# iOS 27.0.1 → 27.2 Patch Analysis

Reconstructing the Calif.io jailbreak surface from a firmware diff

| | |
|---|---|
| Target | iPhone19,2 (v63ap, t8160) |
| Before | iOS 27.0.1 / 24A446 / Train Rave |
| After | iOS 27.2 / 24B119 / Train RaveB (beta) |
| Method | Structural, per-component and string-level binary diff |
| Deliverable | Patch surface + evidence; no exploit construction |

This document records what changed between the two firmware images and what that change implies for the jailbreak demonstrated by Calif.io at Hexacon on 16 October 2026. It is a patch analysis: it describes bug classes and fixes, not an exploit chain.

# 1. Executive summary

The jailbreak is not contained in either firmware image. 27.0.1 is the build that was jailbroken; 27.2 is the build in which most of the underlying bugs were fixed. The pair therefore forms a vulnerable-versus-fixed comparison, and the fixes identify the primitives the jailbreak was built from.

The dominant signal is in **SPTM** (Secure Page Table Monitor), the component that enforces Memory Integrity Enforcement (MIE) on this SoC. 27.2 adds IOMMU/DART page-table reference-count parity checking, GMMU TLB-invalidation verification, and a bounds check on the hibernation handoff page count. Those additions are consistent with an MIE bypass that targets the device-side translation path rather than the CPU-side tag checks.

Secondary signals: a signedness correction in a kernel mbuf length check, replacement of ad-hoc `sockaddr_dl` validation with an invariant, new typed-allocator sites, and a substantial boot-chain reissue (mBoot 20457.2.37 -> 20457.42.5) together with SEP component changes.

**Bottom line.** Apple did not remove the jailbreak; it removed the primitive it rested on. With the IOMMU-side refcount and TLB-invalidation holes closed, and the boot chain reissued, the 27.2 build is not viable for the demonstrated method.

| Area | 27.0.1 | 27.2 | Assessment |
|---|---|---|---|
| SPTM (MIE) | read_refcnt only | + iommu_read_refcnt parity, verified TLBI | Strong |
| Kernel mbuf | unsigned length compare | signed compact compare | Strong |
| sockaddr_dl | per-site validation | invariant assertion | Strong |
| Boot chain | mBoot-20457.2.37 | mBoot-20457.42.5 | Medium |
| SEP | v63.RELEASE | v63.RELEASE (rebuilt) | Unknown (encrypted) |
| Userland / cryptex | AEA encrypted | AEA encrypted | Not accessible |

# 2. Scope, method and limits

**What was compared.** Both IPSW archives were treated as Zip64 containers. Member inventory was taken with 7-Zip, then cross-matched by name and CRC. Security-relevant components were extracted, IM4P containers parsed, LZFSE payloads decompressed, and the resulting Mach-O / kernel collections parsed manually (MH_FILESET, segment and section headers, LC_SYMTAB, LC_FUNCTION_STARTS). String tables were diffed per section (`__cstring`, `__os_log`) and per kext.

**What could not be compared.**

| Item | Reason |
|---|---|
| System / app cryptex DMGs | AEA-encrypted with device-class keys; no plaintext in the bundle |
| SEP firmware payload | Encrypted; no readable structure or strings |
| Stripped code | SPTM/TXM/iBoot carry no symbol or function-start tables |
| Restore ramdisk / OS volume | AEA-encrypted, same constraint as cryptex |

**Analysis-tooling note.** The REA/Ghidra provider was unavailable in this environment (registration missing; installed Ghidra 12.2 is outside the supported 12.1.x line and `analyzeHeadless.bat` is absent; the expected JDK is not installed). All results below come from direct binary parsing and are reproducible with the parsers listed in Appendix A.

**Confidence labels.** every claim is marked Observed (read from the binaries), Inferred (reasoned from the patch), or Unknown (not determinable from these files).

