Assessing CVE-2026-43499 on a Galaxy S20 Ultra: Firmware Evidence and Runtime Limits

Muhammad Azhar · MetaCyberGuru LLC
Lab work: 22 November 2026 · Version 1.2 · Independent technical report (not peer reviewed)

MetaCyberGuru Research | Correspondence and corrections

Abstract

A security patch date can guide vulnerability triage, but it does not establish whether a published issue will have the same effect on a particular device. This case study examines CVE-2026-43499, a previously disclosed Linux kernel use-after-free, on one author-owned Samsung Galaxy S20 Ultra running Android 13. The assessment combined a comparison of network observations with the device's listener table, static examination of matching firmware, and one bounded runtime observation from an already-authorized debugging session. The firmware analysis found behavior consistent with the upstream defect and evidence that Samsung's kernel hardening was configured. The runtime test returned a deadlock error and completed without an observed reboot or privilege elevation. The error alone did not establish that the stale pointer was subsequently used, and the experiment did not measure whether a mitigation intervened. The study separates evidence of code presence, runtime behavior, and demonstrated impact. It shows why conflicting measurements and negative results should remain visible in a vulnerability assessment rather than being compressed into a single verdict.

1. Scope and research question

The question was whether an upstream Linux kernel finding could be mapped to a specific Android firmware sample, and what impact the available evidence justified reporting. The work was limited to one US-market Snapdragon Galaxy S20 Ultra on the author's own Wi-Fi, with the device enrolled in an ordinary debugging session. Nearby hosts were incidentally visible during discovery but excluded from testing. This study does not assess or make findings about a vendor service, other devices on the network, or the Galaxy S20 product line as a whole.

The handset reported Android 13, a Linux 4.19.113-based kernel built on 5 March 2025, and an Android security patch level of 1 March 2025. It reported a locked bootloader, green Verified Boot state, and SELinux in enforcing mode. The shell was unprivileged. The reported patch level describes the tested sample; availability of newer firmware was not checked. Android's Verified Boot documentation describes boot-chain integrity, which is distinct from the runtime behavior assessed here.

2. Method and evidence

Network observations

Two ad hoc desktop scans disagreed with each other and with the phone's own socket table. They were not used as the final inventory. The author compared the device's TCP listener and process-ownership data with a more controlled network scan from the same LAN. The comparison asked two separate questions: whether a service listened locally, and whether it was reachable from the test host. A loopback-only listener was not counted as LAN exposure.

Firmware and public reference

Azhar assessed firmware matching the installed build. The lab notes record that a separate analysis session extracted the kernel from the boot image, recovered symbols and compared the firmware with third-party research code. The upstream Linux patch is cited separately to explain the expected behavior. It was not the original comparison source. The kernel version alone was not used to determine whether the sample matched the defect.

Runtime observation

For the runtime check, Azhar reviewed and built an existing test credited to the researcher MobiusM. Azhar ran it once from the already-authorized, unprivileged debugging session and checked handset uptime before and after. The test returned a deadlock error and exited normally. Both uptime readings showed 2 days and 15 hours. No kernel memory was instrumented, and the run did not show a hardening control intervening. This report omits test artifacts, execution details and device identifiers.

---

3. Results

Network reachability was limited and changed during testing

Wireless debugging was reachable from the lab host, but the shell used in the study was obtained through existing authorization, not through an unauthenticated network path. The assessment also observed app listeners, including one that was no longer reachable on a later check. Their protocols were not established, and their presence was not treated as a vulnerability. The legacy ADB listener was not reachable in the test. This was a point-in-time TCP assessment, not a complete inventory of every interface or protocol.

Static evidence matched the reported defect and identified configured hardening

In the recovered firmware, the reviewed lock rollback code cleaned up state for the task that was running, rather than for the task waiting on the lock. That difference matched the defect addressed by the upstream Linux patch. This was a static finding. The lab notes recorded firmware indicators attributed to Real-time Kernel Protection and Kernel Data Protection. Samsung's RKP documentation explains the general protection model. It does not establish which credential data was protected on this handset or whether a control intervened during the run.

The single runtime observation did not demonstrate impact

The runtime test returned a deadlock error, then completed. The phone did not visibly reboot, and no root access was obtained. A deadlock error is evidence about the returned result; it does not prove a stale reference persisted, a privileged write occurred, or that the device was unaffected. None of those possibilities can be decided from this run alone.

---

4. Interpretation and limitations

- **Observed:** a bounded TCP listener inventory, the device's reported boot and patch state, and one runtime result without an observed crash or privilege elevation.
- **Supported by static analysis:** code behavior consistent with the upstream defect and indications of configured kernel hardening in the firmware sample.
- **Not demonstrated:** runtime memory corruption, unauthenticated remote access, a repeatable crash, root access, or a mitigation bypass.

The sample comprised one handset, one firmware build and one runtime attempt. No independent reproduction or internal memory instrumentation was performed. The assessment neither establishes that every unit with the same marketing name is affected nor rules out other impacts. It does not assign a device-specific CVSS score: the observed local lab conditions and limited outcome do not justify extrapolating the upstream issue's severity to this handset.

5. Defensive implications

For an older device still in service, check its support status and available firmware through the manufacturer or carrier. Turn wireless debugging off when it is not needed and remove pairings that are no longer in use. Use a controlled test network for managed lab devices. If a device no longer receives security updates, reassess whether it should handle sensitive accounts or data. These measures reduce exposure; none is presented as a patch for the kernel defect.

For an analyst evaluating a similar finding, record the exact sample privately, keep device-local listeners separate from network reachability, and state what each measurement can establish. When two instruments disagree, retain the discrepancy and explain how it was resolved. A negative result is most useful when the reader knows the conditions under which it was observed.

6. Disclosure and AI assistance

This is an independent case study of a previously disclosed upstream issue, not a claim of original CVE discovery or a new Samsung vulnerability. No vendor acknowledgment, coordinated disclosure timeline, or fix for the tested firmware was established in the supplied record.

The lab notes identify Claude Code as an analysis aid and record the author's use of Anthropic Cyber Verification Program Defense Access. Muhammad Azhar directed the lab work, ran the handset check and reviewed the conclusions against the available artifacts and observations. OpenAI Codex helped prepare and revise the public report from the supplied notes. The drafting record does not establish that Claude Code drafted the original manuscript. Neither form of assistance is independent replication, peer review or vendor endorsement.

7. Citation and references

Suggested citation: Azhar, Muhammad. "Assessing CVE-2026-43499 on a Galaxy S20 Ultra: Firmware Evidence and Runtime Limits." MetaCyberGuru, independent technical report, version 1.2, 2026. Canonical paper. No DOI has been assigned.

Version 1.2: Separates the firmware comparison from the upstream patch, clarifies the scope of the Samsung documentation, and distinguishes analysis assistance from public drafting. No new tests were conducted for this revision.

1. Linux kernel CVE Numbering Authority. CVE-2026-43499. Upstream defect description and references to the maintainer fixes.
2. Samsung Knox. Real-time Kernel Protection. Vendor documentation for the general protection model, not independent validation of this handset.
3. Android Open Source Project. Verified Boot. Android platform documentation.
4. Android Developers. Android Debug Bridge. Documentation for paired wireless debugging.
5. MobiusM. Credited in the lab notes as the author of the previously published runtime test.
6. Azhar, Muhammad. Lab notes, 22 November 2026. Record of the third-party source comparison and handset observations. The public report does not distribute the comparison code or the runtime test.
7. Linux kernel project. Mainline correction for CVE-2026-43499. Upstream source for the expected lock cleanup behavior.