フォレンジック

フォレンジック機能セットは、リアルタイムのフォレンジックイベントをキャプチャし、ユーザー定義のルールを適用することにより、起こり得るセキュリティインシデントの監視とアラートを可能にします。具体的には、次のことが可能になります。

- 関心のあるフォレンジックイベントを指定するルールの定義
- 一致するフォレンジックイベントに対するトリガーアクションの定義
- 特定のフォレンジック イベントの検索
- イベント生成プロセスとその完全な系統の可視化

警告 **フォレンジック**機能が有効になっている場合、センサーはセンサー構成に応じて追加のホストリソースを消費する場合があります。「ソフトウェアエージェントの設定」セクションを参照してください。

互換性

フォレンジックシグナルは、AIXを除くすべてのプラットフォームの優れた可視性エージェントによって報告されます。詳細については、下記の「フォレンジックシグナル」のセクションを参照してください。

フォレンジック情報は、Linux カーネル API、監査と syslog、Windows カーネル API、Windows イベントなどを通じて提供されます。一般に、OS ベンダーはメジャーリリース内での互換性を保証します。ただし、OSベンダーが機能と修正をバックポートする可能性があるため、API はプラットフォームやマイナーリリース間でわずかに異なる可能性があります。その結果、一部のプラットフォームでは、一部のフォレンジック イベント タイプを使用できない場合があります。また、エージェントは起動時に無効になっている OS サービスの回復や有効化を試みません。

たとえば、Linux 監査フレームワークを使用するフォレンジックシグナルは数多くあります。フォレンジックが有効になっている場合、優れた可視性エージェントは、エージェントの起動後に Secure Workload の監査ルールをシステムに挿入します。ルールを挿入するには、システムにaugenrulesユーティリティがインストールされている必要があります。また、/etc/audit/rules.d ディレクトリが必要です。これらの前提条件のいずれかが満たされていない場合、Secure Workloadの監査ルールは挿入されません。その結果、ファイルアクセスとrawソケット作成を含むフォレンジックシグナルは報告されません。

ユーザーが以前にフォレンジックを有効にしてから無効にした場合、センサーは Cisco Secure Workload によって挿入された監査ルールを削除します。Redhat 7.3 および CentOS 7.3 では、ルールの削除プロセスに影響を与える可能性のあるオペレーティングシステムのバグが確認されました。センサーが監査ルールを削除する過程は次のとおりです。 1. センサーは、*/etc/audit/rules.d/* の taau.rules を削除します。2. センサーは *$service auditd restart* を実行します。OS は、*/etc/audit/rules.d/* 内の *audit.rules* および **.rules* ファイルに基づいてルールセットを再生成します。次に、auditd はルールをシステムにロードします。

オペレーティング システムは、新しいルール セットを挿入する前にすべてのルールをクリアするために、/etc/audit/rules.d/audit.rules ファイルの先頭に -D を追加します。ただし、Redhat 7.3 および CentOS 7.3 マシンでは、*/etc/audit/rules.d/audit.rules* に *-D* がない場合があります。*/etc/audit/rules.d/audit.rules* ファイルが存在せず、*/usr/share/doc/audit-<version>/* のサブディレクトリにデフォルトのルールファイルも存在しない場合、OS が空のファイルを作成するため、このようなことが起こります。*/usr/share/doc/audit- 2.8.4/rules/10-base-config.rules* は、考えられるデフォルトのルールの格納場所の 1 つです。正確な OS の動作は、*$rpm -qf -scripts /etc/audit/rules.d* を実行して RPM 更新スクリプトから確認できます。

Linuxでは、一部のフォレンジックシグナルは64ビットシステムコールの監視に依存しています。32 ビット Linux システムコールは、現在のリリースではサポートされていません。

フォレンジックシグナル

ソフトウェアセンサーがフォレンジックイベントをキャプチャしてレポートするには、フォレンジック機能を有効にする必要があります。この機能は、[ソフトウェアエージェント構成（Software Agent Config）] で有効にできます。詳細については、「ソフトウェアエージェント構成（Software Agent Config）」を参照してください。

フォレンジック機能が有効になっている場合、センサーから次のフォレンジックイベントが報告されます。

信号	説明
特権昇格	Sudo で実行されるコマンドなどの特権昇格
ユーザーログオン	ユーザーログオンイベント
ユーザーログオン失敗	失敗したユーザーログオンの試行
シェルコード	シェルコードの試みに似た不審なシェル実行
File Access	パスワードファイルなどの非常に機密性の高いファイルへのアクセス
ユーザ アカウント	ユーザーアカウントの追加または削除
未確認コマンド	センサーが認識していない新しいコマンド。ユーザーはコマンド異常スコアを使用して、範囲に基づいて結果を調整できます。詳細については、「未確認コマンド」を参照してください。
未確認ライブラリ	以前にプロセスがロードされたことをセンサーが認識していない新しいライブラリ
raw ソケットの作成	raw ソケットを作成するプロセス（ポートノッキングなど）
バイナリ変更	既知のバイナリのハッシュ値または変更時刻の変更
ライブラリ変更	既知のライブラリのハッシュ値または変更時刻の変更
サイドチャネル	サイドチャネル攻撃の試み（Meltdown）
ユーザーログオンの追跡	ユーザーログオンイベント後に分岐または実行される子孫プロセス

信号	説明
移行手順	プロセスに基づくユーザーフォレンジック構成ルールに一致する移行手順イベントのレポートプロセス バイナリパス、コマンド文字列などの属性。
ネットワーク異常	ワークロードのネットワークトラフィックの異常。詳細については、「PCR ベースのネットワーク異常検出」を参照してください。

特権昇格

プロセスがその権限を低から高に変更すると、それは特権昇格と見なされます。Linux において、この変更は、プロセスのユーザーIDがゼロ以外からゼロに変更されたことを意味します。通常のユーザーのパスワードや、Sudo といった特殊目的バイナリの変更など、正当な場合があります。このイベントは現在、Windows では利用できません。Windows での特権昇格は、通常、プロセス自体の権限（整合性レベル）の変更ではなく、他のメカニズムを介して行われます。Windows での権限昇格は、未確認のコマンドや以下のバイナリ変更など、別タイプのフォレンジックイベントの対象となります。

ユーザーログオン

SSH、RDP、およびその他のタイプのログオンを含むユーザーログオンイベント。センサーは誰が、いつ、どのようにユーザーログインしたかを可能な場合は常にキャプチャします。たとえば、LinuxのSSHの場合、センサーはユーザー名、認証タイプ（パスワード、パブリック）、および送信元 IP を報告します。

