Tanium™ Client Recorder Extension ユーザーガイド

バージョン 1.0. 0

2020年 4月 01日

この文書の内容は予告なく変更されることがあります。また、本書に記載の内容は「現状のまま」提供されており、正確には万全を期しておりますが、Taniumの顧客販売契約に規定されている保証を除き、明示または暗黙を問わずいかなる保証もしません。別段の規定がない限り、Taniumはいかなる責任も負いません。TaniumおよびそのサプライヤはTanium Inc.がかかる損害の可能性を事前に通知されていたとしても、本書の使用または使用できないことから生じる、利益損失やデータ損失をはじめとする間接的損害や特別損害、結果的損害、および付随的損害に対して一切の責任を負いません。

本書で使用されているIPアドレスは、実際のアドレスであることを意図していません。本書に記載されている例、コマンド表示出力、ネットワークトポロジ図、およびその他の図は、例示の目的にのみ使用されています。例示コンテンツに実際のIPアドレスが使用されていたとしても、特別な意図はなく、偶然です。

最新のTanium製品のマニュアルについては、https://docs.tanium.com を参照してください。

Taniumは米国およびその他の国におけるTanium,Inc.の商標です。記載されているその他の社名、製品名、サービス名は各社の商標または登録商標です。

目次

Client Recorder Extensionの概要

Client Recorder Extensionは、Tanium Integrity Monitor、Tanium Map、Tanium Threat Response、およびTanium Traceソリューションモジュールに共通する機能です。これは重要なフォレンジックエビデンスを継続的に各エンドポイントに保存します。Client Recorder Extensionは、エンドポイントカーネルとその他の下位レベルのサブシステムを監視し、さまざまなイベントを捕捉します。

従来のディスクやメモリのフォレンジック技術では、エンドポイントのアクティビティの断片を正常に再構築できますが、これは基盤となるオペレーティングシステムがネイティブに保持するエビデンスに限定されます。対象期間からのこのタイプのエビデンスは、時間の経過とともに急速に劣化する可能性があります。対照的に、Client Recorder Extensionは、完全で解釈しやすいイベント履歴を保持するので、最新のシステムイベントを再現することができます。

アイドル状態のシステムでも、迅速にデータを蓄積します。Client Recorder Extensionは、ローカルデータベース内のイベントデータを格納します。デフォルトの構成では、履歴データを数か月間まで保持できます。Client Recorder Extensionが消費するローカルストレージの容量をカスタマイズし、記録されたエビデンスのタイプをフィルタリングすることができます。

記録されたイベントのタイプ

Client Recorder Extensionは、追加のコンテキストとメタデータを含む、広い範囲のイベントを捕捉します。記録されたイベントの例：

- プロセス実行
- ファイルシステムアクティビティ
- レジストリ変更
- ネットワーク接続
- ドライバおよびライブラリの読み込み
- ユーザー認証

エンドポイントのオペレーティングシステムに適用されるかどうかにかかわらず、どのプロセス、レジストリ、ネットワーク、ファイル、セキュリティイベントを記録するか指定できます。

> **ヒント：** データベースをより有用にするため、また、より長い期間データを保持するため、発生回数の多いイベントなどを除外することを検討してください。たとえばLanguageListレジストリ値はWindowsエンドポイント上の冗長なイベントです。

選択したレジストリイベント、ネットワークイベント、ファイルイベント、セキュリティイベント、またはDNSイベントにイベントレコーディングを制限するフィルタを構成できます。

レジストリ

[Windows のみ]レジストリキーと値の作成や変更など、レジストリの変更。関連するプロセスとユーザーコンテキストが含まれます。

ネットワーク

関連するプロセスとユーザーコンテキストを含む、インターネットロケーションへのHTTP要求などのネットワーク接続イベント。すべてのインバウンドとアウトバウンドのTCP接続のイベントが記録されます。

ファイル

エンドポイントのディレクトリに書き込まれたファイルなど、ファイルシステムイベント。関連するプロセスとユーザーコンテキストが含まれます。例：Windows Updateが使用する場所にコピーされたマルウェアファイル、またはファイルに対するコンテンツの変更。

セキュリティ

[Windows と Linux のみ]セキュリティイベントには、認証、特権エスカレーションなどがあります。このイベントタイプには、ログオンイベントが含まれます。

DNS

[Windows 8.1 以降]プロセスパス、ユーザー、クエリ、応答、および操作の種類を含む要求情報。

WindowsでのClient Recorder Extensionデータのソース

Client Recorder Extensionは、複数のソースから、エンドポイントの1つのローカルデータベースにデータを収集します。カーネルイベントはWindowsツールから収集されます。Windowsエンドポイントでは、オプションのMicrosoft Sysmonを設定することで、実行されたプロセスに関する追加情報が提供されます。

Client Recorder Extensionの一部の機能には、特定のバージョンのWindowsが必要です。

表 1：Client Recorder Extensionの機能 - Windows

機能	Windows Server 2008 R2	Windows Server 2012	Windows Server 2012 R2 以降	Windows 7	Windows 8	Windows 8.1以降
DNS イベント	利用不可	利用不可	利用可能	利用不可	利用不可	利用可能
プロセス ハッシュとコマンドライン情報	Taniumドライバまたはsysmonが必要	Taniumドライバを推奨	Taniumドライバを推奨	Taniumドライバまたはsysmonが必要	Taniumドライバを推奨	Taniumドライバを推奨
ドライバ読み込み	利用可能*	利用可能	利用可能	利用可能*	利用可能	利用可能

* Sysmonが設定されている場合、Sysmonによって記録されたドライバの読み込み情報が使用されます。

LinuxでのClient Recorder Extensionデータのソース

Linux向けClient Recorder Extensionは、Linux監査サブシステムを使用してイベントを収集します。Linux向けClient Recorder Extensionは、イベントコレクションに次のコンポーネントを使用します。

カーネルドライバ(kaudit)

このプロセスは、カーネル監査イベントに責任を負うLinuxカーネルの一部であり、監査済みイベントをuauditdプロセスに転送します。監査済みイベントは、ルールファイルが定義します。このルールファイルは、audit.rulesと呼ばれ、`/etc/audit/audit.rules`にあります。kauditdプロセスに読み込むことができる追加のルールファイル、および`audit.rules`ファイルは`/etc/audit/rules.d/`にあります。

監査デーモン(auditd)

このプロセスは、ネットリンクソケットを介してカーネルに通信します。ほとんどのLinuxバージョンでは、これは単一リスナーに限定されます。このプロセスは、監査ログファイルに書き込むか、イベントをaudispdプロセスに転送します。

監査ディスパッチャ(audispd)

このプロセスは、シングルリスナーソケットの制限を克服するためのイベントマルチプレクサです。このプロセスは、監査プロセスから監査イベントを消費し、リアルタイムでイベントを分析したい子プラグインにディスパッチします。Client Recorder Extensionは、これらの子プラグインの一例です。これらの子プラグインの構成は、`/etc/audisp/plugins.d/`でご覧いただけます。

