SK 쉴더스

EQST 그룹이 제안하는

IoT 진단 가이드

목 차

1. 문서개요

본 문서는 SK 쉴더스에서 IoT 진단 시 활용 가능한 수준의 점검 내용을 다루는 것을 목표로 하였습니다.

최근 국내 아파트 단지의 월패드 카메라가 해킹되면서 사생활이 노출되는 사고가 발생했습니다. 또한 5G 를 활용한 IoT 기반 서비스가 확대되면서, 스마트 홈, 스마트 공장, 의료 등 전방위 산업군의 영역에 IoT 가 적용되어 관련 영역에 대한 공격이 증가하고 있습니다. 앞으로도 IoT 기기 및 서비스를 대상으로 정보탈취나 악성코드 유포지 활용, 원격제어 공격이 증가할 것으로 예상되고 있습니다.

본 가이드에는 IoT 디바이스, 근거리 무선통신을 대상으로 한 점검 항목, 점검 예시, 조치 방안이 포함되어 있으며, 점검 예시의 경우 실제(테스트 기기 한정) 분석을 통해 시도한 내역을 포함하여 과정에 대한 이해도를 높이기 위해 노력하였습니다. 조치 방안의 경우 요약하여 다루었으며 다소 범용적인 내용이 포함 되어있습니다.

본 가이드를 통해 IoT 제품의 제조/개발/서비스화를 수행하고 있는 기업에 보안적 사전 대응에 도움이 되길 바랍니다.

2. 점검 항목

점검 항목은 디바이스, 근거리 무선통신, 웹/모바일 점검 항목으로 분류된다. 디바이스는 하드웨어 및 펌웨어 점검 항목, 근거리 무선통신은 NFC, MQTT 의 각 통신 별 특화항목으로 구성된다. 웹/모바일 점검 항목은 따로 첨부되어 있지 않으며, 주요정보통신기반시설 웹/모바일 점검 항목을 기준으로 기기 및 연결된 서비스 점검을 수행한다.

2.1. 디바이스 점검 항목

디바이스 점검 항목은 아래와 같다.

순번	항목 명	항목 설명	비고
1	[DV-001] 물리적 인터페이스 존재 여부	· [인터페이스 존재 유무에 대한 확인] · 디바이스 내/외를 확인하여, 펌웨어 추출 및 정보 획득(Shell 혹은 Bootloader에 접속)에 활용할 수 있는 물리적 포트 존재 여부 확인	-
2	[DV-002] 분해 확인 매커니즘 적용 여부	· [분해 확인 매커니즘 적용 여부 확인] · 디바이스 내부 분석을 위해 분해를 시도하는 경우 이를 식별하기 위한 대비책이 마련되어 있는지 확인	-
3	[DV-003] 펌웨어 추출 가능 여부	· [펌웨어 추출 가능 여부 확인] · 펌웨어를 추출하거나 Device에 존재하는 Filesystem의 추출 가능 여부 확인	-
4	[DV-004] 펌웨어 변조 적용 가능 여부	· [변조된 펌웨어의 적용 가능 여부] · 펌웨어 변조 가능성이 존재하는 기능들을 점검하여 펌웨어 및 Filesystem의 무결성 검증 여부 확인	-
5	[DV-005] 설정 미흡 여부	· [설정 미흡으로 판단되는 취약점들의 존재 유무 확인] · MCU 보안 옵션, 구동되는 서비스들에 대한 보안 설정 적용 여부 확인	-
6	[DV-006] 불필요한 네트워크 서비스 존재 여부	· [불필요한 서비스의 존재 유무 확인] · Device에서 불필요한 서비스의 구동 여부 확인	-
7	[DV-007] 취약한 계정 정보 사용 여부	· [취약한 계정 사용 여부 확인] · Device 관리 및 제어에 취약한 계정 이용 여부 확인	-
8	[DV-008] 중요 정보 출력 여부	· [각종 출력 정보에 중요 정보의 포함 여부 확인] · Device에서 발생하는 출력(인터페이스, 실시간 로그, 저장된 로그)을 통해 중요 정보의 출력 여부 확인	-
9	[DV-009] 중요 정보 평문 저장 여부	· [중요 파일들에 대한 평문 저장 여부 확인] · Device의 저장장치 혹은 펌웨어 내 중요 설정파일, 암호키, 인증정보 등에 대한 평문 저장 여부 확인	-
10	[DV-010] 백업 및 테스트파일 존재 여부	· [백업 및 테스트 파일 존재 유무 확인] · Device 내 제공되는 서비스 혹은 Filesystem 내 백업 및 테스트 파일의 잔존 여부 확인	-
11	[DV-011] 전송 구간 보호 여부 (Device)	· [통신 구간에 대한 평문 전송, 중요정보 노출 여부 확인] · 일반적인 WIFI(무선) 통신 구간에 대해 SSL 적용 여부와 중요 정보 포함 여부 확인	-

12	[DV-012] 기타	· [진단 항목에 정의되지 않은 취약점 존재 유무 확인] · 새로운 공격을 포함한 진단 Device에 해당하는 기타 취약점 존재 여부 확인	-

표 1. 디바이스 점검 항목

2.2. 근거리 무선통신 점검 항목

근거리 무선통신 점검 항목은 다음과 같다.

순번	항목 명	항목 설명	비고
1	[MQ-001] 불필요한 토픽 접근	· [MQTT 통신 시 불필요한 토픽 접근 가능 여부 확인] · 서비스 외 토픽 접근이 가능한지 확인하여, 권한 없는 사용자의 토픽 접근 및 이용이 가능 여부 확인	MQTT
2	[MQ-002] 디폴트 계정 사용 여부	· [MQTT 통신 연결 시 디폴트 계정 사용 가능 여부 확인] · MQTT Broker에 디폴트로 생성되는 계정 사용이 가능하여, 접근 권한 없는 사용자의 디폴트 계정을 통한 서비스 접근 및 이용 가능 여부 확인	MQTT
3	[MQ-003] 전송 구간 보호 여부	· [MQTT 통신 구간에 대한 평문 전송, 중요 정보 노출 여부 확인] · 통신 구간 암호화 적용이 미흡하여 중요 정보 노출 및 노출된 중요 정보 악용 가능 여부 확인	MQTT
4	[NFC-001] 카드 데이터 복제 가능 여부	· [카드 데이터 복제 가능 여부] · 보안성이 낮은 태그/카드를 사용하여 데이터 복제 및 복제한 태그/카드 사용 가능 여부 확인	NFC
5	[NFC-002] 카드 데이터 변조 가능 여부	· [카드 데이터 변조 가능 여부] · 태그/카드의 UUID, 데이터 값 변조 및 변조한 태그/카드 사용 가능 여부 확인	NFC
6	[NFC-003] 취약한 암호화 키 관리 여부	· [취약한 암호화 키 사용 여부] · 알려진 키 또는 취약한 키 사용으로 인한 키 크랙 가능 여부 확인	NFC

표 2. 근거리 무선통신 점검 항목

3. 디바이스 점검 상세

3.1. [DV-001] 물리적 인터페이스 존재 여부

구분	내용
전제조건	· Device 분해/개조
취약점 설명	· [인터페이스 존재 유무 확인] · Device 내/외를 확인하여, 펌웨어 추출 및 정보 획득(Shell 혹은 Bootloader에 접속)에 활용할 수 있는 물리적 포트가 존재 할 경우 디바이스 내 시스템 정보 노출 혹은 펌웨어 Dump 시도 등의 가능성이 존재하는 취약점
판단 기준	· 하기와 같은 Port들을 육안으로 식별하고 취약점 분석에 활용 가능성이 있는 경우 취약 (가능성 있는 포트 확인) [외부] Device 외부에 USB, RS232, Ethernet, SDcard 와 같은 단자 [내부] Device를 분해하여 UART, JTAG/SWD, I2C, SPI, 사전 구성된 회로 (usb, SDcard) 가 있을 경우 ※ PCB 내 주요 Chip의 DataSheet를 확인하고 활용 가능한 Pin들을 Direct로 연결하여 펌웨어 추출이 가능한 지 여부는 테스트가 필요하지만 **취약점으로 포함되지 않음** ※ BGA 형태의 경우 ChipOff (Desoldering)가 필요함. **기판 내 포트들에 대한 전압 체크와 로직 분석기 활용을 통해 Debug에 활용 가능한 포트가 없다면 양호로 판단 가능함.** (실제 Desoldering 작업에는 공수 부족 및 장비 마련, 분석 대상의 고장, A/S불가 등 다양한 문제가 발생할 수 있음)
취약점 영향력	· Shell 접근을 통해 펌웨어 Dump 하고 분석하여 취약점 분석에 활용 · Filesystem에 저장된 주요 파일에서 중요 정보를 획득 · 변조된 펌웨어로 임의 업데이트를 수행하여 기기를 완전하게 제어할 가능성이 있음
보안대책	· 내/외 불필요한 물리적 인터페이스 제거 · MCU에서 지원하는 경우 보안 기능을 사용하여 위협을 최소화 · 전용 프로그램 등을 통해 접근할 수 있도록 구현

표 3. 물리적 인터페이스 존재

3.1.1. 참고 사항

[주의사항]

- 사전 협의되지 않는 이상 제품의 임의 분해 시 A/S 를 받기는 사실상 불가능하다.
- 납땜 수행 시 주변 케이블 및 부품을 건드리면 안되므로 정리 혹은 테이핑 후 안전한 상태에서 진행한다.

※ 잘 알려진 Device 의 경우 분해 전 검색 엔진을 통해 해체, 분해 관련 키워드 (Teardown, Disassemble)등으로 사전 검색하여 정보를 수집을 시도하는 것이 좋다.

[준비물]

* 제품을 분해하는데 있어서는 다음과 같은 기본 공구가 필요하다.
  - '+', '-', '*' 드라이버 (사이즈는 각기 다름)
  - 플라스틱 오프닝 도구 (약한 강도의 분해)
  - 메탈 오프닝 도구 (힘을 가할 필요가 있을 경우)
  - 핀셋 (일반, 절연)
* 납땜을 수행하는데 아래의 공구 및 자재들이 사용된다.

- 멀티테스터 (전압 확인, 통전 확인)
- 납땜용품 (납(유/무연), 일반 점퍼용 단선, 납땜 기구)

* UART 연결을 위해 다음의 제품들이 사용된다.
- FTDI - FT232 (3.3v / Normal USB Type)

3.1.1.1. 외부 인터페이스 확인

Device 외부에서 확인 가능한 Interface 는 대표적으로 USB, RS232, Ethernet (RJ45) , SDcard 가 있다.

USB Type(Male, Female) RS232 SDcard , AUX RJ45 (Ethernet)

그림 1. 외부 단자 예시

Device 종류에 따라서는 AUX, Wired(RJ42) 단자 Smart TV 의 경우 HDMI 단자, 등 유지보수 중 Debug 목적으로 활용 가능한 Port 가 존재하므로 외부에 노출 Port 들에 대한 식별이 우선적으로 필요하다.

그림 2. USB 포트가 외부에 노출된 예시

외부 Interface 의 확인은 쉽게 가능하며 Device 의 표면을 살펴보면 된다.
상단의 Device 는 USB-TypeA (Female) 포트가 있었으며 매뉴얼 확인 시 충전용으로 사용되는 것을 확인할 수 있었다.

※ USB Port 의 경우 매뉴얼에 존재하지 않지만 버튼 조합을 통해 숨겨진 mode 로 변경될 가능성도 있으므로 기본 매뉴얼에 존재하지 않는 패턴으로 버튼을 눌러 보는 작업도 필요하다.

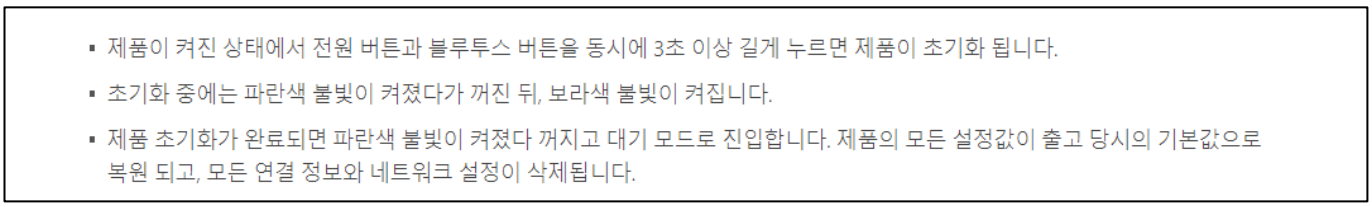


그림 3. 제품 매뉴얼의 초기화 방법 확인

단순히 해당 Port 의 유무 만으로는 취약 판단을 내릴 수는 없으며 Debug 용도로 활용 가능 여부를 추가로 확인이 필요하다. 펌웨어 변조 적용 가능(6.4 항목 취약의 경우) 시 일부 아래와 같이 특정 설정을 변경 적용하여 adb_shell을 활성화하고 외부에 노출된 USB 포트를 adb 연결에 활용할 수 있다.

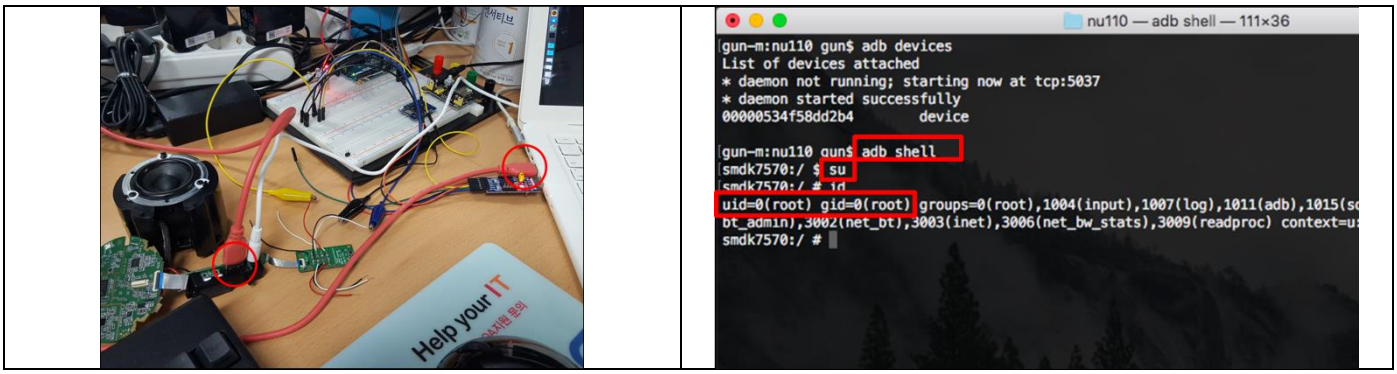


그림 4. 외부 USB 단자 활용 가능 확인 예시

3.1.1.2. 내부 인터페이스 확인

대표적인 내부 인터페이스는 아래와 같다.

No	명칭	주요 특징	주요 단자
1	UART (Universal Asynchronous Receiver/Transmitter)	1:1 비동기 통신 Baudrate : 직렬-병렬 통신 변환 속도 동기화 설정 RS232 활용하는 경우 많음	Tx : 데이터 송신에 사용 Rx : 데이터 수신에 사용
2	JTAG(Joint Test Action Group)/ SWD (Serial Wire Output)	[JTAG]10, 14, 16, 20 Pin 등 다양한 구성이 있음. [SWD] 3 핀 구성으로 디버깅 가능	TDI : 데이터 입력 TMS(SWDIO) : Mode State RTCK(SWCLK) : Clock TRST : Reset TDO(SWO) : 데이터 출력
3	I2C (Inter-Integrated Circuit) / TWI(Two Wire Interface)	1:N 동기 직렬 통신	SDA(TWD) : 데이터 전송 SCL(TWCK) : 클럭
4	SPI (Serial Peripheral Interconnect)	1:N 동기 직렬 통신	SCLK : Serial Clock - MASTER 클럭 MOSI : Master Output Slave Input. MISO : Master Input Slave Output. SS : Slave Select.

표 4. 대표 내부 인터페이스 정리

내부 인터페이스 연결 시 VDD(VCC), GND (Vee, Vss)의 연결이 반드시 필요하다. High, Low 신호를 위한 기준이 있어야 하므로 기본 2 개의 선은 연결이 반드시 필요하다. VCC 의 전압을 알 수 없는 경우에는 직접 멀티테스터를 활용하여 GND, VCC 를 찍어 (DC) 5v, 3.3v, 1.8v 등을 반드시 확인해야 한다.

분석 Device 가 외부로부터 전원을 이미 공급받고 있는 상태라면 VDD(VCC)의 연결은 별도로 수행하지 않고 GND, RX, TX 를 연결하여야 출력 확인이 가능하다.

반드시 교류(AC v220)가 연결되어 있는 회로의 근처는 납땜 및 확인 시 주의해야 한다. 쇼트가 발생하면 동일 AC 전원을 사용하는 라인 전체를 담당하는 차단기가 내려가므로 반드시 주의가 필요하다. (실제 차단기가 동작하여 업무 PC 사용 불가 상황 발생 - 시설 부서 연락 후 조치 대기)

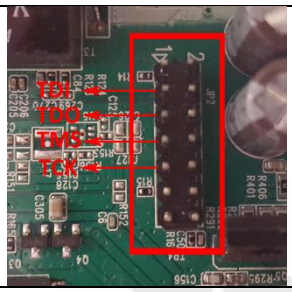
아래는 정상의 경우와 Debug 단자 근처 AC 회로와의 쇼트로 인해 PCB 가 손상된 경우를 보여준다.

정상 PCB - 스마트 플러그

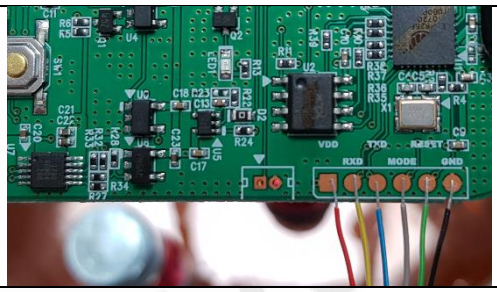
AC 쇼트로 인한 PCB 손상 - 스마트 플러그

그림 5. 내부 포트 확인 시 AC 주의

실제 참고 할 만한 내부 단자의 유형은 핀 헤더가 직접 노출되는 유형이나, 포트만 존재하는 경우이며 아래와 같다.



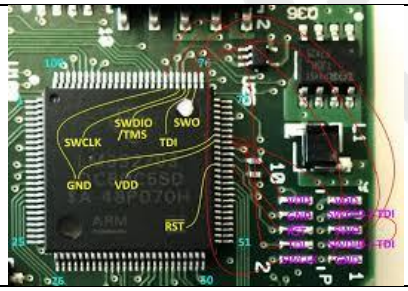
연결 가능 핀 헤더가 존재하는 경우



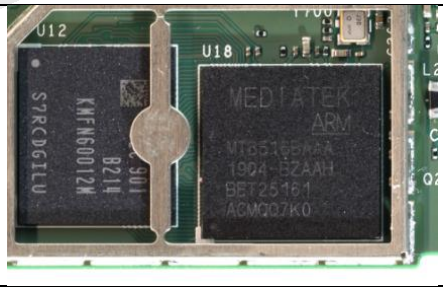
동판 포트 확인

그림 6. 내부 단자 유형

※ [참고] 취약점으로 진단할 수는 없으나 Datasheet 참고를 통해 pin 을 Direct Access 하는 경우가 있다.
BGA 방식의 MCU 를 활용하여 pin 에 접근할 방법이 없는 경우 H/W 개조가 필요하다



Pin 직접접근 (취약점 아님)



BGA 방식 (Chipoff 및 H/W 개조가 필요한 경우)

그림 7. 분석 시 경험 가능한 유형

3.1.2. 양호 Case 1 (AI 스피커)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

그림 8. Device 분해 시도

해당 제품은 분해 결과 제품의 하단에서 활용 가능한 내부 포트가 발견되었다.