ユーザーログオン失敗

前述のユーザーログオンイベントの場合と同様に、センサーは、同様の情報がある場合は常にログイン試行の失敗を報告します。

シェルコード

シェルコードイベントの解釈は、Linux と Windows で異なります。Linux では、センサーはログインセッションまたは端末なしで対話型シェルとして実行されているプロセスを識別します。（ログインセッションの外で対話型シェルを実行する特別な理由はありません。）このリリースでは、システムですでに利用可能なシェルを利用して攻撃されることを想定しているため、シェルコードイベントの検出は制限されています。攻撃によって新しいバイナリがアップロードされた場合、センサーはこれらのバイナリに、未知のコマンドまたはバイナリの変更（既存のバイナリを置き換える場合）のフラグを付けます。Windows では、PowerShell DLL に

リンクされているすべてのプロセスがシェルコードとしてラベル付けされます。ユーザーは、正当なケースを除外するルールを作成できます。

File Access

ファイルアクセスイベントは、パスワードファイルなどの非常に機密性の高いファイルへのアクセスを報告します。このリリースでは、ユーザーは監視対象ファイルのリストを変更できません。Linux では、センサーは /etc/passwd への書き込みアクセスを監視します。センサーは、/etc/shadow への読み取りおよび書き込みアクセスも監視します。このリリースでは、Windows はこのイベントをトリガーしません。

ユーザ アカウント

ユーザーアカウントイベントは、情報が利用可能な場合はいつでも、ローカルユーザーアカウントの作成をレポートします。

未確認コマンド

未確認コマンドイベントは、センサーがこれまで認識していなかったコマンドを報告します。未確認コマンドは、親プロセスから子プロセスへの目に見えない移行/エッジとして定義されます。たとえば、Web サーバー（httpd）が abc.sh という CGI スクリプトを実行していると仮定すると、センサーはそれを初めて検出したときに、abc.sh を未確認コマンドとして報告します。Web サーバーによるそれ以降の abc.sh の実行では、センサーが以前にそれを検出して報告しているため、フォレンジックイベントは発生しません。サービスまたはプロセスがバイナリを実行しない場合、そのサービス/プロセスからの未確認コマンドイベントは、侵害である可能性を示します。センサーは再起動後もステートレスであるため、センサーの再起動後、以前確認されたコマンドが再度報告されることに注意してください。

SaaS クラスタ 3.4 以降の場合、それぞれの未確認コマンドイベントは、0.0 から 1.0 の範囲のコマンド異常スコアに関連付けられています。スコアが低いほど、移行の異常性が高くなります。コマンド移行、つまりタプル（親コマンドライン、コマンドライン）は、以下の同じタプルを持つイベント間の異常な移行についてクロスチェックされます。

- センサーが属する最も狭い範囲。たとえば、次の範囲系統 `Root Scope -> A -> B -> C and Root Scope -> D -> E` に属するワークロード W において、未確認コマンドイベントが観察されます。その後、コマンドは範囲 C と E のすべてのワークロード間でクロスチェックされます（C と E はオーバーラップしている場合もしていない場合もあることに注意してください）。イベントの異常スコアは、これら 2 つの範囲に関するイベントの異常スコアの最大値です。
- 実行中のプロセスの実行パス。
- 親プロセスの実行パス。
- 実行中のプロセスのバイナリハッシュ。

スコア 1.0 は、同じタプル（最も狭い範囲、実行パス、親実行パス、バイナリハッシュ）を持つ、同じコマンド移行が表示されたことを意味します。スコア 0.0 は、実行パス、親実行パス、および実行中のプロセスのバイナリハッシュを使用したコマンド移行が、同じ範囲内のどのホストでも観察されていないことを意味します。異常スコアを使用して、同様の未確認コマンドアラートが同じ範囲内で発生するのを抑制し、誤検知を減らすことができます。このスコアの使用例については、「デフォルトの Secure Workload ルール」を参照してください。

異常スコアは、3.4 の SaaS クラスタでのみ使用できることに注意してください。

未確認ライブラリ

未確認ライブラリイベントは、センサーが以前にロードされたプロセスを認識していないライブラリを報告します。未確認ライブラリは、バイナリ実行パスとライブラリパスの未確認ペアとして定義されます。たとえば、アプリケーションは通常、比較的安定したライブラリのリストをロードします。マシンにアクセスできる攻撃者は、アプリケーションを再起動し、悪意のあるライブラリで LD_PRELOAD を使用する可能性があります。センサーは、このアプリケーションのバイナリ実行パスに新しくロードされた悪意のあるライブラリを初めて検出すると、未確認ライブラリイベントを報告します。センサーが以前に検出して報告しているため、悪意のあるライブラリの後続ロードによってフォレンジックイベントが発生することはありません。正当なケースには、アップグレード後にアプリケーションが新しいライブラリをロードする場合や、アプリケーションが新しいライブラリを動的にロードする場合があります。再起動後、センサーが以前に確認されたライブラリを再度報告する場合があることに注意してください。

これは実験的な機能であり、将来のリリースで変更される可能性があります。

raw ソケットの作成

raw ソケット作成イベントは、このリリースの Linux でのみサポートされています。raw ソケットは通常、トラフィックのスヌーピング、インジェクション、スプーフィングに使用されます。raw ソケットの正当な用途には、 診断ツール（tcpdump）や、 特別な IP パケット（ping、arp）を生成する場合などがあります。ターゲットマシンや被害を受けたマシンによるロギングを回避するステルススキャン、マルウェアポートノッキングなど、悪意を持って使われる場合もあります。また、Secure Workload センサーもフロー関連情報を収集するための raw ソケットを作成します（一貫性を確保するため、センサーは自身のフロー情報収集によってトリガーされたイベントを抑制しません）。

バイナリ変更

バイナリ変更イベントは、実行中のプロセスのファイルコンテンツとバイナリの属性に対する変更を報告します。センサーは、実行中のすべてのプロセスのファイル属性を記録します。プロセスが同じパスでバイナリを実行しているが、ファイル属性（ctime、mtime、サイズ、またはハッシュ）が異なる場合、センサーはプロセスにバイナリ変更のフラグを付けます。正当なケースには、アプリケーションのアップグレードが含まれます。

ライブラリ変更

ライブラリ変更イベントは、実行中のプロセスのファイルコンテンツとライブラリの属性に対する変更を報告します。センサーには、ロードされたライブラリのファイル属性が記録されます。プロセスにより同じパスでライブラリがロードされるが、ファイル属性（ctime、mtime、サイズ、またはハッシュ）が異なる場合、センサーによりプロセスにライブラリ変更のフラグが付けられます。正当なケースには、ライブラリのアップグレードが含まれます。

これは実験的な機能であり、将来のリリースで変更される可能性があります。

サイドチャネル

サイドチャネルイベントは、サイドチャネルの脆弱性をエクスプロイトする実行中のソフトウェアを報告します。このリリースは、選択された Linux プラットフォームでの 1 つのサイドチャネル検出機能を提供します。Meltdown。サポートされているマシン構成については、以下の詳細を参照してください。これらは高度なセキュリティ機能であるため、デフォルトで無効になっています。サイドチャネルレポートが有効になっている場合、CPU 使用率の増加が予想されます。UI で設定された CPU クォータは引き続き適用されます。センサーのフォレンジック コレクション サブプロセスにより、CPU 使用率が高すぎると判断された場合、センサーがシャットダウンし、親センサープロセスが少し遅れて再起動します。古いカーネルやサポートされていないカーネルでこの機能を有効にすると、システムが不安定になる可能性があります。同様の非本番環境でテストすることを強く推奨します。