Client Recorder Extension

Client Recorder Extensionは、audispdプロセスから監査済みイベントを収集するaudispdプラグインとして機能し、`monitor.db`と名付けられたSQLiteデータベースに書き込まれます。このデータベース内のテーブルは生のイベントを保管する、記録されたデータテーブルからデータのクエリを定義する仮想テーブルで、データ収集を簡素化します。Client Recorder Extensionおよび`monitor.db`は`/opt/Tanium/TaniumClient/Tools/Trace/`にあります。

注意：セキュリティとDNSイベントの記録はLinuxでは使用できません。

MacでのClient Recorder Extensionデータのソース

Macエンドポイント上で、レコーダは10.8から最新までのすべてのMacリリースにインストールされているOpenBSM監査システムからデータを収集します。

レコーダは`/dev/auditpipe`のクローンに接続され、イベントをmonitor.dbに記録します。レコーダがMacエンドポイントにインストールされている場合、`/etc/security/audit_class`はTanium Recorderのエントリーで更新され、ランタイムで登録済みイベントをマッピングします。それから、レコーダは`/dev/auditpipe`をコピーし、コピーを構成して`tan`監査クラスを使用します。レコーダ構成が読み取られると、必要なイベントのタイプが適切なシステムコールに翻訳され、監視されます。これらのコールはtan監査クラスにマッピングされます。これにより、レコーダが監査済みイベントをエンドポイントのaudit.logに書き込むことを防ぎ、レコーダが監視するシステムコールを精選することが可能となります。

レコーダは次のイベントタイプをmonitor.dbに書き込みます。

プロセスイベント

プロセスのコマンドライン

プロセスハッシュ

ネットワークイベント

ファイルイベント

注意： セキュリティとDNSイベントの記録はMacでは利用できません。

はじめに

1. Client Recorder Extensionをインストールします。くわしくは、14ページのClient Recorder Extensionのインストールをご覧ください。
2. エンドポイント設定を構成します。くわしくは、20ページのエンドポイント設定の構成をご覧ください。
3. Client Recorder Extensionを使用する複数のモジュールが構成設定を管理する方法を理解します。くわしくは、51ページのモジュール全体の構成の管理をご覧ください。

Client Recorder Extensionの要件

Client Recorder Extensionを含むモジュールをインストールする前に、要件を確認してください。

Taniumの依存関係

Client Recorder Extensionを含む製品モジュールのライセンスに加えて、ご使用の環境が以下の要件を満たしていることも確認してください。

コンポーネント	要件
Tanium Platform	6.5以降 拡張機能は、バージョン7.0.314.6042以降で使用できます。Tanium™ Interactのインストールも推奨されます。 詳細については、Tanium Core Platformインストールガイド：Taniumサーバのインストールを参照してください。
Tanium Client	Client Recorder Extensionは、Tanium Clientと同じLinuxおよびMacエンドポイントでサポートされています。Windowsエンドポイントの場合、最低限のWindows 7またはWindows Server 2008 R2が必要です。Windows 8.1には、DNSイベント記録性能があります。 特定のTanium Clientのバージョンの詳細については、Tanium Clientデプロイガイド：クライアントホストシステム要件を参照してください。
以下のTaniumモジュールの1つ:	
Tanium Module	以下のTaniumモジュールの1つ: • Tanium™ Trace • Tanium™ Threat Response • Tanium™ Integrity Monitor • Tanium™ Map

Tanium Module Server

Client Recorder Extensionをインストールするモジュールは、Module Serverホストコンピュータ上のサービスとしてインストールおよび実行されます。Module Serverへの影響は最小限であり、使用状況によって異なります。

エンドポイント

Client Recorder Extensionは、Windows、Linux、Macのエンドポイントをサポートします。Windowsエンドポイントの場合、最低限のWindows 7またはWindows Server 2008 R2が必要です。Windows 8.1には、DNSイベント記録性能があります。必要な空きディスク領域の量は、Client Recorder Extensionの構成によって異なります。3GBを推奨します。

各エンドポイントデバイスには、最低100MBのRAMが必要です。デフォルトでは、エンドポイントデータベースのサイズは1GBです。空きディスク領域で、最大データベースサイズの3倍が必要です。エンドポイントでのCPU要求は平均1%未満です。

Linuxエンドポイントの場合は、次の作業を行う必要があります。

- エンドポイントを初期化する前に、最新の安定版の監査デーモンとaudispdプラグインをインストールします。手順については、特定のオペレーティングシステムのマニュアルを参照してください。
- 不変の「-e 2」モードを使用する場合、Client Recorder Extensionは不変フラグの前にTanium監査ルールを追加することに注意してください。**-e 2**フラグをLinux上で使用する場合、Client Recorder Extensionを使用する各製品のステータスセンサーは、レコーダを再起動する必要があるかどうかを示します。

Tanium Event RecorderドライバまたはMicrosoft Sysmonを使用して、サポートされているWindowsエンドポイントにプロセスとコマンドラインイベントを記録することができます。以下のオペレーティングシステムは、Tanium Event Recorderドライバをサポートしています。

- Windows 7
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2
- Windows 8.1
- Windows 10、ビルド1607以降
- Windows Server 2016
- Windows Server 2019

注：

- Windows 10、ビルド 1607より前のWindowsオペレーティングシステムは、Tanium署名証明書がオペレーティングシステムにより信頼されることを確保するには KB3033929をインストールする必要があります。KB3033929について詳しくは、

https://support.microsoft.com/en-us/help/3033929/microsoft-security-advisory-availability-of-sha-2-code-signing-supportをご覧ください。

- Windows 10、ビルド 1511は、Tanium Event Recorderドライバはサポートしていません。

サードパーティのソフトウェア

(Windows、オプション) Microsoft Sysmon

Windows 8.1およびWindows Server 2012 R2より前のWindowsエンドポイント上のプロセスハッシュとコマンドライン情報を記録するために、Tanium Event Recorderドライバに加えて、サポートされている最新バージョンのMicrosoft Sysmonを使用できます。Windows 8.1以降およびWindows Server 2012 R2以降では、Sysmonは必要ありません。

ホストとネットワークセキュリティの要件

セキュリティの除外

未知のホストシステムプロセスを監視およびブロックするためにセキュリティソフトウェアが環境内で使用されている場合、セキュリティ管理者はTaniumプロセスを干渉なく実行できるように除外を作成する必要があります。モジュールが期待通りに作動するために必要な、完全な除外の参照用モジュールユーザーガイドをご覧ください。以下のテーブルは、Client Recorder Extensionに必要な除外事項を示しています。

対象デバイス	プロセス
Module Server	`<Tanium Module Server>\services\<ProductName>\node.exe`
エンドポイントコンピュータ (Windows)	`<Installation Location>\sysmon.exe`
エンドポイントコンピュータ (Linux)	`<Tanium Client>/Tools/Trace/recorder`
エンドポイントコンピュータ(Mac)	`<Tanium Client>/Tools/Trace/TaniumRecorder`

Client Recorder Extensionのインストール

Client Recorder Extensionは、イベントデータを記録するために、モジュールがインストールします。**[Distribute Tools (ツールを配信)]**Taniumプラットフォームが配信構成ファイルとすべての対象エンドポイント上のソフトウェアを使用するパッケージ。次のリストに、**[Distribute Tools (ツールを配信)]**パッケージをClient Recorder Extensionを使用するモジュール向けエンドポイント上にインストールした構成ファイルとソフトウェアの詳細があります。

