

# Revisiting Userspace Reachability to Baseband Processors: A Forensic Analysis of MediaTek CCCI Interfaces on MT6765 Platforms

Emmanuel Casarrubias

Independent Security Researcher

`emmanuel.casarrubias.j@gmail.com`

March 2026

## Abstract

This paper presents a forensic analysis of the Cross-Core Communication Interface (CCCI) driver on MediaTek MT6765 platforms, as implemented in Samsung Galaxy A12 (SM-A125M) devices. We identify multiple modem-facing interfaces accessible from userspace processes holding RADIO permission (UID 1001). Experimental results show that structured 48-byte payloads sent via `ioctl` on `/dev/ccci_ioctl0` produce correlated kernel log responses consistent with modem interaction. These interfaces remain active while the device is in airplane mode, with no active cellular or Wi-Fi connectivity. While this work does not claim the presence of an exploitable vulnerability, it documents observable behavior that may expand the reachable attack surface of the baseband processor beyond expected isolation boundaries. These findings highlight the need for clearer documentation and security boundary definitions for modem-facing interfaces in modern mobile platforms. The observations presented in this work contribute to a more precise understanding of modem interface exposure in contemporary mobile systems.

## 1 Introduction

Modern smartphone architectures separate the Application Processor (AP) from the Baseband Processor (BP) to limit risk exposure. Communication between these components is required for normal operation and is typically handled through controlled interfaces. MediaTek platforms implement this communication through the Cross-Core Communication Interface (CCCI), which is used by the Radio Interface Layer (RIL). While expected in design, the exposure and accessibility of these interfaces from userspace merit closer examination.

Prior research has documented vulnerabilities in the CCCI driver. TASZK Security Labs identified CVE-2022-21765 and CVE-2022-21769, out-of-bounds read/write vulnerabilities in the ringbuffer implementation that could be exploited from a compromised baseband to escalate privileges on the AP [1]. CVE-2023-32840 describes another out-of-bounds write in modem CCCI [3]. These findings demonstrate that the CCCI interface has been a recurring subject of security analysis.

This work documents observable behavior related to these interfaces and evaluates their implications from a security architecture perspective, complementing prior research by examining userspace-facing interfaces rather than baseband-to-kernel attack vectors.

## 2 Threat Model

We consider a scenario where execution is obtained within a privileged userspace context (e.g., a compromised system component with RADIO permission). In this context, access to modem-facing interfaces may expand the reachable attack surface beyond typical expectations of AP-BP isolation. This work does not evaluate exploitability, but rather documents the reachability of these interfaces under normal operating conditions.

### 3 Device Under Test

| Parameter            | Value  |
|----------------------|--|
| Device               | Samsung Galaxy A12 (SM-A125M)  |
| Chipset              | MediaTek MT6765  |
| Kernel Version       | 4.19.188-25461540-abA125MUBS6CXJ1                                    |
| Build Fingerprint    | samsung/a12ub/a12:12/SP1A.210812.016/A125MUBS6CXJ1:user/release-keys |
| Android Version      | 12   |
| Security Patch Level | 2024-10-01   |
| Bootloader           | A125MUBS6CXJ1 (unlocked)   |

Table 1: Test Device Configuration

### 4 Methodology

Our analysis employed the following methods:

- **Static Analysis:** Binary examination of system libraries (libsec-ril.so, libril\_sem.so, libccci\_util.so) and CCCI-related binaries (ccci\_mdinit, emdlogger1, meta\_tst)
- **Dynamic Analysis:** Live tracing of ioctl calls, process mappings, and inter-process communication
- **Kernel Log Analysis:** Real-time monitoring of CCCI communication channels with dmesg capture
- **Permission Auditing:** Systematic review of device node permissions and SELinux contexts
- **Thread Analysis:** Identification of thread creation routines in vendor libraries

All experiments were conducted with airplane mode enabled to eliminate dependency on active cellular connectivity.