この機能は、UI の [エージェント構成（Agent Config）] ページからオン/オフの切り替えができ、各エージェント構成プロファイルでオン/オフを切り替えられます。

Meltdown は、CPU の投機的実行とキャッシュ機能を悪用するサイドチャネル攻撃です（https://meltdownattack.com/）。攻撃者は非特権ドメインから特権ドメインデータを読み取ることができます。たとえば、リング 0 特権なしでユーザー空間アプリケーションからカーネルメモリを読み取ることができます。Meltdown 検出は現在、CentOS 7 および Ubuntu 16.04 をサポートしています。

ユーザーログオンの追跡

ユーザーログオンイベントの追跡では、ユーザー ログオン イベント プロセス（SSH、RDP など）の後に実行される子孫プロセス (最大 4 レベル) が報告されます。このユーザーログオンイベントの追跡で報告されるプロセスは監査目的であり、セキュリティイベントは必要ありません。

移行手順

プロセス後イベントは、バイナリパス、コマンド文字列などのプロセス属性に基づいて、ユーザーのフォレンジック構成ルールに一致するプロセスをレポートします。このプロセス後イベントの下でレポートされるプロセスは、監査目的のためのものであり、必ずしもセキュリティイベントを持つとは限りません。

例 1：cmd.exe または powershell.exe によって実行されたプロセスを報告する

Event Type = Follow Process AND (Process Info - Exec Path contains cmd.exe OR Process Info - Exec Path contains powershell.exe)

例 2：winword.exe、excel.exe、または powerpnt.exe によって作成されたプロセスを報告する

Event Type = Follow Process with_ancestor (Process Info - Exec Path contains winword.exe OR Process Info - Exec Path contains excel.exe OR Process Info - Exec Path contains powerpnt.exe)

注：プロセス後イベントは、次のプロセスシグナルのいずれかによって追跡できます。

- Process Info - Exec Path
- Process Info - Command String
- Process Info - Username
- プロセス後 - 親 Exec パス
- プロセス後 - 親コマンド文字列
- プロセス後 - 親ユーザー名

フォレンジック設定

フォレンジック設定では、インテントベースの設定が使用されます。インテントは、フォレンジックプロファイルをインベントリフィルタに適用する方法を指定します。フォレンジックプロファイルは、複数のフォレンジックルールで構成されます。インテント内のプロファイルは、上から順に適用されることに注意してください。

フォレンジックルール

（注） ルート範囲あたりのルールの最大数は 100 です。

フォレンジックルールの追加

このセクションでは、新しいフォレンジックルールを追加する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

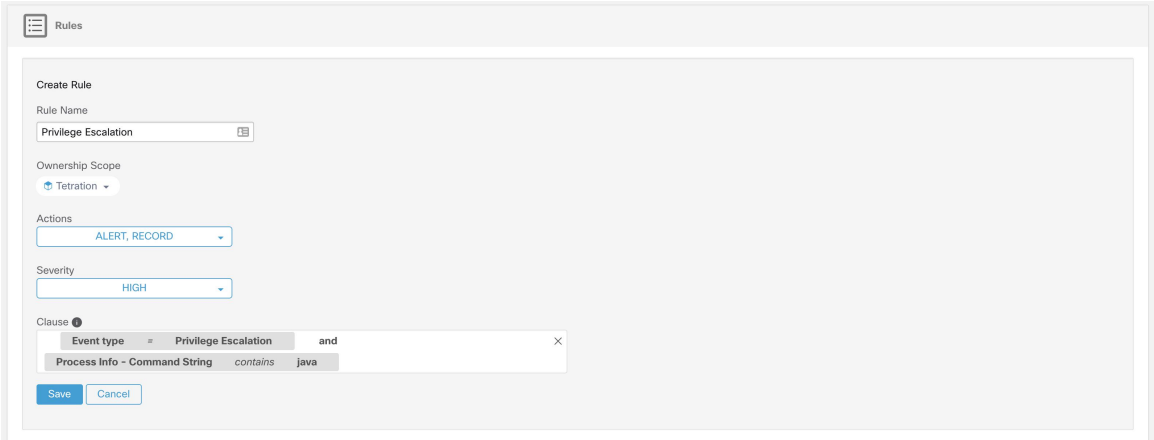
ステップ 1 左側のナビゲーションバーで、**[防御（Defend）] > [フォレンジックルール（Forensic Rules）]**をクリックします。

ステップ 2 [ルールの作成（Create Rule）] をクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
ルール名（Rule Name）	ルールの名前を入力します。名前は空白にできません。
[所有範囲（Ownership scope）]	このルールの所有範囲を入力します。
アクション（Actions）	このルールがトリガーされたときのアクションを選択します。[記録（Record）] は、一致するセキュリティイベントをさらに分析するために保持することを意味します。[アラート（Alert）] アクションは、一致するセキュリティイベントを Secure Workload アラートシステムに公開することを意味します。
重大度	このルールのシビラティ（重大度）レベルを選択します（[低（LOW）]、[中（MEDIUM）]、[高（HIGH）]、[重大（CRITICAL）]、[ただちに対応が必要（REQUIRES IMMEDIATE ACTION）]）。
[句（Clause）]	ルール句を入力します。句には、プロセス フォレンジック イベントまたはワークロードイベントのいずれかからの、セキュリティ イベント シグナルが含まれている必要があります。プロセスシグナルとワークロードシグナルの両方が含まれている句は無効です。

図 *1* : ルールの作成



ステップ 4 [保存（Save）] をクリックします。

基本的なフォレンジックルールの構成

フォレンジックルールには、フォレンジックイベントタイプを**1つだけ**指定する必要があります（例：**Event Type == Unseen Command**）。次のオプション句では、そのイベントの属性を使用する必要があります（例：**Unseen Command - Parent Uptime**）。

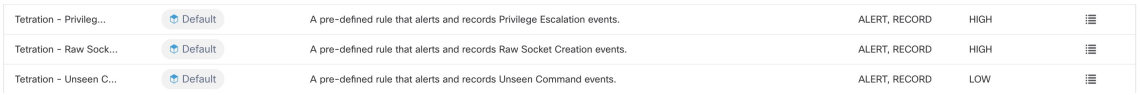
以下は、**Unseen Command** イベントタイプを使用した1つの例です。その他の例については、下記のデフォルトルールと MITER ルールをご覧ください。

EventType = Unseen Command and Unseen Command - Parent Uptime (microseconds) >= 60000000.

