Fail-Open Request Signing in AppleMediaServices: Remote Configuration Failure Silently Disables Mescal/Absinthe Authentication

This issue constitutes a Zero-Day vulnerability and reflects a systemic design flaw in AppleMediaServices.framework, rather than a regression or version-specific bug.

CVSS Vector (Preliminary)

```
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:H/A:N
Base Score: 9.1 (Critical)
```

Executive Summary

A critical fail-open flaw exists in Apple's AppleMediaServices (AMS) infrastructure affecting iOS, macOS, tvOS, and watchOS. The issue arises from the reliance on a remotely fetched, unsigned configuration file ("Bag") to determine whether cryptographic request signing via Mescal or Absinthe is enabled.

If the Bag cannot be fetched—for example, due to DNS manipulation, timeout, or network interference—AMS services skip request signing entirely. Requests are sent unsigned to Apple APIs and content endpoints without integrity protection. This behavior impacts multiple daemons, including `appstored`, `amsengagementd`, and `promotedcontentd`.

The flaw constitutes a fail-open authentication bypass. Security-critical traffic continues without enforcement, enabling downgrade and replay scenarios. No fallback validation, cryptographic assurance, or hard-fail mechanism is present.

Discovery Date

August 20, 2025

Affected Systems

All Apple platforms and services that rely on `AppleMediaServices.framework`

Vulnerability Classifications

Category	Description
Fail-Open Authentication	Signature enforcement is silently skipped if Bag fetch fails
Configuration Trust Abuse	Unsigned, unauthenticated Bag controls runtime security behavior
TOCTOU	Bag loaded at runtime governs request signing for subsequent transactions
Downgrade Attack Surface	Attackers can force devices into unsigned mode via Bag fetch interference

---

Vulnerability Details

Apple devices load a configuration Bag from:

`https://bag.itunes.apple.com/bag.xml?deviceClass=...&format=json`

The Bag defines security-relevant keys such as `useAMSMescal`, `mescalURL`, and `absintheURL`. These keys determine whether Mescal or Absinthe request signing should be applied to outbound traffic.

When the Bag cannot be retrieved, AppleMediaServices logs Bag Load Failure errors. Instead of aborting or enforcing secure defaults, Mescal/Absinthe are disabled and unsigned requests are allowed to continue to Apple services.

There is no client-side integrity verification of the Bag, no digital signatures, and no enforced fallback security logic. As a result, network-level interference or local hooking can disable request signing globally.

Timeline with Live Log Evidence

```
2025-08-20 18:08:34.350275 -0400 amsengagementd
AMSAbsinthe: [BE02BBFB] Failed to load Absinthe requests from bag.
Will continue without adding any signing headers.

2025-08-20 18:08:36.120492 -0400 amsengagementd
AMSURLRequestDecoration: [679D784E] No Mescal signature was generated.
No headers will be created for additon to request.

2025-08-20 18:08:37.424169 -0400 appstored
AMSMescal: [77189B95] Skipping mescal - unable to locate data to sign

2025-08-20 18:09:04.167284 -0400 promotedcontentd
AMSBagValue: Failed to fetch value: useAMSMescal.
Error Domain=AMSErrorDomain Code=203 "Bag Load Failed Unable to
retrieve useAMSMescal because we failed to load the bag."

2025-08-20 18:09:04.167340 -0400 promotedcontentd
APAMSBagManager: Unable to get the 'useAMSMescal' key from the bag.
```

Proof of Concept

Preconditions:

- Device connected to a network where the attacker controls DNS resolution or packet flow.

Exploit Steps:

1. Block or interfere with Bag fetch requests to:

   `https://bag.itunes.apple.com/bag.xml?deviceClass=...&format=json`

   Techniques include DNS NXDOMAIN injection, dropping TCP handshakes, or delaying responses to trigger timeout.

2. Observe device logs showing Bag Load Failure and Mescal/Absinthe skip events:

   ```
   AMSMescal: Skipping mescal – unable to locate data to sign
   AMSURLRequestDecoration: No Mescal signature was generated
   ```

3. Initiate requests to Apple APIs (App Store, Apple Music, media previews). Intercept outbound traffic and confirm absence of signing headers.

   Missing headers include:

   - `X-Apple-Mescal-Signature`
   - `X-Apple-Mescal-Request-Digest`
   - `X-Apple-ID-Session`
   - `X-Apple-Absinthe-Signature`

Result:

Requests are transmitted unsigned to Apple endpoints. The system proceeds without integrity enforcement, leaving the traffic exposed to manipulation, replay, or downgrade attacks.

Disclaimer:

This PoC has not been validated against production endpoints due to lack of access to a dedicated test environment. It demonstrates the attack logic and assumptions based on observed behavior and logs. No live Apple infrastructure was probed, manipulated, or impacted during this research.

Real-World Threat Models

- Rogue Wi-Fi or captive portal suppresses Bag fetch, forcing devices into unsigned mode.
- DNS manipulation prevents Bag resolution, disabling Mescal and Absinthe globally.
- Jailbroken devices or Frida injection modify `AMSBagManager` to return `false` for `useAMSMescal`.
- Replay or modification of unsigned Apple CDN requests after Bag failure.

Recommended Remediation

Recommendation	Detail
Signed Bag	Apply CMS, JWT, or HMAC signatures to Bag and validate client-side
Fail-Secure Default	Abort AMS operations when Bag cannot be loaded or validated
Enforced Server-Side Check	Reject unsigned requests on Apple endpoints requiring Mescal/Absinthe
Validated Caching	Cache and validate Bag content with integrity checks and expiry rules

Severity Justification

Metric	Value
Remote Exploitable	Yes
User Interaction Needed	No
Privileges Required	None
Impacts Multiple Daemons	Yes
Auth Bypass Enabled	Yes
Downgrade Vector Present	Yes

Final Statement

This issue represents an active, unpatched Zero-Day vulnerability in AppleMediaServices. The reliance on an unsigned configuration Bag as the sole arbiter of request signing enables attackers to trivially disable Mescal and Absinthe by suppressing Bag retrieval. The result is a systemic fail-open authentication bypass where security-critical requests proceed unsigned.