Client Recorder Extensionによって追加されたソフトウェアおよび構成ファイル

`/opt/Tanium/TaniumClient/Tools/Trace/recorder` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/TaniumRecorder` (Mac)
`C:\Program Files(x86)\Tanium\Tanium Client\extensions\TaniumCXTrace.dll` (Windows)

Client Recorder Extensionプロセス。このプロセスはaudispdプロセスからのイベントを消費し、`monitor.db`に書き込みます。Windowsエンドポイント上で、Client Recorder ExtensionはTanium Clientが使用するDLLとして実行されます。

Tanium TraceおよびTanium Threat Response

`/opt/Tanium/TaniumClient/Tools/Trace/recorder.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/recorder.json` (Mac)

Client Recorder Extension向けの構成ファイル。このファイルには、モジュールワークベンチに設定されている構成値が含まれています。構成アイテムには、CPU Killswitch値、`monitor.db`サイズ、データを保存する最大日数、auditd RAWログの有効化または無効化、ログレベルの指定、`monitor.db`および`filters.json`へのパスの構成などを含みます。

`/opt/Tanium/TaniumClient/Tools/Trace/filters.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/filters.json` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\filters.json` (Windows)

ネットワーク、プロセス、レジストリおよびファイルイベントを記録から選別するフィルタを含む構成ファイル。これらのフィルタは、Tanium TraceまたはTanium Threat Responseワークベンチから構成されています。

`/etc/audisp/plugins.d/trace.conf` (Linux/Mac)

イベントをClient Recorder Extensionに転送するためのaudispdプロセスの構成ファイル。この構成ファイルは、auditdが停止または再起動する時に、Tanium Recorderを再起動するためにも使用されます。

`/opt/Tanium/TaniumClient/Tools/Trace/monitor.db` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/monitor.db` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\monitor.db` (Windows)

Client Recorder Extensionが作成したデータベース。記録されたイベント詳細の履歴が含まれています。

Tanium Integrity Monitor

Tanium Integrity Monitorのインストール中に、追加ファイルがTraceディレクトリに追加されます。

`/opt/Tanium/TaniumClient/Tools/Trace/im_recorder.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/im_recorder.json` (Mac)
`HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Tanium\Tanium Client\Trace` (Windows)

Client Recorder Extension向けのIntegrity Monitor固有の構成。

`/opt/Tanium/TaniumClient/Tools/Trace/im_filters.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/im_filters.json` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\im_filters.json` (Windows)

記録するイベントのタイプを定義するためのIntegrity Monitor固有の構成。

`/opt/Tanium/TaniumClient/Tools/Trace/watchlist.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/watchlist.json` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\watchlist.json` (Windows)

記録するパスのリスト、パス除外、ファイルイベント操作を含む、Integrity Monitor固有の構成ファイル。このリストの値は、他のTaniumモジュールからの`filters.json`内で見つかったフィルタをすべて上書きします。

Tanium Map

Tanium Mapのインストール中に、追加ファイルがTraceディレクトリに追加されます。

`/opt/Tanium/TaniumClient/Tools/Trace/map_filters.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/map_filters.json` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\map_filters.json` (Windows)

レコーダはTanium Map用のファイルをフィルタします。

`/opt/Tanium/TaniumClient/Tools/Trace/map_recorder.json` (Linux)
`/Library/Tanium/TaniumClient/Tools/Trace/map_recorder.json` (Mac)
`C:\Program Files (x86)\Tanium\Tanium Client\map_recorder.json` (Windows)

この構成ファイルは、Client Recorder Extensionの構成パラメータを保存します。

エンドポイントでの構成変更

[Distribute Tools (ツールを配信)]パッケージは、Client Recorder Extensionを使用するモジュールをインストールする場合に、対象エンドポイント上の監査構成に変更を加えます。

次のリストは、MacおよびLinuxエンドポイントの構成ファイルおよび監査サブシステムに変更の詳細を示しています。

`/etc/audit/auditd.conf` (Linux)

監査デーモンに固有の構成ファイル。モジュールワークベンチへのClient Recorder Extensionのインストールは、RAWログを有効または無効に設定するように管理者に指示します。Client Recorder Extensionは、既存の`auditd.conf`のバックアップコピーをそのエンドポイントに作成し、`/etc/audit/auditd.conf.pretrace`と名前をつけます。

`/etc/audisp/audispd.conf` (Linux)

構成ファイルは、監査イベントディスパッチャプロセスの構成を制御します。Client Recorder Extensionは、既存の`audispd.conf`ファイルのバックアップコピーをそのエンドポイントに作成し、`/etc/audisp/audispd.conf.pretrace`と名前をつけます。Client Recorder Extensionは、q_depth設定を32768に変更します。

`/etc/audit/audit.rules` (Linux/Mac)

このファイルは、カーネル監査システムが記録する監査イベントを指定します。このファイルはカーネル監査システムに読み込まれます。Client Recorder Extensionがエンドポイントにインストールされると、既存の`audit.rules`ファイルのバックアップコピーが作成され、`/etc/audit/audit.rules.pretrace`と名前が付けられます。

Client Recorder Extensionの開始および停止

Client Recorder Extensionを手作業で開始または停止する必要があります。たとえば、`monitor.db`の規模が設定された制限を超えた場合や、CPU使用率がしきい値を超えてClient Recorder Extensionを自動的に無効にした場合は、まず根本的な問題を解決してから、Client Recorder Extensionを手作業で再起動する必要があります。または、Client Recorder Extensionが予想以上に多くのシステムリソースを使用していることが判明した場合は、Client Recorder Extensionを停止して、さらにリソースを消費するリスクのトラブルシューティングを行うことができます。

> **重要:** 停止後、自動的に再起動しないため、Client Recorder Extensionを手作業で再起動する必要があります。

Client Recorder Extensionを停止するには、以下のテーブルのオペレーティングシステムに対応するステップを実行します。

オペレーティングシステム	手順
Windows	a. Windowsのスタートメニューから**[Run (ファイル名を指定して実行)]**をクリックします。 b. `services.msc`と入力して**[OK]**をクリックします。 c. Tanium Clientサービスを指定し、右クリックで**[Stop (停止)]**を選択します。
Linux	`Tools/<module>`ディレクトリから、rootまたはスーパーユーザーで次のスクリプトを実行します。 `./recorder --stop`

オペレーティングシステム	手順
Mac	`Tools/<module>`ディレクトリから、rootまたはスーパーユーザーで次のスクリプトを実行します。 `./TnmRecorder --stop`

Client Recorder Extensionを開始するには、以下のテーブルのオペレーティングシステムに対応するステップを実行します。

オペレーティングシステム	手順
Windows	a. Windowsのスタートメニューから**[Run (ファイル名を指定して実行)]**をクリックします。 b. `services.msc`と入力して**[OK]**をクリックします。 c. Tanium Clientサービスを指定し、右クリックで**[Start (開始)]**を選択します。
Linux	`Tools/<module>`ディレクトリから、rootまたはスーパーユーザーで次のスクリプトを実行します。 `./recorder --start`
Mac	`Tools/Trace`ディレクトリから、rootまたはスーパーユーザーで次のスクリプトを実行します。 `./TnmRecorder --start`

