Driver Metadata, Data Component DC0074

Archived: 2026-04-05 17:26:00 UTC

to contextual data about a driver, including its attributes, functionality, and activity. This can involve details such as the driver's origin, integrity, cryptographic signature, issues reported during its use, and runtime behavior. Examples include metadata captured during driver integrity checks, hash validation, or error reporting. Examples:

- Driver Signature Validation: A driver is validated to ensure it is signed by a trusted Certificate Authority (CA).
- Driver Hash Verification: The hash of a driver is compared to a known good hash stored in a database.
- Driver Compatibility Issues: A driver error is logged due to compatibility issues with a particular version of the operating system.
- Vulnerable Driver Identification: Metadata indicates the driver version is outdated or contains a known vulnerability.
- Monitoring Driver Integrity: Drivers are monitored for any unauthorized modifications to their binary or associated files.

This data component can be collected through the following measures:

Windows

- Windows Event Logs:
  - Event ID 3000-3006: Logs metadata about driver signature validation.
  - Event ID 2000-2011 (Windows Defender Application Control): Tracks driver integrity and policy enforcement.
- Sysmon Logs: Configure Sysmon to capture driver loading metadata (Event ID 6).
- Driver Verifier: Use Driver Verifier to collect diagnostic and performance data about drivers, including stability and compatibility metrics.
- PowerShell: Use commands to retrieve metadata about installed drivers: `Get-WindowsDriver -Online | Select-Object Driver, ProviderName, Version`

Linux

- Auditd: Configure audit rules to monitor driver interactions and collect metadata: `auditctl -w /lib/modules/ -p rwxa -k driver_metadata`
- dmesg: Use `dmesg` to extract kernel logs with driver metadata: `dmesg | grep "module"`
- lsmod and modinfo: Commands to list loaded modules and retrieve metadata about drivers: `lsmod` | `modinfo <module_name>`

macOS

- Unified Logs: Collect metadata from system logs about kernel extensions (kexts): `log show --predicate 'eventMessage contains "kext load"' --info`

- kextstat: Command to retrieve information about loaded kernel extensions: `kextstat`

SIEM Tools

- Ingest Driver Metadata: Collect driver metadata logs from Sysmon, Auditd, or macOS logs into SIEMs like Splunk or Elastic.

Vulnerability Management Tools

- Use these tools to collect metadata about vulnerable drivers across enterprise systems.