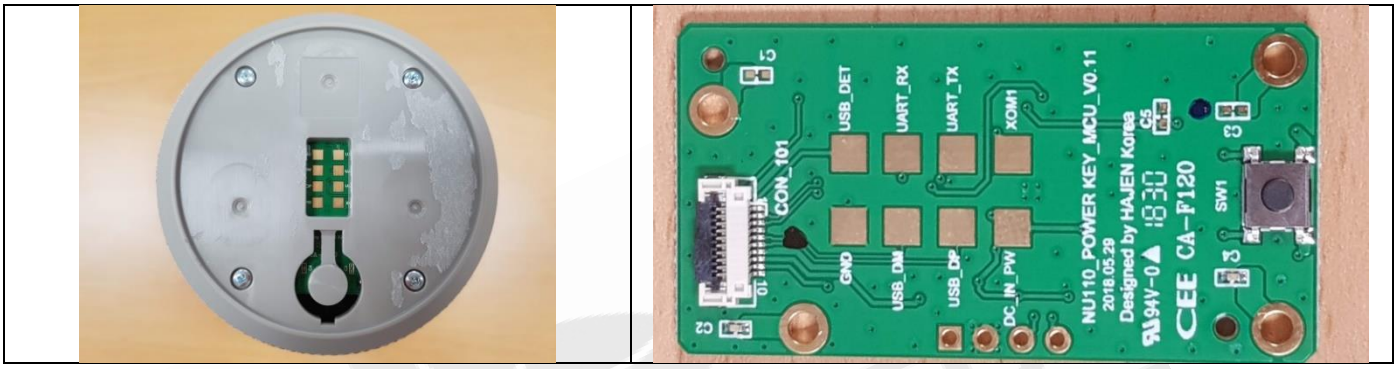
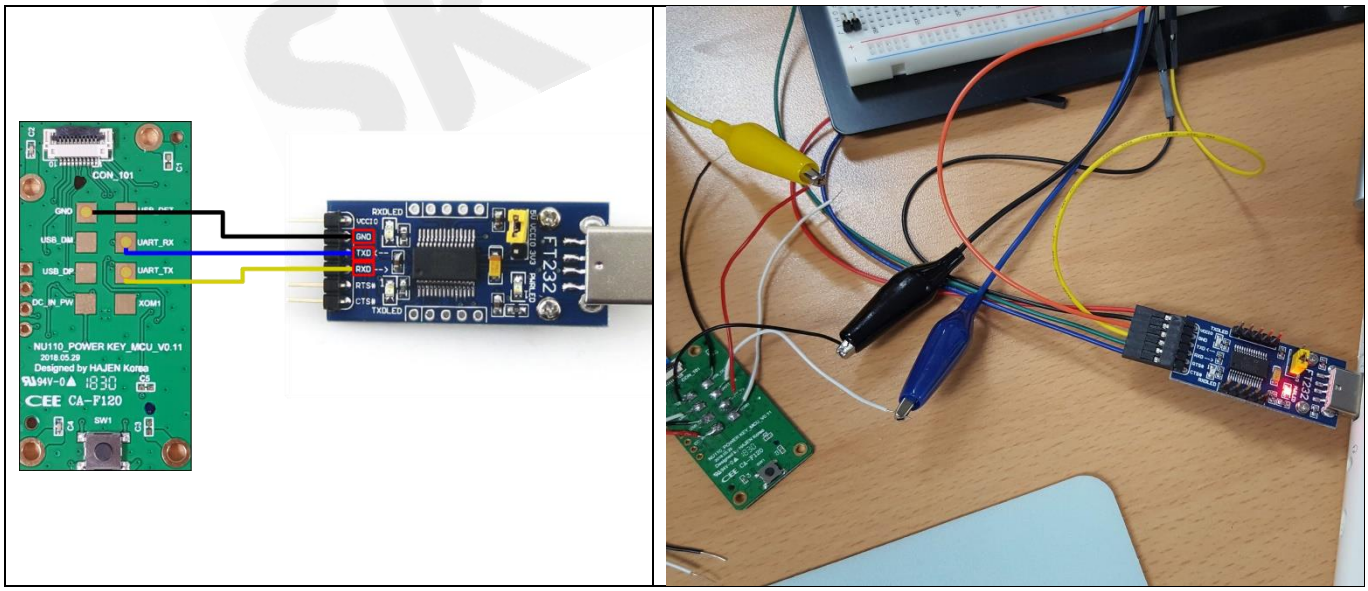


그림 9. 외부 단자 (UART) 확인 예시

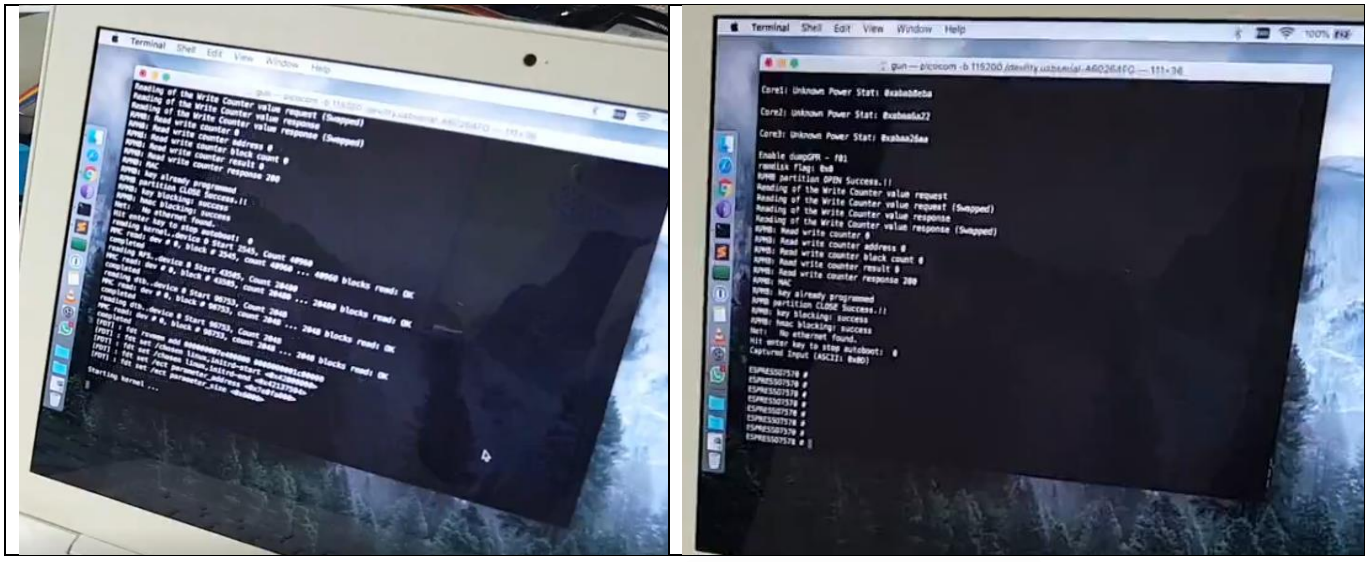
UART RX, TX 확인이 가능하며 VCC, GND 포트가 있다. VCC 전압을 확인한 결과 3.3v 였으며 포트 출력을 확인하기 위해 해당 포트를 납땜하고 PC 에 연결하였다. 일반적인 시리얼 버스의 구성은 2 가닥이며 일반적인 라인 연결과 반대이다. (RX 및 TX 는 서로 반대로 연결해야 하므로 주의가 필요하다)



UART <-> FTDI – FT232 배선 Debug 포트 납땜 및 PC 연결

그림 10. UART <-> PC 연결

터미널의 글자가 깨지는 경우 Baudrate 변경 후 연결 및 전원 인가 시도를 반복하여 수행한다.
(해당 제품에서는 115200 bps 설정 시 부팅 로그 출력이 확인되었다.)



UART TX 출력 확인

엔터(0x0D) 입력 시 쉘(u-boot) Shell 확인
[Boot Parameter 설정 가능 확인]

그림 11. BootLog 출력 및 쉘 접근 확인 (취약)

3.1.3. 양호 Case 2 (AI 스피커)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

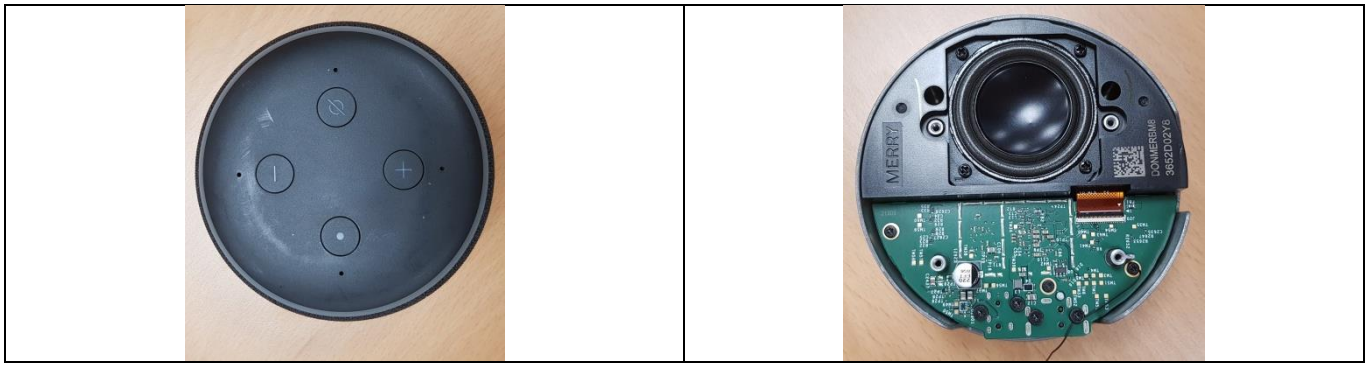
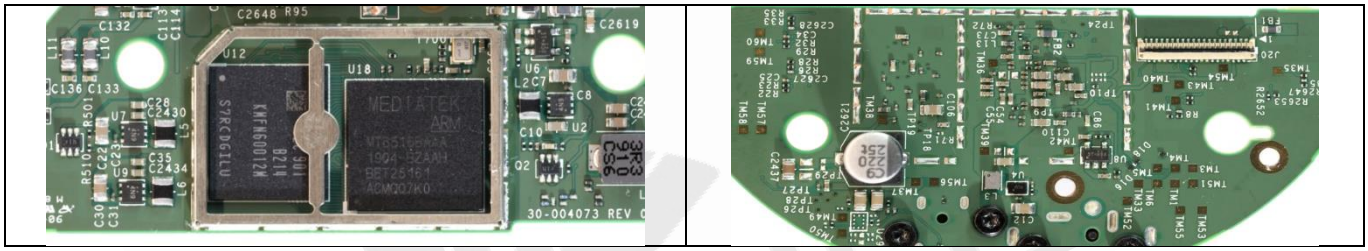


그림 12. Device 분해 시도

※ 분해 전 검색엔진을 통해 해당 제품의 분해 시도 페이지를 발견하여 참고하였다.

전면 확인 시 PCB 전면부 주요 chip 들이 BGA 방식이며 Debug 용 단자가 식별되지 않았다.



기판 내 BGA Chip 확인 (전면) 기판 내 TMxx 형태의 단자 확인 (후면)

그림 13. 주요 기판 확인

후면에 존재하는 각종 TMxx 표시된 단자들의 출력 전압 변동 확인을 진행하였으나 의심되는 변동 폭이 발견되지 않았다. (Debug 단자로 쓰일 혹은 가능성을 체크한다.)

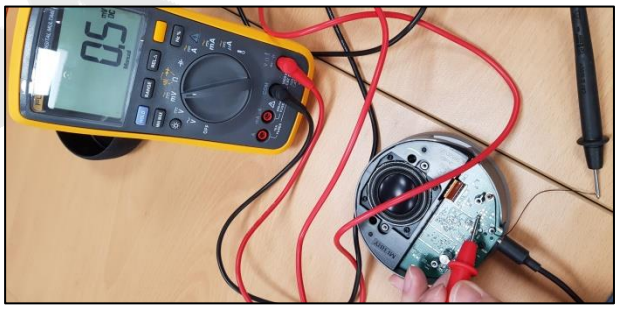


그림 14. 전압 변동 확인

※ 검색 엔진을 통해 분석 관련 외국 포스팅 참고 하여 TMxx 형태 단자들의 분석 내용 추가 확인 시 특별한 부분이 발견되지 않았다.

3.1.4. 취약 Case 3 (문 열림 센서)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

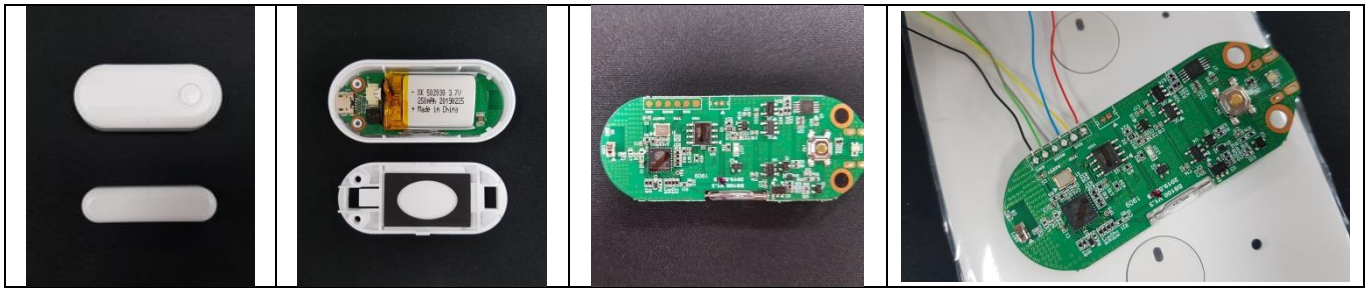


그림 15. Device 분해 및 디버그 포트 확인

PCB 를 확인하면 ESP8266EX 를 사용하며 VDD, RXD, TXD, MODE, RESET, GND 라벨 및 포트 확인이 가능하였다.

VDD 전압을 확인한 결과 3.3v 이며 포트 출력을 확인하기 위해 해당 포트를 납땜하고 PC 에 연결하여 로그 출력을 확인하였다.

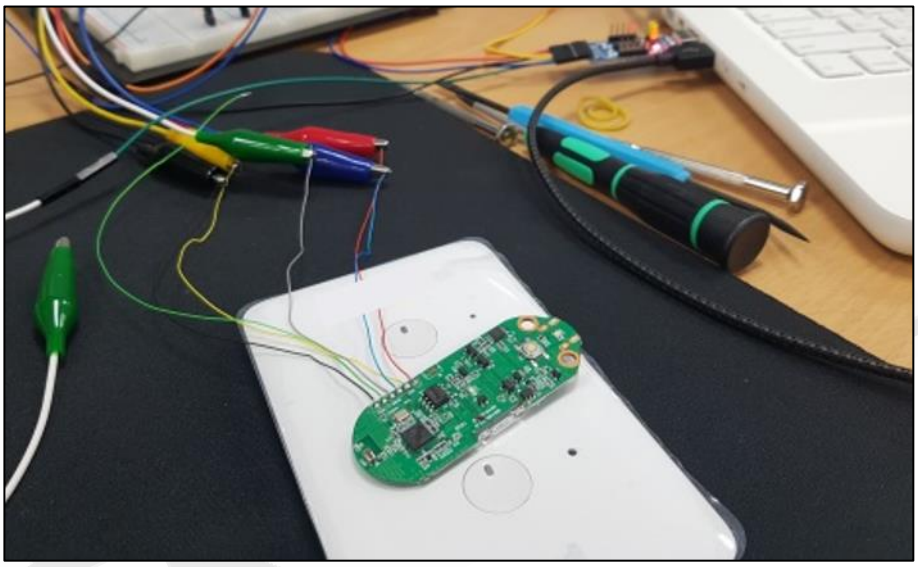


그림 16. 디버그 포트 연결

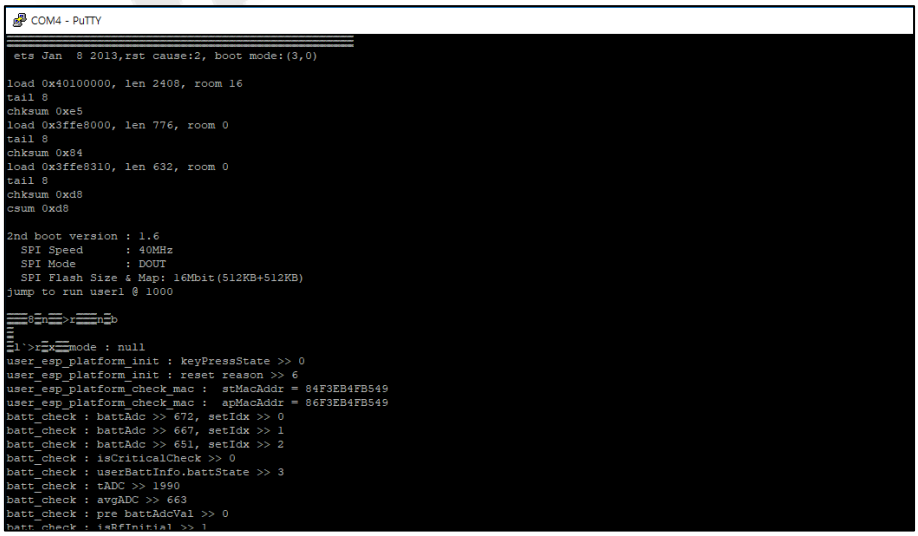


그림 17. Bootlog 출력 확인

3.2. [DV-002] 분해 확인 매커니즘 적용 여부

구분	내용
전제조건	· Device 분해
취약점 설명	· [분해 확인 매커니즘 적용 여부 확인] · Device 내부 접근을 위해 분해를 시도하여 제품의 임의 조작 및 임의 기능 삽입에 악용될 가능성이 존재하는 취약점
판단 기준	· 제품의 분해 시도가 있었는지 확인할 수 있는 방안이 적용되었는지 확인 필요함 · 하기와 같은 다양한 방법이 적용되어 있을 수 있으며 아래의 예시와 같은 방안이 적용되어 있지 않을 경우 취약 판단 - 분해 시 내부 부품이 부러지는 구조적 특징 - 특수 나사 사용 - 별도 확인용 스티커 부착 여부 (봉인 씰) · 또한 설치 시 외/내부 구분이 필요한 장비의 경우 (ex: 도어락) 하기 여부를 추가 확인하여야 함 - **[취약]** 임의의 사용자가 외부에서 Device를 분해할 수 있는 방법이 있을 경우
취약점 영향력	· 제품의 분석/개조를 통해 임의의 기능을 추가하여 악용하거나 조작할 수 있음
보안대책	· 플라스틱 구조의 변경으로 전용 공구를 통해 분리하지 않으면 내부가 부러지도록 설계 (우선 적용 필요) · 특수 나사 적용 · 스티커 적용

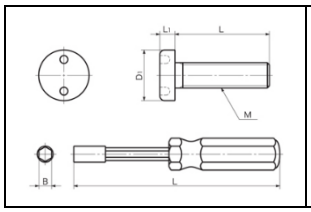
표 5. 분해여부 확인 매커니즘 적용

3.2.1. 참고 사항

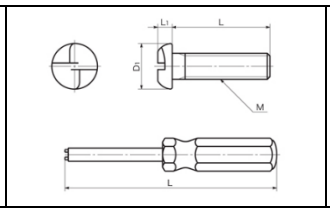
3.2.1.1. 분해 방지 방안

제품의 조립 면에 대해 플라스틱 부분의 구조적 특징을 이용하여 분해 시(A/S 등 수리 목적) 특수 공구를 사용하지 않을 경우 부러지도록 구성하는 것이 제일 우선적으로 제안이 가능하다. (구성품은 더 정밀하게 가공이 필요하다.)

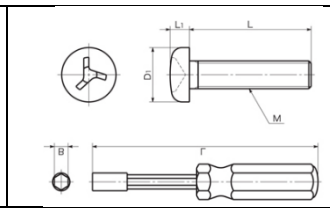
나사 부분에 적용이 가능할 수 있는 대표적인 방안에 대해 기술하였다. (해당 나사 외에도 다양한 나사들이 존재한다.)



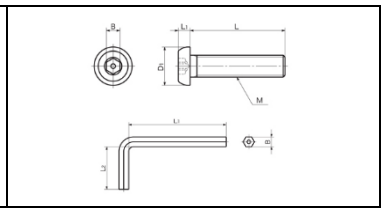
더블 홀 스크류(SRTS) 볼트 및 공구



원사이드 스크류(SROS) 볼트 및 공구



트라이 윙 스크류(SRWS) 볼트 및 공구



6 각 보턴 스크류 (핀 내장) 볼트 및 공구

봉인 라벨(씰) 제품의 유형은 아래와 같다.

그림 18. 봉인 라벨 유형

3.2.2. 취약 Case 1 (AI 스피커)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

그림 19. Device 분해 시도

분해 시도 시 일반 나사 '+' 형태 사용 중이다. 또한 봉인 라벨을 사용하거나 구조적인 특성을 활용하지 않아 재 조립이 쉽게 가능하였다.

3.2.3. 취약 Case 2 (문 열림 센서)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

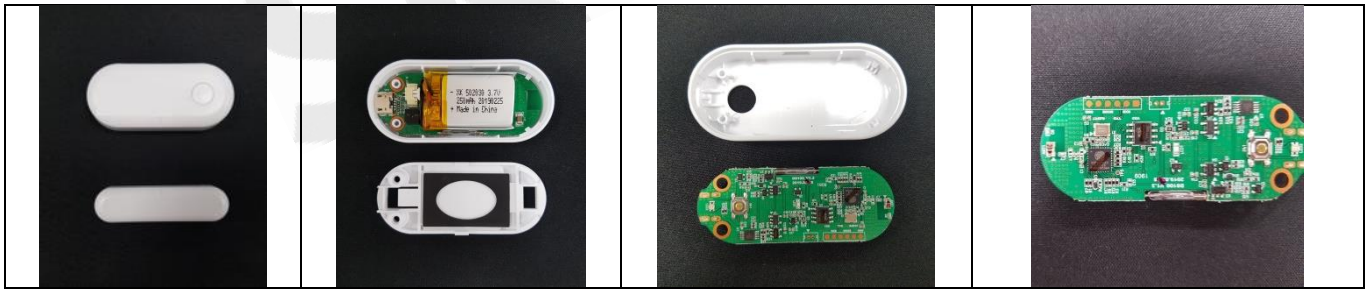


그림 20. Device 분해 시도

분해 시도 시 일반 나사 '+' 형태 사용 중이고 봉인 라벨을 사용하거나 구조적인 특성을 활용하지 않아 재 조립이 쉽게 가능하였다.

3.3. [DV-003] 펌웨어 추출 가능 여부

구분	내용
전제조건	· Device 분해/개조
취약점 설명	· [인터페이스 활용을 통한 펌웨어 추출 가능 여부 확인] · 펌웨어를 추출하거나 Device에 존재하는 Filesystem을 추출할 수 있는 가능성이 있는 취약점
판단 기준	* 펌웨어를 사전에 제공받았을 경우라도 해당 부분을 확인하여야 함 - **[취약]** Device 내부 포트를 통한 쉘 접근이 가능한 경우 중 + Boot Parameter 설정 가능 Shell을 통해 Filesystem Dump가 가능한 경우 + Device의 실제 Shell을 통해 Filesystem Dump가 가능한 경우 - [양호] 격리(chroot)된 별도의 Shell을 사용하고 있으며 이를 우회 하여 상위 경로의 파일을 엑세스 할 수 있는 방법이 없을 경우 - **[취약]** 업데이트 수행 시 네트워크 통신 구간 내에서 펌웨어 획득이 가능한 경우 (업데이트 수행 시 재 요청이 불가능 할 수 있음) · + 펌웨어 획득 시 참고 - 제조사가 제품군 별 펌웨어를 다운로드(파일 획득) 할 수 있도록 제공하고 있음 (분석 시 용이하기 위함이며 취약점은 아님) - Device 내 저장되는 로그, 설정 등에 펌웨어 다운로드 URL 존재 여부 및 직접 다운로드 · ※ Chip Off (Desoldering)을 통해 펌웨어를 추출한 경우는 분석에 도움이 될 수는 있지만 취약점에 포함하지 않음 ※ PCB 내 주요 Chip 들의 DataSheet를 확인하고 활용 가능한 핀들을 통해 펌웨어 추출이 가능한 지 여부는 테스트가 필요하지만 취약점으로 포함되지 않음
취약점 영향력	· Shell 접근을 통해 Filesystem을 Dump 하고 분석하여 취약점 분석에 활용 · Filesystem Dump를 통해 주요 설정 파일들에서 정보 획득
보안대책	· 내/외 불필요한 물리적 인터페이스 제거 · 직접적인 펌웨어 추출이 어렵도록 BGA 방식 활용 · 네트워크 구간 상 펌웨어를 획득하더라도 Filesystem 추출이 어렵도록 암호화 적용

표 6. 펌웨어 추출 가능 여부

3.3.1. 취약 Case 1 (AI 스피커)

Boot Parameter 확인을 위해 아래의 명령어를 실행하고 각 Filesystem 이 Load 되는 영역을 파악한다.

