Oh No, My RAN! Breaking Into an O-RAN 5G Indoor Base Station

Leon Janzen, Lucas Becker, Colin Wiesenäcker,
and Matthias Hollick, *Technical University of Darmstadt (TUDa)*

https://www.usenix.org/conference/woot24/presentation/janzen

This paper is included in the Proceedings of the
18th USENIX WOOT Conference on Offensive Technologies.

August 12–13, 2024 • Philadelphia, PA, USA

ISBN 978-1-939133-43-4

Open access to the
Proceedings of the 18th USENIX WOOT
Conference on Offensive Technologies
is sponsored by USENIX.

Oh No, My RAN! Breaking Into an O-RAN 5G Indoor Base Station

Leon Janzen, Lucas Becker, Colin Wiesenäcker, Matthias Hollick
Technical University of Darmstadt (TUDa)
{ljanzen, lbecker, cwiesenaecker, mhollick}@seemoo.de

Abstract

Indoor base stations are expected to play a crucial role in 5G and beyond, as they are required to provide millimeter wave connectivity in buildings. However, they are a prime target for attacks, as they are difficult to secure against physical access attacks and highly connected within the RAN, especially for Open Radio Access Network (O-RAN) indoor base stations. In this work, we develop and introduce a threat model for indoor base stations. We conduct a security analysis of a proprietary O-RAN Radio Unit and present four novel vulnerabilities. Further, we analyze the Radio Unit regarding its hardware, software, and services, highlighting deviations from the O-RAN standards. The vulnerabilities we discover lead to remote code execution on the Radio Unit, highlighting security issues arising from the novel attack surface introduced by indoor base stations.

1 Introduction

Two trends in the fifth-generation technology standard for cellular networks (5G) make indoor base stations (BSs) a prime target for attacks, especially in the Open Radio Access Network (O-RAN): (1) Achieving physical access control for indoor BSs is hard, if not infeasible, and (2) indoor BSs are highly connected within the O-RAN.

While mobile network operators (MNOs) have thoroughly designed policies regulating the security of outdoor BSs, *physical access control is impractical* for indoor BSs. Unlike outdoor BSs, typically secured with fences, security cameras, and stringent physical access control measures [14], indoor BSs are often deployed on walls or ceilings, similar to enterprise Wi-Fi routers [40]. As a result, only some of the outdoor BS policies apply to indoor BSs. This lack of access control exposes indoor BSs to potential physical port access by attackers, significantly expanding the attack surface of the Radio Access Network (RAN) and the cellular network. The aspect of cellular network security has received limited attention in the research community so far, which motivates us to **introduce a threat model for indoor BSs**.

O-RAN BSs are highly connected using various interfaces to communicate with other cellular network componens. In O-RAN, the BS is disaggregated into several components [52], leaving only the Radio Unit (RU) deployed at the cell site (Figure 1). Within the O-RAN ecosystem, the RU directly interfaces a Distributed Unit (DU) and the Service Management Orchestration Framework (SMO) featuring one of the RAN Intelligent Controllers (RICs) [52]. The O-RAN Alliance has released specifications for the Open Fronthaul Interface surrounding the RU [46, 47]. In this work, **we conduct a security analysis of a proprietary O-RAN RU** to evaluate how vendors implement the specifications in the real world.

Physical access to indoor BS makes adjacent attacks on the RAN feasible, drawing attention to the security of RAN hardware. We deem this novel attack surface one of the major security challenges for RAN vendors and MNOs. To highlight this issue, **we present four vulnerabilities we discovered on a proprietary O-RAN RU**, two exploitable to achieve Remote Code Execution (RCE). In summary, our key contributions are as follows:

- We develop a threat model for indoor BSs (Section 3).
- We conduct a security analysis of a real-world O-RAN RU, highlighting deviations from the Open Fronthaul standards (Section 4).
- We present four vulnerabilities we discovered on a proprietary O-RAN RU (Section 5), which we classify as high or critical.
- We discuss our findings in the context of research trends for future cellular networks, including mitigation of the found exploits (Section 6).

We responsibly disclosed all identified issues to *Airspan Networks Inc.* and are in the process of requesting Common Vulnerabilities and Exposures (CVE) entries for our findings.

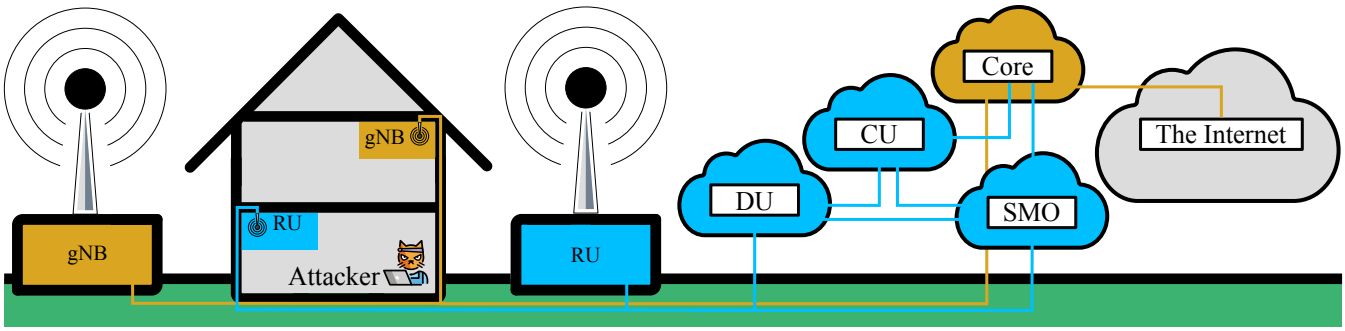


Figure 1: In conventional cellular networks, users connect to indoor or outdoor Next Generation NodeBs (gNBs) that directly connect to the core network (CN), from where traffic is forwarded to the Internet. The Open Radio Access Network (O-RAN) disaggregates the gNB into Radio Unit (RU), Distributed Unit (DU), and Central Unit (CU) with an additional Service Management Orchestration Framework (SMO). DU, CU, and SMO are typically virtualized and deployed remotely.

2 Background and Related Work

This section introduces relevant concepts and terminology of 5G networks (Section 2.1), the Open Radio Access Network (O-RAN) (Section 2.2), and indoor base stations (BSs) (Section 2.3) before summarizing related work (Section 2.4).

2.1 5G Cellular Networks

As depicted in Figure 1, in conventional Radio Access Networks (RANs), the user equipment (UE) connects to a Next Generation NodeB (gNB) that handles all layers of the 3rd Generation Partnership Project (3GPP) stack [2] from the physical layer (PHY) to the Radio Resource Control (RRC) and Service Data Adaptation Protocol (SDAP) and sends user traffic to the core network (CN) [3]. The CN is the central point of the cellular network, providing numerous core network functions (NFs), e.g., user authentication, session management, access- and mobility management, and policy control [5]. When users access the Internet via 5G, the CN converts user plane (U-Plane) traffic from the 3GPP stack to the Internet Protocol (IP) stack and forwards it to the Internet.

2.2 O-RAN and Open Fronthaul

The O-RAN-specific parts of the cellular network are highlighted in blue in Figure 1. One of the innovative ideas of the O-RAN is that the gNB is disaggregated into three components [41], as depicted in Figure 2: The Radio Unit (RU) handles the radio frequency (RF) connectivity and lower PHY [4] before sending user traffic via the Open Fronthaul interface [46, 47] to the Distributed Unit (DU) [41]. The DU handles the remaining upper PHY, the medium access control (MAC) layer, and the Radio Link Control (RLC) layer. Finally, the Central Unit (CU) handles the Packet Data Convergence Protocol (PDCP) and RRC layers before forwarding the traffic to the CN [41, 52]. In contrast to the RU, which is deployed physically at the cell site, DU and CU are typically virtualized [10]. The CN, CUs, DUs, and RUs often build a tree topology where multiple RUs connect to one DU, multiple DUs connect to one CU, and multiple CUs connect to the CN [41, 52]. The O-RAN Alliance uses the data modeling language Yet Another Next Generation (YANG) to model Network Configuration Protocol (NETCONF) configuration and state data of O-RAN components and interfaces. Thus, NETCONF and YANG models facilitate standardization and interoperability between O-RAN vendors.

The Open Fronthaul interface is one of the numerous interfaces in O-RAN, connecting RU and DU. While control and user plane (CU-Plane) traffic is sent via enhanced Common Public Radio Interface (eCPRI), synchronization plane (S-Plane) traffic is sent via Precision Time Protocol (PTP) [46]. Management plane (M-Plane) traffic is sent via NETCONF [21, 47].

The above description suits the O-RAN Split 7.2x [52], where the RU/DU split is within the PHY. Other popular O-RAN splits are Split 6 below the MAC layer [1], also referred to as network functional application platform interface (nFAPI) [56] and preferred by the Small Cell Forum (SCF) [57] or Split 8 above the analog-to-digital and digital-to-analog converter (ADC/DAC) [2].