また、パッケージをアクションとして配置することで、レコーダを停止または開始することもできます。

1. 影響を受けたエンドポイントを絞り込むためにQuestionを使用します。たとえば、`Get Tanium Threat Response Status from all machines`のように尋ねます。
2. 特定のエンドポイントまでドリルダウンします。
3. レコーダを無効にするアクションとして**[Disable Tanium Recorder (Taniumレコーダの無効化)] [*OS*]**パッケージをデプロイします。
4. レコーダを有効化するアクションとして**[Enable Tanium Recorder (Taniumレコーダの有効化)][*OS*]**パッケージをデプロイします。

(オプション) Tanium Event Recorderドライバのインストール

Tanium Event Recorderドライバは、インストールするとプロセスとコマンドラインイベントをより正確に取り込むことができます。

Tanium Event Recorderドライバは、Sysmonと同じエンドポイントに存在することができます。エンドポイントにSysmonがある場合、Tanium Clientがリセットされたときまたはエンドポイントが再起動するときに、WindowsイベントレコーダからTaniumイベントレコーダドライバにSysmonを通じて切り替えできます。エンドポイントに、ForCesysmon=1 レジストリ設定 (HKLM\Software\Wow6432\Tanium\Tanium Client\Trace\)でSysmonを使用し続けさせることができます。

1. メインメニューから、`Get Tanium Driver Status from all machines`のQuestionを実行して、**[Search (検索)]**をクリックします。
2. **[Install Recommended (インストールを推奨)]**を選択します。
3. デプロイアクションページから、**[Install Tanium Driver (Taniumドライバのインストール)]**を選択します。
4. パッケージインストールの終了時に実行する検証クエリを確認することで、インストールがうまくいったことを確認します。
5. Live Responseを使用して検証クエリに失敗したエンドポイントからアクションログを収集します。
6. Tanium Event Recorder ドライバサービスステータスでSERVICE_RUNNING以外を返したエンドポイントでは、**[Remove Tanium Driver (Taniumドライバの削除)]**アクションを実行します。

次にやるべきこと

Client Recorder Extensionの使用については、10ページのはじめにを参照してください。

エンドポイント設定の構成

`recorder.json` (Linux/Mac)またはレジストリエントリ(Windows)で設定を定義することにより、Client Recorder Extensionを構成します。Client Recorder Extensionを使用するモジュールは、デフォルト設定を提供します。Client Recorder Extensionの構成設定を変更する前に、テクニカルアカウントマネージャ(TAM)に相談してください。

> **注意：** 担当のテクニカルアカウントマネージャは、お客様の目的に合わせてClient Recorder Extensionを最適に構成する方法についてアドバイスします。構成パラメータの変更は重大で、取り返しのつかない結果となる可能性があります。

audisp

監査イベントディスパッチャの構成とプラグインを定義します。(Linuxのみ)

audisp構成

q_depth

監査イベントディスパッチャの内部キューのサイズを指定する数値です。より大きなキューは、大量のイベントを処理する能力がありますが、audispデーモンが終了したときに処理されていないイベントを保留できます。ドロップされたイベントに関するsyslogメッセージが報告された場合、イベントのこの値を増加します。デフォルト：80。

overflow_action

内部キュー内のオーバーフローにデーモンがどのように対応するかを指定します。オーバーフローが発生した場合、キューが処理できるより多くのイベントを受け取れます。このパラメータには次の選択肢があります。

- `ignore`
- `syslog`
- `suspend`
- `single`
- `halt`

`ignore`を設定すると、audispデーモンは何もしません。`syslog`を設定すると、syslogに警告が発行されます。`suspend`を設定すると、audispデーモンはイベント処理を停止しますが、デーモンはアクティブなままです。`single`を設定すると、audispデーモンは

エンドポイントを単一ユーザーモードにします。haltを設定すると、audispデーモンがエンドポイントをシャットダウンします。デフォルト：SYSLOG。

priority_boost

監査イベントディスパッチャを適用すべき優先順位を指定するマイナスでない数字です。この優先順位は、監査デーモンから提供される優先順位に追加されます。デフォルト：4。変更なし：0。

max_restarts

監査イベントディスパッチャが非応答のプラグインを再起動できる回数を指定するマイナスでない数字。デフォルト：10。

name_format

エンドポイントノード名が監査イベントストリームに挿入される方法を指定します。このパラメータには次の選択肢があります。

- none
- hostname
- fqd
- numeric
- ユーザー

noneと設定すると、監査イベントにコンピュータ名は挿入されません。hostnameと設定すると、名前はgethostname syscallによって返されます。fqdと設定すると、ホスト名は、エンドポイントのFQDNのDNSで解決されます。numericと設定すると、エンドポイントのIPアドレスが解決されます。userと設定すると、名前オプションからの管理者定義文字列が返されます。名前オプションの詳細については、名前パラメータの文書をご覧ください。デフォルト：None。

name

ユーザーがname_formatオプションとして提供される場合、エンドポイントを指定するadmin定義の文字列。

audispプラグイン

active

監査が開始または再開した場合は、Client Recorder Extensionを再起動するよう指定します。デフォルト：`Yes`。

direction

有効になると、このパラメータは、どの方向にイベントがフローするかのインサイトをイベントディスパッチャに提供します。デフォルト：`In`。

path

プラグイン実行可能への絶対パスを指定します。内部プラグインの場合、プラグインの名前に対応します。

タイプ

プラグインがディスパッチャに実行する方法を指定します。オプションは`builtin`および`always`です。監査イベントディスパッチャの内部にあるプラグインのビルトイン、例えば、`af_unix`や`syslog`を使用します。`always`オプションを、すべてのプラグインではなければ、ほとんどのプラグインに使用してください。デフォルト：`always`。

args

引数を子プログラムに渡す方法を指定します。ほとんどの場合、プラグインは引数を取得せず、構成ファイルを使用して構成方法を指示することができます。2つの引数が上限です。