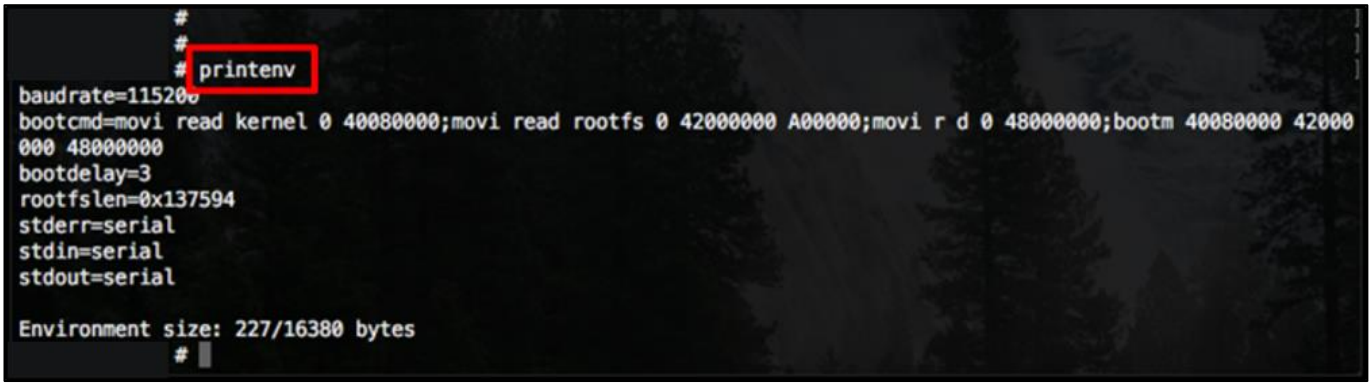


그림 21. Boot Parameter 확인

특정 영역을 Byte 단위로 읽어 로그로 저장하는 형태로 Dump 를 수행하였다.

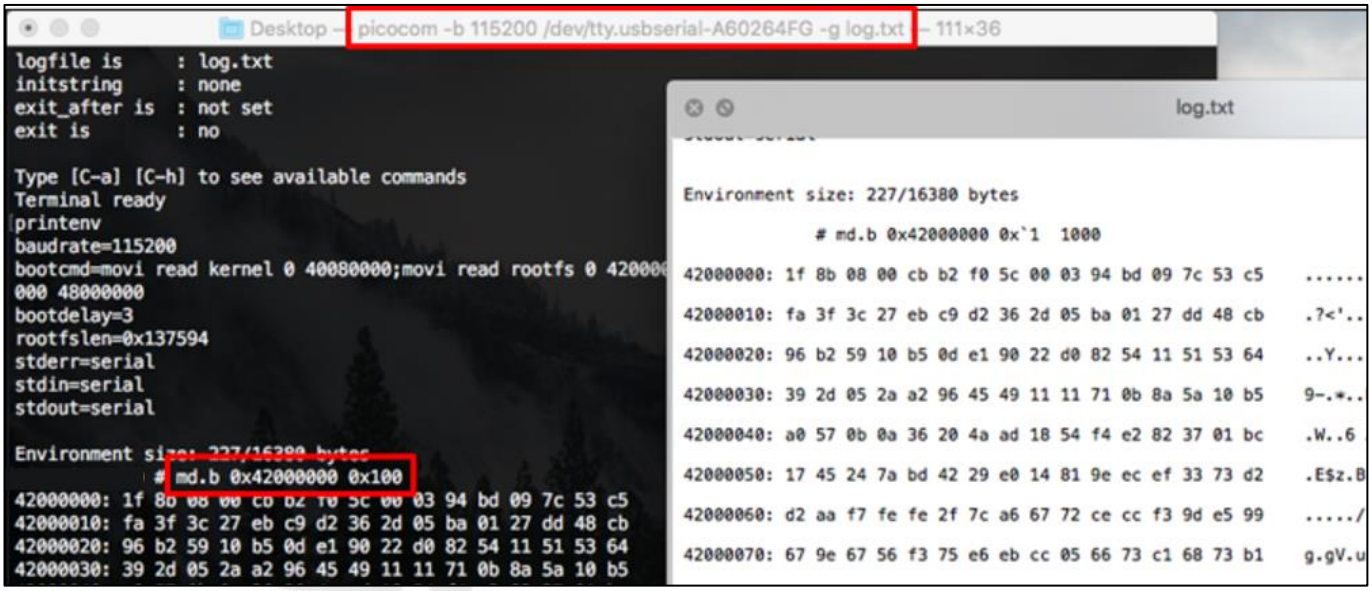


그림 22. md(memory dump).byte 단위

아래는 Kernel 영역 덤프를 바이너리로 만들어 binwalk 를 통해 확인 시 Linux 정보 확인이 가능하였다.

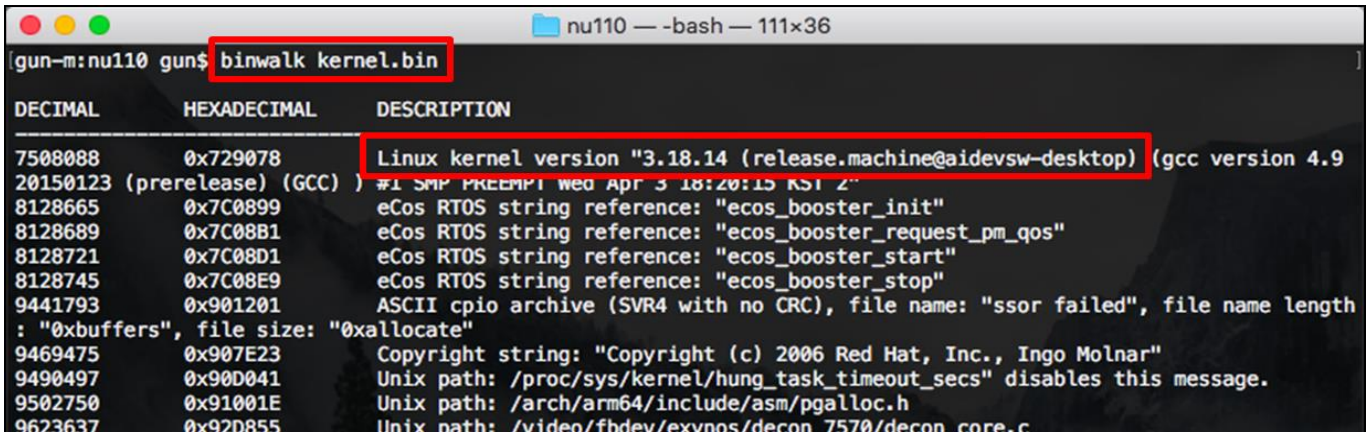


그림 23. Linux Kernel 정보 확인

Root Filesystem 추출 및 파일 확인 (cpio archive)을 진행하였다.

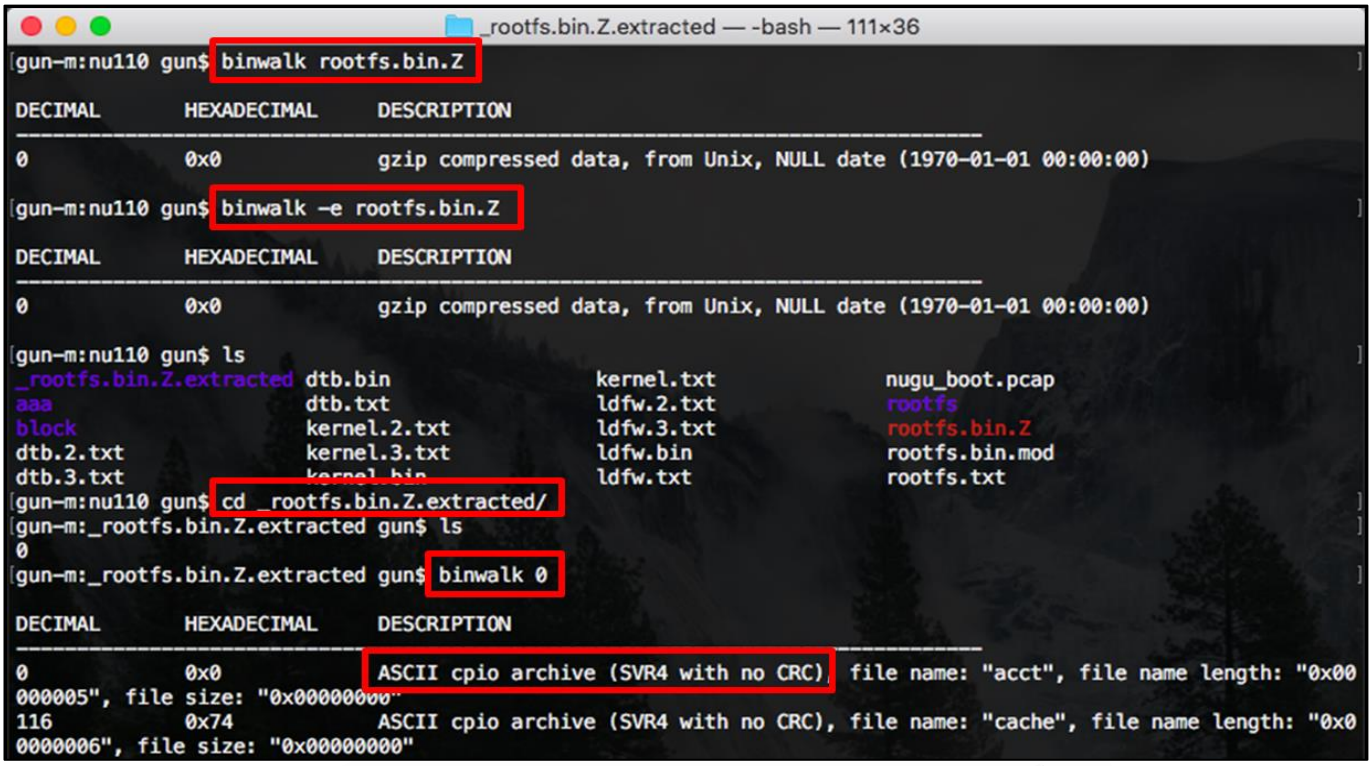


그림 24. 파일 추출 및 확인

해당 파일의 압축해제를 수행하였다.

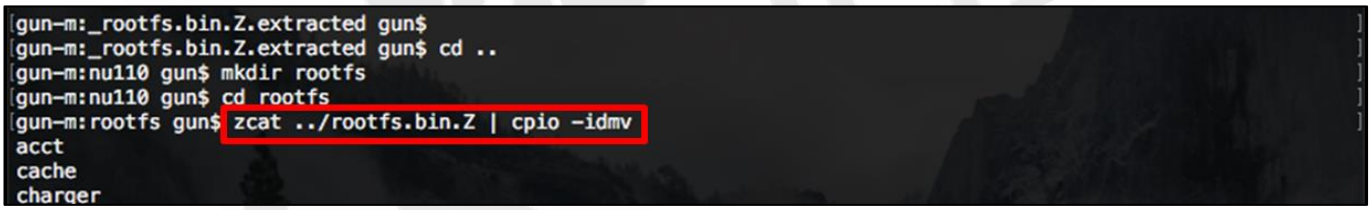


그림 25. 추출 파일에 대한 압축 해제 수행

Filesystem 내부의 파일을 확인하였다.

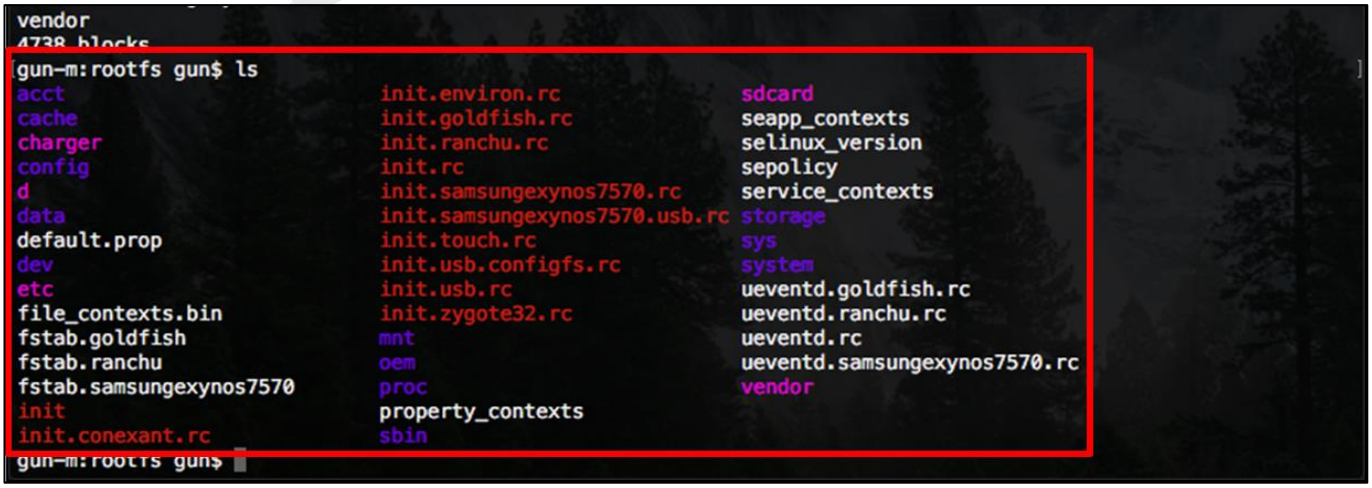


그림 26. 기기내에서 추출된 Rootfs 확인

3.3.2. 양호 Case 2 (AI 스피커)

내부 인터페이스의 확인을 위해 Device 분해를 시도한다.

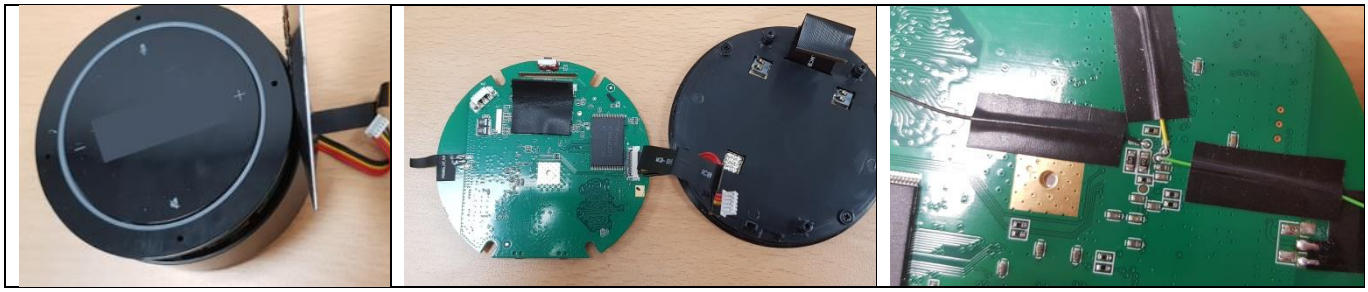


그림 27. Device 분해 시도

RS232 로 연결 시 아래와 같은 Openwrt 콘솔 확인 및 Busybox 구동을 확인하였다.

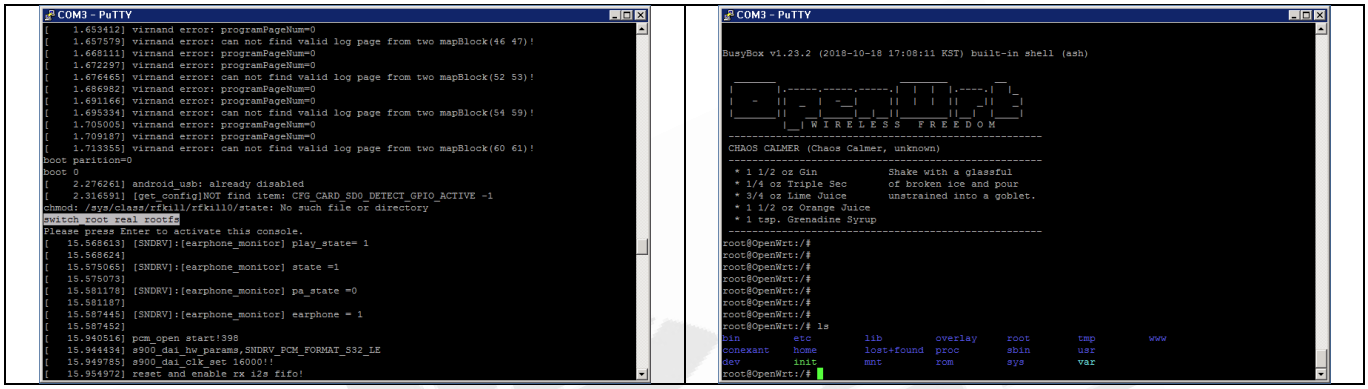


그림 28. Busybox 확인 및 chroot 확인

접근 가능한 Busybox shell 내에서 실제 Filesystem 에서 사용되는 명령어 및 설정 파일이 존재하지 않음 확인하였다.
(/dev, /mnt 내에 상위 폴더에 접근 가능한 별도 구성이 없으며, Busybox 를 통한 Filesystem Dump 가능 명령어 또한 없다.)

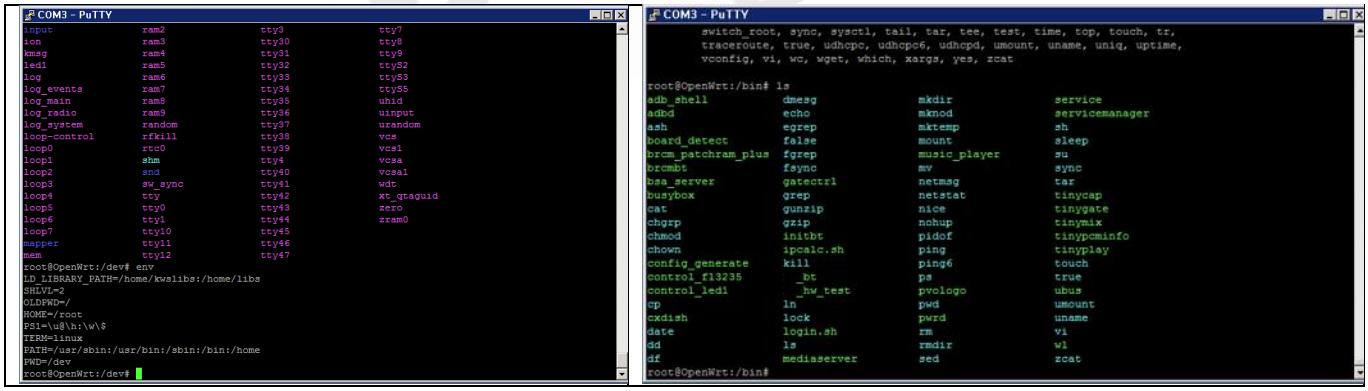


그림 29. Chroot 적용 및 주요 설정 및 명령 확인 불가

※ unchroot 명령어는 제공되지 않으며, 크로스 컴파일을 통해 chroot jailbreak(chw00t)의 실행을 시도하였으나 불가능하였다. (https://github.com/earthquake/chw00t)

3.4. [DV-004] 펌웨어 변조 적용 가능 여부

구분	내용
전제조건	· 펌웨어 획득 혹은 사전 제공
취약점 설명	· [변조된 펌웨어의 적용 가능 여부] · 임의의 펌웨어를 업데이트 하거나 변조된 Filesystem 적용이 가능한 경우 기기의 S/W를 임의 개조하여 구동이 가능한 취약점
판단 기준	· + 다양한 경로를 통해 획득한 펌웨어가 아래와 같이 적용이 가능한 경우 - **[취약]** 펌웨어 업데이트 기능을 통해 변조된 펌웨어를 즉시 적용 가능 한 경우 - **[취약]** 부트로더를 통해 임의의 펌웨어 혹은 Filesystem을 Load하여 실행 가능한 경우
취약점 영향력	· 변조된 펌웨어로 임의 업데이트를 수행하여 기기를 완전하게 제어 할 가능성이 있음
보안대책	· 펌웨어 구동 전 무결성 검증 필요 · Filesystem의 구동 전 변조 여부 검증 필요 · 내/외 불필요한 물리적 인터페이스 제거 · 전용 프로그램 등을 통해 디버그 포트에 접근할 수 있도록 구현

표 7. 펌웨어 변조 적용 가능 여부

3.4.1. 취약 Case 1 (AI 스피커)

Dump 를 통해 획득한 한 주요 파일 중 Android USB 모드 설정파일 변경을 시도한다.

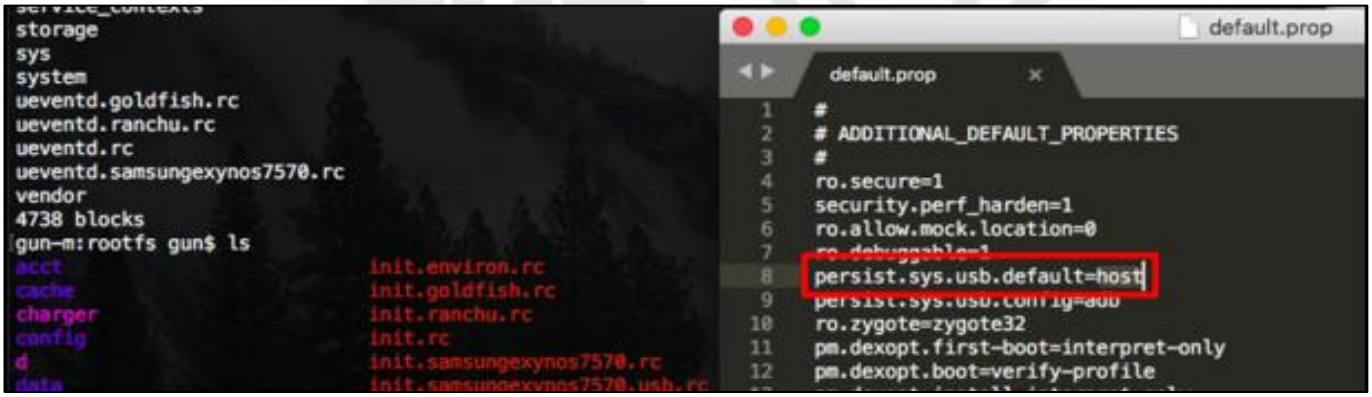


그림 30. USB HOST 설정으로 변경

변경한 파일을 포함해 Filesystem 을 재압축 하였다.

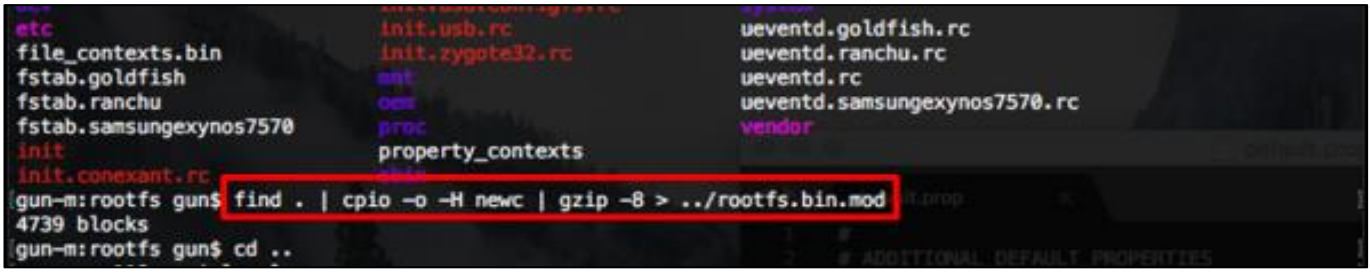


그림 31. FileSystem 재 압축

Kermit 을 활용하여 변경된 Filesystem 을 Device 메모리에 Load 하기 위해 전송 관련 설정을 수행하였다.