2.3 Indoor Base Stations

Indoor BSs are expected to play a crucial role in 5G and beyond to utilize the extremely high frequency (EHF) band for millimeter waves (mmWave) communications [2, 62]. Vendors of indoor BSs include *Airspan Networks Inc.* (Airspan), *Nokia Corporation* (Nokia), *Telefonaktiebolaget LM Ericsson* (Ericsson), and *Hon Hai Precision Industry Co., Ltd.* (Foxconn). **This paper focuses on the Airspan AirVelocity 2700 (AV2700)** because it is intended for indoor deployments, supports mmWave communications, and is compatible with O-RAN Split 7.2x [7].

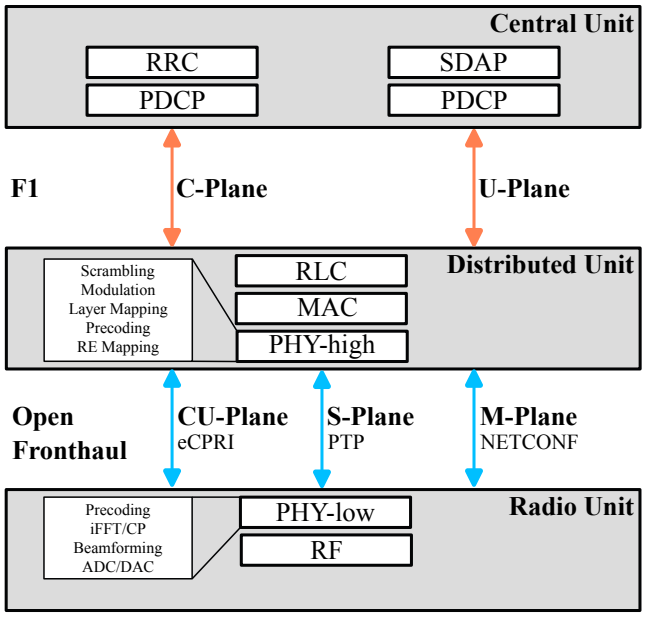


Figure 2: Open Radio Access Network (O-RAN) disaggregation of a Next Generation NodeB (gNB) into a Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU). This figure depicts O-RAN Split 7.2x, where the control and user plane (CU-Plane) of the Open Fronthaul is split within the physical layer (PHY). The F1 interface connects DU and CU. This figure is adapted in parts from [52].

2.4 Related Work

We summarize general O-RAN-security-related publications (Section 2.4.1), address existing work related to the Open Fronthaul and RU security (Section 2.4.2), and distinguish our work from the aforementioned publications (Section 2.4.3).

2.4.1 O-RAN Security

Liyanage et al. [36] analyze security risks and challenges within the O-RAN ecosystem by classifying security-related risks. They offer a detailed overview of various threat categories, including descriptions and evaluations of their applicability to the O-RAN ecosystem. They discuss potential security solutions derived from Cloud Radio Access Network (C-RAN) and delve into design errors while exploring their consequences and available mitigation options for O-RAN. Klement et al. [33] investigate the O-RAN environment, evaluating the security status of its deployed components and proposing measures to ensure their secure operation. They identify critical stakeholders in the O-RAN context and list best practices to enhance O-RAN security. Groen et al. [30] investigate the security aspects of O-RAN systems, adopting a holistic approach, including the O-RAN interfaces and the overall platform. They identify potential threats and offer solutions to address security issues in these areas.

Without a specific focus on the O-RAN architecture, Farooqui et al. [23] present a threat model for 5G-based systems, defining a layered architecture and mapping threats to the respective applicable layers. Sattar et al. [54] model the threats arising from small cells in Long-Term Evolution (LTE) networks. They define trust boundaries including physical security as one aspect.

2.4.2 Open Fronthaul and Radio Unit Security

Abdalla et al. [6] delve into the standardization efforts of the O-RAN Alliance, focusing on network threats with a specific emphasis on the Open Fronthaul. They identify end-to-end security threats affecting the interface and recommend countermeasures and best practices against the identified threats. They detail an attack scenario involving unauthorized access to the physical layer of the Open Fronthaul by compromising the physical connection between the DU and the RU. Liao et al. [35] developed a Denial-of-Service (DoS) attack tool for the Open Fronthaul control plane (C-Plane) by generating C-Plane packets that initiate DoS attacks. Dik et al. [16, 17] contribute two consecutive works on the security of the Open Fronthaul. In their first work [16], the researchers examine the transport security of the Open Fronthaul by investigating threats that can impact the interface. They survey the data types transported over the different data planes and derive necessary security measures. In their second work [17], Dik et al. conduct a more in-depth analysis of the transport network security in the Open Fronthaul. They discuss threats and vulnerabilities of the interface and their network impact. They provide a threat protection solution in MACsec as a layer two security mechanism implemented on field-programmable gate arrays (FPGAs) to secure the Open Fronthaul.

2.4.3 Distinction from Related Work

The publications presented in this section are relevant to our work as they introduce overarching challenges, threats, and vulnerabilities associated with O-RAN, showing the larger attack surface of the ecosystem and shedding light on various approaches attackers can take when attacking the O-RAN components and interfaces. The presented papers all take a theoretical approach to analyzing O-RAN security. In contrast, our work focuses on the security of a single O-RAN component, i.e., the RU. **We investigate the AV2700 as an example of a proprietary RU and present real-world vulnerabilities and security issues of the AV2700**.

3 Threat Model

In contrast to wireless access points [37, 59, 65], RUs pose an especially interesting attack surface with their multiple interfaces to other RAN components. Our threat model is consistent with existing publications [6, 23, 31, 36, 42, 45, 54, 55]

and applicable standards [22,43] tailored towards indoor BSs. It aligns with existing threat models of indoor BSs in conventional RANs, including femtocells [28], for all non-O-RAN aspects. This section defines our system model (Section 3.1) and discusses an adversary's motivation (Section 3.2) and their assumed capabilities (Section 3.3).

3.1 System Model

Figure 1 depicts our system model. We assume an O-RAN infrastructure with one or more RUs deployed indoors. The RUs connect to a corresponding DU via Ethernet to handle control, user, synchronization, and management plane (CUSM-Plane) communication. The RU also connects to the Service Management Orchestration Framework (SMO), where one of the RAN Intelligent Controllers (RICs) is deployed [42,43].

In contrast to physically protected outdoor cell towers [14], RUs are installed akin to prevalent enterprise Wi-Fi routers, implying that they are accessible from within the building [15]. We consider an RU affixed to a wall. The RU might be located within or without the reach of an adversary. The RU might be secured with anti-theft protection means, e.g., a Kensington lock. Surveillance measures might be implemented to mitigate undiscovered interactions with the RU. The network infrastructure might be configured so an adversary can achieve Ethernet access from an adjacent Ethernet port connected to the RU. Depending on these deployment options, the adversary can gain different capabilities (Section 3.3). Possible scenarios enabling such access include installations in shared or multi-tenant buildings, e.g., office complexes, shopping centers, or universities.

3.2 Adversary Motivation

The adversary we consider aims to attack the 5G network, using the O-RAN RU for their initial foothold. Note that the cellular network is classified as a critical infrastructure [24] and, hence, is particularly interesting to adversaries. In the context of this work, **the adversary aims to gain complete control of an RU to facilitate further attacks.**

While any attacks beyond controlling the RU are outside this work's scope, the adversary's next steps might include local operation or lateral movement: (1) On the RU, the adversary might manipulate in-transit traffic by recording, manipulating, or redirecting, potentially targeting UEs [12,34,51]. Additionally, the adversary might extract sensitive configuration data. (2) The adversary might prepare attacks for lateral movement in the O-RAN by escalating attacks from the controlled RU to the DU [6] or SMO [58,60].

3.3 Adversary Capabilities

We consider an adversary targeting the RAN by abusing physical access to an indoor RU, directly achieving physical access, or connecting to an adjacent Ethernet port connected to the RU's Open Fronthaul interfaces. In doing so, our assumed adversary achieves a subset of the following four capabilities summarized in Table 1:

Table 1: Summary of the adversary's potential actions achievable with capabilities $\mathbb{C}_1$ - $\mathbb{C}_4$. Checkmarks imply that the adversary is capable of the potential action and crosses imply that the adversary is not.

Potential Action	$\mathbb{C}_1$	$\mathbb{C}_2$	$\mathbb{C}_3$	$\mathbb{C}_4$
Accees to the RU's Ethernet ports	✓	✓	✓	✓
Access to the RU's power socket	✗	✓	✓	✓
Access to the RU's debug ports	✗	✓	✓	✓
Evaluation in own environment	✗	✗	✓	✓
Redeployment of a modified RU	✗	✗	✗	✓

Ethernet Access With access to the RU via Ethernet ($\mathbb{C}_1$), the adversary can communicate with the RU's Open Fronthaul interface (Figure 3). This access enables the adversary to take the logical position of another RAN component, e.g., the DU or SMO, to target exposed services on the RU and any attack surface provided by the Open Fronthaul interface. By exploiting vulnerabilities in this attack surface, they attempt to obtain control over the RU. The adversary can achieve $\mathbb{C}_1$ with access to an adjacent Ethernet port connected to the RU, regardless of surveillance and anti-theft protection means in place.

