Detection of Mutex-Based Execution Guardrails Across Platforms, Detection Strategy DET0132

Archived: 2026-04-05 17:44:26 UTC

AN0372

Adversary-created named mutex using system APIs (e.g., CreateMutexW) followed by conditional process termination or alternate code path indicating malware avoiding reinfection.

Log Sources

Mutable Elements

Field	Description
mutex_name_entropy_threshold	Filter out common benign mutex names; highlight suspicious high-entropy/dynamic names.
parent_process_path	Limit alerting to non-standard parent-child relationships indicative of malware staging or self-spawning.
TimeWindow	Correlate mutex creation + rapid process exit or lack of further activity within a short timeframe.

AN0373

File lock acquired via open() + flock() or lockf() on predictable path (e.g., /tmp/.lock123) followed by conditional early exit or divergent process behavior.

Log Sources

Mutable Elements

Field	Description
lockfile_path_regex	Detect patterns like /tmp/.lock*, /var/run/*lock used by malware.
exit_code	Track specific exit codes (e.g., 1, 2) that signal lock acquisition failure.
TimeWindow	Correlate lockfile access + early process termination within N seconds.

AN0374

User-mode application uses flock() or NSDistributedLock to gain exclusive access to a resource file (e.g., /tmp/guard.lock), conditional logic alters execution if already locked.

Log Sources

Mutable Elements

Field	Description
lockfile_path	Path to mutex file (e.g., /tmp/*, /private/tmp/*), tune per environment.
user_context	Flag non-user processes using these APIs.
TimeWindow	Detection correlation across short time intervals between lock attempt and process exit.