デフォルトの Secure Workload ルール

ユーザーが自分の環境に合ったルールを構築する際の手助けとなるように、デフォルトの Secure Workload ルールが用意されています。これらのルールは、フォレンジック設定ページに表示され、編集できません。ルールはすべてのルート範囲で使用できます。

図 *2:* デフォルトルール



このリリースには、4 つの Secure Workload フォレンジックルールが組み込まれています。

1. **ルール名** Secure Workload - 特権昇格

   Clause EventType = Privlege Escalation and (ProcessInfo - ExecPath *doesn't contain* **sudo and ProcessInfo - ExecPath** *doesn't contain* **ping and Privilege Escalation Is≒ Type - Suid Binary**)

   説明：このルールは、setuid バイナリによって生成されない特権昇格イベントを報告します。setuid バイナリを確実に除外するためには、「ProcessInfo - ExecPath」に基づいて **sudo** と **ping** も除外します。Secure Workload ユーザーは、独自のルールを定義することで、他の setuid バイナリを除外することもできます

2. **ルール名** Tetration - 未確認のコマンド

   Clause EventType = Unseen Command and Unseen Command - Parent Uptime (microseconds) >= 60000000 or ProcessInfo - ExecPath *contains* **/bash or ProcessInfo - ExecPath** *contains* **/sh or ProcessInfo - ExecPath** *contains* **/ksh or Parent - ExecPath** *contains* **httpd or Parent - ExecPath** *contains* **apache or Parent - ExecPath** *contains* **nginx or Parent - ExecPath** *contains* **haproxy**

   説明：このルールは、次の条件のいずれかに一致する未確認のコマンドイベントを報告します。

   1. プロセスの親が **60,000,000** マイクロ秒を超えて稼働している
   2. プロセス ExecPath には、何らかのタイプのシェル（**/bash**、**/sh**、**/ksh** など）が含まれている。
   3. プロセスの親 ExecPath には、何らかのタイプのサーバーアプリケーション（**httpd**、**apache**、**nginx**、**haproxy** など）が含まれている。

3. **ルール名** Tetration - raw ソケット

   Clause EventType = Raw Socket Creation and (Raw Socket - ExecPath *doesn't contain* **ping and Raw Socket - ExecPath** *doesn't contain* **iptables and Raw Socket - ExecPath** *doesn't contain* **xtables-multi**)

   説明：このルールは、**ping** や **iptables** によって生成されない raw ソケット作成イベントを報告します。Secure Workload ユーザーは、独自のルールを定義して、他のバイナリを除外することもできます。

4. **ルール名** Tetration - 未確認コマンドによるネットワーク異常

   Clause EventType = Network Anomaly and Network Anomaly - Unseen Command Count > 3 and Network Anomaly - Non-seasonal Deviation > 0

   説明：このルールは、次の条件に一致するネットワーク異常イベントを報告します。

   1. 15 分以内に同じワークロードで 3 つ以上の未確認コマンドイベントがある。
   2. ルールの属性が 0 より大きい（6.0 はすべてのネットワーク異常イベントについて報告される最小偏差であるため、6.0 以上であることも意味します）。

5. **ルール名** Tetration - 異常な未確認コマンド

   Clause EventType = Unseen Command and Unseen Command - Anomaly - Score < 0.6

   説明：このルールは、異常スコアが 0.6 未満の未確認のコマンドイベントを報告します。これは、コマンドが以前に観察されたコマンドと類似していないように見える非常に異常なイベントのみが報告されることを意味します。しきい値 0.6 は、類似コマンドが異なるしきい値でどのように動作するかに関する Cisco Secure Workload での実験に基づいて決定されます。スコアの詳細な説明については、「未確認コマンド」を参照してください。

6. **ルール名** Tetration - SMSS の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **smss.exe and (Follow Process - ParentExecPath** *doesn't contain* **smss.exe and Follow Process - ParentExecPath** *doesn't contain* **System**)

   説明：これは Windows 独自のルールです。このルールは、smss.exe の別のインスタンスまたはシステムプロセスとは異なる親が smss.exe にある場合に警告します。

7. **ルール名** Tetration - wininit の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **wininit.exe and Follow Process - ParentExecPath** *doesn't contain* **smss.exe**

   説明：これは Windows 独自のルールです。このルールは、wininit.exe に smss.exe とは異なる親がある場合に警告します。

8. **ルール名** Tetration - RuntimeBroker の異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **RuntimeBroker.exe and Follow Process - ParentExecPath** *doesn't contain* **svchost.exe**

説明：これは Windows 独自のルールです。このルールは、RuntimeBroker.exe に svchost.exe とは異なる親がある場合に警告します。

9. **ルール名** Tetration - サービスの異常な親

   Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **services.exe and Follow Process - ParentExecPath** *doesn't contain* **wininit.exe**

   説明：これは Windows 独自のルールです。このルールは、services.exe に wininit.exe とは異なる親がある場合に警告します。

10. **ルール名** Tetration - lsaio の異常な親

    Clause EventType = Follow Process and ProcessInfo - ExecPath *contains* **lsaio.exe and Follow Process - ParentExecPath** *doesn't contain* **wininit.exe**

    説明：これは Windows 独自のルールです。このルールは、lsaio.exe に wininit.exe とは異なる親がある場合に警告します。

11. **ルール名** Tetration - lsass の異常な子

    Clause (EventType = Follow Process and ProcessInfo - ExecPath *doesn't contain* **efsui.exe and ProcessInfo - ExecPath** *doesn't contain* **werfault.exe) with ancestor Process Info - ExecPath** *contains* **lsass.exe**

    説明：これは Windows 独自のルールです。このルールは、lsass.exe に efsui.exe または werfault.exe 以外の子孫がある場合に警告します。

デフォルトの MITRE ATT&CK ルール

デフォルトの MITRE ATT&CK ルールは、MITRE ATT&CK フレームワーク（https://attack.mitre.org/）からのアラート手法に対して指定されています。安全を脅かす行動に関連した 24 のルールがあり、それらのほとんどは特定の MITER 手法にマッピングされています。ルールの完全なリストを以下に示します。

1. **名前** 疑わしい MS Office の動作

   Clause (Event type = Follow Process and (Process Info - Exec Path doesn't contain **Windowssplwow64.exe) and (Process Info - Exec Path** *doesn't contain* **chrome.exe) and (Process Info - Exec Path** *doesn't contain* **msip.executionhost.exe) and (Process Info - Exec Path** *doesn't contain* **msip.executionhost32.exe) and (Process Info - Exec Path** *doesn't contain* **msosync.exe) and (Process Info - Exec Path** *doesn't contain* **ofcccaupdate.exe) with ancestor (Process Info - Exec Path** *contains* **winword.exe or Process Info -Exec Path** *contains* **excel.exe or Process Info -Exec Path contains powerpnt.exe)**

   説明 このルールは、Microsoft Office プロセス（WIN-WORD.exe/EXCEL.exe/POWERPNT.exe）が子プロセスを作成した場合にアラートを出し、記録します。シスコの調査に基づき、誤検知の量を減らすため、これらの MS Office バイナリによって作成されることが知られているいくつかの一般的な子プロセスを許可しています。