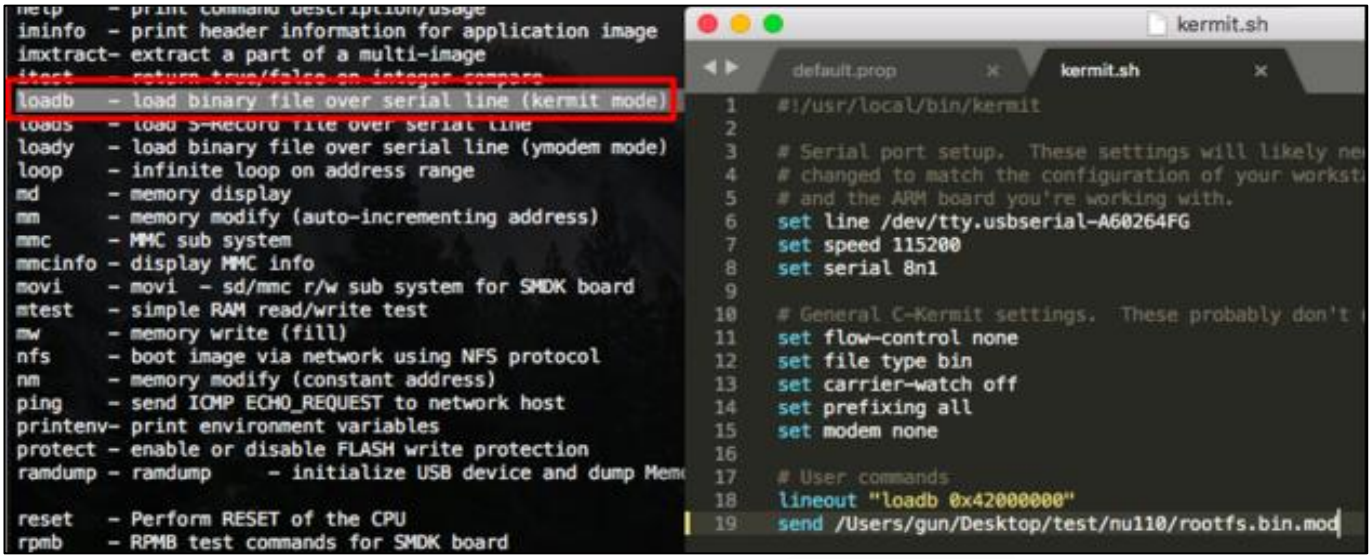


그림 32. 변경 내역 반영을 위한 Device 전송 설정

Device 로 전송을 시도하였다.

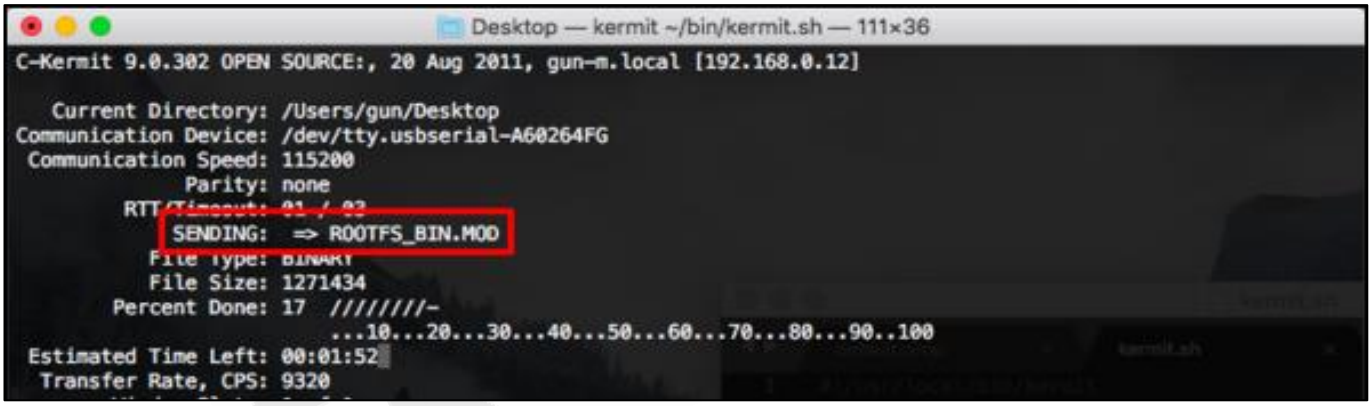


그림 33. Kermit 을 통한 전송

전송 완료 후 수동 부팅 진행이 가능한 것을 확인하였다.

그림 34. 전송 및 로드 완료 후 수동 Boot 진행

Device 외부 USB <-> PC 연결을 재 시도하였다.

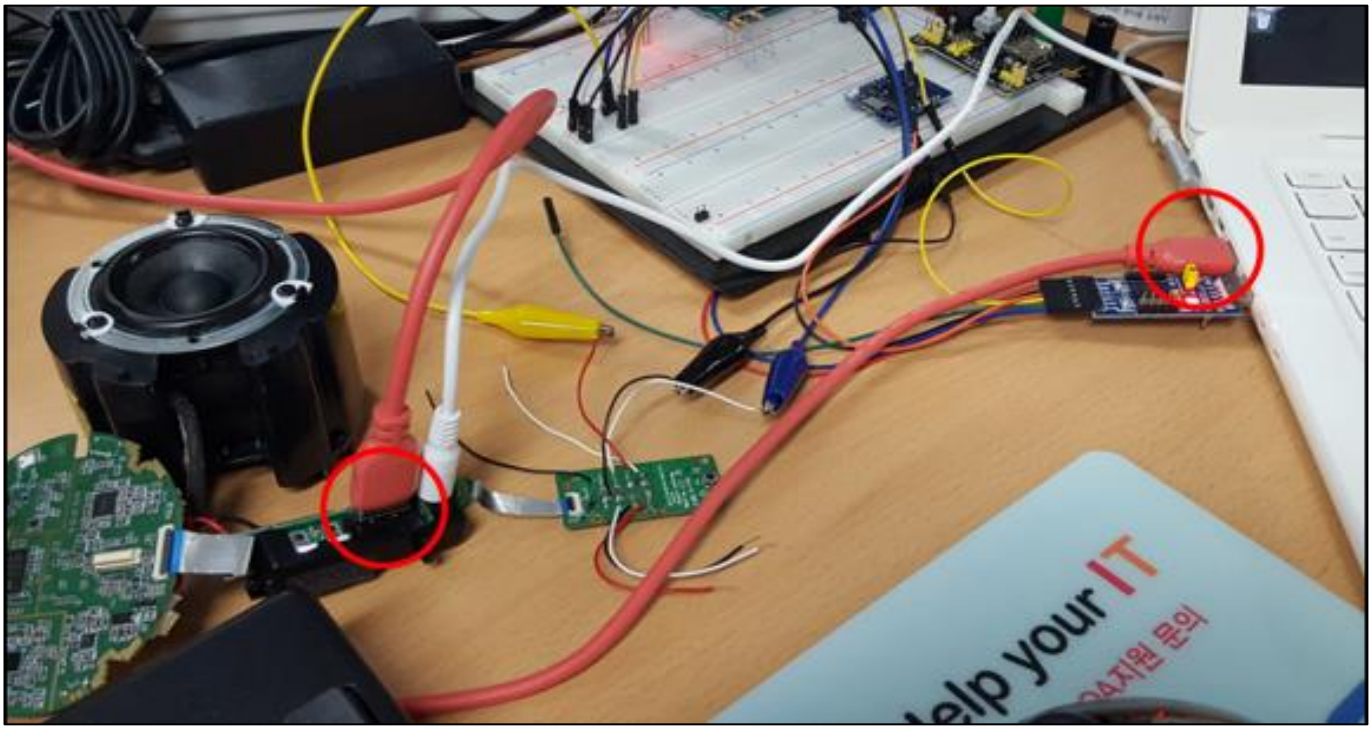


그림 35. 외부 USB 포트를 활용한 adb 연결 시도

설정 변경 적용을 통해 Android 용 adb(Android) 접근이 가능함을 확인하였다.

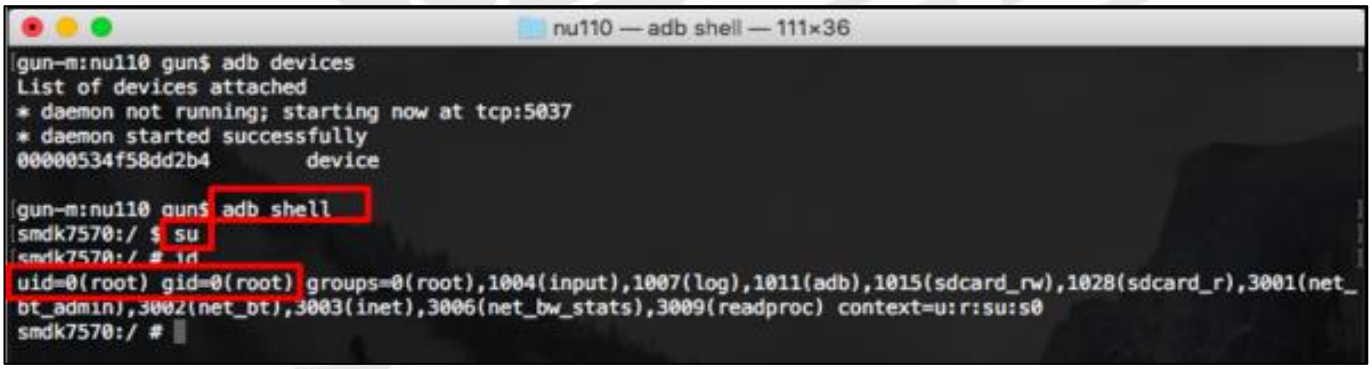


그림 36. 변조 Filesystem Load 시 정상 동작 확인

3.5. [DV-005] 설정 미흡 여부

구분	내용
전제조건	· Device 분해/개조 · MCU 종류에 따른 디버거 필요
취약점 설명	· [설정 미흡으로 판단되는 취약점들의 존재 유무 확인] · Device Chipset, 부팅 과정, 구동되는 서비스들에 대한 보안 설정이 미흡할 경우 정보 노출/침입 등에 악용 가능한 취약점
판단 기준	· + Datasheet 및 공식 홈페이지의 Document 확인을 통해 MCU 보안 기능의 제공여부 확인 - [양호] 해당 보안기능이 적용되어 Debugging 및 Dump가 불가능한 경우 · + Busybox 와 같은 제공 도구들의 확인 - **[취약]** nc, wget, telnet, ftp와 관련된 파일 전송이 가능한 명령어 들이 존재하는 경우 · + 서비스 설정 미흡 관련 취약점 확인 - **[취약]** Device에서 접근 가능한 서비스(FTP, SSH, Web 등)가 보안 설정이 미흡하여 발생하는 문제로 판단 되는 경우 · ※ 부가적으로 Shell 접근이 가능하고, 설정 파일의 평문 조작이 가능한 경우 - 서비스 내 주요 설정 파일들을 확인하여 취약한 설정이 있는지 확인
취약점 영향력	· 변조된 펌웨어 Load를 통해 사용 중 공격자의 임의대로 Device 제어가 가능할 수 있음. · 서비스의 취약점을 통해 주요 파일들을 탈취하거나 shell 획득이 가능할 수 있음.
보안대책	· 제공되는 보안기능 설정을 확인하고 이를 적용할 수 있도록 함 (ex: Secure lock, jtag password 설정 등) · Busybox 구성(menuconfig)을 활용하여 관련 명령어 제거 설정 후 Build 수행 · 불필요 서비스일 경우 서비스를 제거함 · 서비스의 설정을 변경하여 취약점을 제거함 · + 가능한 경우 - Secure boot 기능 활용 - 하드웨어 보안 모듈 (Trusted Platform Module) 사용

표 8. 서비스 설정 미흡

3.5.1. 양호 Case 1 (도어락)

[주의사항]

- Chip 의 제조사 마다 서로 다른 Debugger 가 필요할 수 있다. (ST-Link, J-link 등)

※ 잘 알려진 Device 의 경우 분해 전 검색 엔진을 통해 해체, 분해 관련 키워드 (Teardown, Disassemble)등으로 사전 검색하여 정보를 수집을 시도하는 것이 좋다.

[준비물]

* EFM(ARM) 계열 MCU 의 Debug, Dump 수행을 위한 장비
  - SEGGER J-Link v8, v9, v10 Debugger [상용 제품이며 version 외 Edition 별기능 차이 있음), 유사클론존재]
* S/W
  - SEGGER 홈페이지 참고 (https://www.segger.com/downloads/jlink/)

EFM32G232XXX 계열 MCU 를 확인하였고 DataSheet 확인 및 공식 wiki 를 확인하여 보안 관련 옵션이 있는 것으로 확인하였다.

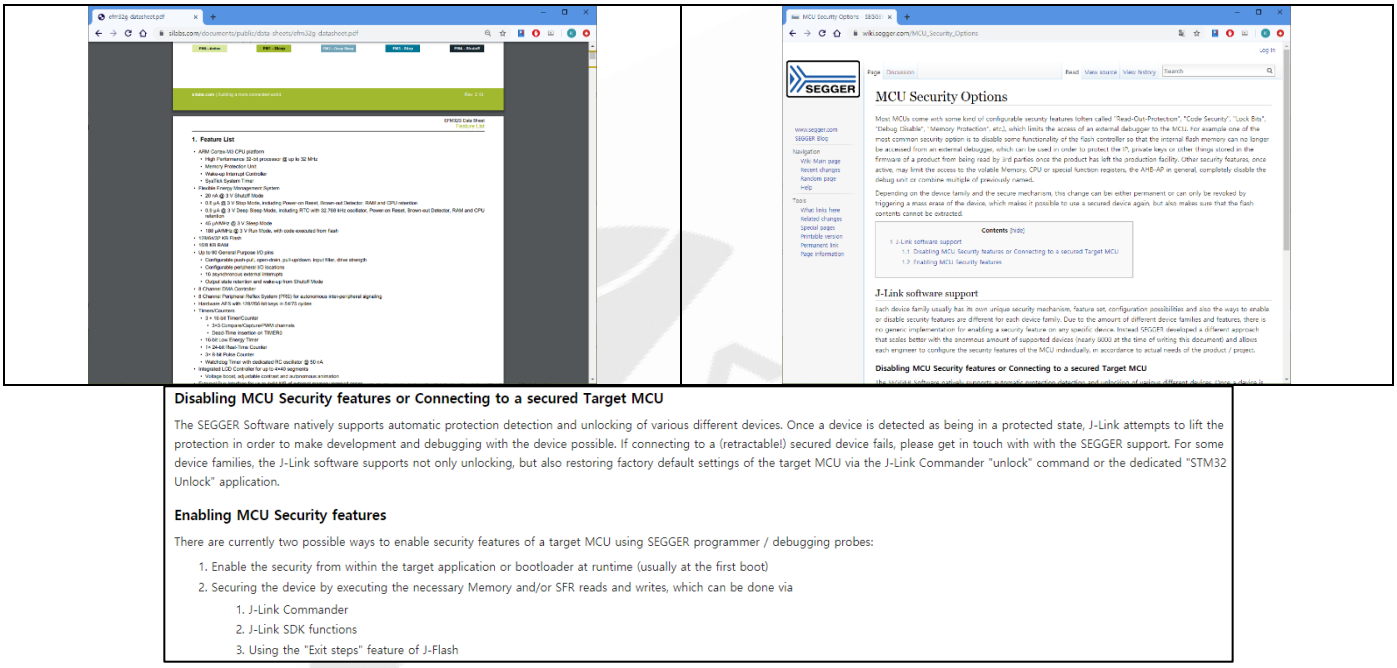


그림 37. MCU 보안 옵션 확인

PCB 내 내부 포트를 통해 SWDIO, SWCLK, SWO, GND 등의 단자를 Debugger(J-link)에 연결하였다.

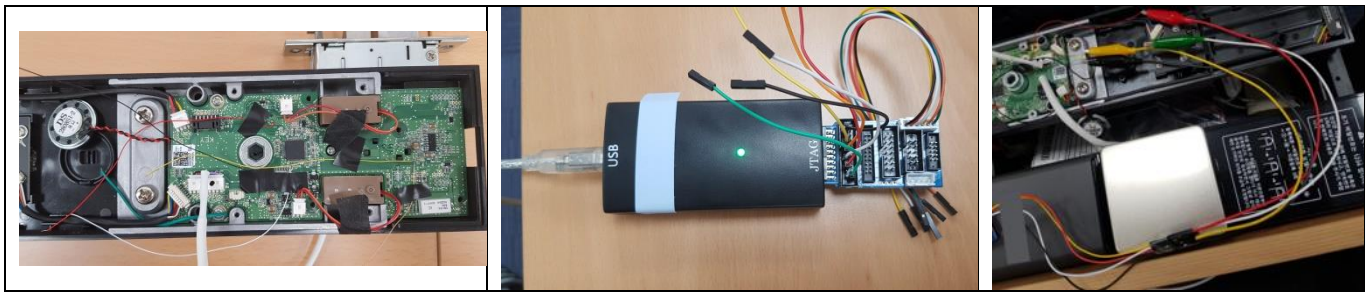


그림 38. SWD 연결 시도

Debugger 를 통해 메모리 확인 시도를 위해 제조사 제공 도구들을 활용하였으나 MCU(Micro Controller Unit) 보안기능에 의한 펌웨어 보호 활성화를 확인하였다.

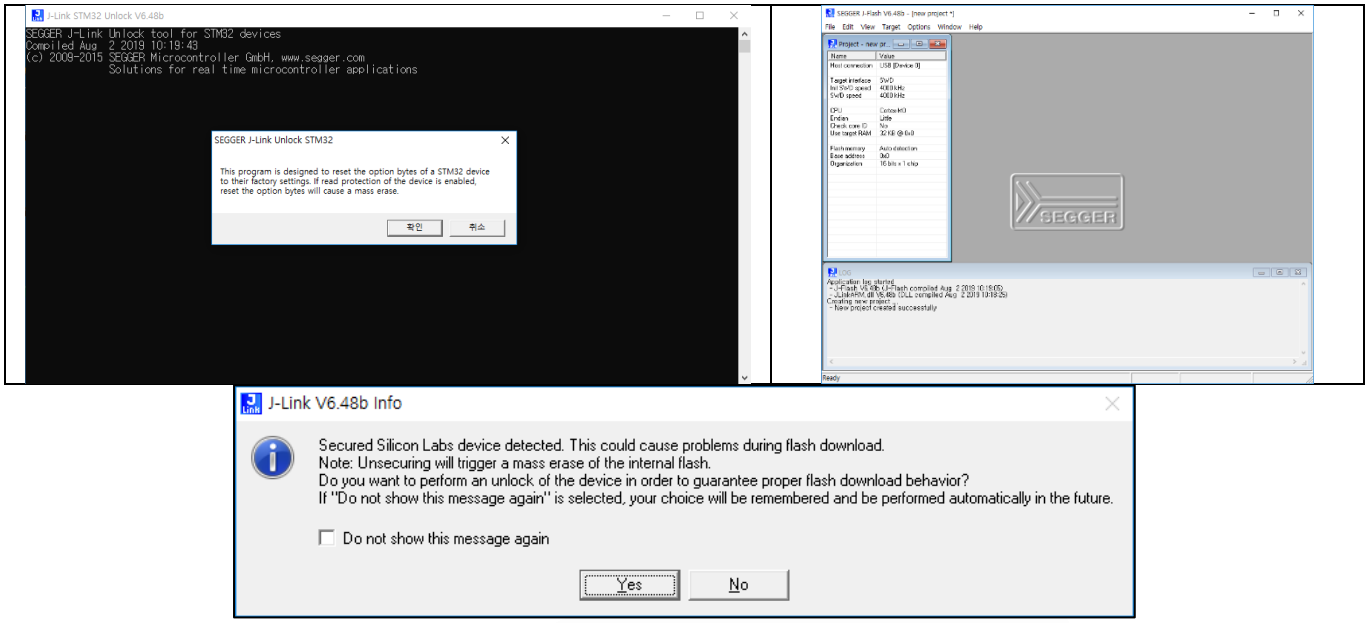


그림 39. SEGGER commander, J-Flash 구동 시 보호 메시지 확인

※ [주의] 검색 엔진을 통해 확인한 임의의 Debug lock 해제를 위한 스크립트 구동 시 펌웨어가 제거됨을 확인하였다. (A/S 불가능). – (https://nathan.vertile.com/blog/2017/03/05/unlocking-stm32-chips/)

3.6. [DV-006] 불필요한 네트워크 서비스 존재 여부

구분	내용
전제조건	· Device 분해/개조 · 펌웨어 획득 혹은 사전제공
취약점 설명	· [불필요한 서비스의 존재 유무 확인] Device 사용 시 불필요한 서비스들을 통해 Device 내의 정보가 노출되거나 디바이스를 장악할 가능성이 존재하는 취약점
판단 기준	· 불필요한 서비스 및 데몬 구동 여부를 확인해야 함 (부득이한 경우 사용자 인증 제공 여부 체크 필요) - FTP, Telnet, SSH, NFS, Upnp · Shell 접근이 불가능 한 경우 - 해당 Device에 ip가 할당되어 있으면 해당 IP에 대한 포트 스캐닝을 수행 - 포트가 존재할 경우 접속 수행하여 확인 - 해당 포트들에 대한 알려진 취약점 확인 ※ 부가적으로 Shell 접근이 가능한 경우 - shell(sh, busybox)에서 netstat 등의 명령어를 활용하여 외부에서 접근 가능한 포트가 있는지 확인
취약점 영향력	· 제품 내의 임의로 제공되는 서비스를 통해 디바이스 장악에 활용 (Backdoor) · 사용자가 인지하지 못하는 서비스를 공격자가 악용할 수 있음
보안대책	· 불필요한 네트워크 서비스 제거

표 9. 불필요한 네트워크 서비스

3.6.1. 양호 Case 1 (AI 스피커)

해당 장치에 IP 가 할당이 되어 있어서 아래와 같이 포트 스캔을 수행하였다.