形式

このパラメータのオプションは、バイナリと文字列です。バイナリは、監査イベントディスパッチャが監査デーモンから取得したとおりのデータを渡します。文字列オプションは、監査構文解析ライブラリで解析するのに適した文字列にイベントを完全に変更するように監査ディスパッチャに指示します。デフォルト：`string`。

auditd

構成ファイルの監査セクションには、監査デーモンに固有の情報が含まれています。1行当たり1つの構成キーワード、等号、そのあとに適切な構成情報を含める必要があります。(Linuxのみ)

log_file

監査レコードが格納されるログファイルにフルパス名を指定します。symlinkではなく、通常のファイルである必要があります。

log_format

ログ情報をディスクに保存する方法を指定します。次の2つのオプションがあります。

- RAW
- NOLOG

RAWに設定する場合、監査レコードは、カーネルが送信する形式で格納されます。このオプションがNOLOGと設定されている場合、すべての監査情報は破棄されますが、監査イベントディスパッチャに送信されたデータには影響しません。デフォルト：RAW。

log_group

ログファイルへのアクセス許可に適用されるグループを指定します。グループ名は、数値またはスペルアウトのどちらかにできます。デフォルト：root。

priority_boost

監査デーモンに指示するマイナスでない数字で、割り当てる優先順位ブーストの量を指定します。デフォルト：4。変更なし: 0。

flush

監査レコードをフラッシュするときに指定します。値は次のとおりです。

- none
- incremental
- data
- sync

noneに設定すると、監査記録をディスクにフラッシュするための特別な作業は行われません。incrementalに設定すると、freqパラメータは、ディスクに対して明示的なフラッシュが発行される頻度を指定します。dataに設定すると、監査デーモンは、ディスクファイルのデータ部分を常に同期させます。syncに設定すると、監査デーモンは、データとメタデータの両方をディスクへの書き込みすべてに完全に同期させます。

freq

ディスクコマンドへの明示的なフラッシュを発行する前に、書き込みレコードの数を指定するマイナスでない数値。この値は、フラッシュキーワードが増分に設定されている場合にのみ有効です。

num_logs

rotateがmax_log_file_actionとして提供される場合、保存するログファイルの数を指定します。数値が2未満の場合、ログはローテーションしません。この数値は99以下でなければなりません。ローテーションするログファイルの数を増やすと、カーネルバックログ設定は、ファイルをローテーションするのに必要な時間に合わせて増やすことができます。カーネルバックログ設定は通常、/etc/audit/audit.rulesで指定されます。デフォルト：0 (ローテーションなし)。

disp_qos

監査デーモンとディスパッチャ間のブロック/ロスレスまたはブロック解除/ロシー通信を指定します。監査デーモンとディスパッチャ間には128kバッファがあります。lossyを選択した場合、このキューがフルになっていると、ディスパッチャへの入力イベントが破棄されます。しかし、log_formatがNOLOGに設定されている場合、イベントはディスクに書き込まれます。そうでなければ、ディスクにログする前に、監査デーモンが空きスポットをキューで待ちます。リスクは、デーモンがネットワークIOを待機中に、イベントがディスクに記録されないことです。有効な値は次のとおりです。lossyおよびlosslessです。デフォルト：lossy

dispatcher

ディスパッチャは、監査デーモンが開始するプログラムです。すべての監査イベントのコピーをこのプログラムのstdinに渡します。このラインに追加するアプリケーションは、ルート権限で実行するので信頼する必要があります。

name_format

コンピュータノードを監査イベントストリームに挿入する方法を指定します。name_formatパラメータは、次のオプションをサポートします。

- none
- hostname
- fqd

- numeric
- ユーザー

noneと設定すると、監査イベントにコンピュータ名は挿入されません。hostnameを設定する場合、gethostnameシステムの呼び出しによって返された名前は、監査イベントに挿入されます。fqdと設定すると、ホスト名は、エンドポイントのFQDNのDNSで解決されます。numericと設定すると、ホスト名はエンドポイントのIPアドレスで解決されます。このオプションを使用して、「hostname -i」または「domainname -i」が数値アドレスを返すことをテストします。このオプションは、同じエンドポイントで時間が経過すると異なるアドレスを持つ可能性があるため、DHCPには推奨されません。Userは、名前オプションからの管理者定義文字列です。デフォルト：None。

名前

userがname_formatオプションとして提供される場合、エンドポイントを指定する管理者定義の文字列を指定します。

max_log_file

メガバイトで最大ファイルサイズを指定します。この上限に達すると、構成可能なアクションをトリガーします。この値は数値である必要があります。

max_log_file_action

最大ファイルサイズ制限に達したことをシステムが検出したときに取るアクションを指定します。max_log_file_actionパラメータは、次のオプションをサポートします。

- ignore
- syslog
- suspend
- ratate
- keep_logs

ignoreを設定すると、監査デーモンは何もしません。syslogを設定すると、監査デーモンはsyslogに警告を発行します。suspendを設定すると、監査デーモンはレコードをディスクへ書き込みを中止しますが、デーモンは引き続き有効です。rotateを設定すると、監査デーモンは、高い数値のログが低い数値のログより古くなるようにログをローテーションさせます。keep_logsオプションは、ローテーションと同様で、違うのは監査ログが上書きされるのを防止するためにnum_logs設定を使用しないことです。デフォルト：SYSLOG。

action_mail_acct

有効なEメールアドレスまたはエイリアスを指定します。Eメールアドレスがエンドポイントのローカルでない場合、ローカルコンピュータおよびネットワーク上で電子メールが適切に構成されていることを確認します。このオプションでは、`/usr/lib/sendmail`がローカルコンピュータに存在することが必要です。デフォルト：`root`。

space_left

ローカルコンピュータがディスクスペースが欠乏した状態で開始している場合、設定可能アクションをいつ実行するか監査デーモンに指示するメガバイト数値です。

space_left_action

ディスクスペースが欠乏し始めたのをローカルシステムが検知した場合、とるべき対応を指定します。space_left_actionパラメータは、次のオプションをサポートします。

- `ignore`
- `syslog`
- `email`
- `exec`
- `suspend`
- `single`
- `halt`

`ignore`を設定すると、監査デーモンは何もしません。`syslog`を設定すると、監査デーモンはsyslogに警告を発行します。`email`と設定すると、監査デーモンは、`action_mail_acct`で指定されたメールアカウントに警告を送信し、syslogにメッセージを送信します。`exec`と設定すると、監査デーモンは提供されたスクリプトを実行します。注意：パラメータをスクリプトに渡すことはできません。`suspend`を設定すると、監査デーモンはレコードをディスクへ書き込むのを中止しますが、デーモンは引き続き有効です。`single`を設定すると、監査デーモンはエンドポイントを単一ユーザーモードにします。`halt`を設定すると、監査デーモンがエンドポイントをシャットダウンします。

admin_space_left

ローカルコンピュータがディスクスペースが欠乏した状態で実行している場合、設定可能アクションをいつ実行するか監査デーモンに指示するメガバイト数値です。これは、ディスク領域がなくなる前に何かを実行する最後のチャンスと考えてください。このパラメータの数値は`space_left`の数値より小さくする必要があります。

admin_space_left_action

ディスクスペースが欠乏し始めたのをシステムが検知した場合、とるべき対応を指定します。admin_space_left_actionパラメータは、次のオプションをサポートします。

- ignore
- syslog
- email
- exec
- suspend
- single
- halt

ignoreを設定すると、監査デーモンは何もしません。syslogを設定すると、監査デーモンはsyslogに警告を発行します。emailと設定すると、監査デーモンは、action_mail_acctで指定されたメールアカウントに警告を送信し、syslogにメッセージを送信します。syslogを設定すると、監査デーモンはsyslogに警告を発行します。execと設定すると、提供されたスクリプトを実行します。パラメータをスクリプトに渡すことはできません。suspendを設定すると、監査デーモンはレコードをディスクへ書き込むのを中止しますが、デーモンは引き続き有効です。singleを設定すると、監査デーモンはエンドポイントを単一ユーザーモードにします。haltを設定すると、監査デーモンがエンドポイントをシャットダウンします。デフォルト：SYSLOG。

disk_full_action

ログファイルが書き込まれたパーティションがフルになっているのをシステムが検出したときに取るアクションを指定します。disk_full_actionパラメータは、次のオプションをサポートします。