# 3. Artifact identity

| Property | 27.0.1 | 27.2 |
|---|---|---|
| Product version | 27.0.1 | 27.2 |
| Build | 24A446 | 24B119 |
| Restore long version | 24.1.446.0.0 | 24.2.119.0.0 |
| Build train | Rave | RaveB |
| Image name | V63RestoreIPSW | V63V64RestoreIPSW |
| Supported devices | iPhone19,2 | iPhone19,2 / 19,3 / 19,7 |
| Zip size (bytes) | 12,761,486,095 | 13,722,855,612 |
| Members | 204 | 247 |
| Board config | v63ap (BDID 10, CPID 0x8160) | v63ap included |

SHA-256 digests of the archives themselves are recorded in Appendix A. Because 27.2 is a multi-device beta image, all component comparisons below use the manifest identity that matches `iPhone19,2 / v63ap / Customer Erase Install` in both files.

**Note.** The 27.2 archive carries three kernel caches (v63, v64, v6x-pandia) while 27.0.1 carries only v6x-pandia. The per-device comparison therefore uses `kernelcache.release.v6x-pandia`, which exists in both.

# 4. Archive-level diff

| Metric | Count |
|---|---|
| Members in 27.0.1 | 204 |
| Members in 27.2 | 247 |
| Same name, byte-identical (CRC + packed size) | 115 (2.26 MB) |
| Same name, different content | 66 |
| Only in 27.0.1 | 23 |
| Only in 27.2 | 66 |
| Manifest components changed (iPhone19,2) | 123 of 187 |

The system images are renumbered between builds and none are byte-identical under a new name:

```
043-69473-662.dmg.aea   9,261,023,232  ->  141-68953-104.dmg.aea   9,529,458,688
043-70107-711.dmg.aea   2,348,810,240  ->  141-68931-108.dmg.aea   2,415,919,104
043-69851-764.dmg.aea     251,658,240  ->  141-68940-107.dmg.aea     260,046,848
043-68478-764.dmg.aea     188,743,680  ->  141-68945-108.dmg.aea     553,648,128
```

Only 2.26 MB of the payload is common and unchanged; effectively the entire multi-gigabyte system content is different. Consequently a byte-level comparison of the archives is not informative, and the meaningful comparison is component by component.

# 5. Boot chain and SEP

The boot chain moved substantially, which matters for any claim of reboot persistence.

| Component | 27.0.1 | 27.2 |
|---|---|---|
| iBoot / LLB / iBSS / iBEC | mBoot-20457.2.37 | mBoot-20457.42.5 |
| DeviceTree v63ap | unchanged hardware tree | build timestamp only |
| sep-firmware v63.RELEASE | 12,983,271 B | 12,966,887 B |
| sep-patches v63 | 21,137 B | rebuilt |
| exclavecore_bundle | 33,259,754 B | 33,489,130 B |
| txm | 557,088 B | 557,088 B (different UUID) |

**Observed.** The bootloader version jumps from the .2 series to the .42 series. That is a security-branch reissue, not a cosmetic change. LLB and iBoot payloads remain the same size class and are the same image on this platform. The DeviceTree carries no new functional property, so the change is in code, not in hardware description.

**Observed.** SEP firmware and SEP patch containers were both rebuilt, and the SEP-related kernel drivers changed (AppleSEPManager, AppleSEPKeyStore, AppleSEPCredentialManager, AppleMesaSEPDriver).

**Unknown.** The SEP payload is encrypted and yields no readable strings or structure, so the specific SEP change cannot be characterised from these files. The boot-chain change is likewise not attributable to a single function because the image is stripped.

**Inference.** Persistence across reboot and across an OS update cannot be provided by a kernel-only bug. It requires either boot-chain trust or a component that survives system-volume replacement. The simultaneous reissue of the boot chain and SEP components is consistent with closing that link, but the evidence stops at "reissued", not at "this specific check".

# 6. Kernel collection

Version: `KernelManagement_host-514.2.2` → `514.40.8`. Decompressed size 78,315,520 → 78,675,968 bytes (+360 KB). 304 fileset entries (kexts) in 27.2 versus 303 in 27.0.1; the difference is `com.apple.driver.ApplePencilDriver`.