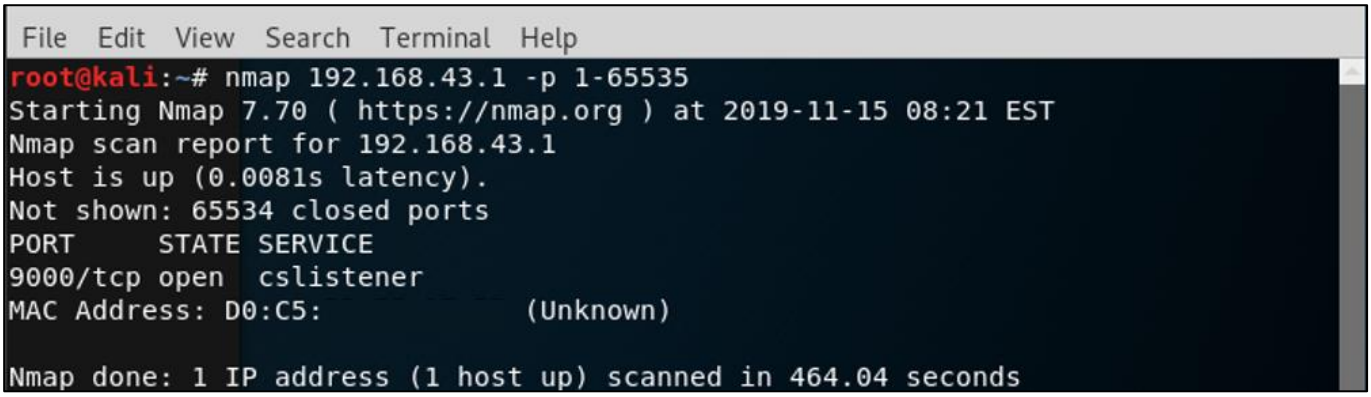


그림 40. Nmap 을 활용한 포트 스캔 수행

최종적으로 9000 포트에 접근이 가능하지만 해당하는 포트에 대한 알려진 접근 방법으로는 해당 포트에서 제공하는 서비스에 접근할 수 없었다. 또한 해당 포트에 대해 일반적이지 않은 데이터 전송을 시도하였으나 동작 없음을 확인하였다.

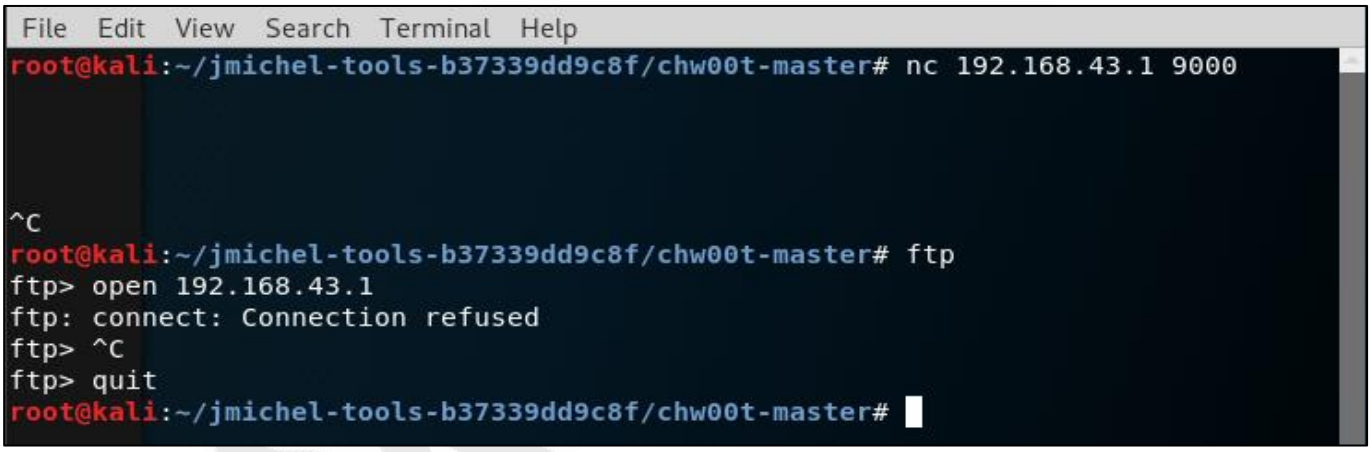


그림 41. 알려진 서비스 접근 방식으로 접근 시도

내부 포트를 통해 네트워크 서비스를 확인한 결과 /home/media 바이너리에 의해 서비스되고 있음을 확인 (netstat –atp) 하였다.

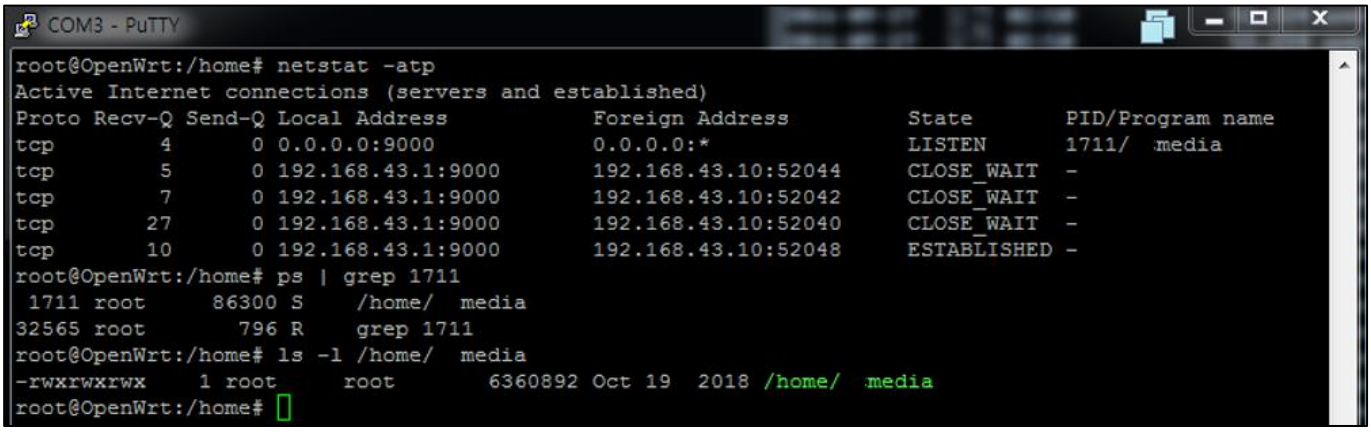


그림 42. 디버깅 포트를 통한 네트워크 서비스 확인

해당 바이너리 내의 심볼 확인이 가능하였으며, recv, bind 등의 통신관련 API 기준으로 분석을 하였으나 별다른 정보가 없음을 확인하였다.

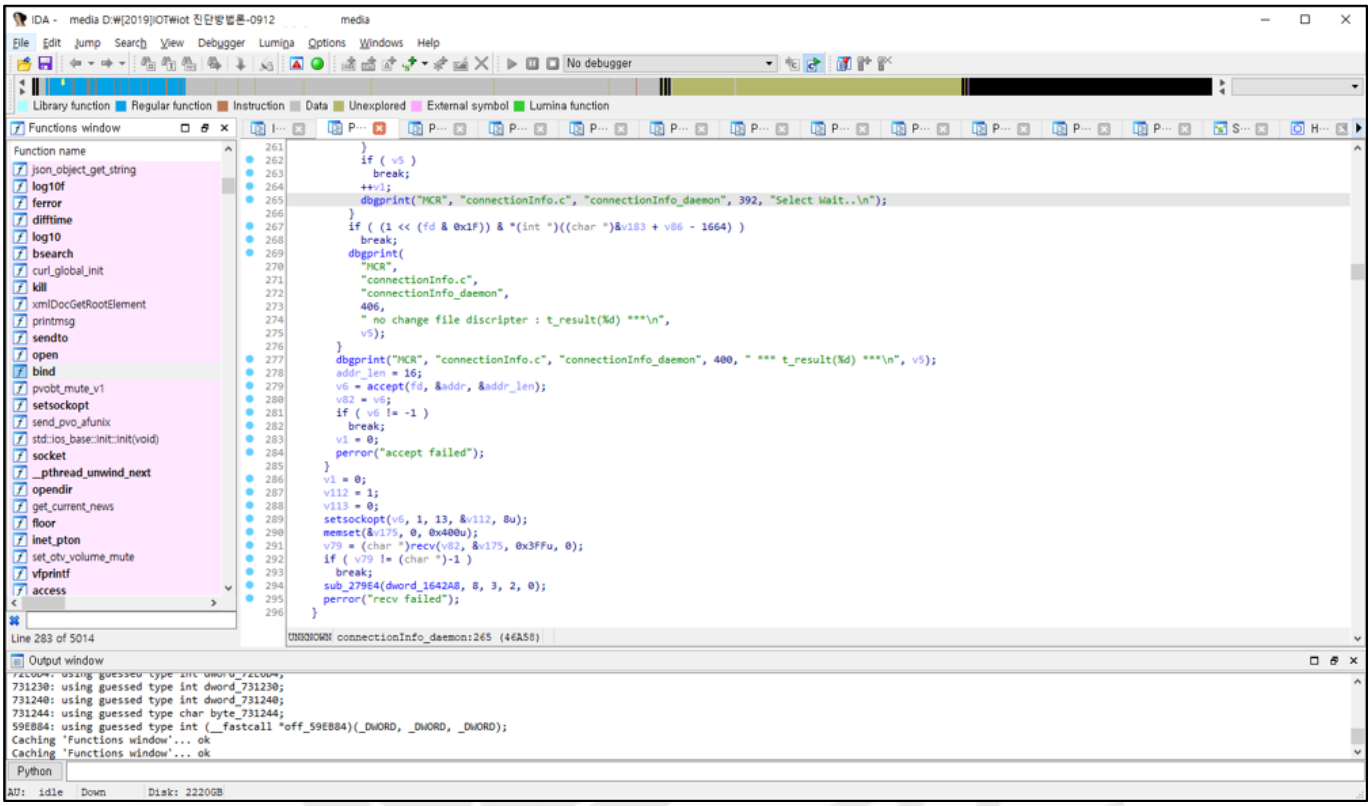


그림 43. 바이너리 확인

또한, 9000 번 포트에 대한 알려진 취약점을 검색하였으나 해당 Device 에 해당하는 내용이 아니므로 양호로 판단하였다.

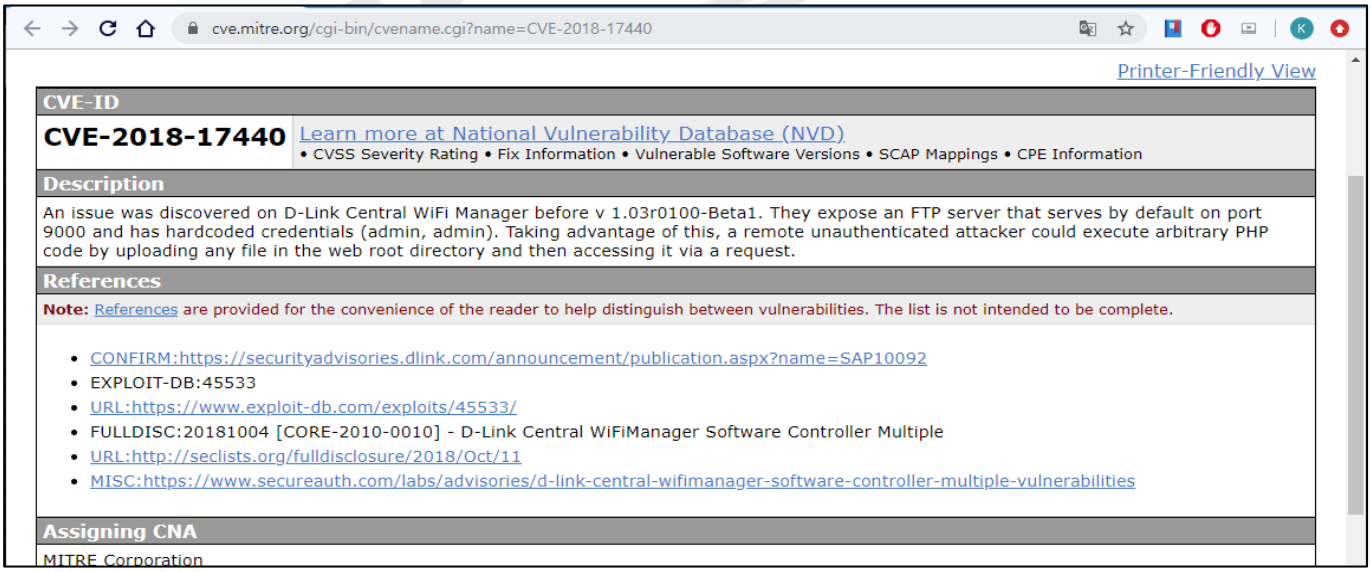


그림 44. 알려진 취약점 검색

3.7. [DV-007] 취약한 계정 사용 여부

구분	내용
전제조건	· 없음
취약점 설명	· [취약한 계정 사용 여부 확인] · Device 관리 및 제어에 취약한 계정을 사용하는 취약점
판단 기준	· **[취약]** 초기 설치 시 제공되는 디폴트 계정 사용 · **[취약]** 인터넷에 잘 알려진 계정을 활용하여 디바이스의 설정, 접근제어 등의 설정 변경에 활용할 수 있는 경우 ※ 최근의 경우 등록 Device들은 Cloud를 통해 관리되며, 별도의 관리 페이지를 제공하지 않는 추세이지만 존재할 경우 확인이 반드시 필요함
취약점 영향력	· 기기 설정 접근을 통해 사용자의 Device 사용 시간대, 현황 파악하여 악용 가능 · 무작위 대입 공격, 사전 대입 공격 등에 의해 장비 접근 가능성이 있음
보안대책	· 최초 초기 제공되는 디폴트 계정의 경우 패스워드를 반드시 변경하도록 함 · 디폴트 계정의 패스워드를 변경하여 사용 (ex: 최소 8자 이상 대문자, 소문자, 숫자 혼합)

표 10. 디폴트, 취약한 기준의 계정 사용

3.7.1. 취약 Case 1 (IP cam)

IPcam 구동 시 웹 인터페이스가 존재하며 해당 인터페이스에 디폴트 계정을 통하여 로그인이 가능하였다.

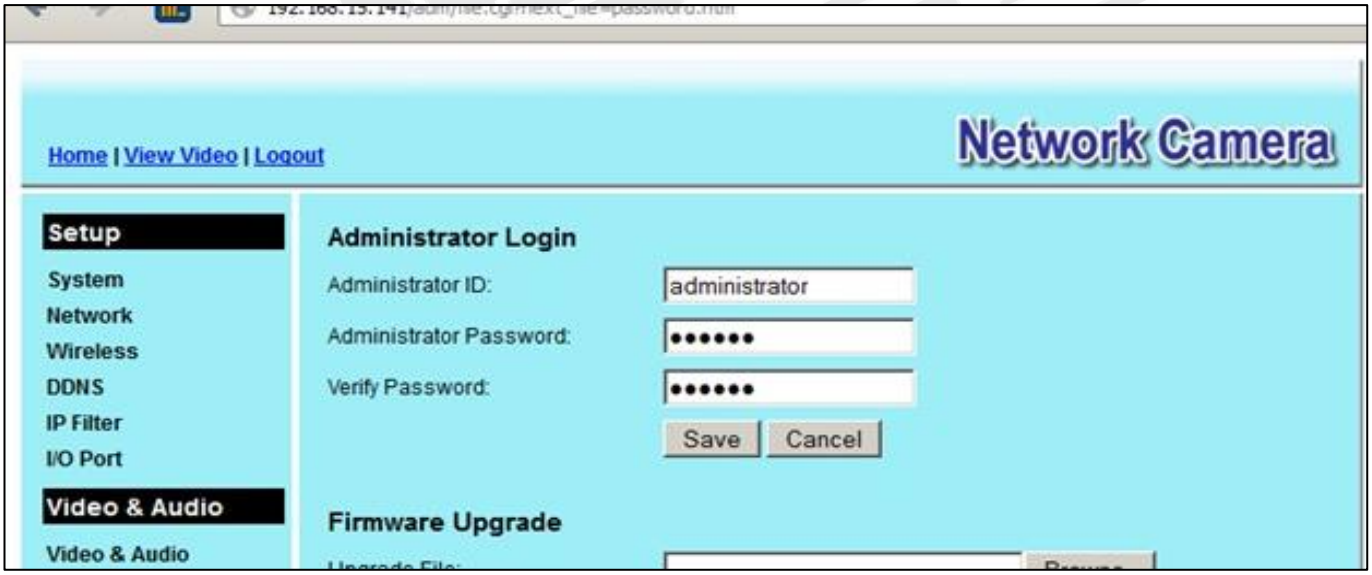


그림 45. 취약한 IP Cam Default 계정 로그인 가능 확인

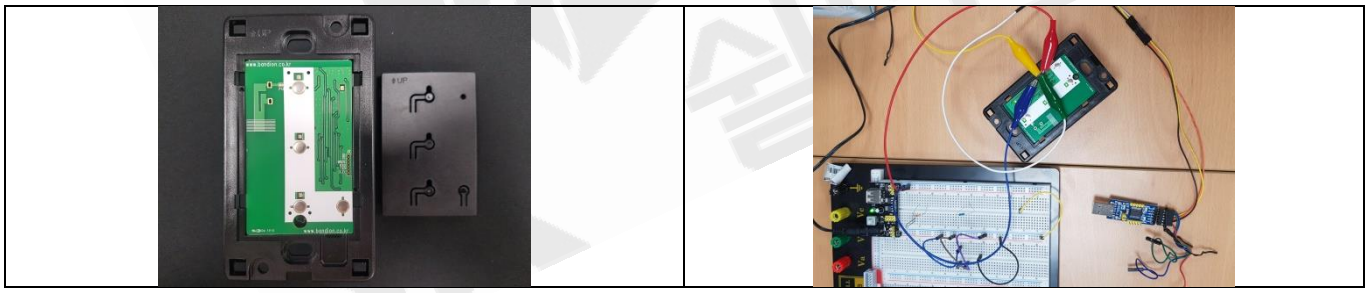
3.8. [DV-008] 중요 정보 출력 여부

구분	내용
전제조건	· Device 분해/개조 · MCU 종류에 따른 디버거 필요
취약점 설명	· [출력되는 정보에 중요 정보의 포함 여부 확인] · Device에 의해 발생하는 출력을 통해 중요 정보가 확인되어 노출되는 취약점
판단 기준	· **[취약]** PCB 내 주요 H/W의 DataSheet를 확인하고 디버깅용 핀을 확인했을 때 통해 주요 정보가 출력되는 경우 · **[취약]** Device 내 쉘 접근이 가능한 경우 디바이스 내 저장되는 로그에 주요 정보가 출력 출력되는 경우 · **[취약]** BootLog에 주요 정보가 출력 되는 경우
취약점 영향력	· 출력되는 중요 정보를 수집하여 다른 공격에 활용할 수 있음
보안대책	· 디버그 출력 내 주요 정보 제거

표 11. 서비스 설정 미흡

3.8.1. 취약 Case 1 (스마트 스위치)

스마트 스위치 앱 연동 이후 Debug 포트를 통한 로그를 확인한다.



Wifi 정보 및 제품의 트래픽 암호화에 사용되는 AES 대칭 Key 정보가 포함된 로그 출력을 확인하였다.

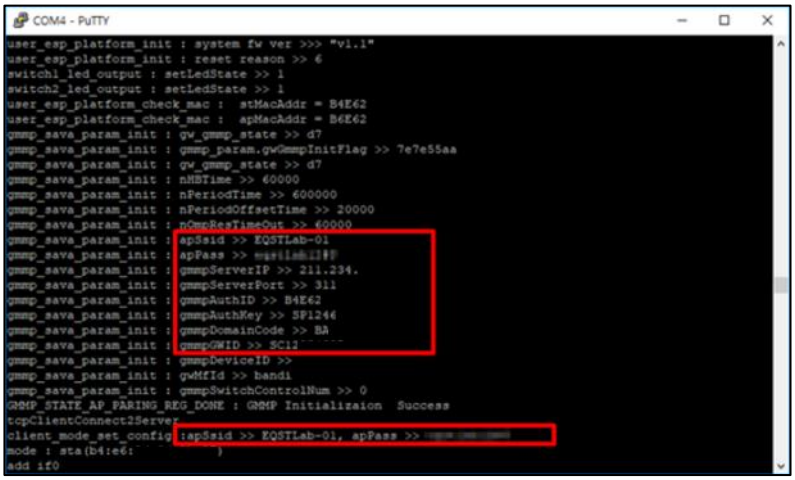


그림 46. 중요 정보 출력 확인

3.9. [DV-009] 중요 정보 평문 저장 여부

구분	내용
전제조건	· 물리 인터페이스를 통한 접근, 펌웨어 제공
취약점 설명	· [중요 정보에 대한 평문 저장 여부 확인] · Device의 저장장치 내 중요 설정파일, 암호키, 인증 정보 등에 대해 평문으로 저장하고 있을 경우 공격자가 이를 확인하여 악용할 가능성이 있는 취약점
판단 기준	· **[취약]** 중요 파일을 암호화하여 저장하지 않는 경우 · **[취약]** Filesystem 혹은 펌웨어 내에 Hard Coding 된 불필요한 계정 정보 (FTP, SSH 등의 계정 정보)가 존재할 경우 · **[취약]** SDCARD등의 외장 스토리지에 설정 파일들을 평문 저장하는 경우
취약점 영향력	· 출력되는 중요 정보를 수집하여 2차 공격에 활용할 수 있음
보안대책	· 중요 파일이 존재할 경우 암호화하여 저장할 수 있도록 권고 · 개발 시 중요 계정 정보(테스트, 운영)를 제거함 · 불필요 정보 시 제거

표 12. 주요 파일 평문 저장

3.9.1. 취약 Case 1 (스마트 스위치)

스마트 스위치 앱 연동 이후 Debug 포트를 통해 (ESP82XX) Esptool.py 를 활용하여 FlashDump 를 수행한다.
Dump 내의 User Data 로 판단되는 영역에 String 중 FTP 포트, 계정정보를 확인 가능하였다.

