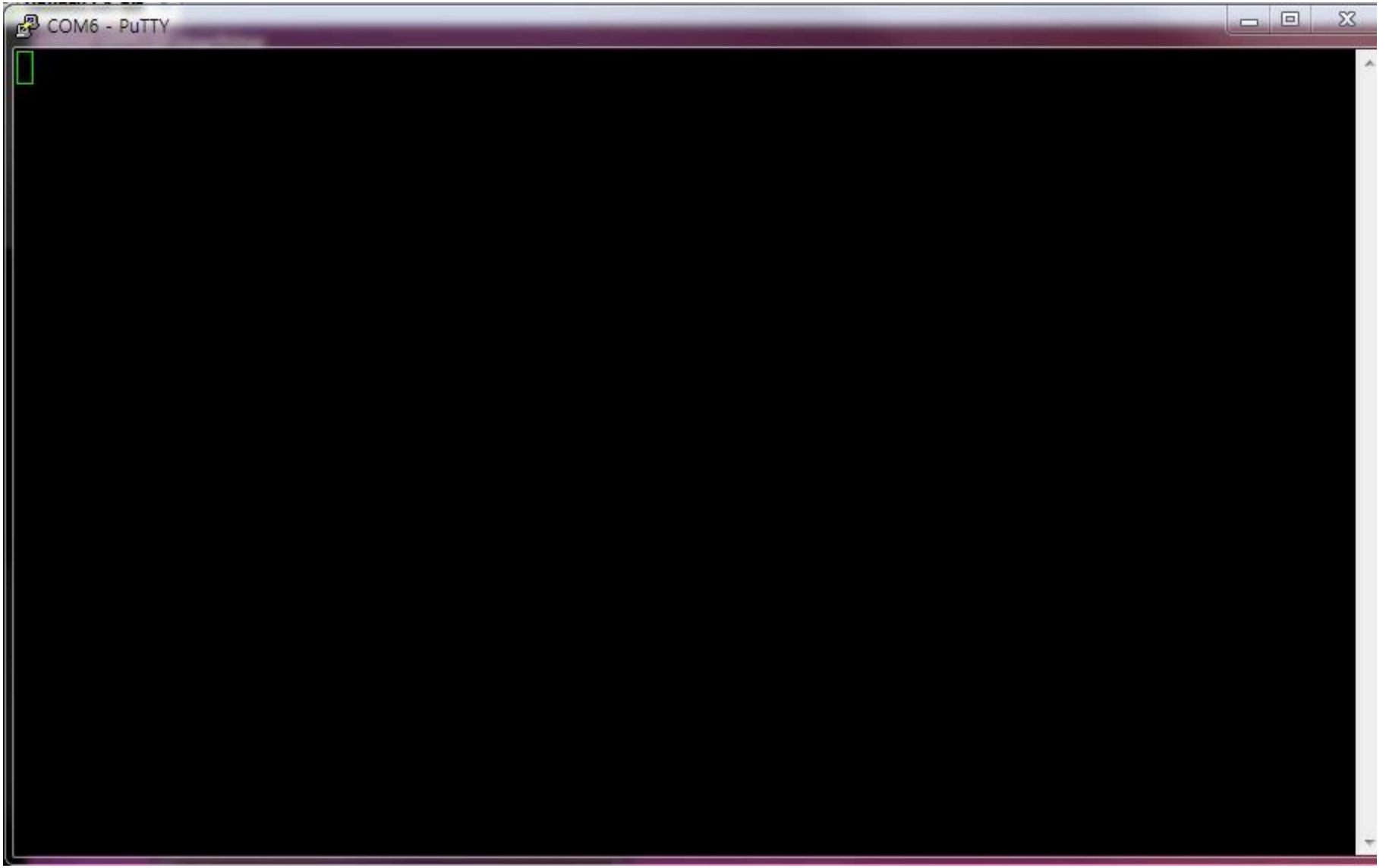
- 결론
  - 기존의 검증도구와 제안하는 검증도구의 비교

	기존의 검증도구	제안하는 검증도구
검증 시점	운영체제 부팅 전	운영체제 부팅 전
검증도구에 사용하는 알고리즘	**CRC**	**SHA1, ECDSA**
검증 대상	플래시메모리 내의 펌웨어 파일	플래시메모리 내의 펌웨어 파일
검증 도구 구현	U-boot	U-boot
순정 파일의 위치	**플래시 메모리 내 존재**	**네트워크를 통해 전달받음**

