Cross-Platform Behavioral Detection of File Timestomping via Metadata Tampering, Detection Strategy DET0591

Archived: 2026-04-05 18:07:32 UTC

AN1626

Detects attempts to modify file timestamps via API usage (e.g., `SetFileTime`), CLI tools (e.g., `w32tm`, PowerShell), or double-timestomp behavior where $SI and $FN timestamps are mismatched or reverted.

Log Sources

Mutable Elements

Field	Description
TimeWindow	Correlate timestamp change with preceding file creation or suspicious access
APINamePattern	Include SetFileTime, NtSetInformationFile, or other timestamp APIs
TimestampDeltaThreshold	Trigger on excessive backdating (e.g., >90 days)

AN1627

Detects use of timestamp-altering commands like `touch -a -m -t` or `touch -r`, particularly when executed by unusual users or in suspicious directories.

Log Sources

Mutable Elements

Field	Description
MonitoredCommandList	Commands like `touch -r`, `debugfs`, `stat` used in sequence
FilePathRegex	Suspicious paths like `/tmp/`, `/var/lib/`, `/mnt/esxi/`
DeltaThreshold	Mismatch between timestamp and file activity time

AN1628

Detects timestamp changes using `touch`, `SetFile`, or direct metadata tampering (e.g., xattr manipulation) from Terminal, scripts, or low-level APIs.

Log Sources

Mutable Elements

Field	Description
CommandMatch	Touch/setfile and backdated timestamps
UserContext	Detects execution under non-interactive/system accounts

AN1629

Detects abuse of busybox commands (e.g., `touch`) or log timestamp tampering during backdoor persistence or evasion.

Log Sources

Mutable Elements

Field	Description
TimestampAgeComparison	Unusual backdating to match legit files
PersistenceOverlap	Overlap with known persistence paths