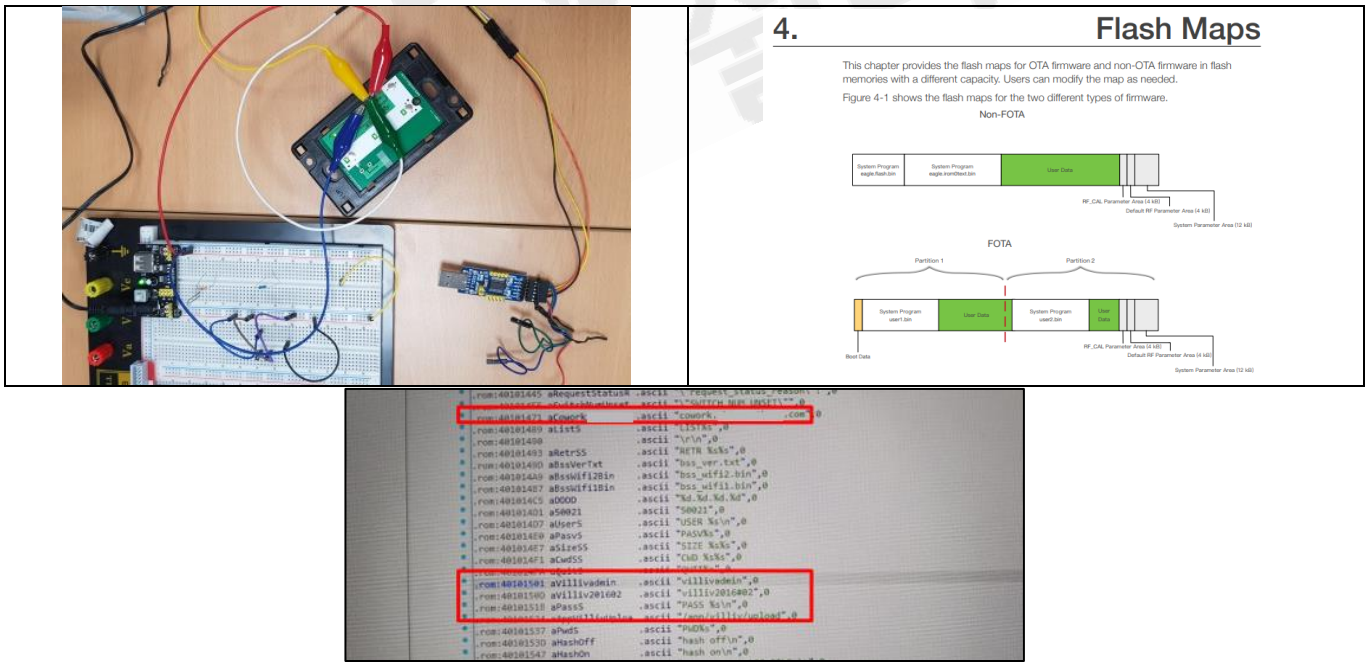


그림 47. 펌웨어 내 FTP 접속 계정 정보 확인

※ 해당 정보를 통해 직접 연결을 시도하였으나 계정 정보는 올바르지 않아 실제 접근은 불가능 하였다..

(참고 : https://github.com/esp8266/esp8266-wiki/wiki/Memory-Map
https://www.espressif.com/sites/default/files/documentation/2a-esp8266-sdk_getting_started_guide_en.pdf)

3.10. [DV-010] 백업 및 테스트 파일 존재 여부

구분	내용
전제조건	· 물리 인터페이스를 통한 접근, 펌웨어 제공
취약점 설명	· [백업 및 테스트 파일 존재 유무 확인] · Device 내 제공되는 서비스 혹은 Filesystem 내 백업 및 테스트 파일에서 의도하지 않은 정보가 누출되는 취약점
판단 기준	· + Device 내 아래에 해당하는 **파일이 존재하는 경우 취약** - 테스트 파일 - 백업 파일 - 제공되는 서비스의 Default 파일
취약점 영향력	· 백업 파일 내 중요정보를 수집하여 2차 공격에 활용할 수 있음
보안대책	· 개발 단계 이후 불필요파일의 경우 제거 필요

표 13. 백업 및 테스트파일

3.10.1. 참고 사항

3.10.1.1. 백업 및 테스트 파일 유형

기본적으로 백업 파일이 포함할 수 있는 문자열들은 아래와 같다.

주요 문자열				
.bak.	*.bak	*.backup	*.org	*.zip
*.old	*.tar.gz	*.zip	*.log	*.copy
*.txt	*.new	*.tmp	*.temp	*.db.old
*.orig	~*,	*.gzip	*.!	*.gz

…

표 14. 백업 및 테스트 파일 유형

3.10.2. 취약 Case 1 (AI 스피커)

Shell 접근을 통해 주요 확장자에 대한 파일을 검색하여 백업 파일로 유추되는 파일을 확인한다.
(해당 파일 내 중요 정보로 판단할 수 있는 내용은 없으나 제거가 필요하다.)

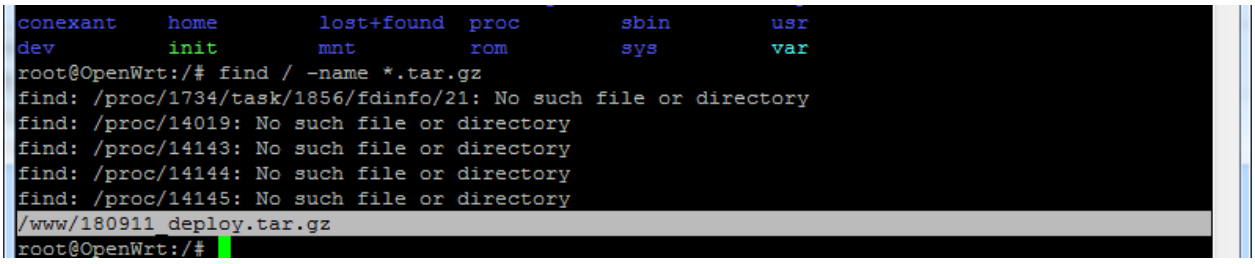


그림 48. 백업 파일 확인

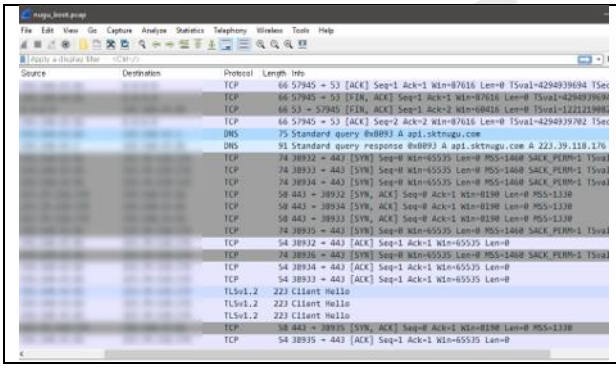
3.11. [DV-011] 전송 구간 보호 여부 (Device)

구분	내용
전제조건	· 별도 사용 가능한 공유기, 펌웨어 변조
취약점 설명	· [통신 구간에 대한 평문 전송, 중요정보 노출 여부 확인] · WIFI(무선) 통신 구간에 대해 SSL 미적용 시 평문으로 전송되는 정보가 유출될 가능성이 있는 취약점
판단 기준	· + 사용자의 중요 정보를 전송하는 메뉴들이 있는지 확인함 · - 결제 정보, 주소지 등록, 패스워드 등록 등 · + [양호] 통신 구간 SSL을 적용하고 있을 경우 · + **[취약]** HTTP를 통해 평문으로 전송되는 경우 · + **[취약]** 전송구간에서 사용자의 중요정보가 포함되어 확인이 가능한 경우
취약점 영향력	· 유출되는 사용자 중요 정보를 실제 결제에 악용 · 중요정보를 수집하여 2차 공격에 활용
보안대책	· 평문 전송 시 암호화 통신(SSL) 적용 필요 · 전송 구간 내 사용자의 중요정보 암호화 · 불필요하게 전송하는 사용자 정보 제거

표 15. 주요 정보 전송 (Device)

3.11.1. 양호 Case 1 (AI 스피커)

전송 구간 SSL 적용 여부를 확인하기 위해 AI 스피커의 트래픽을 확인하였으나 SSL 이 적용되어 있는 것을 확인하였다.



SSL 적용 확인

도메인	IP 주소	포트	TLS 통신	비고
no[redacted].com	211.[redacted].64	443	TLSv1.2	로그 서버로 추측
api.[redacted].com	223.[redacted].176	443	TLSv1.2	API 서버
pi[redacted]cloud.co.kr	223.[redacted].191	443	TLSv1.2	앱에서 텍스트 명령 송신
as[redacted]cloud.co.kr	223.[redacted].193	8100	None	사용자 음성 명령 송신
rd[redacted]cloud.co.kr	223.[redacted].194	8281	TLSv1.2	디바이스 제어 명령 수신 (무드등 켜/꺼 등)
tt[redacted]cloud.co.kr	223.[redacted].245	7000	None	음성 데이터 수신

전체 통신 현황 확인

그림 49. 전송 구간 보호 여부 확인

4. 근거리 무선통신 점검 상세

4.1. [MQ-001] 불필요한 토픽 접근 가능 여부

구분	내용
전제조건	· MQTT 통신을 사용하는 디바이스
취약점 설명	· [불필요한 토픽 접근 가능 여부] · 토픽 접근 제한이 적용되어 있지 않아 접근 권한 없는 사용자가 토픽 접근 및 이용이 가능한 취약점
판단 기준	· + [양호] 토픽 접근 제한이 적용된 경우 · + **[취약]** 타 사용자의 토픽 접근 및 이용이 가능한 경우 · + **[취약]** 계정 없는 사용자의 토픽 접근 및 이용이 가능한 경우
취약점 영향력	· 타 사용자의 기기에 악의적인 값 전달 · 중요 정보를 수집하여 2차 공격에 활용
보안대책	· 등록되지 않은 사용자의 토픽 publish/subscribe 방지 · ACL을 통한 타 사용자 토큰 접근 방지

4.1.1. 취약 Case 1 (HiveMQ)

토픽 접근 제한이 적용되어 있지 않은 경우 공격자는 Broker 에 접근하는 모든 메시지를 확인할 수 있게 된다. HiveMQ 는 애플리케이션에서 직접 클라이언트에 대한 권한을 설정하거나, RBAC 를 사용하여 적용할 수 있다.
접근 권한 설정이 제대로 되어 있지 않는 경우 다음과 같이 타 사용자의 토픽에 접근할 수 있다.
클라이언트의 사용자 정보 [User ID : user1 , Password : pass1]

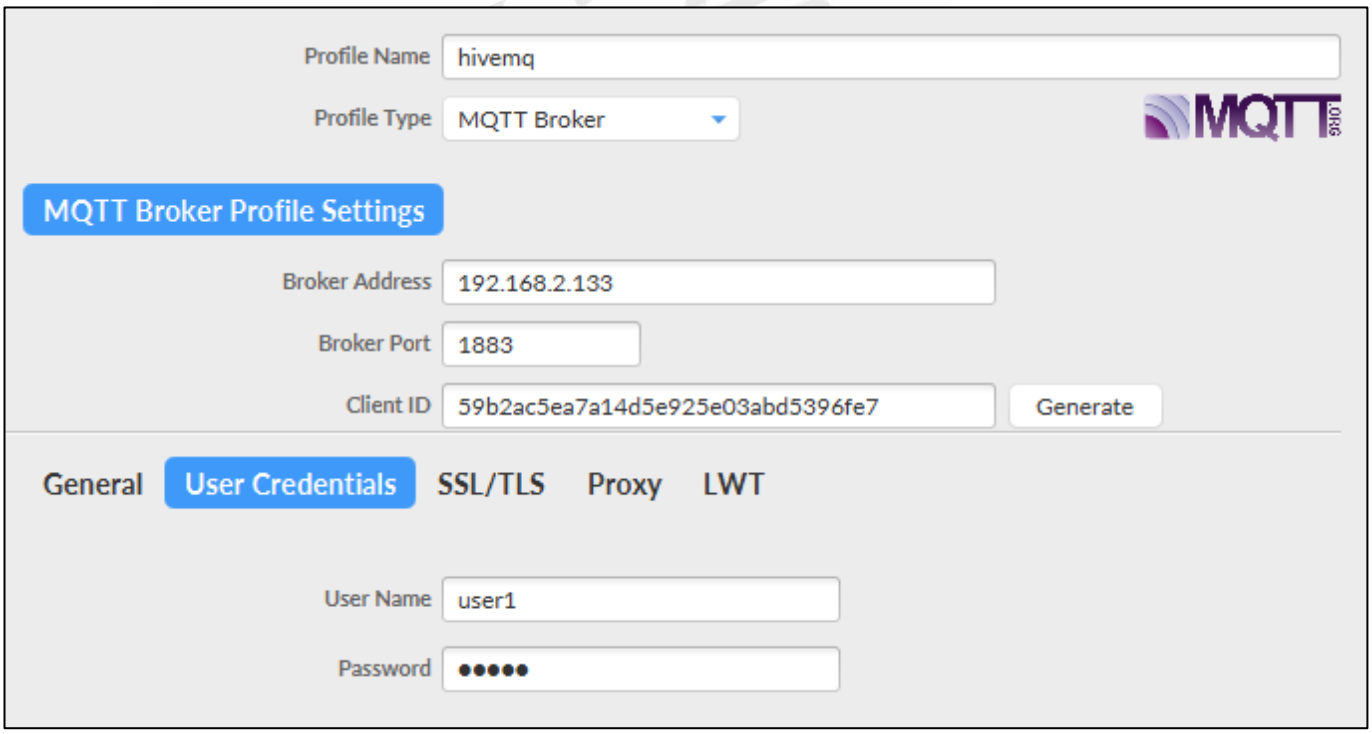


그림 50. MQTT 접속

클라이언트에서 user1 의 topic 으로 메시지를 publish 하였다. [topic : user1/test , message : test]

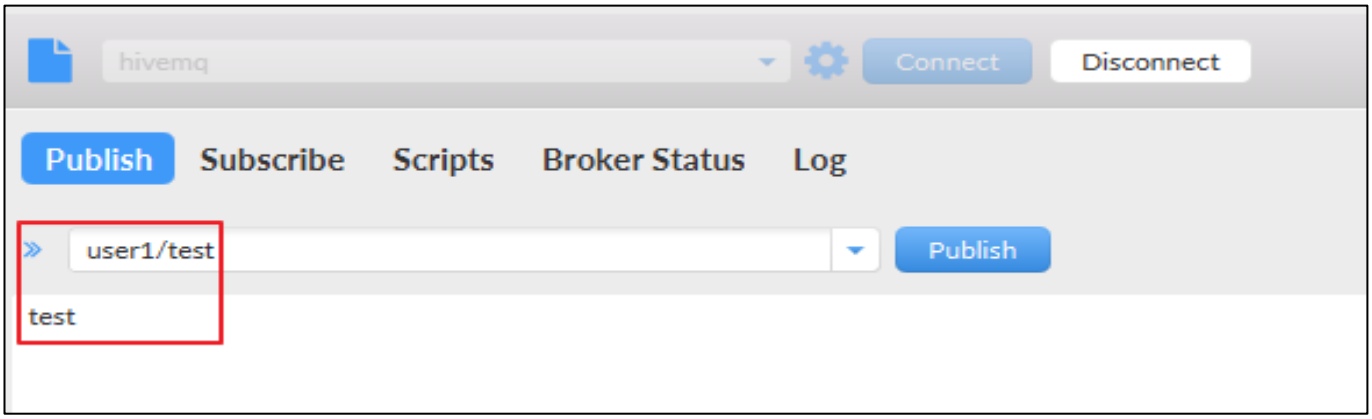


그림 51. user1 의 topic Subscribe

다른 클라이언트에서 다른 사용자 user2 로 user1 이 publish 한 topic 의 구독을 시도하였다. 별도의 권한 설정이 없어 타 사용자의 메시지를 구독할 수 있었다. [User ID: user2 , subscribe topic: user1/test (user1 의 topic)]

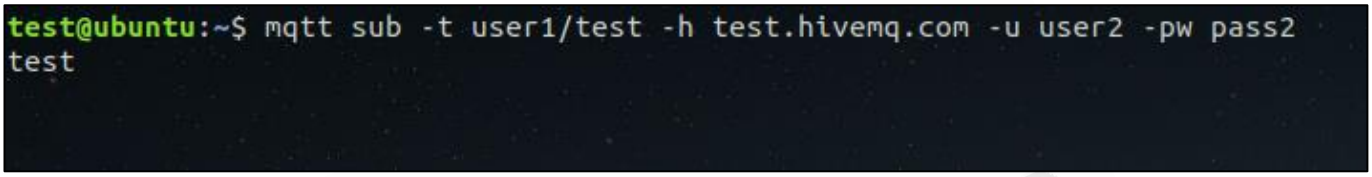


그림 52. 타 사용자 메시지 구독 확인

4.1.2. 취약 Case 2 (RabbitMQ)

user01 계정으로 접속하여 user02 에 지정된 topic publish 가능여부를 확인하기 위해 '/client/device/user02' topic 으로 "ON" 메시지를 publish 해보았다. [User ID: user01 , publish topic: /client/device/user02]

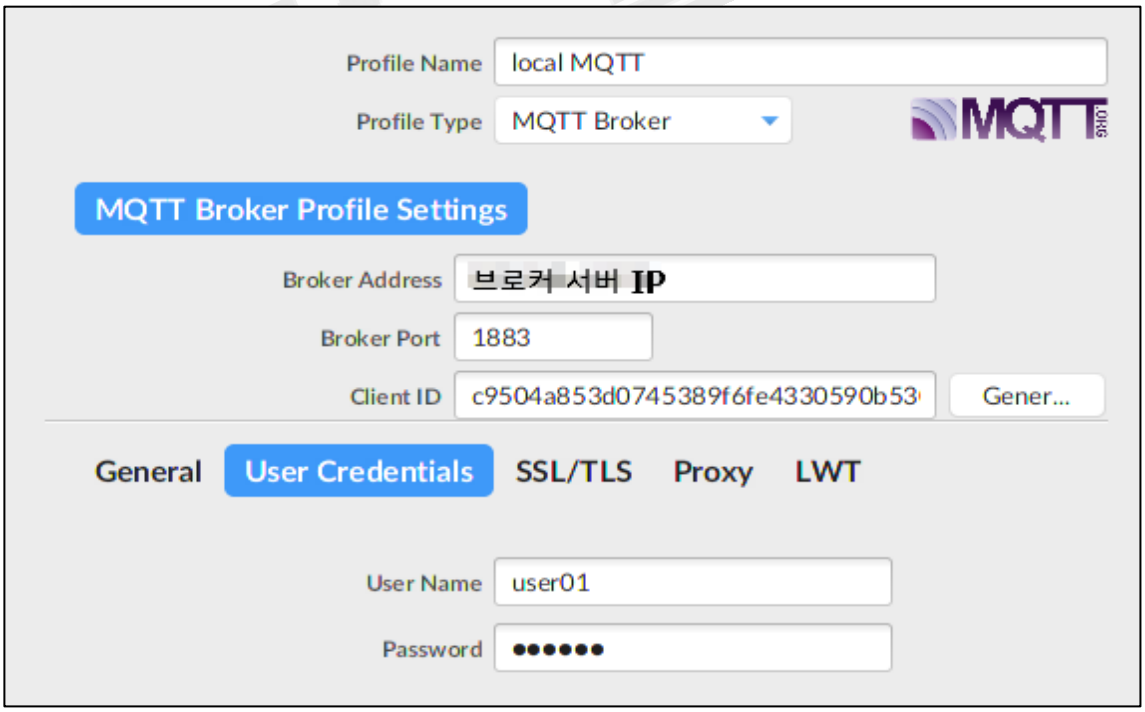


그림 53. MQTT 접속

user01 계정으로 user02 topic 접근이 가능하여 subscriber 로부터 "Turn On" 응답을 확인할 수 있다.

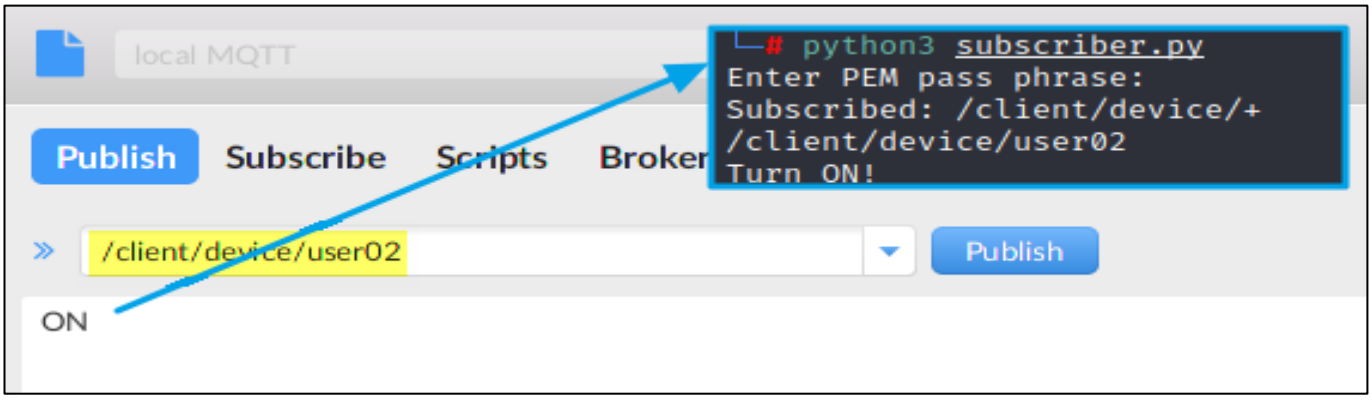


그림 54. Topic Publish (user01 -> user02)

user01 계정으로 topic wildcard 를 이용하여 다른 사용자들의 publish 메시지를 subscribe 해보았다.
user01 계정으로 wildcard 를 이용한 subscribe 가 가능하여 다른 사용자들의 publish 메시지를 확인할 수 있다.
[User ID: user01 , subscribe topic: /client/device/+]

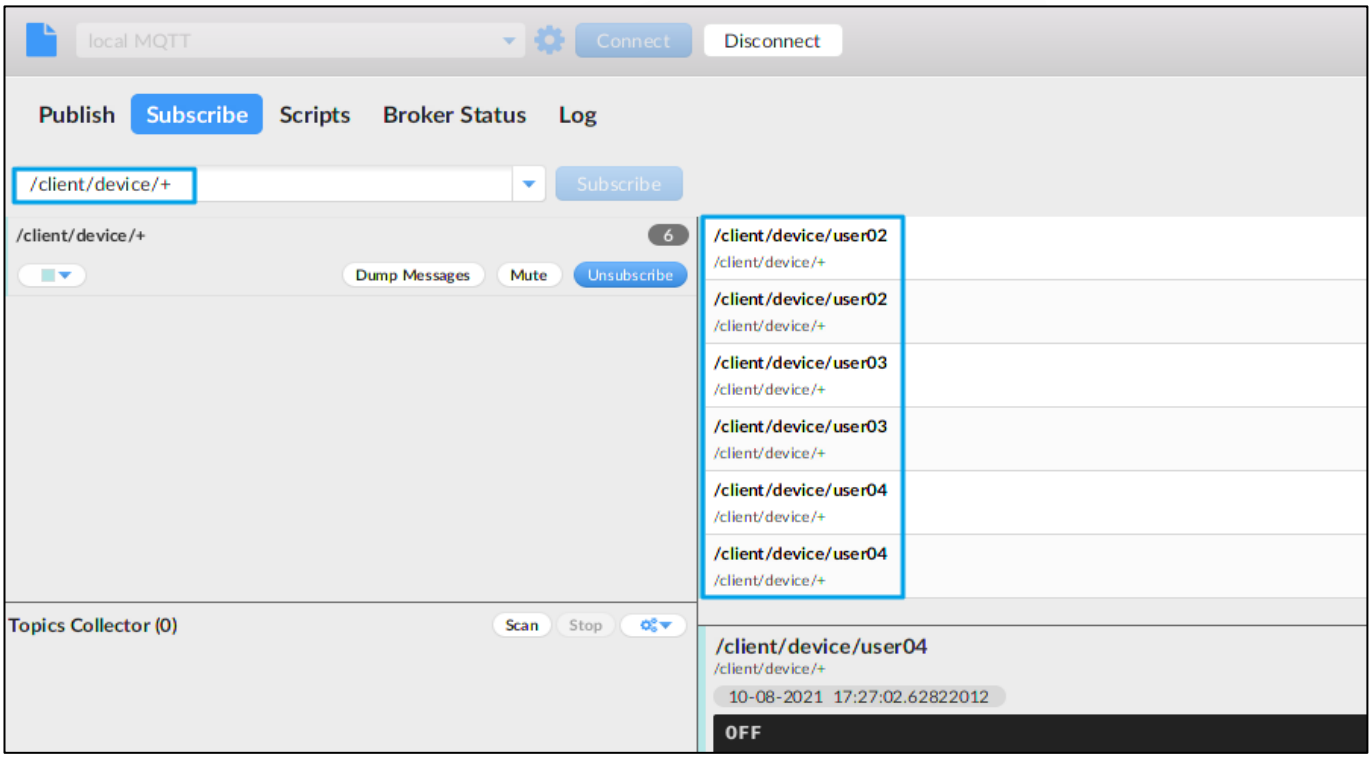


그림 55. 다른 사용자 publish 메시지 확인

4.1.3. 양호 Case 3 (RabbitMQ)

Client 에서 "ON"이라는 publish topic 전송 시 "Turn ON!"으로 응답하는 Subscriber 가 있는 MQTT Broker 서버이다. user01 계정으로 접속하여 '/client/device/user01' topic 으로 "ON" 메시지를 publish 해보았다.