Full Interface Access Access to all of the RU's interfaces ($\mathbb{C}_2$) grants the adversary all of $\mathbb{C}_1$ and access to the RU's power socket and, potentially, to debug ports. With the power socket, the adversary can shut down and restart the RU, e.g., for trivial DoS attacks and to activate the RU's start-up procedure. Additionally, the adversary can inspect the RU's High-Definition Multimedia Interface (HDMI) debug port, potentially facilitating gaining control of the RU. Capability $\mathbb{C}_2$ requires physical access to the RU, the feasibility of which depends on surveillance and access control means in place.

RU Theft If the adversary can remove the RU ($\mathbb{C}_3$), they can conduct further attacks in a prepared environment. This enables the adversary, on top of $\mathbb{C}_1$ and $\mathbb{C}_2$, to perform more intrusive operations that require disassembly. If no hardware security features exist, they can use this access to extract secrets from the device. In addition, they can inspect and modify the firmware running on the device. After probing the RU, e.g., to extract non-default secrets and potentially tamper with the software and hardware of the device, the adversary can use the findings to attack other RUs. Capability $\mathbb{C}_3$ requires direct physical access to the RU and an unguarded deployment.

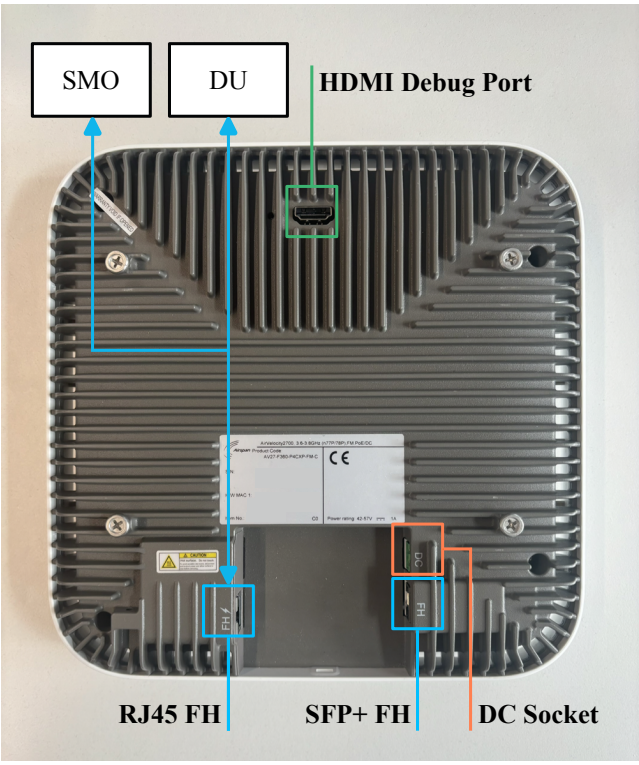


Figure 3: The AV2700's interfaces feature a power socket, an HDMI debug port, and two Open Fronthaul interfaces (Small Form-factor Pluggable+ (SFP+) and RJ45; blue) that connect to the Distributed Unit (DU) and Service Management Orchestration Framework (SMO).

RU Redeployment Redeploying the RU ($\mathbb{C}_4$) grants the adversary all of $\mathbb{C}_1$ - $\mathbb{C}_3$ and the option to set up a manipulated RU into the O-RAN. After probing, modifying, and possibly gaining control over the removed RU, the adversary can redeploy the device to its designated spot. Taking the position of the RU enables interaction with the other components in the RAN and lays the basis for further attacks. Capability $\mathbb{C}_4$ requires physical access to the RU and a deployment environment that is unguarded and unsecured over an extended period.

4 Analysis

The AV2700's hardware and software structure is not publicly disclosed, so we decided to learn as much as possible about its inner workings to understand its attack surface. We connected a computer to the AV2700 (Section 4.1) to understand the network interfaces, remotely connected to the AV2700 to explore the file system, and reverse-engineered firmware parts. We report insights into the AV2700's hardware and software structure (Section 4.2) and its exposed services (Section 4.3).

4.1 Setup

Our hardware setup comprises two components interconnected via an Ethernet cable: A commercial off-the-shelf (COTS) computer and an AV2700. We utilize the computer to communicate with the AV2700, investigate the device, and capture network traffic for analysis. The AV2700 connects to the computer via Ethernet. In this connection, we observed unencrypted traffic between the AV2700 and the PC for different reasons: (1) During start-up, the RU initiates a call-home procedure to the DU, and (2) some services running on the RU's host system are not directly related to O-RAN. In normal operation, RU and DU communicate over an encrypted channel. Apart from the AV2700, our setup solely comprises COTS hardware, highlighting that only minimal resources are necessary to replicate our findings.

4.2 Hardware and Software Structure

The AV2700 hardware (Figure 3) is based on the *Mercury+ XU8* System-on-Chip (SoC) [20] containing a *Xilinx Zynq UltraScale+*, which includes an FPGA [64], an *ARM Cortex A53* [8] and an *ARM Cortex R5F* [9].

We investigated the AV2700 with firmware `19.6.3` of System Release `1.6.37`. The operating system (OS) on the AV2700 is an embedded Linux solution built and deployed using `PetaLinux 2019.1`. PetaLinux is an embedded software development kit (SDK) for Xilinx FPGA-based SoC designs that includes auxiliary functions for building Linux solutions for embedded systems [63]. Further, `BusyBox v.1.29.2`, a Unix software suite for embedded systems and mobile devices [11], provides Unix functionality on the AV2700.

Besides a power socket, the AV2700 has three physical interfaces (Figure 3), two connecting to other RAN components, while we assume the third to be a physical debug interface:

SFP+ port The first physical port is an SFP+ Ethernet port, providing high-speed connectivity, which is ideal for the Open Fronthaul control, user, and synchronization plane (CUS-Plane). It requires an SFP+ module and connector.

RJ45 port The second physical interface is an RJ45 Ethernet port, allowing communication to the AV2700.

Debug port The third physical interface is an HDMI port, which we suspect to be an HDMI-muxed debug port similar to [50] and compatible with HDMI-muxed debug cables [49].

4.3 Services

Figure 4 outlines the service architecture deployed on the AV2700. Seven ports are open, out of which four are unauthenticated. The most notable components are:

Table 2: Summary of our findings. The impact of $\mathbb{F}_1$ - $\mathbb{F}_4$ is a combination of reconfiguration (Reconf.), Denial-of-Service (DoS), and Remote Code Execution (RCE). The CVSS scores refer to the Common Vulnerability Scoring System (CVSS) version 4.

	Finding	Impact	CVSS[1]	Affected Services	Mitigation
$\mathbb{F}_1$	Exposed TCF Agent	RCE	9.3	`tcf-agent`	Remove before deployment
$\mathbb{F}_2$	Missing Access Control	DoS, Reconf.	8.4	`clish_agentd`, `mosquitto`	Implement authentication
$\mathbb{F}_3$	Memory Corruption	DoS/RCE	8.3	All management daemons	Secure coding best practices, bound checking
$\mathbb{F}_4$	Command Injection	RCE	9.3	`itf-mgmt`	Sanitize user input

[1] We self-assigned the CVSS scores to vulnerabilities $\mathbb{F}_1$ - $\mathbb{F}_4$ according to [25].

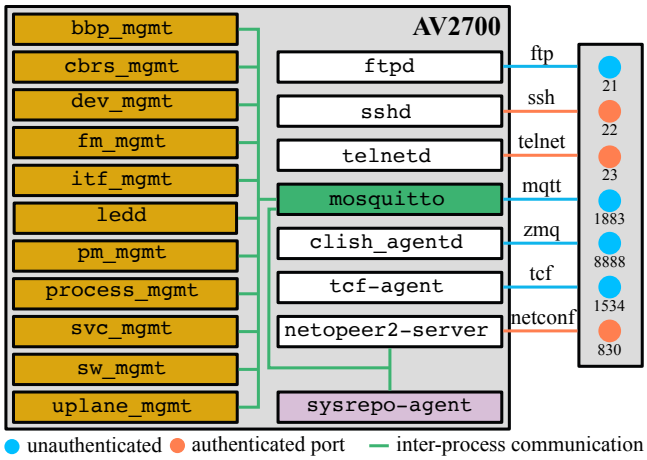


Figure 4: Service architecture of the AV2700. The central point is the `mosquitto` service communicating to the management daemons via inter-process communication (IPC). Three of the ports are authenticated and four unauthenticated.