## Section growth in com.apple.kernel

| Section | 27.0.1 | 27.2 | Delta |
|---|---|---|---|
| __TEXT_EXEC | 0x8f3718 | 0x8fa9d8 | +0x62c0 |
| __cstring | 0x8bf19 | 0x8ca7e | +0xb65 |
| __os_log | 0x41afb | 0x41fdc | +0x4e1 |
| __kalloc_type | 0x14fc0 | 0x15080 | +0xc0 |
| __assert | 0x148c | 0x1630 | +0x1a4 |
| __DATA_SPTM | 0x4c000 | 0x4c000 | unchanged |

Growth in `__kalloc_type` means new type-isolated allocation sites were added, which is adjacent to MIE hardening. Growth in `__assert` means new invariants were introduced.

## New kernel messages (verbatim)

```
%s: mbuf %p len (%d) < off+len (%d+%d) @%s:%d
sdl->sdl_alen == size
External objects are not implemented for CoW. %p %i %i %llx %llx @%s:%d
Exclaves requirements which have been relaxed
memorystatus: swap is disabled, bypassing fast-wake warmup
utun_flowswitch_attach - ifnet_detach failed: %d @%s:%d
```

## Removed in the same area (verbatim)

```
%s: mbuf %p len (%u) < off+len (%u+%u) @%s:%d
%s: rejecting AF_LINK gateway on index %u with unusable link address (nlen %u, alen %u, sdl_len %u)
%s: route %s on %s%d link address does not fit (sdl_alen %u, error %d)
Page modified after framebank withdraw %p
```

**Observed.** The mbuf bounds message changes from unsigned (%u) to signed (%d) formatting, i.e. the comparison itself changed signedness. This is a length-check correction in `bsd/kern/kpi_mbuf.c`.

**Observed.** Two per-call-site `sockaddr_dl` validations in the routing path were removed and replaced by the invariant `sdl->sdl_alen == size`. Link-address length handling in `sockaddr_dl` is a known heap-overflow surface; the patch converts scattered checks into a single precondition.

**Observed.** A new VM CoW sanity check was added for external objects.

**Notable removal.** `Page modified after framebank withdraw` disappears. "Framebank" refers to the EMTE tag-storage side of MIE. A check that detects modification of a page after tag withdrawal is directly MIE-adjacent; its removal between builds warrants examination of the 27.0.1 binary rather than being read as a regression.

# 7. SPTM: the MIE bypass surface

SPTM is the component that enforces memory integrity on this SoC. It grew from 1,261,600 to 1,327,136 bytes decompressed, with __LATE_CONST increasing by 64 KB. This is the single most significant change in the comparison.

## Added in 27.2 (verbatim strings)

```
%s(%s:%d) - fte(%p), new_type(%s), page_flags(%u), read_refcnt(%d) iommu_read_refcnt(%u)
%s: %s: iommu_read_refcnt (0x%hhx) > read_refcnt(0x%x) for FTE %p
%s: Failed to verify TLBI
%s: The GMMU TLBI verification register should be 8B-aligned
%s: gmmu-tlbi-verification-bit (%llu) is unset or out of range [0, 63]
%s: verify-gmmu-tlbis-at-sync is enabled but gmmu-tlbi-verification-reg is not set
%s: UAT Dekker gate acquisition timed out after %llu ns (spun for %llu cycles)
%s: UAT Dekker lock acquisition timed out after %llu ns (spun for %llu cycles)
%s: dart %p (%s:%u): relaxed_rw_protections and allow_pte_remap are not supported together
%s: dart %p (%s:%u): Inconsistent unganged clock protection configuration
%s: unexpected size %u for property '%s'
/chosen/manifest-properties
hib_header_copy->handoffPageCount < HIB_HANDOFF_PAGECOUNT_LIMIT
```

## Removed from 27.2 relative to 27.0.1

```
%s(%s:%d) - fte(%p), new_type(%s), page_flags(%u), read_refcnt(%d)
```

**Observed.** 27.0.1 tracks a single `read_refcnt` on its FTE bookkeeping structure. 27.2 adds a parallel `iommu_read_refcnt` and asserts that `iommu_read_refcnt > read_refcnt` is not reachable. That is a reference-count consistency fix between the CPU-side and DART-side views of the same page-table entry.

**Observed.** 27.0.1 issues GMMU TLB invalidations without verification. 27.2 adds verification at synchronisation points, enforces an 8-byte alignment requirement for the verification register, and rejects a configuration where verification is requested without a valid register.