[User ID: user01 , publish topic: /client/device/user01]

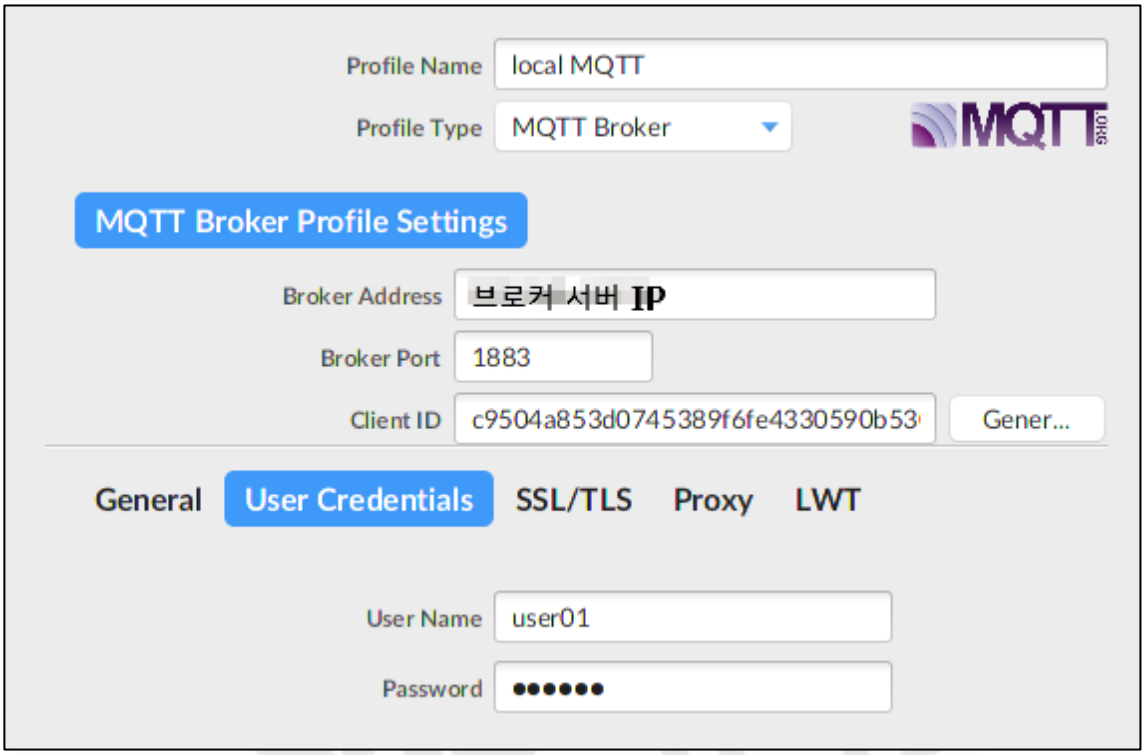


그림 56. MQTT 접속

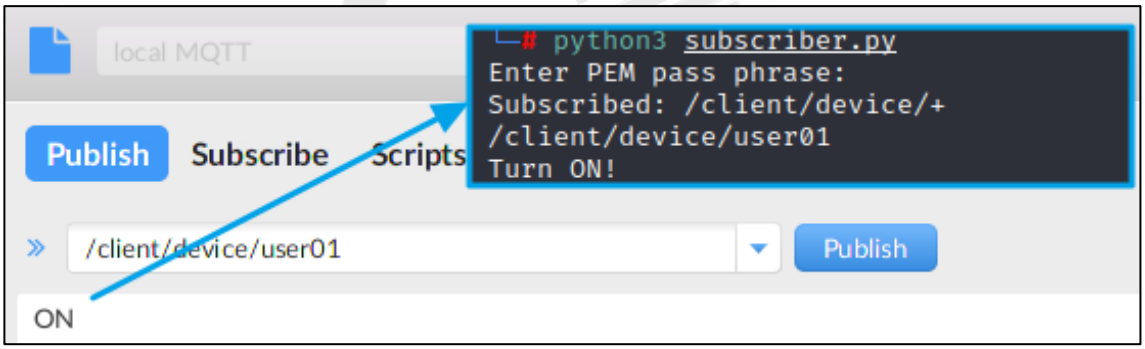


그림 57. Topic Publish

user01 계정으로 접속하여 user02 에 지정된 topic publish 가능여부를 확인하기 위해 '/client/device/user02' topic 으로 "ON" 메시지를 publish 해보았다.

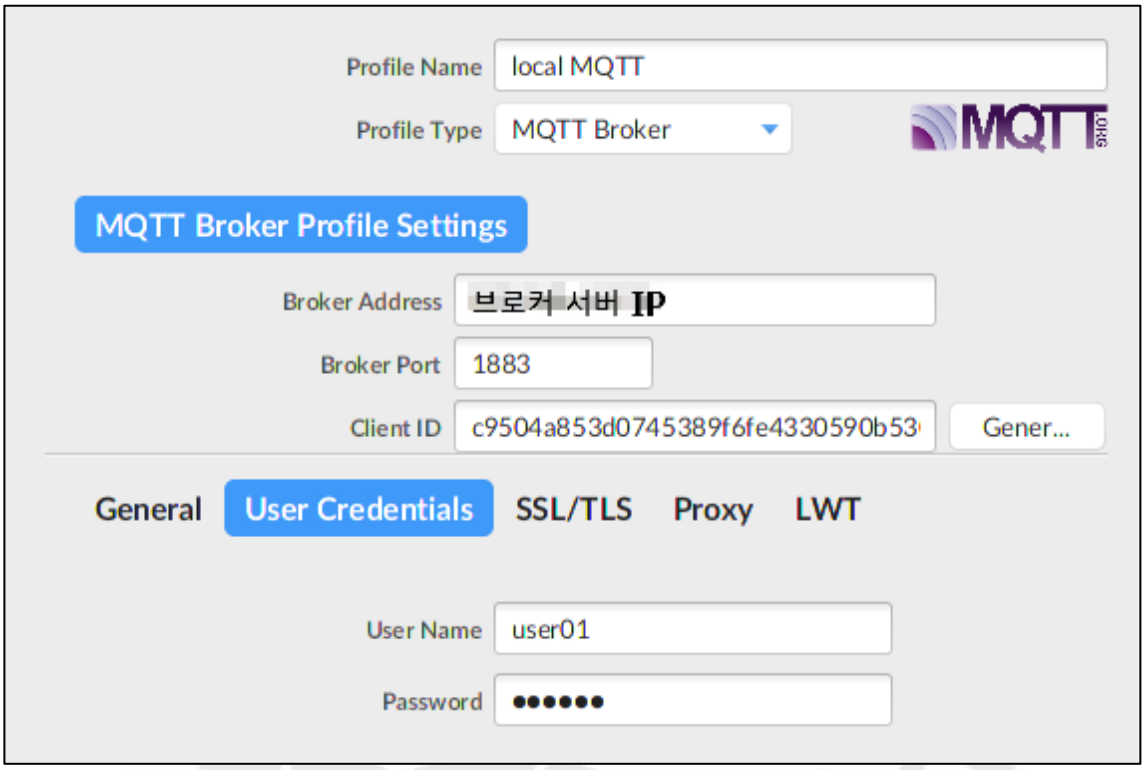


그림 58. MQTT 접속

user01 계정으로 user02 topic 사용 시 연결이 해제되는 것을 확인할 수 있다.

[User ID: user01 , publish topic: /client/device/user02]

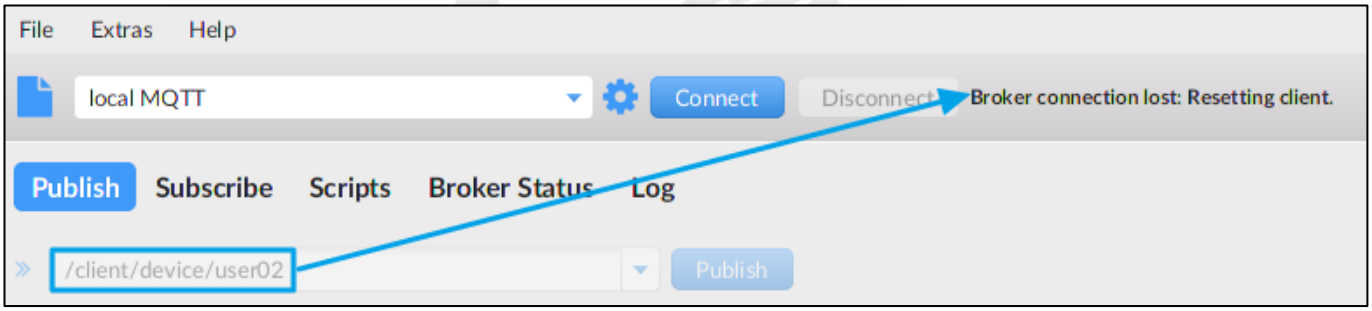


그림 59. user02 topic 사용 불가

user01 계정으로 topic wildcard 를 이용하여 다른 사용자들의 publish 메시지를 subscribe 해보았다.

user01 계정으로 wildcard 를 이용한 subscribe 시도 시 연결이 해제되는 것을 확인할 수 있다.

[User ID: user01 , subscribe topic: /client/device/+]

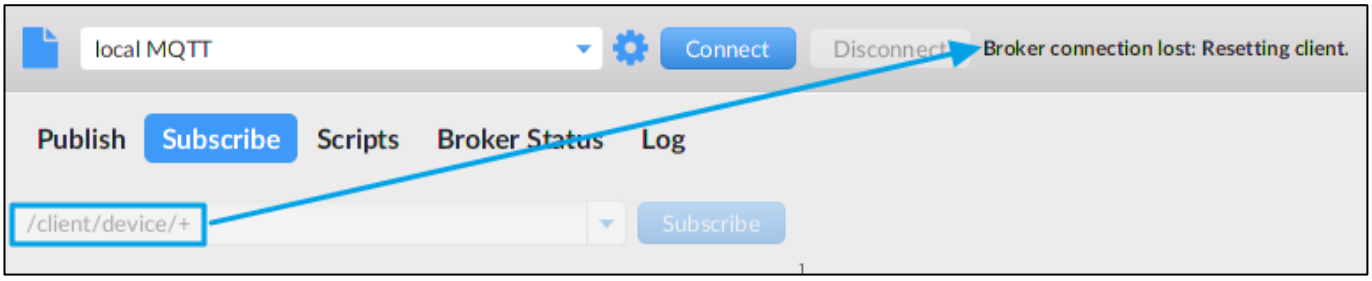


그림 60. wildcard 를 이용한 subscribe 불가

4.2. [MQ-002] 디폴트 계정 사용 여부

구분	내용
전제조건	· MQTT 통신을 사용하는 Device
취약점 설명	· [디폴트 계정 사용 여부 확인] · MQTT Broker에 디폴트로 생성되는 계정 사용이 가능하여 접근 권한이 없는 사용자가 디폴트 계정을 이용하여 서비스에 접근 및 이용이 가능한 취약점
판단 기준	· + [양호] 디폴트 계정이 존재하지 않거나 접속이 제한된 경우 · + **[취약]** 디폴트 계정이 존재할 경우 · + **[취약]** 디폴트 계정을 이용하여 서비스 접근 및 이용이 가능할 경우
취약점 영향력	· 디폴트 계정으로 임의의 사용자가 토픽 접근 가능 · 중요정보를 수집하여 2차 공격에 활용
보안대책	· 디폴트 계정 삭제 · 일부 MQTT Broker의 경우 일정 버전 이상 기본 게스트 사용자가 로컬 호스트에서만 연결 가능하므로 해당 버전 이상 사용 (ex RabbitMQ 3.3.0)

4.2.1. 양호 Case 1 (RabbitMQ)

MQTT Client 를 이용하여 RabbitMQ 서버에 디폴트 계정 접속을 시도하였다. [User ID: guest , Password: guest]

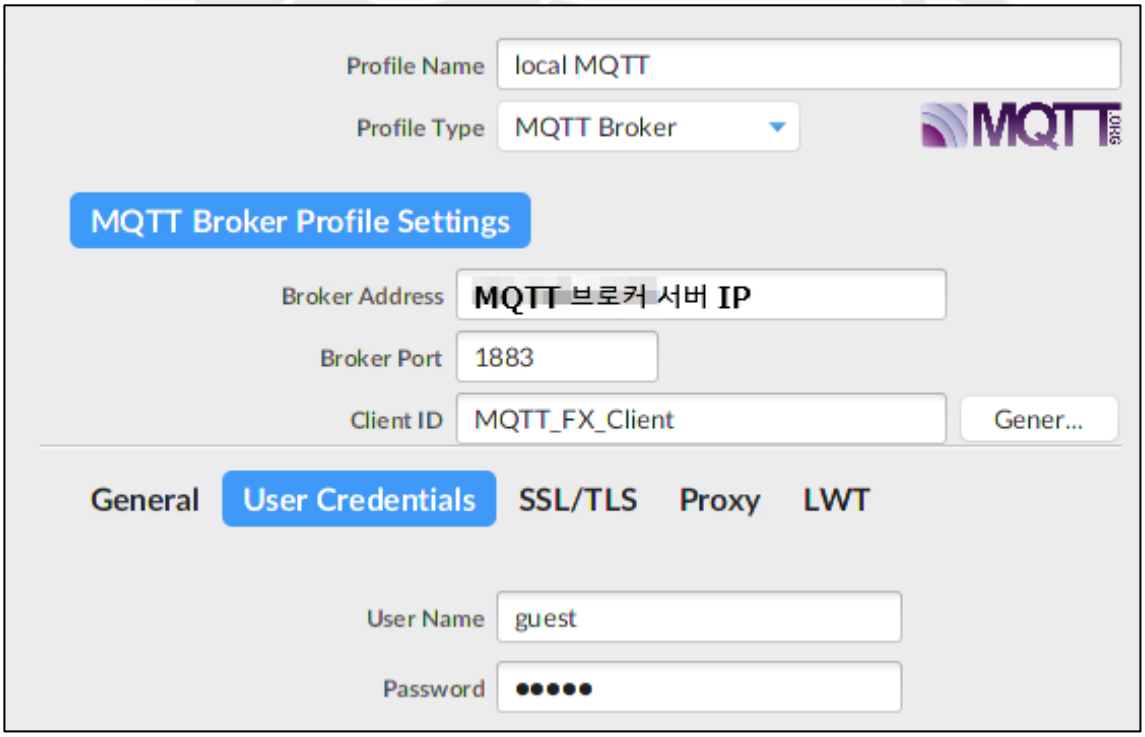


그림 61. 디폴트 계정 접근 시도

디폴트 계정이 존재하지만 해당 계정으로 접근이 불가한 것을 확인하였다.

(RabbitMQ 버전 3.3.0 이상부터 디폴트 계정(게스트) 사용자는 로컬 호스트를 통해서만 연결이 가능하다.)

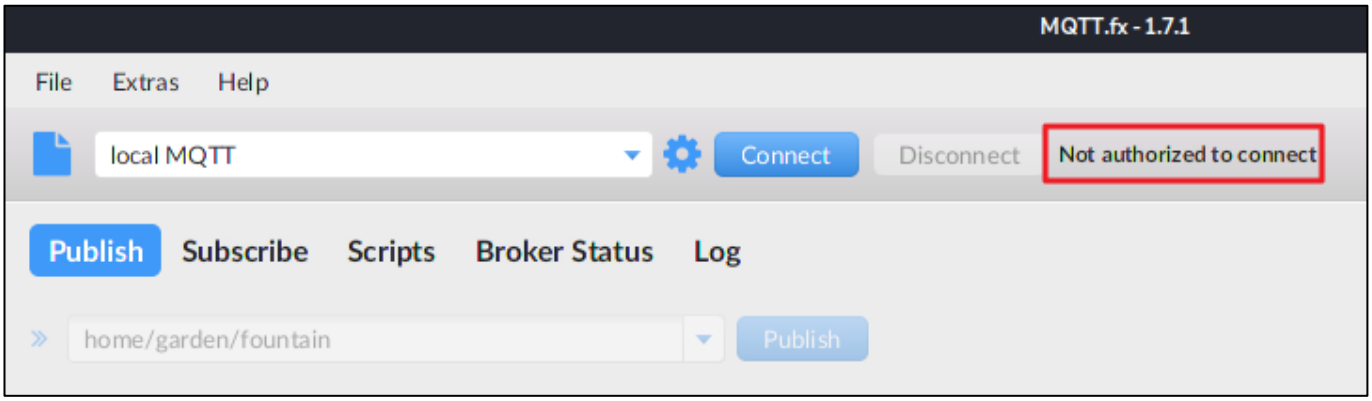


그림 62. 접근 불가

4.2.2. 취약 Case 2 (HiveMQ)

HiveMQ 의 RBAC 확장 프로그램을 설치하면 디폴트 계정이 설정되어 있다.

[일반 사용자 ID : user1 , Password : pass1 / admin ID : admin-user , Password : admin-password]

```
<users>
    <user>
        <name>user1</name>
        <!-- password hash for "pass1" -->
        <password>WFNQUVB0UkxjM04xa0hSR1BQNGhuOTJKVzdlbXA4bjk=:100:FY12nwpUEbBK9EKQ/Aw/rQKSoA7jXsC0HKE
LwU2mLCVU39bJVK0zf4NemuFeDOHPO4BW1nOjxi6NporkC6rUog==</password>
        <roles>
            <id>role1</id>
        </roles>
    </user>
    <user>
        <name>admin-user</name>
        <!-- password hash for "admin-password" -->
        <password>Vjc1a0lxQ3Nvb0ljNFVHNE9WRnM3RG1IZmdNUFcwVGY=:100:PL2FLqfpdhONG7qXjAMmdVn4wlMiXnypdXi
FW09zqorFhKgoiixFQw2EVJJfE9Zn79q45V7Xpc6JeKLp0ntmYA==</password>
        <roles>
            <id>superuser</id>
        </roles>
    </user>
</users>
```

그림 63. HiveMQ 디폴트 계정

디폴트 계정으로 접속 시도하여 topic 에 메시지를 보내는 것이 가능하였다. [Default User ID: user1 , topic: test]

```
test@ubuntu:~$ mqtt pub -t test -m test -h test.hivemq.com
CONNECT failed as CONNACK contained an Error Code: BAD_USER_NAME_OR_PASSWORD.
test@ubuntu:~$ mqtt pub -t test -m test -h test.hivemq.com -u user1 -pw pass1
test@ubuntu:~$
```

그림 64. 디폴트 계정 접속 및 topic 전송

4.3. [MQ-003] 전송 구간 보호 여부

구분	내용
전제조건	· MQTT 통신을 사용하는 Device
취약점 설명	· [MQTT 통신 구간에 대한 평문 전송 및 중요 정보 노출 여부 확인] · 통신 구간 암호화 적용이 미흡하여 중요 정보 노출 및 노출된 중요 정보 악용이 가능한 취약점
판단 기준	· + [양호] 통신 구간 SSL을 적용하고 있을 경우 · + [양호] 중요정보 암호화 적용하고 있을 경우 · + **[취약]** 중요 정보 전송 시 중요 정보 암호화 적용 미흡
취약점 영향력	· 중요정보를 수집하여 2차 공격에 활용
보안대책	· 평문 전송 시 암호화 통신(SSL) 적용 필요 · 전송 구간 내 사용자의 중요정보 암호화

4.3.1. 취약 Case 1 (RabbitMQ)

MQTT 브로커 서버로 publish 메시지를 전송하고 Wireshark 를 이용하여 MQTT 트래픽 확인 시 평문으로 전송되는 것을 확인하였다.

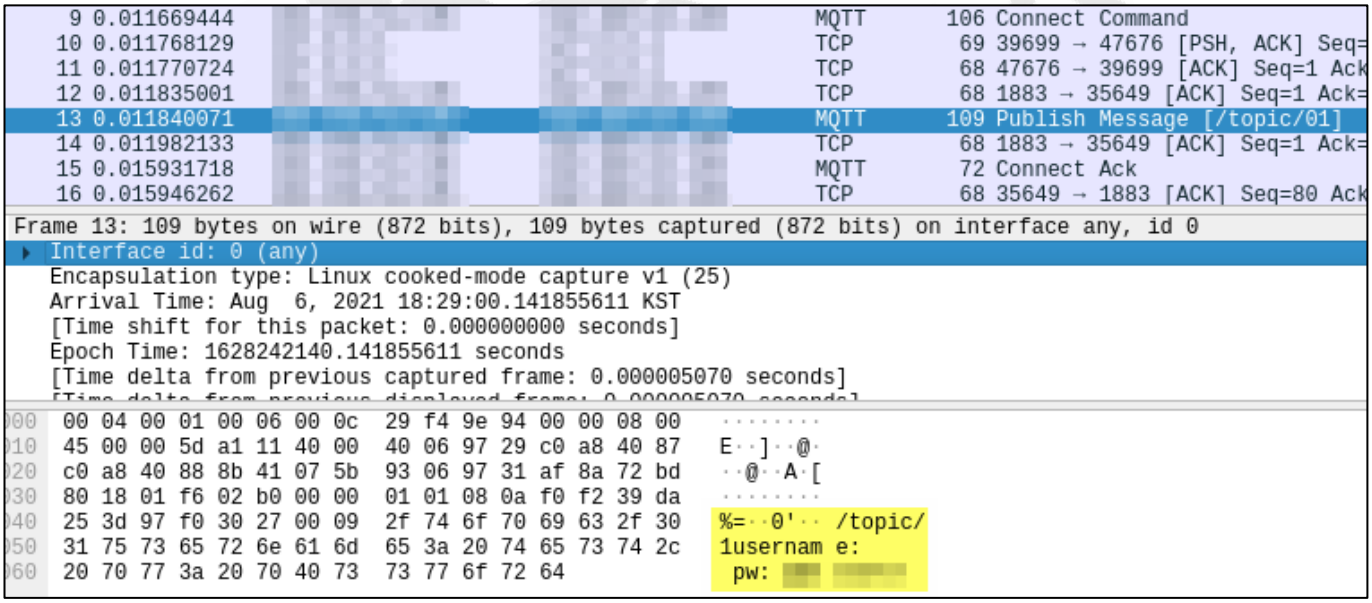


그림 65. 데이터 평문전송

4.3.2. 양호 Case 2 (RabbitMQ)

전송구간 TLS 적용 여부를 확인하기 위해 MQTT 트래픽 확인 결과 TLS 가 적용되어 있는 것을 확인하였다.