FTP Server The AV2700 runs the `ftpd` of `BusyBox` started as an `inetd` server. The firmware misconfigures the File Transfer Protocol (FTP) server using an unsupported argument without a directory to serve the supplied files, preventing successful file transfers. Consequently, its purpose is unclear, especially when considering that the O-RAN standard mandates Secure File Transfer Protocol (SFTP) and File Transfer Protocol Explicit-mode Secure (FTPES), which secure FTP with Secure Shell (SSH) and Transmission Control Protocol (TCP), respectively, see Section 5.1 of [47]. Furthermore, per standard, the FTP service is located on the DU, where the RU is supposed to connect for file uploads.

SSH server The `sshd` service is provided by the `OpenSSH 7.8` SSH server. This service is required for secure M-Plane connections; see Section 5.4 of [47]. However, the SSH server with enabled shell access also allows command execution on the RU, which is not a functionality described in the standard.

Telnet The built-in `telnetd` of BusyBox provides Teletype Network (Telnet) remote access. It offers functionality similar to the SSH server without the confidentiality or integrity protection of the transmitted data. Identical to the SSH server, the remote access capabilities offered by Telnet are not mandated by any standard covering the RU, nor are they part of the M-Plane. However, when considering Telnet as a vendor extension of the M-Plane specification, it violates the end-to-end security requirement stated in [47].

Mosquitto MQTT server The `mosquitto` Message Queuing Telemetry Transport (MQTT) server listens on all interfaces and is externally reachable. In extension, the IPC functionality to the manager daemons can also be called from outside, indirectly exposing the manager daemons. The MQTT server appears to be unrelated to any O-RAN standard. We assume it to be a leftover implementation detail of the IPC mechanism that the internal manager daemons (described below) use to communicate.

Clish-agent service The `clish-agentd` implements the functionality of a local `oru-shell` over ZeroMQ [66]. It dispatches commands sent to the shell to the corresponding internal manager daemon by publishing them on the MQTT topic. No O-RAN standard describes the `oru-shell`, but it is directly derived from the NETCONF YANG models. Consequently, supported options overlap with settings configured over NETCONF. The `clish-agentd` violates the mandatory end-to-end security of the M-Plane because it uses an unauthenticated plain-text protocol [47].

TCF debugger The `tcf-agent` exposes debugger functionality for the FPGA. The Target Communication Framework (TCF) is an open-source network protocol to communicate with embedded devices [18]. The `tcf-agent` is unrelated to any O-RAN specification and appears to be a leftover development artifact. We checked recent Board Support Packages (BSPs) for the *Zynq UltraScale+* and found the `tcf-agent` enabled by default. Therefore, its presence might be unintentional and not the result of a deliberate decision during development.

NETCONF The `netopeer2-server` implements the NETCONF protocol over SSH. NETCONF over SSH is required by the M-Plane specification [47]. In addition, the specification also requires NETCONF over Transport Layer Security (TLS), which we found missing (Section 5.5.1).

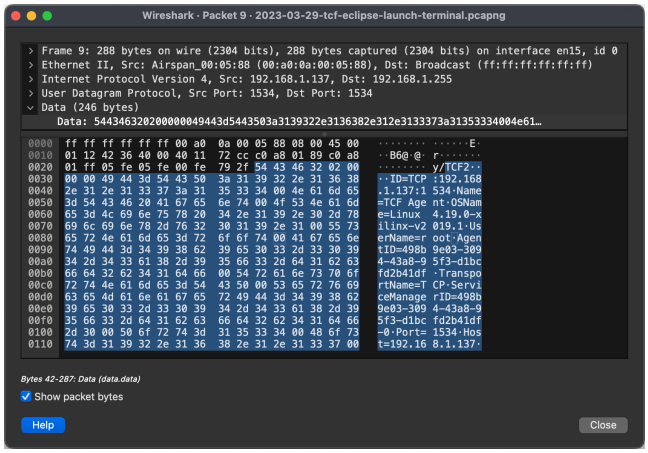


Figure 5: Target Communication Framework (TCF) packet broadcasted periodically by the TCF agent on port 1534, disclosing the TCF agent with ID, port, and version, and the operating system `Linux 4.19.0-xilinx-v2019.1` with user `root`, leading to $\mathbb{F}_1$.

Sysrepo agent The `sysrepo-agent` is another central component that implements the NETCONF protocol's internal YANG datastore as part of the O-RAN M-Plane [47]. Furthermore, it connects to the MQTT server to trigger manager functions as a reaction to NETCONF remote procedure calls (RPCs). This component directly results from the M-Plane specification, which requires NETCONF support for device management.

Manager daemons The AV2700 software is structured into multiple manager daemons (the leftmost box in Figure 4), which are used to configure and monitor device aspects related to the O-RAN: The `sw_mgmtd`, e.g., is used to update, install, and activate firmware archives. These managers use the publish-and-subscribe-based MQTT protocol for IPC to exchange messages encoded with JavaScript Object Notation (JSON) via the shared `mosquitto` [26] server.

5 Findings

This section presents four novel vulnerabilities we discovered on the AV2700, which are summarized in Table 2: An exposed TCF agent (Section 5.1), missing access control (Section 5.2), multiple memory corruption vulnerabilities (Section 5.3), and an OS command injection vulnerability (Section 5.4). We also discuss deviations from the O-RAN standards identified on the AV2700 (Section 5.5). **All vulnerabilities $\mathbb{F}_1$ - $\mathbb{F}_4$ are exploitable for adjacent adversaries with capability $\mathbb{C}_1$ using low-complexity attacks without user interaction, special privileges, or additional attack requirements** (Figure 6).

```python
import zmq

context = zmq.Context()

#  Socket to talk to server
print("Connecting to remote server...")
con = context.socket(zmq.DEALER)
con.connect("tcp://o-ran-ru-ip:8888")

con.send(b"view=system-view subview=all reboot\n")

while 1:
    message = con.recv()
    print(message.decode(), end="")
```

Listing 1: Python code to interact with ORU shell views. In this example, the `reboot` command in the `system` view is invoked to disrupt the device ($\mathbb{F}_2$).

5.1 Exposed TCF Agent

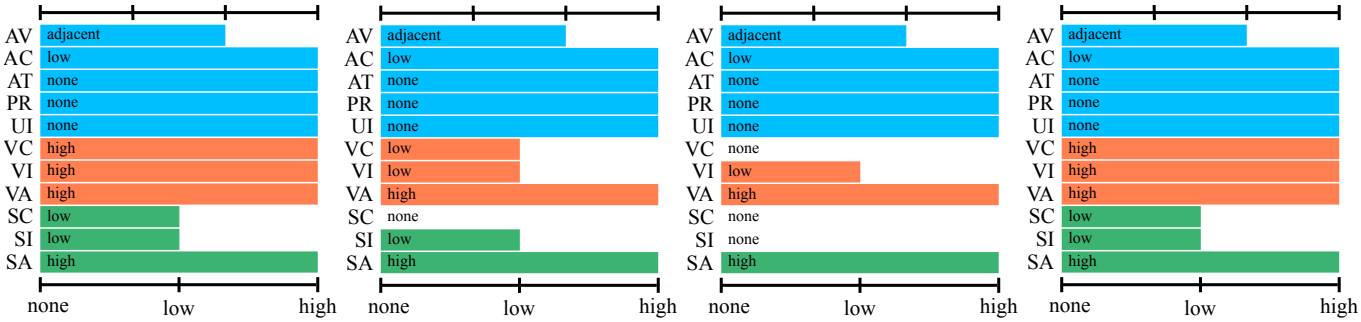
In the context of the AV2700, TCF enables developers to communicate with the built-in FPGAs [19], e.g., from within Eclipse with the TCF debugger add-on that opens a terminal for debugging. In the background, TCF opens a terminal on the device and runs commands as `root`.

The AV2700's TCF agent periodically sends out User Datagram Protocol (UDP) packets on port `1534` (Figure 5) while it waits for connections. The O-RAN standards do not describe the use of TCF to communicate to the RU, so we assume it to be a leftover service that the vendor unintentionally enabled during development. As the TCF agent was not removed before deployment, adversaries can abuse its functionality with at least capability $\mathbb{C}_1$, yielding $\mathbb{F}_1$. The TCF packets disclose detailed information about the device, including the host and port. After recording the handshake between Eclipse and the AV2700, we reconstructed the TCF messages required to execute arbitrary shell commands on the embedded device.

Finding $\mathbb{F}_1$ gives an adversary full control over the RU, which we further discuss in Section 6.1. While mitigation is straightforward, i.e., removing the TCF agent before deployment, we assign $\mathbb{F}_1$ a *critical* CVSS score of 9.3 (Figure 6a) with a high impact regarding all security goals with low subsequent impact on confidentiality and integrity and a high subsequent impact on availability.