2. **名前** T1015 - アクセシビリティ機能 1

Clause Event type = Follow Process (Process Info - Exec Path *contains* **cmd.exe or Process Info -Exec Path** *contains* **powershell.exe or Process Info - Exec Path** *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and (Follow Process - Parent Exec Path** *contains* **winlogon.exe or Follow Process - Parent Exec Path** *contains* **atbroker.exe or Follow Process - Parent Exec Path** *contains* **utilman.exe**)

説明 このルールは、アクセシビリティ機能のバイナリ（スクリーンキーボード、拡大鏡、固定キーなど）のいずれかが悪用され、cmd/powershell/cscript/wscript を開くように誘導された場合にアラートを出し、記録します。アクセシビリティバイナリの呼び出しは、どこから呼び出されるか（ログオン画面から、またはユーザーのログイン後）に応じて、winlogon、atbroker、または utilman のいずれかのプロセスによって制御されます。このルールは、アクセシビリティプロセス（winlogon.exe、utilman.exe、および atbroker.exe）の疑わしい子プロセス（cmd.exe、powershell.exe、cscript.exe、wscript.exe）をキャプチャします。これを [T1015 -アクセシビリティ機能2（T1015 - Accessibility features 2）] で使用して、4 つの疑わしい子プロセスの追加の子プロセスも捕捉します。**

3. **名前** T1015 - アクセシビリティ機能 2

   Clause Event type = Follow Process with ancestor ((Process Info - Exec Path *contains* **cmd.exe or Process Info - Exec Path** *contains* **powershell.exe or Process Info - Exec Path** *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and (Follow Process - Parent Exec Path** *contains* **winlogon.exe or Follow Process - Parent Exec Path** *contains* **atbroker.exe or Follow Process - Parent Exec Path** *contains* **utilman.exe))**

   説明 このルールは、アクセシビリティ機能のバイナリ（スクリーンキーボード、拡大鏡、固定キーなど）のいずれかが悪用され、cmd.exe/powershell.exe/cscript.exe/wscript.exe を開くように誘導された場合にアラートを出し、記録します。アクセシビリティバイナリの呼び出しは、どこから呼び出されるか（ログオン画面から、またはユーザーのログイン後）に応じて、winlogon、atbroker、または utilman のいずれかのプロセスによって制御されます。このルールは、これらのプロセス（winlogon、utilman、および atbroker）の疑わしい子プロセスの子プロセスをキャプチャします。アクセシビリティバイナリの疑わしい子プロセスについてアラートを出す [T1015 -アクセシビリティ機能1（T1015 - Accessibility features 1）] でこれを使用する必要があります。

4. **名前** T1085 - rundll32

   Clause (Event type = Follow Process and Process Info Exec Path *doesnt contain* **msiexec.exe and Process Info Exec Path** *doesnt contain* **WindowsSystem32SystemPropertiesRemote.exe with ancestor (Process Info - Exec Path** *contains* **rundll32.exe and Follow Process - Parent Exec Path** *doesnt contain* **msiexec.exe and not (Process Info -command string** *contains* **Windowssystem32shell32.dll or (Process Info -command string** *contains* **Windowssyswow64shell32.dll or (Process Info -command string** *contains* **WindowsSystem32migrationWinInetPlugin.dll))**

   説明 このルールは、rundll32.exe が子プロセスを作成した場合にアラートを出し、記録します。このバイナリは、任意のバイナリ/dll を実行するために呼び出されたり、悪意のあるコントロールパネル項目をインストールするために control.exe によって使用されたりする可能性があります。ただし、msiexec.exe が rundll32.exe の親または子である場合は許可しています。また、よく知られた dll を利用するいくつかの一般的な rundll32 コマンドも許可しています。

5. **名前** T1118 - InstallUtil

   Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **installutil.exe**

   説明 このルールは、InstallUtil.exe が子プロセスを作成した場合にアラートを出し、記録します。

6. **名前** T1121 - Regsvcs/Regasm

   Clause Event type = Follow Process and (Process Info - Exec path *doesn't contain* **fondue.exe or Process Info - Exec path** *doesnt contain* **regasm.exe or Process Info - Exec path** *doesnt contain* **regsvr32.exe with ancestor (Process Info - Exec Path** *contains* **regasm.exe or Process Info - Exec Path** *contains* **regsvcs.exe)**

   説明 このルールは、regsvcs.exe または regasm.exe が子プロセスを作成した場合にアラートを出し、記録します。ただし、誤検知の数を減らすため、fondue.exe/regasm.exe/regsvr32.exe が regasm.exe または regsvcs.exe によって生成される場合は許可しています。

7. **名前** T1127 - 信頼できる開発者ユーティリティ - msbuild.exe

   Clause (Event type = Unseen Command with ancestor Process Info - Exec Path *contains* **MSBuild.exe) and (Process Info - Exec Path** *doesn't contain* **Tracker.exe) and (Process Info -Exec Path** *doesn't contain* **csc.exe) and (Process Info - Exec Path** *doesn't contain* **Microsoft Visual Studio) and (Process Info - Exec Path** *doesn't contain* **al.exe) and (Process Info - Exec Path** *doesn't contain* **lc.exe) and (Process Info - Exec Path** *doesn't contain* **dotnet.exe) and (Process Info - Exec Path** *doesn't contain* **cvtres.exe) and (Process Info - Exec Path** *doesn't contain* **conhost.exe) and not (Event type = Unseen Command with ancestor (Process Info - Exec Path** *contains* **Tracker.exe or Process Info - Exec Path** *contains* csc.exe or Process Info - Exec Path *contains* **Microsoft Visual Studio or Process Info - Exec Path** *contains* **al.exe or Process Info - Exec Path** *contains* **lc.exe or Process Info - Exec Path** *contains* **dotnet.exe or Process Info - Exec Path** *contains* **cvtres.exe))**

   説明 このルールは、msbuild.exe が、通常作成する子プロセスの許可リストに属していない子プロセスを作成した場合にアラートを出し、記録します。Follow Process はプロセスサブツリーの許可をまだサポートしていないため、このルールは現在、Follow Process ではなく Unseen Command に基づいています。現在のルールでは、次のプロセスとその子が許可されます。tracker.exe、csc.exe、「Microsoft Visual Studio」パスからのプロセス、al.exe、lc.exe、dotnet.exe、および cvtres.exe。このルールでは conhost.exe も許可されます。これらのプロセスは、MSBuild.exe の通常の使用中に表示されます（たとえば、Visual Studio を使用したプロジェクトのコンパイルなど）。MSBuild.exe の上記以外の子プロセスすべて（通常の動作ではない）に関してアラートが生成されます。

8. **名前** T1127 - 信頼できる開発者ユーティリティ - rcsi.exe

   Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **rcsi.exe**

   説明 このルールは、rcsi.exe が子プロセスを作成した場合にアラートを出し、記録します。

9. **名前** T1127 - 信頼できる開発者ユーティリティ - tracker.exe

Clause (Event type = Unseen Command with_ancestor Process Info - Exec Path *contains* **tracker.exe) and not (Event type = Unseen Command with_ancestor Process Info - Exec Path** *contains* **MSBuild.exe)**