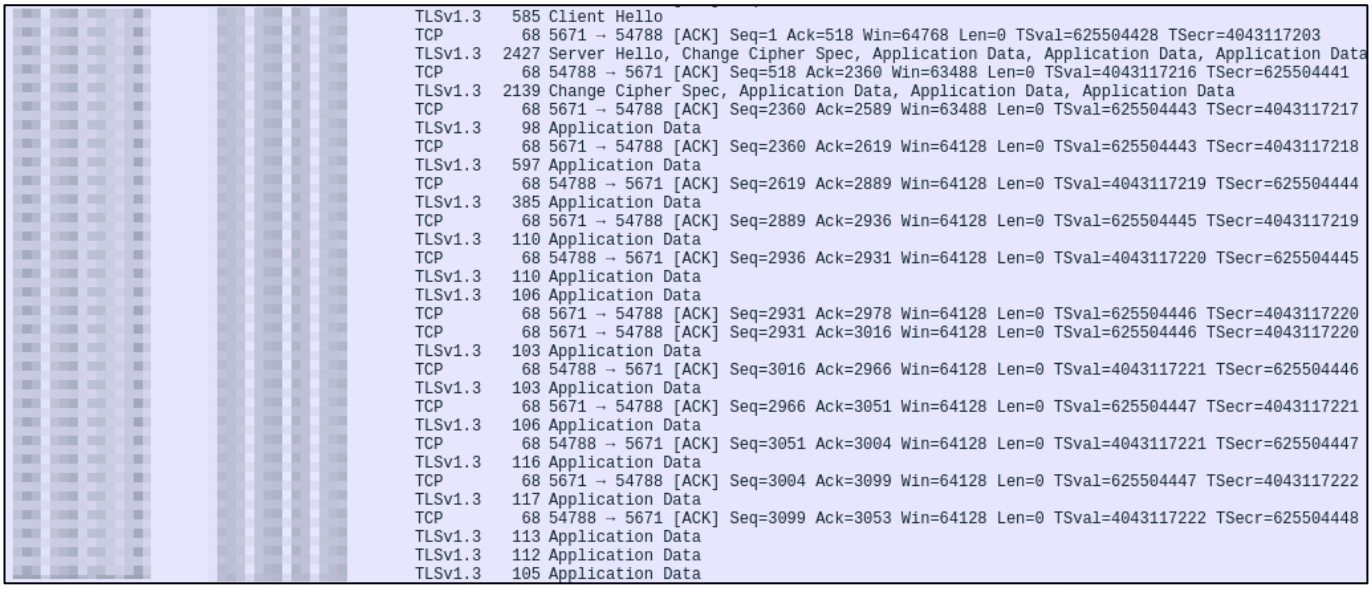


그림 66. TLS 적용 확인

4.4. [NFC-001] 카드 데이터 복제

구분	내용
전제조건	· NFC 통신을 사용하는 카드
취약점 설명	· [카드 데이터 복제 가능 여부] · 카드 내 데이터 복제 후 복제한 카드 사용이 가능한 취약점
판단 기준	· + **[취약]** 데이터 복제 및 복제 카드 사용이 가능한 경우
취약점 영향력	· 복제한 카드를 이용한 출입 인증, 결제 등 악용이 가능
보안대책	· 안전한 암호화 키를 사용하여 데이터 덤프를 방지함. - 제조사에서 제공하는 default key 변경 - 안전한 암호화 기능을 제공하는 태그 사용 (태그마다 지원하는 기능이 상이하므로 타입 별 확인 필요) · + 가능한 경우 - 유효성 검증을 수행하는 태그 사용 (태그마다 지원하는 기능이 상이하므로 타입 별 확인 필요) ex) NTAG의 경우 UID에 NXP에서 제공하는 디지털 서명 값을 추가하여 공격자가 복제하여 사용할 시 유효성 검증을 통해 복제 여부를 확인함.

4.4.1. 취약 Case 1 (도어락 출입카드)

복제 대상의 카드 타입이 MIFARE Classic 1k 임을 확인하였다. MIFARE 타입의 카드는 신용카드, 교통카드, 출입카드 등 실생활에서 많이 사용되고 있으며, MIFARE Classic 타입은 취약한 타입의 태그 중 하나이다.

그림 67. 태그 타입 확인

대상 카드는 모든 섹터가 알려진 키 값을 사용하고 있었으며, 알려진 키 값을 사용하고 있지 않은 경우 nested, darkside 등의 별도의 공격을 통해 키 값을 크랙하는 과정을 거쳐야한다. 대상 카드는 모든 섹터에 알려진 키 값을 사용하고 있었으며, 별도의 공격 없이 키 값 덤프를 실행하였다. (hf-mf-UID-key.bin 파일이 생성된다.)

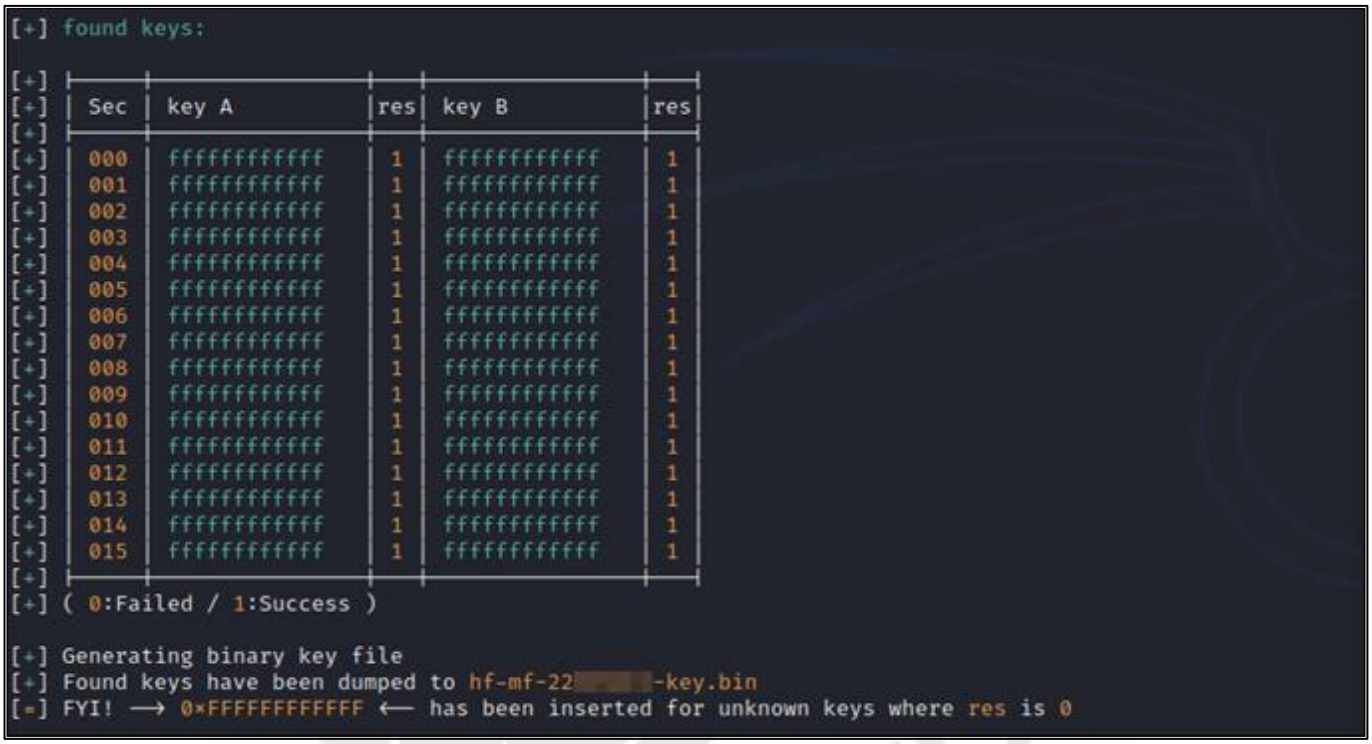


그림 68. 키 확인 및 키 파일 덤프

덤프한 키 파일을 이용하여 카드 내 데이터를 덤프하였다. (.bin, .eml, .json 세 개의 파일이 생성된다.)

그림 69. 키 파일을 이용하여 데이터 덤프

덤프한 데이터 파일을 빈 카드에 복제하였다.

그림 70. 데이터 덤프파일 복제

그림 71. 빈 카드에 덤프파일 복제

복제한 카드를 이용한 출입 인증이 가능하였다.

그림 72. 복제 카드 사용

4.5. [NFC-002] 카드 데이터 변조

구분	내용
전제조건	· NFC 통신을 사용하는 카드
취약점 설명	· [카드의 데이터 변조 가능 여부] · 카드 내 UUID, 데이터 값 변조 후 변조한 카드 사용이 가능한 취약점
판단 기준	· + **[취약]** 데이터 변조 및 변조한 카드 사용이 가능한 경우
취약점 영향력	· 데이터 변조한 카드를 이용한 출입 인증, 결제 등 악용이 가능
보안대책	· 안전한 암호화 키를 사용하여 데이터 변조를 방지함. - 제조사에서 제공하는 default key 변경 - 안전한 암호화 기능을 제공하는 태그 사용 (태그마다 지원하는 기능이 상이하므로 타입 별 확인 필요) · + 가능한 경우 - Lock bytes 설정하여 영구적으로 쓰기가 불가하도록 함 - 태그 또는 리더기에 위변조 방지 적용 (태그마다 지원하는 기능이 상이하므로 타입 별 확인 필요) ex) 버스카드의 경우 암호화 모듈이 설치된 리더기를 이용하여 탑승정보 암호화 저장

4.5.1. 양호 Case 1 (교통카드)

대상 카드 타입이 MIFARE Plus 이고, SL mode 가 3 으로 설정되어 AES 암호화가 되어있음을 확인하였다.

```
[usb] pm3 --> hf mfp info

[=] --- Tag Information ---------------------------
[=] ------------------------------------------------------------
[!] No card response.

[+]  UID: 00 52 [redacted]
[+] ATQA: 00 04
[+]  SAK: 20 [1]
[+] Possible types:
[+]    MIFARE Plus EV1 2K/4K in SL3
[+]    MIFARE Plus S 2K/4K in SL3
[+]    MIFARE Plus X 2K/4K in SL3
[+]    MIFARE Plus SE 1K
[+]    NTAG 4xx

[=] --- Fingerprint
[=]           SIZE: 2K (4 UID)
[=]            SAK: MIFARE Plus SL0/SL3 or MIFARE DESFire
[!] No card response.
[=] Send copy to iceman of this command output!
[=] data:
[=]         result: MIFARE Plus SL0/SL3
[=] --- Security Level (SL)
[+]        SL mode: SL3
[=]   SL 3: 3-Pass authentication based on AES, data manipulation commands secured by AES encryption and an AES
 based MACing method.
```

그림 73. 태그 타입 확인

알려진 키를 사용하고 있지 않음을 확인하였다.

```
[usb] pm3 --> hf mfp chk
[=] Loaded 26 keys
Search keys
.RE
[=] No keys found(
```

그림 74. 태그 타입 확인

키 파일 획득이 불가하여 데이터 확인 및 변조가 불가함을 확인하였다.

4.6. [NFC-003] 취약한 암호화 키 사용 여부

구분	내용
전제조건	· NFC 통신을 사용하는 카드
취약점 설명	· [취약한 암호화 키 사용 여부] · 알려진 키를 사용하거나 보안성이 낮은 카드 카드를 사용하여 키 크랙이 가능한 취약점
판단 기준	· + **[취약]** 모든 섹터에 알려진 키 사용 · + **[취약]** 1개 이상의 알려진 키를 사용하여 nested, hardnested 공격 등을 이용한 · 키 크랙이 가능한 경우 · + **[취약]** NACK 취약점이 존재하여 darkside 공격을 이용한 키 크랙이 가능한 경우 · + **[취약]** 알기 쉬운 키 사용하여 dictionary 공격을 이용한 키 크랙이 가능한 경우 · + **[취약]** 기타 추가적인 공격을 통한 키 크랙이 가능한 경우
취약점 영향력	· 획득한 키를 이용하여 카드 내 데이터 복호화 및 복제 등 악용에 활용
보안대책	· 제조사에서 설정한 default key 변경하여 사용 · 안전한 암호화 키 사용하여 데이터 암호화 · 안전한 버전의 태그 사용 (태그마다 지원하는 보호기법이 상이하므로 타입 별 확인 필요) ※ 주로 사용되는 MIFARE 계열 태그의 경우 MIFARE PLUS 이상의 태그를 사용하며, MIFARE PLUS 사용 시 보안 수준을 SL3으로 설정

4.6.1. 취약 Case 1(신용카드)

대상 카드 타입이 MIFARE Classic 1k 이고, prng detection 이 weak 인 것을 확인하였다.

```
[usb] pm3 --> hf search
 ⓘ  Searching for ISO14443-A tag ...
[+]  UID: E7 [redacted]
[+] ATQA: 00 04
[+]  SAK: 88 [2]
[+] Possible types:
[+]    MIFARE Classic 1K
[=] proprietary non iso14443-4 card found, RATS not supported
[+] Prng detection: weak
[#] Auth error
[?] Hint: try `hf mf` commands


[+] Valid ISO 14443-A tag found
```

그림 75. 태그 타입 확인

섹터 암호화 키 확인 시 일부 섹터에서 알려진 키를 사용하고 있음을 확인하였다. (FFFFFFFFFFFF 를 사용중이다.)

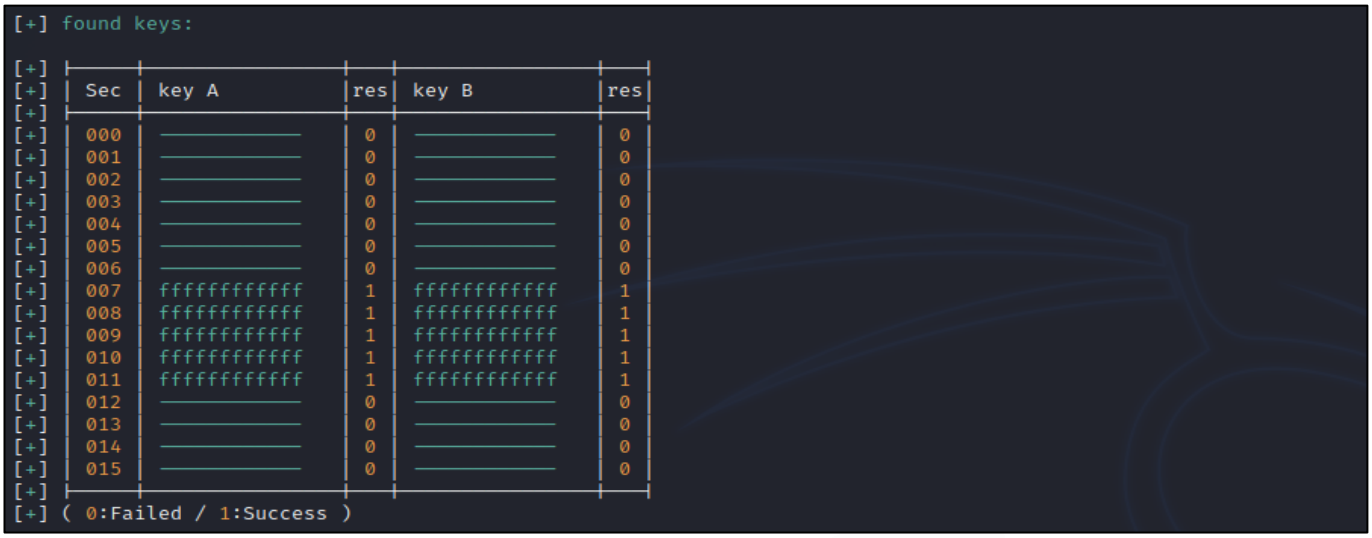


그림 76. 알려진 키 사용 여부 확인

알려진 키(FFFFFFFFFFFF)를 사용하여 nested 공격을 시도하였다.

그림 77. nested 공격 시도

nested 공격에 성공하여 나머지 섹터의 키 획득이 가능하였다.

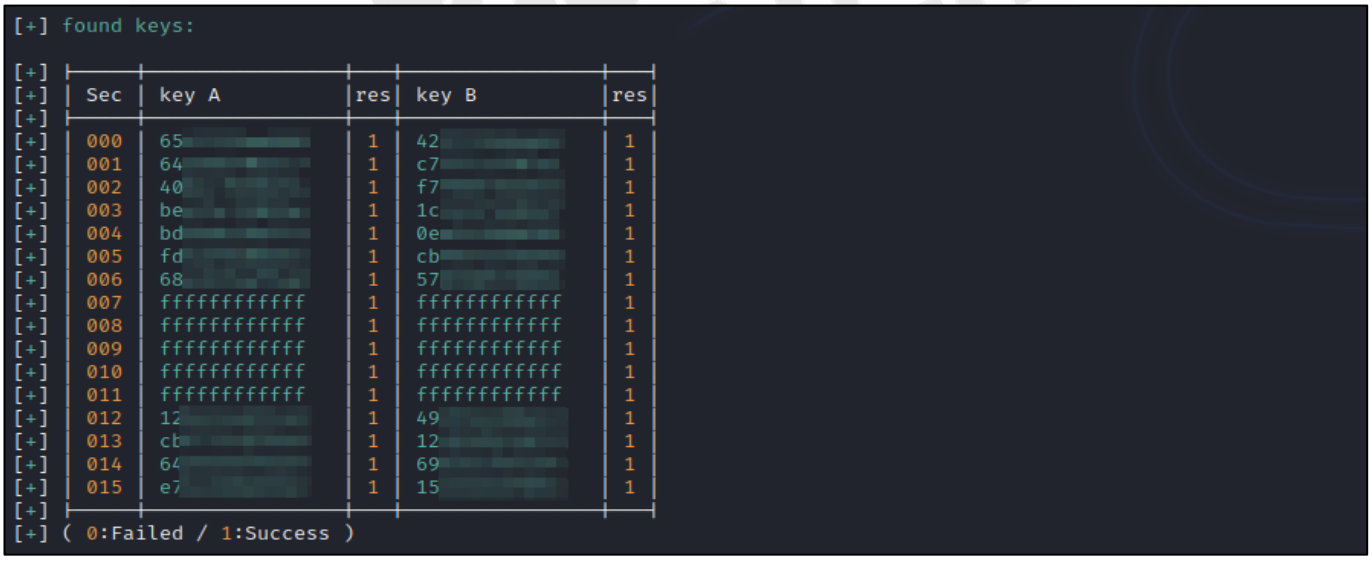


그림 78. nested 공격을 이용한 모든 섹터 키 크랙

획득한 키를 이용하여 데이터 덤프 및 데이터 확인이 가능하였다.

```
[+] Succeeded in dumping all blocks

[+] saved 1024 bytes to binary file hf-mf-E7[redacted]-dump.bin
[+] saved 64 blocks to text file hf-mf-E7[redacted]-dump.eml
[+] saved to json file hf-mf-E7[redacted]-dump.json
```

그림 79. 키 파일을 이용한 데이터 덤프

```
┌──(root💀kali)-[/home/kali/iot/proxmark3]
└─# xxd hf-mf-E79B78DA-dump.bin
00000000: [redacted]  ..x.....C'......
00000010: [redacted]  32.0............
00000020: [redacted]
00000030: [redacted]
00000040: [redacted]
00000050: [redacted]
00000060: [redacted]
00000070: [redacted]
00000080: [redacted]
00000090: [redacted]
000000a0: [redacted]
000000b0: [redacted]
000000c0: [redacted]
000000d0: [redacted]
000000e0: [redacted]
```

그림 80. 데이터 확인

덤프 데이터를 빈 카드에 복제한 결과 복제 카드의 사용이 가능하였다.

그림 81. 복제 카드를 이용하여 교통카드 잔액 확인

그림 82. 잔액 조회 성공

5. 웹/모바일 점검 상세

구분	내용
전제조건	· Mobile App을 통한 IoT 기기 제어 수행
취약점 설명	· [Server, App 취약점 항목 확인] · Device와 연동되는 Mobile APP 및 Server에 대한 취약점 점검을 수행
참고 사항	· 일반적으로 진단 시 확인 가능한 구성은 아래와 같음 - [Server] API Server, Cloud Server, Auth Server, Commnad Server 등 - [APP] Mobile APP (Cloud) - [Device] IoT Device ※**[Server]** IoT 연동 Server는 대부분 UI가 별도 제공되지 않으며 Json, xml 등의 형태로 통신을 수행한다. 이에 따라 웹 점검 항목을 선정하여 반드시 점검이 필요하다. 대표적으로 반드시 보아야 하는 취약점 들은 다음과 같다. - 사용자 입력 값에 의한 취약점 (SQL Injection, XSS, 파일 업로드 등) - 인증관련 항목 (인증 누락, 정보누출 등) - 서버 설정, 백업파일 관련 항목 ※**[Mobile]** 참고 사항으로 아래와 같은 사항을 고려하여 위험 발생 요소가 있을 경우 취약으로 볼 수 있다. - 임의로 디바이스 등록, 해제, 초기화 가능 여부 - 사용자 결제 정보 등록, 변경, 해제 시 확인 절차가 없어 임의로 변경 - 디바이스 분실 가능성에 의한 초기화 혹은 Lock 기능 없음
취약점 영향력	· 주요정보통신기반시설 웹/모바일 가이드 참고하여 작성
보안대책	· 주요정보통신기반시설 웹/모바일 가이드 참고하여 작성

표 16. 웹/모바일 점검

EQST

2022.02

SK쉴더스㈜ 13486 경기도 성남시 분당구 판교로227번길 23, 4&5층
https://www.skshieldus.com

발행인 : SK쉴더스 EQST그룹
제　작 : SK쉴더스 커뮤니케이션그룹