- ignore
- syslog
- exec
- suspend
- single
- halt

ignoreを設定すると、監査デーモンはsyslogメッセージを発行しますが、その他のアクションは実行しません。syslogを設定すると、監査デーモンはsyslogに警告を発行します。execと設定すると、提供されたスクリプトを実行します。パラメータをスクリプトに渡すことはできません。suspendを設定すると、監査デーモンはレコードをディスクへ書き込むのを中止しますが、デーモンは引き続き有効です。singleを設定すると、監査デーモンはエンドポイントを単一ユーザーモードにします。haltを設定すると、監査デーモンがエンドポイントをシャットダウンします。デフォルト：SYSLOG。

disk_error_action

監査イベントをディスクまたはローテーションログに書き込む際に、エラーを検知した場合にとるべきアクションを指定します。disk_error_actionパラメータは、次のオプションをサポートします。

- ignore
- syslog
- exec
- suspend
- single
- halt

ignoreを設定すると、監査デーモンは抑制する前に最大5つのsyslogメッセージを発行し、その他のアクションは実行しません。syslogを設定すると、監査デーモンはsyslogに警告を発行します。execと設定すると、提供されたスクリプトを実行します。パラメータをスクリプトに渡すことはできません。suspendを設定すると、監査デーモンはレコードをディスクへ書き込むのを中止しますが、デーモンは引き続き有効です。singleを設定すると、監査デーモンはエンドポイントを単一ユーザーモードにします。haltを設定すると、監査デーモンがエンドポイントをシャットダウンします。デフォルト：SYSLOG。

tcp_listen_port

1～65535の範囲の数値で指定した場合、auditdはリモートシステムから監査記録用に対応するTCPポートをリッスンします。監査デーモンはtcp_wrappersとリンクできます。このオプションを使用して、hosts.allowおよびdenyファイルへの入力のアクセスを制御します。

tcp_listen_queue

接続の保留(要求されているが、未承認)がいくつ許容されるかを示す数値。これを小さく設定しすぎると、停電の後など、同時に開始するホストが多すぎる場合に接続が拒否されます。デフォルト：5。

tcp_max_per_addr

1つのIPアドレスからの同時接続がいくつ許可されているかを示す数値。最大値は16です。この値を大きく設定すると、ログサーバ上のサービス妨害攻撃の可能性が増大します。カスタムリカバリスクリプトが未送信イベントに転送されない限り、ほとんどの場合は

デフォルトで十分です。この場合、問題が起きないように数値のみを増やします。デフォルト：1。

use_libwrap

tcp_wrappersを使用するかどうかを指定して、許可されたエンドポイントからの接続試行を識別します。オプションはyesおよびnoです。デフォルト：yes

tcp_client_ports

単一の数値、またはダッシュ(スペースなし)で区切られた2つの値。受信接続に使用するクライアントポートを示します。指定されていない場合、すべてのポートが許可されます。サポートされている値は1～65535です。例えば、クライアントに特権ポートを使用させるには、このパラメータを1～1023で指定します。local_portオプションをaudisp-remote.confファイルで設定します。クライアントが特権ポートから送信したことを認証するのは、信頼できないユーザーによるログインジェクション攻撃を防止するセキュリティ機能です。

tcp_client_max_idle

監査がアクションを実行する前に、クライアントをアイドル状態にできる秒数を指定します。クライアントエンドポイントで接続を完全に切断できない問題がある場合は、このオプションを使用して、無効な接続をシャットダウンします。これはグローバル設定です。個々のクライアント**heartbeat_timeout**設定よりも高くなければなりませんし、2倍高いのが好ましいです。デフォルト：0 (無効にする)。

enable_krb5

yesと設定すると、認証および暗号化にKerberos 5が使用されます。デフォルト：no

krb5_principal

主たるサーバを指定します。サーバは、認証にあたり次の形式で名前を付けたキーを期待します。/etc/audit/audit.keyに保存されている auditd/hostname@EXAMPLE.COM。ホスト名は、IPアドレスのDNSルックアップが返すサーバのホストの正規名です。デフォルト：auditd

krb5_key_file

クライアントの主たるキーの場所を指定します。キーファイルは、ルートおよびモード0400が所有している必要があります。デフォルト：/etc/audit/audit.key

CPU Killswitchパラメータ

cpuKillAfterNumViolations

Client Recorder Extensionは、違反の上限数を超えるまでシステムイベントを記録し続けます。デフォルト：1。

cpuKillEnabled

cpuThresholdパラメータが提供した値に合わせて、CPUキルスイッチを切り替えます。falseに設定すると、しきい値を超えてもレコーダはシャットダウンしません。デフォルト：True。

cpuThreshold

プロセッサあたりのエンドポイントのCPU使用率が1分間にわたってこの値を超えると、Client Recorder Extensionが無効になり、監査ルールが削除されます。デフォルト：25%。

cpuThresholdMinutes

CPU使用計算を複数分ウィンドウに分配します。最大設定は1440です。デフォルト：1。

データベースおよびフィルタロケーションパラメータ

dbLocation

Client Recorder Extensionデータベースの名前とロケーション。デフォルトでは、レコーダデータベースは`config.json`と同じディレクトリにあります。デフォルト：`monitor.db`。

filterLocation

フィルタ構成の名前とロケーション。 デフォルトでは、このファイルは`config.json`と同じディレクトリにあります。デフォルト：`filters.json`。

ログパラメータ

logLevel

Client Recorder Extensionプロセスに適用するログのレベル。デフォルト：Information。

logMaxSize

ログファイルの最大サイズ。デフォルト：10 MB。

logRotations

ログファイルの最大サイズに達したらログファイルをロールする回数。デフォルト：3。

最大データベースサイズパラメータ

maxSizeMB

Client Recorder Extensionデータベースの最大サイズ(`monitor.db`)。デフォルト：1024 MB。

RAWログパラメータ

rawLogging

trueに設定すると、監査ルールは未処理ログに書き込まれ、保存されます。この設定により、エンドポイントの監査ログボリュームが増加します。falseに設定すると、ログのディスクへの書き込みは無効になります。イベントスループットを改善し、CPU使用率を下げるには、この設定を使用します。生の監査ログの読み取りに依存する他の非Taniumプロセスがないことを確認してください。デフォルト：True。

その他のClient Recorder Extensionパラメータ

throttle

Trueに設定した場合、情報が`/var/log/messages`および`recorder.log`に記録されます。リソースを保護し、混乱の可能性を最小限に抑えるために、処理はより長い時間にわたってイベントを記録するよう調整されます。デフォルト：False。

onlyWatchFileRoot

レコーダにルートフォルダを強制的に監視させます。Integrity Monitorでは、これが設定されていない場合は、Integrity Monitorのみがファイルウォッチモードになります。デフォルト：True。

Windowsレジストリエントリ

Windowsエンドポイント上で、Client Recorder Extension構成データは、`HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Tanium\Tanium Client\Trace`のレジストリエントリに含まれています。