説明 このルールは、tracker.exeが子プロセスを作成した状況で、tracker自体がMSBuild.exeの子ではない場合にアラートを出し、記録します。したがって、Visual Studioを使用したtrackerの正当な呼び出しは承認されますが、他の呼び出しにはアラートが出されます。tracker.exeルールおよび以前のMSBuild.exeルールの限界の1つに、攻撃者がMSBuild手法を使用してtrackerを作成し、次いでtrackerに悪意のある子を作成させた場合、MSBuildを先祖とするtrackerは正当であるため、どちらのルールによってもアラートが出されない点があります。

10. **名前** T1128 - Netsh ヘルパー Dll

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **netsh.exe**

    説明 このルールは、netsh.exeが子プロセスを作成した場合にアラートを出し、記録します。

11. **名前** T1136 - アカウントの作成

    Clause Event type = User Account

    説明 このルールは、新しいユーザーが作成された場合にアラートを出し、記録します。

12. **名前** T1138 - アプリケーションシミング

    Clause Event type = Follow Process Process Info - Exec Path *contains* **sdbinst.exe**

    説明 このルールは、sdbinst.exeが呼び出された場合にアラートを出し、記録します。

13. **名前** T1180 - スクリーンセーバー

    Clause Event type = Follow Process AND with ancestor Process Info - Exec Path *contains* **.scr**

    説明 このルールは、execパスで「.scr」を使用してプロセスが作成された場合にアラートを出し、記録します。

14. **名前** T1191 - CMSTP

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **cmstp.exe**

    説明 このルールは、cmstp.exeが子プロセスを作成した場合にアラートを出し、記録します。

15. **名前** T1202 - 間接コマンド実行 - forfiles.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **forfiles.exe**

    説明 このルールは、forfiles.exeが子プロセスを作成した場合にアラートを出し、記録します。

16. **名前** T1202 - 間接コマンド実行 - pcalua.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **pcalua.exe**

    説明 このルールは、pcalua.exeが子プロセスを作成した場合にアラートを出し、記録します。

17. **名前** T1216 - 署名付きスクリプトプロキシ実行 - pubprn.vbs

    Clause Event type = Follow Process with ancestor ((Process Info - Exec Path *contains* **cscript.exe or Process Info - Exec Path** *contains* **wscript.exe) and Process Info - Command String** *contains* **.vbs and Process Info - Command String** *contains* **script)**

    説明 このルールは、パラメータ「script」を含む wscript.exe または cscript.exe を使用して vbs スクリプトが実行され、新しいプロセスが作成された場合にアラートを出し、記録します。攻撃者はこの手法を使用して、コードを実行する悪意のある sct ファイルを指すスクリプトパラメータを含む pubprn.vbs を実行する可能性があります。

18. **名前** T1218 - 署名付きバイナリプロキシ実行 - msiexec.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **msiexec.exe**

    説明 このルールは、msiexec.exe が子プロセスを作成した場合にアラートを出し、記録します。

19. **名前** T1218 - 署名付きバイナリプロキシ実行 - odbcconf.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **odbcconf.exe**

    説明 このルールは、odbcconf.exe が子プロセスを作成した場合にアラートを出し、記録します。

20. **名前** T1218 - 署名付きバイナリプロキシ実行 - Register-CimProvider

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **Register-CimProvider.exe**

    説明 このルールは、Register-CimProvider.exe が子プロセスを作成した場合にアラートを出し、記録します。

21. **名前** T1220 - XSL スクリプト処理 - msxsl.exe

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **msxsl.exe**

    説明 このルールは、msxsl.exe が子プロセスを作成した場合にアラートを出し、記録します。

22. **名前** T1220 - XSL スクリプト処理 - wmic

    Clause Event type = Follow Process and (Process Info - Exec Path *contains* **wmic.exe and Process Info - Command String** *contains* **.xsl**)

    説明 このルールは、xsl スクリプトが wmic によって使用されている場合にアラートを出し、記録します。これは任意のバイナリを起動するために使用できます。

23. **名前** T1223 - コンパイル済み HTML ファイル

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **hh.exe**

    説明 このルールは、hh.exe が子プロセスを作成した場合にアラートを出し、記録します。

24. **名前** T1003 - ログイン情報ダンピング - Lsass

Clause Event type = Follow Process and Process Info - Exec Path *contains* **procdump.exe and Process Info - Command String** *contains* **lsass**

説明 このルールは、lsass プロセスのメモリをダンプするために procdump.exe が使用されている場合にアラートを出し、記録します。

25. **名前** T1140 - ファイルまたは情報の難読化解除/復号化

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **certutil.exe and (Process Info - Command String** *matches* **.*encode\s.* or Process Info - Command String** *matches* **.*decode\s.***

    説明 このルールは、ファイルのエンコードまたはデコードに certutil.exe が使用されている場合にアラートを出し、記録します。この手法は、攻撃者が攻撃対象のマシンでエンコードされたペイロードをデコードするためによく使用されます。

26. **名前** T1076 - リモート デスクトップ プロトコル

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **tscon.exe**

    説明 このルールは、tscon.exe が実行された場合にアラートを出し、記録します。攻撃者は、tscon.exe を使用して既存の RDP セッションをハイジャックできます。

27. **名前** T1197 - BITS ジョブ - Powershell

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **powershell.exe and Process Info - Command String** *contains* **Start-BitsTransfer**

    説明 このルールは、powershell.exe を使用して cmdlet Start-BitsTransfer を実行し、ファイルをコピー/移動した場合にアラートを出し、記録します。

28. **名前** T1170 - MSHTA

    Clause Event type = Follow Process with ancestor Process Info - Exec Path *contains* **mshta.exe**

    説明 このルールは、子プロセスを生成する悪意のある HTA スクリプトを実行するために mshta.exe が使用されている場合にアラートを出し、記録します。

29. **名前** T1158 - 非表示ファイルおよびディレクトリ

    Clause Event type = Follow Process and (Process Info - Exec Path *contains* **attrib.exe and Process Info - Command String** *contains* **+h)**

    説明 このルールは、attrib.exeを使用してファイル/ディレクトリを非表示に設定した場合にアラートを出し、記録します。

30. **名前** T1114 - 電子メール収集

    Clause Event type = Follow Process (Process Info - Command String *matches* **.*.(ost|pst)(\s|"|').* or Process Info - Command String** *matches* **.*.(ost|pst)$) Process Info - Exec Path** *doesn't contain* **outlook.exe**

    説明 このルールは、outlook.exe 以外のプロセスから電子メールファイル（.ost および .pst）にアクセスした場合にアラートを出し、記録します。

31. **名前** T1070 - ホストでのインジケータ削除 - イベントログ

Clause Event type = Follow Process and Process Info - Exec Path *contains* **wevtutil.exe and Process Info - Command String** *matches* **.*\s(cl|clear-log)\s.***