5.2 Missing Access Control

While the NETCONF, Telnet, and SSH interfaces require authentication, the `mosquitto` MQTT and `clish-agentd` services can be accessed unauthenticated, yielding $\mathbb{F}_2$. In contrast to NETCONF, these services are not required by the applicable O-RAN standards. The `clish-agentd` is a more user-friendly way to access the NETCONF settings and can, therefore, be considered a vendor-specific O-RAN extension. The MQTT server is an exposed implementation detail with-



(a) Exposed TCF Agent ($\mathbb{F}_1$) (b) Missing Access Control ($\mathbb{F}_2$) (c) Memory Corruption ($\mathbb{F}_3$) (d) Command Injection ($\mathbb{F}_4$)

Figure 6: The Common Vulnerability Scoring System (CVSS) scores of $\mathbb{F}_1$ - $\mathbb{F}_4$. The colored sectors depict the three CVSS metrics: Exploitability, vulnerable system impact, and subsequent impact metrics. Each figure shows, starting from the top, the following items: attack vector (AV), attack complexity (AC), attack requirements (AT), privileges required (PR), user interaction (UI), confidentiality (VC), integrity (VI), availability (VA), confidentiality (SC), integrity (SI), availability (SA). The individual scores of the exploitability metric are as follows: None, low, high for PR; none, passive, active for UI; low, high for AC; none, present for AT; and network, adjacent, local, physical for AV.

out significant user benefits. As a result of the missing authentication, the entire interface of the `clish-agentd` is available to a remote adversary with at least $\mathbb{C}_1$. It can be used to set relevant configuration options of the AV2700 (Listing 1). While there is no built-in option to execute arbitrary commands, the shell can be abused to misconfigure the device. Since the available configuration options include vital system parameters such as the sending power, which affects the transmission of user data, this attack vector endangers the system's availability. Communication with the `mosquitto` server grants comparable capabilities to accessing the `clish-agentd` since the majority of commands implemented by this agent are also dispatched via MQTT. However, it poses a higher risk because direct access to the underlying MQTT broker allows the adversary to control message contents fully. A remote adversary possessing at least $\mathbb{C}_1$ can exploit vulnerabilities in the supposedly internal management daemons by carefully crafting messages, which we discuss further in Sections 5.3 and 5.4. We assign $\mathbb{F}_2$ a *high* CVSS score of 8.4 (Figure 6b) with a high impact on availability, low impact on confidentiality and integrity, high subsequent impact on availability, and low subsequent impact on integrity.

5.3 Memory Corruption Vulnerabilities

The custom-written management daemons of the AV2700 (the leftmost box in Figure 4) are most likely written in C, judging by the libraries used, some strings referring to filenames with a `.c` extension, and the overall observable programming paradigms in place. Consequently, these components suffer from a lack of language-based memory safety, leading to $\mathbb{F}_3$. Examples of such issues include multiple null pointer de-references crashing the affected components.

In the custom components, bounds checking is done implicitly using the fortified versions of security-relevant functions such as `__strcpy_chk` instead of `strcpy()` [27]. These functions perform checks to ensure sufficient buffer sizes and prevent the exploitation of buffer overflows. However, this means that all services using such functionality immediately crash when encountering an out-of-bounds error, resulting in straightforward attacks on availability. We found this a problem in almost every case where input is copied from an MQTT message containing a user-supplied JSON payload. While one can argue that in our threat model, the adversary can always attack availability by pulling the plug that supplies the unit with energy, this attack vector allows for the disruption of specific sub-services in a stealthier manner.

```
1  void* buffer = calloc(1, 0x102c);
2  void* build_id = cJSON_GetObjectItem(json_obj,
3                                       "build_id");
4  if (build_id != 0) {
5     // Fortified version of strcpy (safe)
6     __strcpy_chk(buffer, *(build_id + 32), 64);
7     [...]
8     void* buffer_ptr = buffer + 0x188;
9     void* filename_field = cJSON_GetObjectItem(
       json_obj, "file-name");
10    if (filename != 0) {
11       // strcpy call with indirect pointer,
12       // not fortified (unsafe!)
13       strcpy(buffer_ptr - 0x84,
14              *(file_name_field + 32));
15    }
16 }
```

Listing 2: Reverse-engineered code surrounding the heap buffer overflow in Line 13 due to unfortified functions ($\mathbb{F}_3$).

Fortified functions can only be used if the length of the target buffer is known during compile time. This requirement

```
1  /* Function is called with user supplied input
2     extracted from the JSON payload in an MQTT
3     message. */
4  void create_interfaces(char *inf, int vlan_id) {
5     char if_name[10];
6     char cmd_buff[100];
7
8     /* This formatting call limits the size of
9        `inf` to 7. `if_name` is not used for the
10       system() call, but the program crashes
11       if the size is exceeded. */
12    __sprintf_chk(if_name, 1, 10,
13                  "%s.%d", inf, vlan_id);
14    if(!check_if_inf_exists(if_name)) {
15        __sprintf_chk(cmd_buff, 1, 100,
16                      "vconfig add %s %d",
17                      inf, vlan_id);
18        system(cmd_buff);
19    }
20 }
```

Listing 3: Reverse-engineered code excerpt of the operating system (OS) command injection vulnerability in Line 18 ($\mathbb{F}_4$).

is violated when accessing dynamically allocated memory areas such as heap buffers with pointers, and their fortified counterparts cannot replace unsafe functions. We attribute the existence of $\mathbb{F}_3$, a heap buffer overflow in the `sw_mgmtd` daemon, to this fact (Listing 2): Surrounding code relies on `__strcpy_chk()` in combination with stack-located buffers, while the problematic code copies data from an MQTT message into a heap buffer (Line 13). Here, the unfortified `strcpy()` function is used, thus producing an overflow bug when copying from untrusted input. The impact of such issues goes beyond simple DoS attacks against services on the RU and can lead to full RCE [32] with severe consequences for the whole O-RAN (Section 6.1). At least $\mathbb{C}_1$ is required to exploit this finding since the service is exposed on the Open Fronthaul interface. We assign $\mathbb{F}_3$ a *high* CVSS score of 8.3 (Figure 6c) with a high impact on availability, a low impact on integrity, and a high subsequent impact on availability. We base this score on the conservative assumption that exploitation for full RCE might be infeasible due to insufficient primitives.

5.4 Command Injection Vulnerabilities

Memory-related issues are not the only area where user input sanitization is lacking. Generally, we noticed that commands executed by the `system` function were built using string formatting techniques. A review of associated input parameters uncovered an exploitable command injection vulnerability in one of the management daemons (Listing 3), resulting from the passing of untrusted user input to `system` ($\mathbb{F}_4$). Since the management daemons run as `root`, this enables the execution of arbitrary commands in the context of the super-user. Similar to memory corruption issues, this vulnerability is also externally exploitable with at least $\mathbb{C}_1$ and no authentication due to the missing access control on the MQTT server. As described in Section 4.3, no O-RAN standard mandates the MQTT server. Instead, it is an implementation detail of the RU vendor. The exposure of internal services that implement O-RAN-specific functionality leads to additional attack surfaces that could have been avoided. The specific command injection vulnerability we found gives an adversary-controlled buffer of seven bytes (Line 12). Only five usable bytes remain after accounting for two bytes to terminate the previous command and cut off trailing characters. Although this length restriction prevents straightforward execution of arbitrary code, known techniques exist to exploit exactly such scenarios to gain full RCE [61], with effects on the whole O-RAN, which we describe in Section 6.1.

We adapt this idea to create empty files with controlled filenames by using the shell's output redirection operator (`>`). After creating the necessary files, we use `ls *> 0` to create a file containing the chosen payload. Note that we get the trailing zero for free due to the virtual local area network (VLAN) ID appended to the injectable interface name (Line 15), allowing us to stay within the payload length constraints. We force `ls` to list the files in the order of most recent creation by creating a file called `-tx` beforehand, which is parsed as an argument to `ls`, controlling the sorting precedence of the output. This order allows our files to appear first in the directory listing, enabling us to ignore trailing characters. As `ls` adds whitespace between filenames, we facilitate a combination of `tr` and `sed` invocations to remove whitespaces and construct arbitrary payloads, inserting `${IFS}` whenever we require spaces in our payload. We assign $\mathbb{F}_4$ a *critical* CVSS score of 9.3 (Figure 6d) with a high impact on confidentiality, integrity, and availability, a low subsequent impact on confidentiality and integrity, and a high subsequent impact on availability.

5.5 Open Fronthaul Standard Deviations

The investigation of the AV2700 RU revealed several deviations from the concepts and functionalities introduced in the O-RAN M-Plane specification for the Open Fronthaul. Specifically, various features specified in the standard for the RU startup procedure were absent in the AV2700. This section addresses the missing TLS option (Section 5.5.1) and the persistent creation of users (Section 5.5.2) before discussing the use of default credentials (Section 5.5.3). **While none of these deviations are exploitable, they can facilitate follow-up attacks.**

5.5.1 Missing NETCONF via TLS Option

The RU performs a call-home procedure during start-up, which leads to the DU establishing a NETCONF connection to the RU. The M-Plane specification mandates TLS encryption as an alternative to SSH for establishing the NETCONF

connection [47]. However, we discovered that the NETCONF via TLS option is missing in the AV2700. As a result, only NETCONF via SSH is available during the initiation of the call-home procedure. To use this deviation in combination with a flaw affecting the SSH implementation, an adversary will need $\mathbb{C}_2$ to restart the RU and trigger the start-up procedure. $\mathbb{F}_1$ - $\mathbb{F}_4$ would still be exploitable when using TLS encryption.