Path

Client Recorder Extensionの作業パスを指定します。

タイプ: REG_SZ

LogVerbosityLevel

モニターログの詳細度を指定します。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 100

LogBufferSizeInMessages

Client Recorder Extensionログファイル内のメッセージの数を指定します。

タイプ: REG_DWORD

デフォルト値 = 1000

下限値 = 0

上限値 = 1000 * 1000

LogFileSize

Client Recorder Extensionログファイルサイズのサイズを KB単位で指定します。

タイプ: REG_DWORD

LogFileSizeInBytes

Client Recorder Extensionログファイルサイズのサイズをバイト単位で指定します。

タイプ: REG_DWORD

デフォルト値 = settings["LogFileSize"]

下限値 = 0

上限値 = (1000 * 1024 * 1024)

LogFileSizeInMessages

Client Recorder Extensionログファイルサイズのサイズをメッセージ数で指定します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = (1000 * 1000 * 1000)

IMToolsPath

Tanium Integrity Monitorツールへのパスを指定します。

タイプ: REG_SZ

settings["path"] + \Tools\IM

TraceToolsPath

Tanium Traceツールへのパスを指定します。

タイプ: REG_SZ

settings["path"] + \Tools\Trace

MapToolsPath

Tanium Mapツールへのパスを指 定します。

タイプ: REG_SZ

settings["path"] + \Tools\Map

ForceIMOnlyMode

Client Recorder ExtensionはTanium Integrity Monitorのみのモードで実 行 するか指 定します。

タイプ: REG_DWORD

デフォルト値 = 0

下 限 値 = 0

上 限 値 = 1

MonitorDatabasePath

モニタデータベースへのパスを指 定します。

タイプ: REG_SZ

MonitorRegistry

レジストリ監 視 を有 効 または無 効 にするかどうかを指 定します。

タイプ: REG_DWORD

デフォルト値 = 1

下 限 値 = 0

上 限 値 = 1

MonitorNetwork

ネットワーク監 視 を有 効 または無 効 にするかどうかを指 定します。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

MonitorFiles

ファイル監視を有効または無効にするかどうかを指定します。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

MonitorDNS

DNS監視を有効または無効にするかどうかを指定します。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

MonitorImageLoad

画像の読み込み監視を有効または無効にするかどうかを指定します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

MonitorDays

データベースに保持する監視データの日数を指定します。

タイプ: REG_DWORD

デフォルト値 = 90

下限値 = 0

上限値 = 65536

MaxStorageSizeMB

`monitor.db`データベースの最大サイズを指定します。0は制限されません。

タイプ: REG_DWORD

デフォルト値 = 1024

下限値 = 0

上限値 = -1

MaxRuntimeStorageSizeMB

実行中にデータベースがこの値を超えると、DisableTraceを設定して終了します。デフォルトは 2 * MaxStorageSizeMBです。

タイプ: REG_DWORD

デフォルト値 = settings["MaxStorageSizeMB"] * 2

下限値 = 0

上限値 = -1

AbsoluteMaxStorageSizeMB

`monitor.db`データベースの最大サイズを指定します。デフォルト：`MaxStorageSizEmB`のサイズ*2。

タイプ: REG_DWORD

デフォルト値 = settings["MaxStorageSizeMB"] * 2

下限値 = 0

上限値 = -1

CleanPercentOverLimit

`MaxStorageSizeMB`を超えるデータベースを削減する割合を指定します。デフォルト10。

タイプ: REG_DWORD

デフォルト値 = 10

下限値 = 0

上限値 = 100

SysmonHistoryLimit

起動時に読み込むSysmon履歴の量を秒単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 3600

下限値 = 0

上限値 = 86400

DatabaseCleanupIntervalMinutes

各 データベースのクリーンアップ間 の時 間 を指 定します。

タイプ: REG_DWORD

デフォルト 値 = 30

下 限 値 = 1

上 限 値 = 2880

FilterDefinitionFile

フィルタ定 義 JSONファイルへのパスを指 定します。

タイプ: REG_SZ

DisableParentProcessFilter

DLLとして実 行 される場 合 、親 プロセスのフィルタリングを無 効 にします。

タイプ: REG_DWORD

デフォルト 値 = 0

下 限 値 = 0

上 限 値 = 1

BatchWriteDelayMS

各 バッチ保 存 操 作 間 の遅 延 をミリ秒 単 位 で指 定します。

タイプ: REG_DWORD

デフォルト 値 = 1000

下 限 値 = 0

上 限 値 = 5000

DisableSetDebug

SE_DEBUG_PRIVILEGEフラグ設定を無効にします。これを1に設定すると、プロセスリストの包括性が低下する可能性があります。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

ResetIntervalMinutes

DLLがリセットされるまでの実行時間を分単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 240

下限値 = 0

上限値 = 10080

NewEventLimit

作成するイベントの最大数を指定します。この値を小さくすると、CPU使用率は増加しますが、メモリ使用率は低下します。

タイプ: REG_DWORD

デフォルト値 = 2000

下限値 = 0

上限値 = 100000

MaxCleanTries

データベースのクリーニングを試行する回数を指定します。この試行回数でクリーンアップできない場合、データベースはアーカイブされます。

タイプ: REG_DWORD

デフォルト値 = 3

下限値 = 0

上限値 = 10

DatabaseArchiveLimit

保持するデータベースアーカイブの最大数を指定します。この制限を超えている場合、最も古いものを消去します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 100

DisableTrace

Client Recorder Extensionを無効にします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

WMICTimeout

WMICコマンドのタイムアウトを秒単位で指定します。0はコマンドを無効にします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 30

上限値 = 300

DatabaseErrorWaitMinutes

待機時間を指定して、高リソースの使用状況を確認する時間を指定します。モニター.db データベース作成に失敗しました。

タイプ: REG_DWORD

デフォルト値 = 5

下限値 = 0

上限値 = 60

PendingEventCap

負荷時のCPU/メモリ使用量を制御します。現在処理中のイベントの数がこの値を超えると、イベントの数がこのしきい値よりも少なくなるまで、入ってくるイベントはすべてドロップされます。

タイプ: REG_DWORD

デフォルト値 = 5000

下限値 = 0

上限値 = -1

DropRegCreateKey

RegCreateKeyのログを無効にします。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

WatchlistFile

Tanium Integrity Monitorウォッチリストファイルへのパスを指定します。

タイプ: REG_SZ

デフォルト値 = Tanium Clientパス \watchlist.json

ProcessStateFile

プロセス状態ファイルへのパスを指定します。

タイプ: REG_SZ

デフォルト値 = Tanium Clientパス\Trace\process_state.bin

ImageHashStateFile

画像状態ファイルへのパスを指定します。

タイプ: REG_SZ

デフォルト値 = Tanium Clientパス\Trace\image_state.bin

ProcessExpirationSeconds

終了後メモリにプロセス情報を保持する秒数を指定します。

タイプ: REG_DWORD

デフォルト値 = 180

下限値 = 0

上限値 = -1

SignalsJSONFile

シグナルJSON定義へのパス。

タイプ: REG_SZ

デフォルト値 = Tanium Clientパス\Trace\signals.json

SignalsBinFile

エンコードされたシグナルデータへのパスを指定します。

タイプ: REG_SZ

デフォルト値 = Tanium Clientパス \Trace\signals.bin

DisableSignals

Taniumシグナル処理を無効にします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

DisableSignalsTrie

contains/begins with/ends withのシグナル処理を無効にします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