説明 このルールは、wevtutil.exeを使用してイベントログが消去される場合にアラートを出し、記録します。

32. **名前** T1070 - ホストでのインジケータ削除 - USN

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **fsutil.exe and Process Info - Command String** *matches* **.*\susn\s.* and Process Info - Command String** *matches* **.*\sdeletejournal.***

    説明 このルールは、fsutil.exe を使用して USN ジャーナルが削除される場合にアラートを出し、記録します。

33. **名前** T1053 - スケジュールされたタスク

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **schtasks.exe and Process Info - Command String** *contains* **create**

    説明 このルールは、schtasks.exeを使用して新しいスケジュールされたタスクが作成される場合にアラートを出し、記録します。

34. **名前** T1003 - ログイン情報ダンピング - Vaultcmd

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **vaultcmd.exe and Process Info - Command String** *matches* **.*\/list.***

    説明 このルールは、vaultcmd.exe を使用して Windows Credentials Vault にアクセスされた場合にアラートを出し、記録します。

35. **名前** T1003 - ログイン情報ダンピング - レジストリ

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **reg.exe and ((Process Info - Command String** *contains* **save or Process Info - Command String** *contains* **export) and (Process Info - Command String** *contains* **hklm or Process Info - Command String** *contains* **hkey_local_machine) and (Process Info - Command String** *contains* **sam or Process Info - Command String** *contains* **security or Process Info - Command String** *contains* **system))**

    説明 このルールは、reg.exeを使用して特定のレジストリハイブをダンプされた場合にアラートを出し、記録します。

36. **名前** T1201 - パスワードポリシー検出 1

    Clause Event type = Follow Process and Process Info - Exec Path *contains* **chage and Process Info - Command String** *contains* **-l**

    説明 このルールは、chageユーティリティを使用してLinuxマシンのパスワードポリシー（パスワード有効期間ポリシー）が一覧表示された場合にアラートを出し、記録します。

37. **名前** T1081 - ファイル内のログイン情報 - Linux

    Clause Event type = Follow Process and (Process Info - Exec Path *contains* **cat or Process Info - Exec Path** *contains* **grep) and (Process Info - Command String** *contains* **.bash_history or Process Info - Command String** *contains* **.password or Process Info - Command String** *contains* **.passwd)**

説明 このルールは、Linux マシン上のファイルに保存されているパスワードを検索する試みが実行された場合にアラートを出し、記録します。

38. **名前** T1081 - ファイル内のログイン情報 - Windows

Clause Event type = Follow Process and Process Info - Exec Path *contains* **findstr.exe and Process Info - Command String** *contains* **password**

説明 このルールは、Windows マシン上のファイルに保存されているパスワードを検索する試みが実行された場合にアラートを出し、記録します。

39. **名前** T1089 - セキュリティツールの無効化

Clause Event type = Follow Process and ((Process Info - Exec Path *contains* **fltmc.exe and Process Info - Command String** *contains* **unload sysmon) or (Process Info - Exec Path** *contains* **sysmon.exe and Process Info - Command String** *contains* **/u))**

説明 このルールは、fltmc.exe または sysmon.exe を使用して sysmon ドライバをアンロードする試みが実行された場合にアラートを出し、記録します。

フォレンジックプロファイル

プロファイルの追加

このセクションでは、新しいフォレンジックプロファイルを追加する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

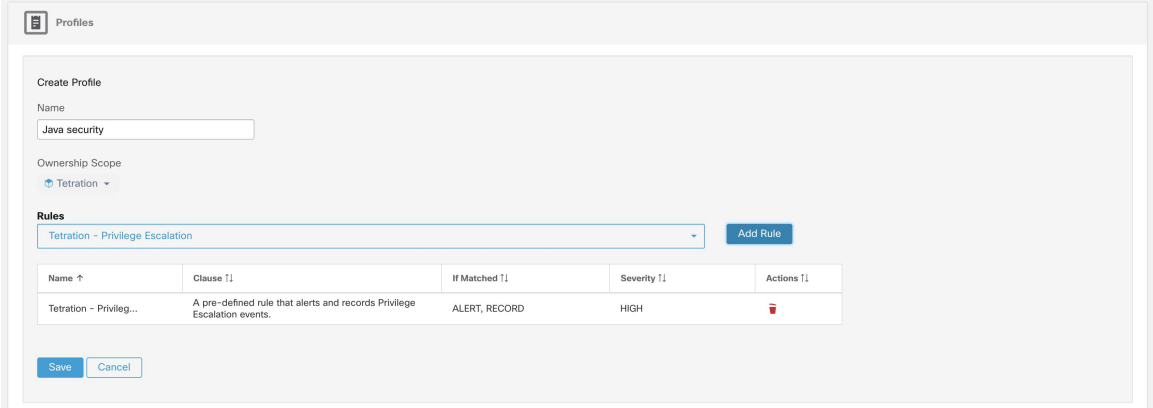
ステップ 1 左側のナビゲーションバーで、**[防御（Defend）]** > **[フォレンジックルール（Forensic Rules）]**をクリックします。

ステップ 2 [プロファイルの作成（Create Profile）] をクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
Name	プロファイル名を入力します名前は空白にできません。
[所有範囲（Ownership scope）]	このプロファイルの所有範囲を入力します。
ルール（Rule）	このプロファイルにルールを追加します。

図 *3:* プロファイルの作成



ステップ 4 [保存（Save）] をクリックします。

プロファイルの編集

このセクションでは、ユーザーがフォレンジックプロファイルを編集する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

ステップ 1 左側のナビゲーションバーで、**[防御（Defend）]** > **[フォレンジックルール（Forensic Rules）]**をクリックします。

ステップ 2 編集するプロファイルを見つけて、右側の列にある**鉛筆**アイコンをクリックします。

ステップ 3 以下のフィールドに適切な値を入力します。

フィールド	説明
Name	プロファイルの名前を更新します。名前は空白にできません。
[所有範囲（Ownership scope）]	このプロファイルの所有範囲を更新します。
ルール（Rule）	このプロファイルにルールを追加/削除します。

ステップ 4 [保存（Save）] をクリックします。

プロファイルの複製

このセクションでは、ユーザーがフォレンジックプロファイルを複製する方法について説明します。

ステップ 1 左側のナビゲーションバーで、**[防御（Defend）]** > **[フォレンジックルール（Forensic Rules）]**をクリックします。

ステップ 2 複製するプロファイルを見つけて、右側の列にある [複製（clone）] アイコンをクリックします。

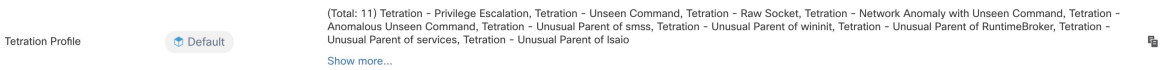
ステップ 3 複製されたプロファイルの名前を入力します。

ステップ 4 [保存（Save）] をクリックします。

デフォルトプロファイル：Secure Workload プロファイル

Secure Workload プロファイルには 11 のデフォルト フォレンジック ルールが含まれており、インテントに追加できます。ユーザーによる編集はできませんが、複製は可能です。複製されたデフォルト フォレンジック プロファイルは編集できます。