### 5 Airplane Mode Confirmation

All identified interfaces remained active when the device was in airplane mode. This suggests that these communication channels are not gated by radio state transitions, but operate independently at the driver level.

```
1 $ adb shell getprop persist.radio.airplane_mode_on
2 1
3
4 $ adb shell dumpsys telephony.registry | grep -i "serviceState"
5 mVoiceRegState=3(POWER_OFF), mDataRegState=3(POWER_OFF)
```

Listing 1: Airplane Mode Verification

### 6 Observed Interfaces

The following interfaces were observed to be connected to modem-facing components. This does not imply direct data flow semantics for all entries, but indicates potential communication paths:

```
1 crw-rw---- 1 audio  audio      498, 159 /dev/ccci_raw_audio
2 crw-rw---- 1 radio  radio      498, 12  /dev/ccci_ioctl0
3 crw-rw---- 1 radio  radio      498, 20  /dev/ccci_rpc
4 crw-rw---- 1 radio  radio      498, 28  /dev/ccci_umts_ipc0
5 crw-rw---- 1 system radio      10, 122 /dev/umts_router
6 crw-rw---- 1 system radio      10, 124 /dev/umts_ipc0
```

Listing 2: Active CCCI Device Nodes

These remained accessible during airplane mode.

## 7 Critical Process Analysis

The following processes were observed maintaining active modem communication channels:

| Component   | PID     | UID           | Modem Interface                |
|-------------|---------|---------------|--------------------------------|
| ccci_mdinit | 756     | system (1000) | /dev/ccci_ioctl0               |
| ccci_rpcd   | 763     | radio (1001)  | /dev/ccci_rpc                  |
| smdexe      | 879     | system (1000) | /dev/umts_router               |
| rild        | 924     | radio (1001)  | /dev/umts_ipc0                 |
| emdlogger1  | 1160    | shell         | @com.mediatek.mdlogger.socket1 |
| audioserver | 667/729 | audioserver   | /dev/ccci_raw_audio            |

Table 2: Persistent Modem-Facing Processes

## 8 ioctl Interaction

A structured 48-byte payload sent via `ioctl 0xc004c404` on `/dev/ccci_ioctl0` produced kernel log responses correlated with the transmitted data:

```
1 [136473.704356] mif: RX: 00 00 00 00 29 00 00 00 c1 00 2c d1 00 00 00 00 19 00 ff ff 07
   06 03 73 63 63 00 73 11 8c 00 ff
2 [136473.704386] mif: RX: 19 00 ff ff 07 06 03 73 63 63 00 73 11 8c 00 ff ff fb ff ff ff
   ff ff ff 00 00 00 00 5c 00 4e 00
```

Listing 3: Kernel Logs Showing Correlated Modem Response

The correlation between transmitted payloads and received logs is consistent with bidirectional communication across the AP-BP boundary. While internal baseband handling cannot be directly verified, observable behavior supports the conclusion of interface reachability.

### 8.1 Proof of Concept

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 #include <unistd.h>
5 #include <sys/ioctl.h>
6 #include <string.h>
7
8 #define CCCI_IOCTL_SEND 0xc004c404
9 #define PAYLOAD_SIZE 48
10
11 int main(int argc, char *argv[]) {
12     int fd;
13     unsigned char payload[PAYLOAD_SIZE];
14
15     memset(payload, 0x41, PAYLOAD_SIZE);
16     *(uint32_t*)payload = 0x00000029;
17
18     fd = open("/dev/ccci_ioctl0", O_RDWR);
19     if (fd < 0) {
20         perror("open");
21         return 1;
22     }
23
24     int ret = ioctl(fd, CCCI_IOCTL_SEND, payload);
25     close(fd);
26     return ret;
27 }
```

Listing 4: PoC Implementation

## 9 Permission Model

Access to these interfaces requires RADIO permission (UID 1001), which is typically restricted to system components and privileged applications. However, this still represents a broader exposure than expected for modem-facing interfaces.

## 10 Diagnostic Socket

A persistent diagnostic socket was observed:

```
1 shell          1160          1 2264360    1364 0          0 S emdlogger1
2 Socket: @com.mediatek.mdlogger.socket1 (reference 12262) - LISTENING
```

Listing 5: Diagnostic Socket

No authentication mechanism was observed during testing. Further analysis is required to determine whether access control is enforced at other layers. This interface was not identified in any public documentation reviewed during this investigation.

## 11 Binary Analysis

Examination of system binaries revealed the presence of thread creation routines:

```
1 === libril_sem.so ===
2 pthread_create
3 ...
4 _Z13ril_event_addP9ril_event
5 _Z13ril_event_setP9ril_eventibPFvisPvES1_
6 _Z14ril_event_initv
7 _Z14ril_event_loopv
```

Listing 6: Thread Creation Strings

## 12 Discussion

### 12.1 Architectural Implications

The findings indicate that modem-facing interfaces remain reachable under conditions where higher-level connectivity is disabled. This does not imply a vulnerability by itself, but highlights a security boundary that may be less restrictive than commonly assumed.

The persistence of these interfaces in airplane mode suggests that these communication channels are not gated by radio state transitions, but remain operational at the driver level independent of network connectivity.

This distinction is important, as it shifts the security discussion from vulnerability presence to boundary definition and attack surface exposure.

### 12.2 Comparison with Prior Work

The interfaces documented in this paper differ from previously reported CCCI vulnerabilities in their accessibility. CVE-2022-21765 and CVE-2022-21769 required baseband compromise to exploit [2]. In contrast, the interfaces examined here are accessible from privileged userspace contexts without requiring prior baseband compromise.

### 12.3 Limitations

- The analysis was conducted on a single device model (SM-A125M) with a specific chipset (MT6765)
- The study did not include analysis of the baseband firmware itself
- The investigation did not attempt to exploit these interfaces beyond confirming data reachability
- The behavior of these interfaces on other MediaTek chipsets was not verified

## 12.4 Recommendations

1. Implement capability-based access control for modem communication interfaces
2. Require explicit authorization beyond Unix permissions for sensitive ioctl operations
3. Audit device nodes exposed to userspace
4. Establish SELinux policies restricting modem communication to necessary system components
5. Review the security implications of maintaining these interfaces active in airplane mode

## 13 Responsible Disclosure Timeline

| Date           | Event  |
|----------------|--|
| March 13, 2026 | Initial report submitted to MediaTek Security Team                                   |
| March 16, 2026 | MediaTek acknowledges receipt  |
| March 18, 2026 | Complete forensic evidence provided  |
| March 19, 2026 | PoC and kernel logs demonstrating data reachability provided                         |
| March 20, 2026 | MediaTek closes report: "does not meet the bar for inclusion in a security bulletin" |
| March 24, 2026 | Public disclosure (this paper)   |

Table 3: Disclosure Timeline

MediaTek's final assessment stated that access to `/dev/ccci_ioctl0` by apps holding radio permission is "intended behavior" and that no security issue exists. This paper documents the observed behavior for independent evaluation.

## 14 Conclusion

This paper documents observable userspace reachability to modem-facing interfaces on MediaTek MT6765 platforms. Structured input produces responses that are consistent with modem interaction, even in airplane mode. While this work does not claim the presence of an exploitable vulnerability, it documents behavior that may expand the reachable attack surface of the baseband processor beyond expected isolation boundaries.

These findings highlight the need for clearer documentation and security boundary definitions for modem-facing interfaces in modern mobile platforms. The observations presented in this work contribute to a more precise understanding of modem interface exposure in contemporary mobile systems.

## References

- [1] TASZK Security Labs, "Full Chain Baseband Exploits, Part 3," December 2023. [https://labs.taszk.io/articles/post/full\\_chain\\_bb\\_part3/](https://labs.taszk.io/articles/post/full_chain_bb_part3/)
- [2] TASZK Security Labs, "CVE-2022-21769: Mediatek CCCI Kernel Driver OOB Read," November 2023. [https://labs.taszk.io/blog/post/82\\_mtk\\_ccci2\\_oob\\_read/](https://labs.taszk.io/blog/post/82_mtk_ccci2_oob_read/)
- [3] NVD, "CVE-2023-32840," <https://nvd.nist.gov/vuln/detail/CVE-2023-32840>

## A Reproduction Steps

1. Obtain device with MT6765 chipset (Samsung Galaxy A12 SM-A125M) and unlock bootloader
2. Enable airplane mode
3. Verify device nodes: `adb shell ls -la /dev/ccci_ioc10`
4. Monitor kernel logs: `adb shell dmesg -w | grep "mif: RX"`
5. Build and execute the PoC binary
6. Observe modem responses in kernel logs

## B Contact Information

- Email: `emmanuel.casarrubias.j@gmail.com`
- GitHub: <https://github.com/EmmanuelCasarrubias>