MaxFileHandlesPerProcess

プロセスに対して保持するファイルハンドルの最大数を指定します。

タイプ: REG_DWORD

デフォルト値 = 500

下限値 = 0

上限値 = -1

SysmonLowerToleranceMS

Sysmonイベントをプロセスイベントにマッピングするときに使用される下限許容値をミリ秒単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 200

下限値 = 0

上限値 = -1

SysmonUpperToleranceMS

Sysmonイベントをプロセスイベントにマッピングするときに使用される上限許容値をミリ秒単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 200

下限値 = 0

上限値 = -1

MappingLowerToleranceMS

イベントをプロセスイベントにマッピングするときに使用される許容値をミリ秒単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 200

下限値 = 0

上限値 = -1

MappingUpperToleranceMS

イベントをプロセスイベントにマッピングするときに使用される上限許容値をミリ秒単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 200

下限値 = 0

上限値 = -1

DNSEventIDList

監視対象のDNSクライアントイベントのイベントIDのコンマ区切りリスト。セット表記{3008,3013,3018,3020}である必要があります。

タイプ: REG_DWORD

デフォルト値 =「3008,3013,3018,3020」

ForceSysmon

代替手段がある場合でも、Sysmonの使用を強制します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

ForceRecorderDriver

Client Recorder Extensionドライバの使用を強制します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

UseAuditCommandLine

コマンドラインにセキュリティ監査ログを使用するよう指定します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

VerboseSignals

シグナルの詳細なデバッグを有効にします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

CommandLineMappingRetry

コマンドラインが試行する保存バッチの数を指定します。

タイプ: REG_DWORD

デフォルト値 = 10

下限値 = 0

上限値 = 10

SysmonCheckIntervalMinutes

Sysmonステータスがチェックされる間隔を分単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 5

下限値 = 0

上限値 = 300

ETWCustomBuffers

カスタムETWバッファサイズを有効にします。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

ETWBufferSize

各 イベントトレースセッション バッファに割り当 てられたメモリの量 をキロバイト で指 定しま す。最 大 バッファサイズは1MB です。

タイプ: REG_DWORD

デフォルト 値 = 1024

下 限 値 = 0

上 限 値 = 1024

ETWMinBuffer

イベントトレースセッションのバッファプールに割り当 てられるバッファの最 小 数 を指 定しま す。

タイプ: REG_DWORD

デフォルト 値 = settings["ETWMaxBuffer"]

下 限 値 = プロセッサ数 * 2

上 限 値 = 128

ETWMaxBuffer

イベントトレースセッションのバッファプールに割り当 てられるバッファの最 大 数 を指 定しま す。

タイプ: REG_DWORD

デフォルト 値 = minBufferCount + 20

下 限 値 = プロセッサ数 * 2

上 限 値 = 128

RecordSignalDefinitions

コンパイルされたシグナル定 義 の`monitor.db`データベースでの保 存 を有 効 化 します。

タイプ: REG_DWORD

デフォルト値 = 1

下限値 = 0

上限値 = 1

RecordSignalMatches

`monitor.db`データベースに格納される一致したシグナルの数を指定します。

タイプ: REG_DWORD

デフォルト値 = 5

下限値 = 0

上限値 = -1

DisableImageLoadHashing

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

FileExtraFlagsFilter

`ExtraFlags`フィールドにあるこのマスクに一致するファイル書き込みイベントをフィルタリングします。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = -1

ConfirmFileWrites

ファイル書き込みでのファイル修正時間の変更を確認します。

タイプ: REG_DWORD

デフォルト値 = 0

下限値 = 0

上限値 = 1

SQLiteChunkSizeMB

`monitor.db`データベースのサイズが増加したときにファイルシステムから割り当てられるチャンクのサイズをMB単位で指定します。

タイプ: REG_DWORD

デフォルト値 = 100

下限値 = 0

上限値 = -1

EventQueueExpirationSeconds

保留中のイベントをパージするまでの、プロセスイベントを待つ秒数を指定します。

タイプ: REG_DWORD

デフォルト値 = 120

下限値 = 1

上限値 = -1

モジュール全体の構成の管理

Client Recorder Extensionは<Tanium Client Dir>/Tools/Traceディレクトリにある<product name>_recorder.jsonおよび<product_name>_filters.jsonを読み込みました。Tanium Traceは現在、これらの構成ファイルを<product name>_なしで使用しています。これらの構成ファイルはClient Recorder Extensionが読み取り、それぞれの製品目標を達成するためにその情報をまとめます。

フィールド	戦略	デフォルト値
maxSizeMB	最大値	1024
logRotations	最大値	3
logMaxSize	最小値	10 MB
rawLogging (Linuxのみ)	いずれかがfalseである場合、false	構成ファイルに設定されていない場合は設定されません。
cpuThreshold	最小値	0.25
cpuKillEnabled (LinuxおよびMacのみ)	いずれかがfalseである場合、キルスイッチが無効化されている	True

Client Recorder Extensionのトラブルシューティング

auditdが見つからないLinuxエンドポイントを特定する

Linuxエンドポイントイベントが記録されていない場合、監査デーモンとaudispdが欠落している可能性があります。Traceモジュールをインストールする前に監査デーモンをインストールして設定するのが理想的ですが、後でエンドポイントをオンラインにすることも可能です。

1. (任意) auditdパッケージを作成します。
   一般的なインストールパッケージを作成してロジックをスクリプトに入れるか、単純なスクリプトを作成してロジックをTaniumクエリに入れることができます。Tanium Core Platform ユーザーガイド：パッケージの作成と管理を参照してください。

   > **ヒント：** 将来このパッケージを定期的にチェックして展開する保存されたアクションを作成します。

2. 次のQuestionを実行をします：`Get Installed Application Exists from all machines with Is Linux containing "true" [audit]`
3. 適切なauditdパッケージを識別されたエンドポイントに配備するには、お好みの方法を使用してください。

   > **重要：** パッケージを多数のエンドポイントに配布する必要がある場合は、ネットワークへの悪影響を避けるために、変更に要する時間を長い期間に延長してください。

エンドポイントデータベースを再作成する

データベースが破損していると識別された場合は、エンドポイントデータベースの内容を消去する必要があります。

1. 影響を受けたエンドポイントを絞り込むためにQuestionを使用します。
   たとえば、`Get Trace Invalid File Operations from all machines`と尋ねます。
2. `True`を返すエンドポイントまでドリルダウンします。
3. **[Trace - Recreate Database (Trace - データベースの再作成)][OS]**または**[Threat Response - Recreate Database(Threat Response - データベースの再作成)][OS]**パッケージをアクションとしてデプロイします。

詳細については、Tanium Core Platformユーザーガイド：パッケージの管理と作成またはTanium Interactユーザーガイド：デプロイアクションの使用を参照してください。