**Observed.** New mutual-exclusion and configuration checks on DART page-table remapping (`relaxed_rw_protections` versus `allow_pte_remap`), new Dekker-gate locking with timeouts on the UAT path, and a hard upper bound on `handoffPageCount` in the hibernation header.

**Inference.** EMTE applies to CPU accesses. DART translates device DMA through page tables that are not tag-checked by EMTE. A reference-count desynchronisation between the CPU view and the device view permits a page to be freed and reused while a DART mapping still resolves to it; an unverified or misaligned TLB invalidation leaves a stale translation live. Either yields device-side access to memory the kernel believes it has reclaimed. This is a bypass *around* MIE rather than a defeat of EMTE, which is consistent with the public claim that the jailbreak bypassed MIE.

**Inference.** The hibernation bound is an independent primitive: without it, a crafted hibernation header can specify an out-of-range handoff page count, giving memory corruption very early in the boot path. Its addition is a straightforward bounds fix.

**Unknown.** SPTM is stripped (no symbol or function-start tables), so these changes cannot be attributed to named functions from the images alone. Function-level attribution requires pattern matching on the __TEXT_EXEC range.

# 8. Kext-level changes (string triage)

Every kext differs byte-for-byte after a relink, so raw byte comparison is not a useful triage signal. Comparing only real `__cstring`/`__os_log` content narrows 304 entries to 124 with any change, and 53 with meaningful message changes. The largest deltas are listed below; a kext with no string delta is most likely unmodified apart from layout.

| Kext | + new | - removed |
|---|---|---|
| com.apple.driver.AppleH16ANEInterface | 25 | 165 |
| com.apple.driver.AppleAVE2 | 14 | 64 |
| com.apple.kernel | 6 | 53 |
| com.apple.driver.AppleMobileDispH19P-DCP | 4 | 42 |
| com.apple.driver.RTBuddy | 2 | 26 |
| com.apple.iokit.IOTimeSyncFamily | 17 | 8 |
| com.apple.filesystems.apfs | 6 | 19 |
| com.apple.driver.AppleSARService | 7 | 17 |
| com.apple.driver.AppleHIDTransportSPI | 11 | 11 |
| com.apple.plugin.IOgPTPPlugin | 20 | 0 |
| com.apple.driver.AppleProcessorTrace | 9 | 9 |
| com.apple.driver.AppleM2ScalerCSCDriver | 4 | 10 |
| com.apple.iokit.IOMobileGraphicsFamily-DCP | 0 | 12 |
| com.apple.filesystems.lifs | 5 | 7 |
| com.apple.driver.AppleDisplayCrossbar | 0 | 12 |
| com.apple.security.sandbox | 5 | 5 |
| com.apple.driver.AppleCamera | 3 | 7 |
| com.apple.driver.AppleSPMIPMU | 4 | 4 |
| com.apple.driver.AppleCS42L79Audio | 4 | 4 |
| com.apple.driver.AppleT8160MCC | 0 | 7 |
| com.apple.iokit.IONVMeFamily | 1 | 5 |
| com.apple.driver.SecureRTBuddyProxy | 0 | 6 |
| com.apple.driver.AppleSmartBatteryManagerEmbedded | 1 | 5 |
| com.apple.driver.AppleiPhoneACSKPPM | 0 | 5 |
| com.apple.driver.AppleSEPManager | 2 | 3 |
| com.apple.driver.AppleActuatorDriver | 1 | 4 |
| com.apple.EXBrightKext | 2 | 3 |
| com.apple.security.AppleImage4 | 2 | 2 |

Full ranking and all messages are preserved in the analysis artifacts listed in Appendix A.