5.5.2 Persistent Creation of Users

The second discrepancy occurs when creating a new user account with super-user privileges on the AV2700. The M-Plane specification states that upon creating a new user account and assigning it super-user privileges, the default `root` account on the device should be deactivated, and the active NETCONF connection should be disconnected [47]. However, we found that after creating a new user account and assigning super-user privileges on the AV2700, the device neither disconnects the active NETCONF connection with the default account nor deactivates the default `root` account. This behavior deviates from the specification and can facilitate follow-up attacks, e.g., for adversaries that manage to create a user via the debug port ($\mathbb{C}_2$).

5.5.3 Default Credentials

The O-RAN Alliance has identified the use of default credentials on RUs as a main security issue [44]. Considering the prevalence of default password lists [38] and the associated risks in network equipment [13], the failure to deactivate the default account poses severe security risks. Adversaries can gain access to deployed AV2700s by brute-forcing devices with default password lists as long as the default super-user remains active, which can be attacked by adjacent adversaries with access to a connected Ethernet port ($\mathbb{C}_1$).

6 Discussion

This section discusses the requirements of our findings and their impact on the operation of the O-RAN (Section 6.1). We outline mitigation means for $\mathbb{F}_1$ - $\mathbb{F}_4$ (Section 6.2). We discuss the security implications of our findings considering technological trends related to indoor BS (Section 6.3), 5G and beyond (Section 6.4), and the O-RAN ecosystem (Section 6.5). Finally, we address limitations of our work, future work (Section 6.6), and the responsible disclosure process (Section 6.7).

6.1 Impact on the Cellular Network

In Section 3, we defined the goal of our presumed adversary as full control of an RU running in an O-RAN. This section summarizes the exploitation requirements of findings $\mathbb{F}_1$ - $\mathbb{F}_4$ (Section 6.1.1) and their impact (Section 6.1.2) on the cellular network. Figure 7 depicts which capabilities are required to exploit $\mathbb{F}_1$ - $\mathbb{F}_4$ and what level of control they enable. Finally, we point out follow-up attacks (Section 6.1.3).

6.1.1 Requirements

Findings $\mathbb{F}_1$ - $\mathbb{F}_4$ are all exploitable via the RU's Open Fronthaul interface. Thus, adversaries with access to an adjacent Ethernet port connected to the RU can exploit them ($\mathbb{C}_1$). The adversary is not required to have specific knowledge of any credentials.

6.1.2 Impact

In the following, we discuss what adversaries can achieve with $\mathbb{F}_1$ - $\mathbb{F}_4$ and how close that brings them to fully controlling an RU running in O-RAN.

Reconfiguration With $\mathbb{C}_1$, adversaries can reconfigure the running RU with $\mathbb{F}_2$. Note how $\mathbb{F}_1$ and $\mathbb{F}_4$ also enable reconfiguration of the RU.

Denial of Service A DoS attack on the RU leads to users losing access to the cellular network. An adversary that is limited to $\mathbb{C}_1$ can exploit $\mathbb{F}_2$ to achieve DoS by reconfiguration of settings in the `oru-shell`: (1) data transmission can be interrupted by modifying RF parameters, (2) access to the RU can be hindered by configuring a VLAN tag unknown to the network operator, or (3) the device can be rebooted repeatedly with the reboot option to disrupt availability. Note how, with $\mathbb{C}_2$, $\mathbb{C}_3$, or $\mathbb{C}_4$, adversaries can trivially achieve DoS by repeatedly restarting or shutting down the RU.

Full Access Findings $\mathbb{F}_1$ and $\mathbb{F}_4$ both grant the ability to execute arbitrary code on the RU. Notably, both findings only require $\mathbb{C}_1$ and allow RCE in the security context of the `root` user. Thus, **$\mathbb{F}_1$ and $\mathbb{F}_4$ give the adversary full control of the RU.** Assuming it is feasible to gain RCE with $\mathbb{F}_3$, that finding also gives the adversary full control of the RU.

6.1.3 Follow-Up Attacks

Figure 1 depicts to which O-RAN components the RU is connected. With full control over an RU, there are three potential follow-up goals: (1) targeting users via their UEs, (2) attacking the O-RAN DU on the Open Fronthaul CUSM-Plane, or (3) attacking the O-RAN SMO on the Open Fronthaul M-Plane. Adversaries can target users by injecting downlink traffic to attack UEs. While no known attacks targeting users from an O-RAN RU exist, similar attacks exist for LTE [12, 34, 51]. Lateral movement in the O-RAN is possible towards the DU and the SMO. Adversaries can conduct the Open Fronthaul C-Plane DoS attack against the DU described

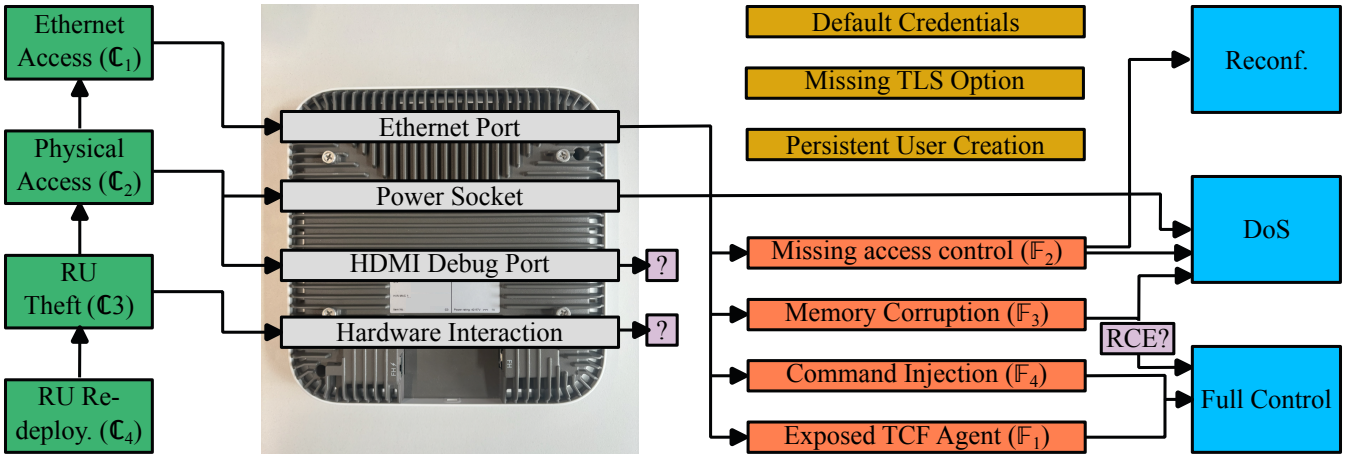


Figure 7: The Requirements and impact of our findings $\mathbb{F}_1$ - $\mathbb{F}_4$. From left to right, the adversary's capabilities $\mathbb{C}_1$ - $\mathbb{C}_4$ determine the level of access to the Radio Unit (RU). Our findings enable the adversary to achieve attacker goals, i.e., reconfiguration, Denial-of-Service (DoS), or full control of the RU. While not required for the shown attacks, the Open Fronthaul standard deviations facilitate further attacks. We also highlight angles for future work.

by Liao et al. [35]. They can also attempt to get in control of a DU [6]. Adversaries can attack the SMO on the Open Fronthaul M-Plane [58, 60].

6.2 Mitigating the Discovered Vulnerabilities

Mitigating $\mathbb{F}_1$ is straightforward by removing the exposed TCF agent before deployment. To mitigate $\mathbb{F}_2$, we recommend limiting internal services to local addresses to avoid exposing them to external threats. Regarding $\mathbb{F}_3$, we recommend performing explicit bound checking on all untrusted user input and considering switching away from the programming language C [48]. Vulnerability $\mathbb{F}_4$ is addressable by sanitizing user input before passing it to functions that evaluate commands, such as the `system()` function. Limiting the internal services to local addresses, as suggested for $\mathbb{F}_2$, also restricts the exploitability of this issue but still enables a low-privileged user to escalate their privileges.

Generally, we recommend applying a reasonable threat model (Section 3) during the software design phase to limit the external attack surface from an architectural point of view. Furthermore, we identified several deviations from the O-RAN and Open Fronthaul specifications (Section 5.5). Coherency to these standards, especially regarding security, ensures the implementation of the best practices, thus mitigating vulnerabilities in general.

6.3 Indoor Base Stations

The high number of security-related issues emphasizes the need for an updated threat model for indoor BSs. This shift voids the assumption that only trusted entities can directly communicate with the RU. Combined with a system architecture that exposes many services without authentication, as described in Section 4.3, the AV2700 presents a vast attack surface. Large parts of the AV2700's internal code are probably written in C, requiring high security awareness and rigorous security testing [53]. Without such precautions, memory safety bugs that lead to vulnerabilities are very likely.

