Airwatch/Workspace ONE Mobile Device Management (MDM) Auto-Lock Enforcement Bypass

AVALON DIGITAL FORENSICS

AVALON

Airwatch/Workspace ONE Mobile Device Management (MDM) Auto-Lock Enforcement Bypass

Subject: iOS (2024-2026) Airwatch/Workspace ONE Mobile Device Management (MDM) Auto-Lock Enforcement Bypass: Browser Screen-Awake Behavior Can Prevent Lock State Transition on Managed iOS Devices

Classification: Coordinated Exploitation Disclosure/Public Dissemination for Security Awareness

Executive Summary

Organizations commonly rely on MDM-enforced Auto-Lock and passcode policies to ensure iOS devices reliably transition to a locked state after a maximum idle period. On iOS devices running iOS 26.2, this control intent can be undermined and defeated because a foreground browser session can sustain an "awake" display state and prevent the device from reaching the lock state for extended periods.

This paper explains the issue in detail, outlines the potential risks, and recommends mitigation steps, along with a coordinated disclosure approach for Apple and MDM vendors. Importantly, this is not a security breach such as a passcode crack, encryption bypass, or hardware exploit. Instead, it represents a gap between what administrators expect the auto-lock policy to enforce and how the operating system or browser behaves in certain situations. As a result, devices may remain unlocked longer than intended.[1]

Affected Configuration and Observed Conditions

The behavior was confirmed on the following configuration:

Device model: iPhone 11 (A2111)

OS: iOS 26.2 (released December 12, 2025)

The relevant policy context is an MDM posture where Auto-Lock settings are restricted, and a maximum auto-lock interval is expected to force a lock transition. For this device it is "three minutes."

Control Intent Versus Observed Outcome

MDM passcode and lock policies are designed to reduce risk if a device is left unattended. Most administrators assume that the "maximum auto-lock" setting means a device will automatically lock after a certain amount of inactivity, no matter what.

However, testing shows that this doesn't always happen. If a web browser is open and active on the

1 https://support.apple.com/en-us/125884

screen, it can keep the display awake, preventing the device from locking when the auto-lock timer should normally trigger. One possible reason is a feature similar to a "screen wake lock," which allows apps or websites to keep the screen on when they need to continue running.

This behavior can also occur in more data-heavy scenarios; for example, when a long video is playing on a site like YouTube, which keeps the screen active for extended periods. The difference in the scenario described here is that the screen can remain awake without obvious media playing, which may reduce the amount of activity captured during a live forensic extraction of the device.

Security Impact

This issue can increase the risk of unauthorized access if a device is left unattended but remains awake longer than expected. Because the screen does not lock when administrators think it will, someone with physical access could potentially use the device during that extended window. It can also create compliance and audit concerns, since the real-world behavior of the device does not match the security policy administrators believe is being enforced.

In some situations, this behavior can actually be useful. For example, it may help keep a device active during legitimate tasks such as device servicing, data migration, or authorized data collection. However, that convenience does not remove the underlying risk: under certain conditions, a browser session can prevent a security policy from locking the device when it should.

In simple terms, the issue weakens a security feature designed to automatically lock a device and protect sensitive data if a user walks away and leaves it unattended.

Non-Operational Proof Description

This section explains the general nature of the issue for security review, without providing details that would allow someone to easily reproduce it.

In testing, a managed iPhone was configured with MDM policies that enforce automatic locking after a set period of inactivity and limit the user's ability to change those settings. The device was then left with a web browser open in the foreground. In this state, the browser kept the screen awake, and the device sometimes remained unlocked longer than the maximum idle time administrators expected – or behaved inconsistently across repeated tests – despite the MDM policy.

To prevent misuse, the exact website, URL structure, settings, and step-by-step actions used to confirm the behavior are not included in this document. Those technical details are reserved for a controlled disclosure appendix that will only be shared with the affected vendors and relevant enterprise stakeholders on a need-to-know basis.

AVALON

Airwatch/Workspace ONE Mobile Device Management (MDM) Auto-Lock Enforcement Bypass

Root Cause Hypothesis

The most likely explanation is a definitional mismatch between "idle" as interpreted by the OS or browser runtime and "idle" as assumed by MDM policy administrators. Web platform features exist to keep screens awake for legitimate use cases, and the OS may treat that condition as "in use," even in the absence of touch interaction.[2]

If an MDM "maximum auto-lock" policy does not override features that keep the screen awake – such as wake locks or similar mechanisms – the policy may not be enforced as intended. In those cases, the device may stay unlocked longer than administrators expect.

When a device does not reliably lock itself and require Face ID, Touch ID, or a passcode, its ability to protect sensitive data is weakened. As a result, the MDM control meant to enforce automatic locking may not be as effective as administrators assume.