## Other components with security-relevant deltas

| Component | Observation |
|---|---|
| AppleMobileFileIntegrity | DER-entitlement handling messages removed |
| com.apple.security.sandbox | 5 messages changed (rootless flag reporting) |
| AppleImage4 | 2 messages changed |
| AppleLockdownMode | messages changed |
| apfs / lifs | new validation and invalidation messages |
| AppleH16ANEInterface | large refactor: new endpoint size validation, DART flush errors |
| AppleAVE2 | large refactor: parameter validation restructured |

# 9. Reconstruction of the chain

The following is a link-by-link reconstruction assembled from the patch evidence plus public reporting. Each link is labelled with its evidence strength.

| # | Link | Evidence | Status in 27.2 |
|---|---|---|---|
| 1 | PDF 1-click entry (iMessage / Signal / AirDrop) | Public reporting only | Not visible (cryptex) |
| 2 | Sandbox / platform escape | AMFI, sandbox, Image4 touched | Touched, weak |
| 3 | IOMMU / DART page-table refcount | SPTM iommu_read_refcnt parity | Closed (strong) |
| 4 | Stale GMMU TLB translation | SPTM TLBI verification | Closed (strong) |
| 5 | Kernel foothold | mbuf signedness, sockaddr_dl invariant | Closed (strong) |
| 6 | Persistence across reboot + update | Boot chain and SEP reissued | Unknown specifics |

**What killed it.** A chain fails when any single link is removed. Here several were removed at once. The decisive one for the MIE claim is link 3/4: SPTM now enforces parity between the CPU-side and IOMMU-side reference counts, and now verifies that TLB invalidation actually occurred and is correctly aligned. With those in place, the device-side translation path is no longer a way around EMTE. The boot-chain and SEP reissue removes the persistence foothold, which is what made the demonstration survive reboot.

**What is not claimed.** This document does not identify a CVE, does not attribute a specific function to a specific stage of the chain, and does not describe how to construct any primitive. Attribution requires the userland images, which are encrypted inside the restore bundle.

# 10. Defensive implications

**For device operators.** The exposure is the 27.0.1 build, not 27.2. Any device still on 24A446 remains exposed to the demonstrated method; the 27.2 build carries the fixes listed here. Where a device cannot be updated, treat it as potentially compromised and prioritise update over any configuration-level mitigation, because the bypass is below the layer configuration controls.

**For detection.** The fixes define observable behaviour: SPTM refcount parity violations, TLBI verification failures, and out-of-range hibernation handoff counts are all now surfaced as diagnostics in 27.2. Their presence in logs on a 27.2 device indicates either an attempted exploit or memory corruption; their absence on 27.0.1 devices is expected and not reassuring.

**For researchers.** The most productive follow-up is function-level attribution inside the 27.0.1 SPTM image around the DART/FTE paths, and a structural comparison of the hibernation handoff parser before and after the bound was added. Both are tractable: SPTM is ~1.3 MB and self-contained.

# Appendix A. Artifacts and reproduction

All intermediate results are retained under the analysis working directory. The parsers are self-contained Python and require only `lzfse` (and optionally `capstone` for disassembly).

| Artifact | Purpose |
|---|---|
| diff.json | Member-level archive diff (names, sizes, CRCs, classification) |
| cstr.py / cstr2.txt | Per-kext section-level string diff and full ranking |
| kextdiff.txt | Per-kext segment hashes and string delta for all 304 kexts |
| macho.py | MH_FILESET / segment / section / symtab parser |
| im4p.py | IM4P (and bare-DER kernelcache) parser with LZFSE decompression |
| funcdiff.py | Function-start enumeration and per-function hashing |
| bindiff.py | Generic payload string differ for flat images |
| bin/a, bin/b | Decompressed kernelcache, SPTM, TXM, iBoot, LLB, DeviceTree, exclavecore |

**Reproduction outline.** (1) list both archives and cross-match members by name and CRC; (2) select the manifest identity matching the target product type; (3) extract the changed components; (4) parse IM4P and decompress LZFSE payloads; (5) parse the kernel collection and enumerate filesets and sections; (6) diff only cstring/log sections per component; (7) for flat images, diff printable-string sets. This pipeline reproduces every observation in this document without any device interaction.

## Archives (SHA-256).

27.0.1 — iPhone19,2_27.0.1_24A446_Restore.ipsw, 12,761,486,095 bytes
53310AFBC73D3868F6DB19A92BE3EF7C8AC6150AC200143E6B4385E4AA7A9C41

27.2 — iPhone19,2,iPhone19,3,iPhone19,7_27.2_24B119_Restore.ipsw, 13,722,855,612 bytes
5702869F40603DC0B8F1A42DFFE45A2F78E6A65AFF9AE6A8588A041FA78B2F11