As we show in Section 5, the security weaknesses affecting the AV2700 spread beyond memory corruption issues, including missing access control for dangerous services and an OS command injection. These problems can also occur in software written in memory-safe languages. Therefore, it is necessary to follow security best practices and apply a proper threat model during the development phase, reflecting the reality that adversaries might have physical access to the RU when deployed as an indoor BS in a public space.

6.4 Technologies of 5G and Beyond

The intended application areas of 5G, namely enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and Massive Machine Type Communications (mMTC), incur high requirements on the RAN. 5G facilitates novel technology, such as FPGAs for mmWave beamforming to fulfill these requirements. However, using novel technology in gNBs and O-RAN RUs introduces new challenges for RAN vendors and mobile network operators (MNOs), e.g., more complex hardware in indoor BSs (Section 4.2) and other RAN components.

While $\mathbb{F}_1$ is not a vulnerability within the cellular network itself, it is exploitable by an adjacent adversary to gain access to the AV2700's host system, from where escalation to the AV2700 is trivial with `root` privileges. The exposed TCF agent vulnerability (Section 5.1) is a direct cause of developers not removing the TCF agent from the AV2700 before

deployment. As FPGAs are included in O-RAN RUs to fulfill the requirements of 5G in the application areas, $\mathbb{F}_1$ is a consequence of the novel technologies of 5G and beyond. Additionally, with the TCF debugger enabled by default for the widespread *Zynq UltraScale+*, $\mathbb{F}_1$ can likely be reproduced on other RUs.

6.5 Complexity of the Open RAN Ecosystem

The O-RAN ecosystem is complex with its new open interfaces and introduced features. The different components are highly interconnected through the various interfaces, and research identified that adversaries can use the interconnectivity to their advantage to escalate attacks [36, 39]. As detailed in [6], establishing control of an RU provides an adversary with the means to escalate attacks upwards, penetrating the O-RAN through the DU and beyond, consequently impacting the entire O-RAN ecosystem, including the CU, the SMO, and the RICs. The security implications of this intrusion into the O-RAN are critical as adversaries might access user- and other sensitive data. They might manipulate the O-RAN to transmit malicious packets and data to users, potentially affecting user devices. Additionally, an adversary might bring down the entire O-RAN with a DoS attack, leading to a large-scale outage in 5G, classified as a critical infrastructure.

6.6 Limitations and Future Work

We did not fully evaluate the RU's HDMI debug port. An adversary with access to all interfaces ($\mathbb{C}_2$) might use the RU's debug port to perform a DoS attack or prepare follow-up attacks that lead to RCE, e.g., creating a new super user. An adversary capable of removing the RU ($\mathbb{C}_3$) can perform more intrusive operations to achieve full control of the RU, e.g., firmware modifications or hardware fault injection. However, if these attacks lead to RCE in the security context of `root`, the adversary still needs to redeploy the RU into the running O-RAN to achieve their goal, requiring $\mathbb{C}_4$.

We analyzed the AV2700 as an example of a proprietary indoor O-RAN RU. We focused on the capabilities of an adversary abusing physical access to an indoor RU, potentially stealing, modifying, and redeploying the RU. As we did not analyze an RU in a live O-RAN, future work might provide valuable insights into how much CU-Plane traffic an adversary with full control of the RU can access.

While we aimed to highlight general issues with indoor O-RAN RUs, our evaluation considered only one product, the AV2700. Future work might reproduce our findings on other indoor RUs and assess to which extent our findings are generalizable.

6.7 Responsible Disclosure

We privately reported $\mathbb{F}_1$ to Airspan on April 19, 2023. After waiting for an acknowledgment or response, we sent a follow-up email on February 13, 2024, with a revised deadline of April 13, 2024, marking 360 days from the initial reporting. On February 14, 2024, an Airspan executive responded to our email, who acknowledged dismissing our initial email as a phishing attempt. We were assured that the responsible team at Airspan had been informed about our report and that they would contact us regarding the vulnerability and next steps. On February 21, 2024, we privately reported $\mathbb{F}_2$ - $\mathbb{F}_4$ to Airspan. We set a deadline for May 21, 2024, marking 90 days from the day of reporting, which complies with recommended industry practice [29]. To the best of our knowledge, Airspan is now working on patches for $\mathbb{F}_1$ - $\mathbb{F}_4$.

7 Conclusions

With this paper, we contribute to the RAN security of 5G and beyond, especially regarding the deployment of indoor BSs. We introduce a threat model for indoor BSs, considering they are more easily accessible than outdoor BSs. Our security analysis of the Airspan AirVelocity 2700 (AV2700) results in multiple deviations from the O-RAN and Open Fronthaul standards. We find four vulnerabilities on the AV2700 that we, due to the lack of official scores, self-assign high or critical CVSS scores ($\mathbb{F}_1$ - $\mathbb{F}_4$) and recommend mitigation means for all of them. Our findings show that vulnerabilities in the host system of a state-of-the-art indoor BS are exploitable to Remote Code Executions (RCEs), which facilitate follow-up attacks on the RAN. This highlights the importance of securing not only the RAN-related implementations of a RAN component but also the underlying host.

Acknowledgments

We thank our shepherd and the anonymous reviewers for their helpful suggestions. This work had been co-funded by the Federal Ministry of Education and Research of Germany in the project Open6GHub (grant number: 16KIS014) and the German Research Foundation (DFG) in the project CRUST (grant number: 503199853).

References

[1] 3rd Generation Partnership Project (3GPP). Study on CU-DU lower layer split for NR, Technical Report (TR) 38.816, version 15.0.0. Technical report, December 2017.

[2] 3rd Generation Partnership Project (3GPP). Study on New Radio Access Technology: Radio Access Archi-

tecture and Interfaces, Technical Report (TR) 38.801, version 14.0.0. Technical report, March 2017.

[3] 3rd Generation Partnership Project (3GPP). NR; NR and NG-RAN Overall description; Stage-2, Technical Specification (TS) 38.300, version 18.0.0. Technical specification, December 2023.

[4] 3rd Generation Partnership Project (3GPP). NR; Physical Layer; General Description, Technical Specification (TS) 38.201, version 18.0.0. Technical specification, September 2023.

[5] 3rd Generation Partnership Project (3GPP). System architecture for the 5G System (5GS), Technical Specification (TS) 23.501, version 18.4.0. Technical specification, December 2023.

[6] Aly Sabri Abdalla and Vuk Marojevic. End-to-End O-RAN Security Architecture, Threat Surface, Coverage, and the Case of the Open Fronthaul, 2023.

[7] Airspan Networks Inc. 5G Products. https://airspan.com/5g-products/, 2024. Accessed: 2024-03-10.

[8] ARM Ltd. Cortex-a53 – the most widely used low-power processor. https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a53, 2024. Accessed: 2024-03-10.

[9] Arm Ltd. Cortex-r5 – seamless, real-time embedded processors. https://www.arm.com/products/silicon-ip-cpu/cortex-r/cortex-r5, 2024. Accessed: 2024-03-10.

[10] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead. *Computer Networks*, 182:107516, December 2020.

[11] BusyBox Developers. BusyBox. https://busybox.net/. Accessed: 2024-03-04.

[12] Merlin Chlosta, David Rupprecht, Christina Pöpper, and Thorsten Holz. 5g suci-catchers: Still catching them all? In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 359–364, 2021.

[13] Min-kyu Choi, Rosslin John Robles, Chang-hwa Hong, and Tai-hoon Kim. Wireless Network Security: Vulnerabilities, Threats and Countermeasures. *International Journal of Multimedia and Ubiquitous Engineering*, 3(3):77–86, 2008.

[14] Jeffrey Cichonski, Joshua M Franklin, and Michael Bartock. Guide to LTE Security.

[15] Charles Clancy and Scott G. Kelly. Control And Provisioning of Wireless Access Points (CAPWAP) Threat Analysis for IEEE 802.11 Deployments. Informational RFC 5418, Internet Engineering Task Force.

[16] Daniel Dik and Michael Stübert Berger. Transport Security Considerations for the Open-RAN Fronthaul. In *2021 IEEE 4th 5G World Forum (5GWF)*, pages 253–258, 2021.

[17] Daniel Dik and Michael Stübert Berger. Open-RAN Fronthaul Transport Security Architecture and Implementation. *IEEE Access*, 11:46185–46203, 2023.

[18] Eclipse Foundation. Target Communication Framework (TCF). https://wiki.eclipse.org/TCF. Accessed: 2024-02-15.

[19] Eclipse Foundation. TCF/RISC-V: Connect the TCF Debugger. https://wiki.eclipse.org/TCF/RISC-V#Connect_the_TCF_Debugger, 2020. Accessed: 2024-02-15.