Validation Methodology and Acceptance Criteria

Testing should be performed with a consistent time source and documented start and stop times. The objective is to compare lock behavior under a baseline condition versus a screen-awake condition, holding all other variables constant.

Baseline condition (expected behavior):

- Place the device in a neutral, non-screen-awake state (for example, a static home screen or settings screen) and do not interact with it.
- Measure elapsed time until the device locks.
- The expected result under the enforced posture is a lock transition at approximately three minutes.

Test condition (screen-awake behavior present):

- Place the device in a foreground browser context that sustains an awake display state, then do not interact with the device.
- Measure elapsed time until the device locks.
- The gap is considered present if lock is prevented or materially delayed beyond the three-minute threshold compared to baseline.

Acceptance criteria for confirming the gap: The issue is considered confirmed if the device stays unlocked for longer than the enforced three-minute auto-lock setting during the test condition, while

2 https://developer.mozilla.org/en-US/docs/Web/API/Screen_Wake_Lock_API

the baseline test (without the condition present) locks at or close to the expected three-minute mark within normal timing variations.

Evidence to document: To support the finding, the following evidence should be collected: a time-stamped screen recording or external video showing the device state and elapsed time; a snapshot of the MDM policy confirming the three-minute auto-lock setting; the device's iOS build version; and any relevant acquisition tool logs if the behavior occurs during a data collection workflow.

Risk and Mitigation Guidance

Risk framing: A three-minute auto-lock setting is commonly used to limit how long a device can remain accessible if it is left unattended. If a browser session in the foreground can keep the device from locking beyond that three-minute limit – without meaningful user interaction – the actual window of exposure becomes longer than the organization intends, especially when the device is physically accessible.

It's important to clarify that the issue affects the auto-lock timer itself. The three-minute example is just one scenario observed in 2025; similar behavior has been seen with other configured time limits as well.

Mitigations (near-term): Organizations can reduce risk by treating browser-based "keep-awake" situations as a temporary exception that requires additional safeguards. When a device needs to remain unlocked for legitimate operational reasons, it should remain under continuous physical supervision. The start and end of this exception period should also be documented. In addition, organizations should require the device to prompt for a passcode immediately once it locks, with no grace period. This ensures the device returns to a secure state as soon as the screen finally locks.

Mitigations (mid-term): Where possible, organizations should limit browser access or restrict unattended browser use on higher-risk devices through supervised device controls and internal policies. For situations that require a device to remain unlocked for extended periods, such as authorized data collection or technical servicing, organizations should establish a formal, case-specific process. This process should include clear approvals and audit documentation rather than relying on informal or ad hoc practices.

Remediation request (platform and MDM): The goal of remediation is to give organizations a way to ensure devices lock at the configured time limit, even if a browser or app is trying to keep the screen awake. At a minimum, systems should be able to detect and report when a screen-awake request causes a device to stay unlocked longer than the configured auto-lock limit. This would allow administrators to identify when the policy is not being enforced as expected.

Airwatch/Workspace ONE Mobile Device Management (MDM) Auto-Lock Enforcement Bypass

Conclusion

The observed behavior on iPhone 11 devices running iOS 26.2: The three-minute MDM auto-lock setting may not reliably trigger when a browser session keeps the screen awake. While this is not a passcode or encryption bypass, it reduces the effectiveness of an administrative control designed to limit the time a device remains unlocked when unattended.

Organizations should promptly verify whether their managed iOS devices exhibit this behavior under existing lock policies, paying particular attention to high-risk devices and those in physically accessible environments. If the issue is confirmed, compensating controls should be put in place to maintain continuous physical oversight during any authorized workflow that requires extended unlocked time, and to avoid relying on Auto-Lock as an absolute security boundary in risk assessments and compliance reporting.

Coordinated Disclosure and Remediation Path

This issue should be reported through coordinated disclosure to Apple Product Security and the relevant MDM vendor(s), including controlled test artifacts, policy snapshots, OS build information, and observed results. The remediation goal is to provide an enterprise-level solution that either enforces device lock at the configured threshold regardless of browser screen-awake behavior or, at a minimum, delivers administrative telemetry that detects and records when screen-awake activity causes a device to remain unlocked longer than policy allows.

Limitations

This paper reflects testing on a limited configuration and should be treated as an initial advisory pending broader validation across device models, iOS builds, supervision states, and MDM platforms. Additional testing may identify variance based on OS version, browser engine behavior, or specific MDM payload interactions.

For further technical details, reproduction artifacts under controlled distribution, or coordination of remediation efforts, contact:

Jonathan Edwards

Senior Forensic Investigator

jonathan.edwards@teamavalon.com

AVALON

www.teamavalon.com