네트워크를 통한 펌웨어 무결성 검증 도구

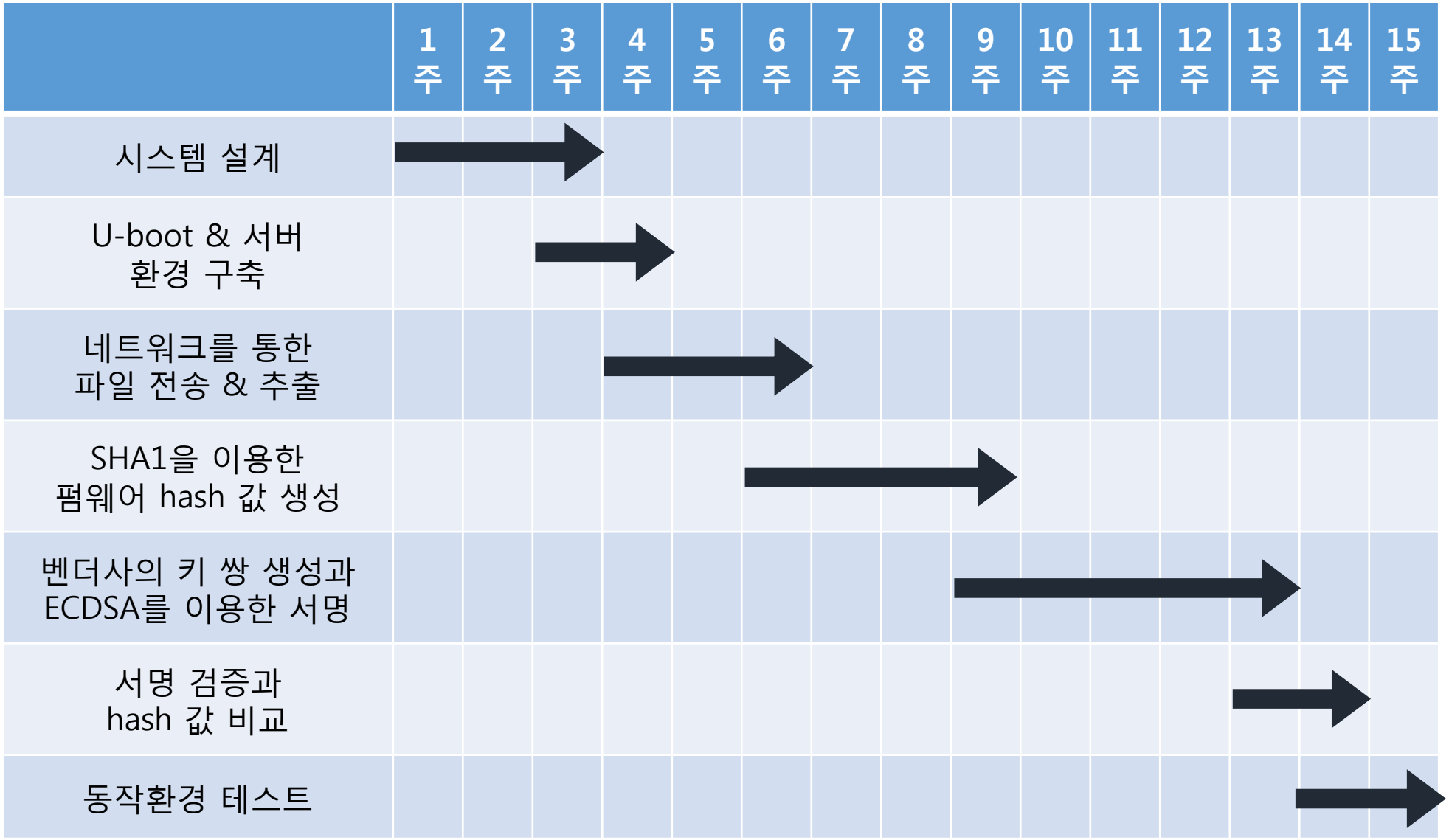
- 결론
  - 기대효과
    - 펌웨어의 보안성 강화
      - 펌웨어는 현재 대부분의 전자기기에서 사용
        (ex. PC, 스마트폰, 공유기, 프린터, 라우터, USB 등)
      - 부팅 시 자동으로 펌웨어 무결성 검증 가능
      - 외부에서 안전하게 적재된 순정 펌웨어 파일을 통해 무결성을 검증하므로 기존 검증 도구와 비교하여 보안성 증대
  - 활용방안
    - 펌웨어가 존재하는 임베디드 기기들에 대한 안정성을 검증

시연 영상



Q & A

개발 일정


1 주
2 주
3 주
4 주
5 주
6 주
7 주
8 주
9 주
10 주
11 주
12 주
13 주
14 주
15 주
시스템 설계
U-boot & 서버 환경 구축
네트워크를 통한 파일 전송 & 추출
SHA1을 이용한 펌웨어 hash 값 생성
벤더사의 키 쌍 생성과 ECDSA를 이용한 서명
서명 검증과 hash 값 비교
동작환경 테스트

개발 환경

- 벤더사
  - PC
    - OS : Ubuntu 16.04
    - CPU : x86-64
- Host
  - PC
    - OS : Ubuntu 16.04
    - CPU : x86-64
    - U-boot : 2016.09-rc2
  - Raspberry Pi 2(타겟 디바이스)
    - OS : Raspbian jessie
    - CPU : ARM