[20] Enclustra FPGA Solutions. Mercury+ XU8 | Xilinx Zynq UltraScale+ MPSoC Module. https://www.enclustra.com/en/products/system-on-chip-modules/mercury-xu8/, 2024. Accessed: 2024-03-10.

[21] Rob Enns, Martin Björklund, Andy Bierman, and Jürgen Schönwälder. Network Configuration Protocol (NETCONF). RFC 6241, June 2011.

[22] European Telecommunications Standards Institute (ETSI). 5G; Security architecture and procedures for 5G System (3GPP TS 33.501 version 17.12.0 Release 17). Technical specification, January 2024.

[23] Muhammad Najmul Islam Farooqui, Junaid Arshad, and Muhammad Mubashir Khan. A Layered Approach to Threat Modeling for 5G-Based Systems. 11(12):1819.

[24] Federal Office for Information Security (BSI). Open-RAN Risk Analysis. Technical report, Federal Office for Information Security, Germany, 2 2022. Accessed: 2024-03-10.

[25] Forum of Incident Response and Security Teams (FIRST). Common Vulnerability Scoring System Version 4.0 Calculator. https://www.first.org/cvss/calculator/4.0, 2024. Accessed: 2024-03-10.

[26] Eclipse Foundation. Eclipse Mosquitto. https://mosquitto.org/.

[27] Free Software Foundation, Inc. Source Fortification (The GNU C Library). https://www.gnu.org/software/libc/manual/html_node/Source-Fortification.html.

[28] Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. Weaponizing femtocells: The effect of rogue devices on mobile telecommunications. In *NDSS*, 2012.

[29] Google. About google's app security. https://about.google/appsecurity/, 2024. Accessed: 2024-03-10.

[30] Joshua Groen, Salvatore DOro, Utku Demir, Leonardo Bonati, Michele Polese, Tommaso Melodia, and Kaushik Chowdhury. Implementing and Evaluating Security in O-RAN: Interfaces, Intelligence, and Platforms, 2023.

[31] Sebastian Haas, Mattis Hasler, Friedrich Pauls, Stefan Köpsell, Nils Asmussen, Michael Roitzsch, and Gerhard Fettweis. Trustworthy computing for o-ran: Security in a latency-sensitive environment. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 826–831. IEEE, 2022.

[32] Sean Heelan, Tom Melham, and Daniel Kroening. Automatic Heap Layout Manipulation for Exploitation. pages 763–779.

[33] Felix Klement, Stefan Katzenbeisser, Vincent Ulitzsch, Juliane Krämer, Slawomir Stanczak, Zoran Utkovski, Igor Bjelakovic, and Gerhard Wunder. Open or not open: Are conventional radio access networks more secure and trustworthy than Open-RAN?, 2022.

[34] Gyuhong Lee, Jihoon Lee, Jinsung Lee, Youngbin Im, Max Hollingsworth, Eric Wustrow, Dirk Grunwald, and Sangtae Ha. This is your president speaking: Spoofing alerts in 4g lte networks. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 404–416, 2019.

[35] Shu-Hua Liao, Chih-Wei Lin, Fransiscus Asisi Bimo, and Ray-Guang Cheng. Development of C-Plane DoS Attacker for O-RAN FHI. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, page 850–852. Association for Computing Machinery, 2022.

[36] Madhusanka Liyanage, An Braeken, Shahriar Shahabuddin, and Pasika Ranaweera. Open RAN security: Challenges and opportunities. *Journal of Network and Computer Applications*, 214:103621, 2023.

[37] Ahmed Redha Mahlous. Threat model and risk management for a smart home iot system. *Informatica*, 47(1), 2023.

[38] Daniel Miessler, Jason Haddix, and g0tmi1k. SecLists. https://github.com/danielmiessler/SecLists, 2024. Accessed: 2024-03-07.

[39] Dudu Mimran, Ron Bitton, Yehonatan Kfir, Eitan Klevansky, Oleg Brodt, Heiko Lehmann, Yuval Elovici, and Asaf Shabtai. Evaluating the Security of Open Radio Access Networks, 2022.

[40] David Muirhead, Muhammad Ali Imran, and Kamran Arshad. Insights and approaches for low-complexity 5g small-cell base-station design for indoor dense networks. *IEEE access*, 3:1562–1572, 2015.

[41] O-RAN Working Group 1. O-RAN Architecture Description. Technical Specification OAD-R003-v11.00, O-RAN ALLIANCE, 2024. Available online at https://www.o-ran.org/specifications.

[42] O-RAN Working Group 11. O-RAN Security Threat Modeling and Risk Assessment. Technical Report TR.0-R003-v02.00, O-RAN ALLIANCE, 2024.

[43] O-RAN Working Group 11. Security Requirements and Controls Specifications. Technical Report TR.0-R003-v08.00, O-RAN ALLIANCE, 2024.

[44] O-RAN Working Group 11. Study on O-RU Centralized User Management. Technical Report TR.0-R003-v01.00, O-RAN ALLIANCE, 2024.

[45] O-RAN Working Group 11. Study on Security for Shared O-RU. Technical Report TR.0-R003-v04.00, O-RAN ALLIANCE, 2024.

[46] O-RAN Working Group 4. Control, User, and Synchronization Plane Specification. Technical Specification CUS.0-R003-v14.00, O-RAN ALLIANCE, 2023.

[47] O-RAN Working Group 4. Management Plane Specification. Technical Report MP.0-R003-v14.00, O-RAN ALLIANCE, 2024.

[48] Office of the National Cyber Director. Back to the Building Blocks: A Path Toward Secure and Measurable Software. Technical report, White House, Washington, DC, 2 2024. Accessed: 2024-03-10.

[49] Mark Alan Overby. HDMI-muxed debug cable methods and apparatuses, October 7 2014. US Patent 8,856,744.

[50] Mark Alan Overby. HDMI-muxed debug port methods and apparatuses, April 7 2015. US Patent 9,003,369.

[51] CheolJun Park, Sangwook Bae, BeomSeok Oh, Jiho Lee, Eunkyu Lee, Insu Yun, and Yongdae Kim. {DoLTEst}: In-depth downlink negative testing framework for {LTE} devices. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1325–1342, 2022.

[52] Michele Polese, Leonardo Bonati, Salvatore D'oro, Stefano Basagni, and Tommaso Melodia. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys & Tutorials*, 2023.

[53] Alex Rebert and Christoph Kern. Secure by Design: Google's Perspective on Memory Safety. Technical report, March 2024.

[54] Danish Sattar, Ashraf Matrawy, Troy Bryant, and Marc Kneppers. Threat Modeling in LTE Small Cell Networks. In *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pages 1–5. IEEE.

[55] Chih-Ting Shen, Yu-Yi Xiao, Yi-Wei Ma, Jiann-Liang Chen, Cheng-Mou Chiang, Shiang-Jiun Chen, and Yu-Chuan Pan. Security threat analysis and treatment strategy for oran. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 417–422. IEEE, 2022.

[56] Small Cell Forum. 5G nFAPI Specifications. https://scf.io/en/documents/225_5G_nFAPI_specifications.php. Accessed: 2024-03-11.

[57] Small Cell Forum. Small Cell Forum. https://www.smallcellforum.org/. Accessed: 2024-03-11.

[58] Kashyap Thimmaraju, Altaf Shaik, Sunniva Flueck, Christian Werling, and Jean-Pierre Seifert. Security testing the o-ran near-real time ric & a1 interface. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'24)*, 2024.

[59] Aju Mathew Thomas, Gowtham Akshaya Kumaran, R Ramaguru, R Harish, and K Praveen. Evaluation of wireless access point security and best practices for mitigation. In *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, pages 422–427. IEEE, 2021.

[60] Walter Tiberti, Eleonora Di Fina, Andrea Marotta, and Dajana Cassioli. Impact of man-in-the-middle attacks to the o-ran inter-controllers interface. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 367–372. IEEE, 2022.

[61] Orange Tsai. BabyFirst Revenge v2 Challenge Writeup. https://github.com/orangetw/My-CTF-Web-Challenges/blob/325ab4e4b4888a7ca73092b8f9e4af70844a09e9/README.md#babyfirst-revenge-v2. Accessed: 2024-02-21.

[62] Sichao Wen and Yuandan Dong. A Low-Profile Wideband Antenna With Monopolelike Radiation Characteristics for 4G/5G Indoor Micro Base Station Application. *IEEE Antennas and Wireless Propagation Letters*, 19(12):2305–2309, 2020.

[63] Xilinx Inc. *PetaLinux Tools Documentation*. Xilinx Inc., 2019. Accessed: 2024-03-04.

[64] Xilinx, Inc. Zynq ultrascale+ mpsoc. https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html, 2024. Accessed: 2024-03-10.

[65] Bin Yang, Yue Hou, Yefeng Zhang, Shiying Feng, and Yong Zhang. Security architecture of wireless private networks for smart grid. *Electrical Engineering and Computer Science (EECS)*, 2:95–98, 2019.

[66] ZeroMQ authors. ZeroMQ. https://zeromq.org/.