図 *4:* デフォルトプロファイル



デフォルトのプロファイル：MITRE ATT&CK プロファイル

MITRE ATT&CK プロファイルには 39 の MITER ATT&CK ルールが含まれており、インテントに追加できます。ユーザーによる編集はできませんが、複製は可能です。複製されたプロファイルは編集できます。MITRE ATT&CK プロファイルには、次のルールが含まれます。

1. 疑わしい MS Office の動作
2. T1015 - アクセシビリティ機能 1
3. T1015 - アクセシビリティ機能 2
4. T1085 - rundll32
5. T1118 - InstallUtil
6. T1121 - Regsvcs/Regasm
7. T1127 - 信頼できる開発者ユーティリティ - msbuild.exe
8. T1127 - 信頼できる開発者ユーティリティ - rcsi.exe
9. T1127 - 信頼できる開発者ユーティリティ - tracker.exe
10. T1128 - Netsh ヘルパー Dll
11. T1136 - アカウントの作成

12. T1138 - アプリケーションシミング
13. T1180 - スクリーンセーバー
14. T1191 - CMSTP
15. T1202 - 間接コマンド実行 - forfiles.exe
16. T1202 - 間接コマンド実行 - pcalua.exe
17. T1216 - 署名付きスクリプトプロキシ実行 - pubprn.vbs
18. T1218 - 署名付きバイナリプロキシ実行 - msiexec.exe
19. T1218 - 署名付きバイナリプロキシ実行 - odbcconf.exe
20. T1218 - 署名付きバイナリプロキシ実行 - Register-CimProvider
21. T1220 - XSL スクリプト処理 - msxsl.exe
22. T1220 - XSL スクリプト処理 - wmic
23. T1223 - コンパイル済み HTML ファイル
24. T1003 - ログイン情報ダンピング - Lsass
25. T1140 - ファイルまたは情報の難読化解除/復号化
26. T1076 - リモート デスクトップ プロトコル
27. T1197 - BITS ジョブ - Powershell
28. T1170 - MSHTA
29. T1158 - 非表示ファイルおよびディレクトリ
30. T1114 - 電子メール収集
31. T1070 - ホストでのインジケータ削除 - イベント ログ
32. T1070 - ホストでのインジケータ削除 - USN
33. T1053 - スケジュールされたタスク
34. T1003 - ログイン情報ダンピング - Vaultcmd
35. T1003 - ログイン情報ダンピング - レジストリ
36. T1201 - パスワードポリシー検出 1
37. T1081 - ファイル内のログイン情報 - Linux
38. T1081 - ファイル内のログイン情報 - Windows
39. T1089 - セキュリティツールの無効化

ログの変更

ルート範囲で SCOPE_OWNER 機能を持つ**サイト管理者**およびユーザーは、以下に示すアイコンをクリックすることで、各フォレンジックルール、プロファイル、およびインテントの変更ログを表示できます。

図 *5*: 変更ログ

これらのユーザーは、対応するテーブルの下にある [削除済みルール/プロファイル/インテントの表示（View Deleted Rules/Profiles/Intents）] リンクをクリックして、削除されたルール、プロファイル、およびインテントのリストを表示することもできます。

変更ログの詳細については、「変更ログ」を参照してください。ルート範囲の所有者は、その範囲に属するエンティティの変更ログ エントリの表示に制限されます。

法医学の可視化

フォレンジックページへのアクセス

このセクションでは、フォレンジックページにアクセスする方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

ステップ 1 左側のパネルで [セキュリティ（Security）] リンクをクリックします。

ステップ 2 [フォレンジック（Forensics）] 項目をクリックします。フォレンジックページが表示されます。

図 *6:* セキュリティフォレンジック

フォレンジックイベントの閲覧

このセクションでは、一致するフォレンジックイベントを参照する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインし、フォレンジックページに移動する必要があります。

ステップ 1 ページ上部の [時間範囲ピッカー（Time Range Picker）] で特定の範囲を選択します。

ステップ 2 [重大度（Severity）] ドロップダウンを選択します。

ステップ 3 [フィルタ（Filters）] で、一致するフォレンジックイベントのフィルタを入力し、[フォレンジックイベントのフィルタ処理（Filter Forensic Events）] をクリックします。

ステップ 4 選択した時間範囲、重大度、およびフィルタに従って、一致するフォレンジックイベントのテーブルが更新されます。

（注） フォレンジックイベントはルート範囲レベルで表示され、サブ/子範囲に切り替えると表示されません。

フォレンジックイベントの検査

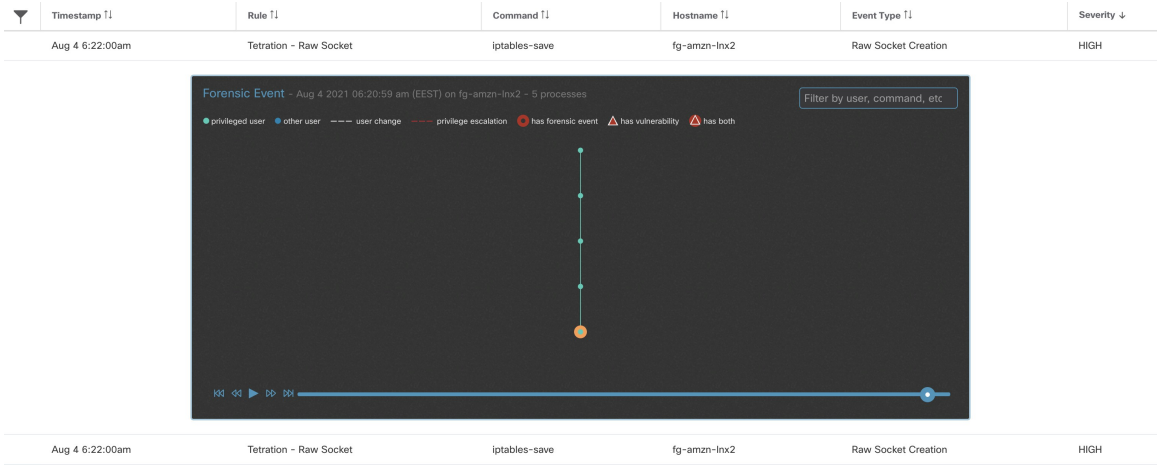
このセクションでは、フォレンジックイベントを検査する方法について説明します。

はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または[範囲所有者（ルート範囲）（Scope Owner（Root Scope））] としてシステムにログインする必要があります。

ステップ 1 検査するイベントをクリックします。[プロセスの詳細（Process detail）] ペインが表示されます。

図 *7:* フォレンジックイベントの表



ステップ 2 系統ツリーで、検査するプロセスをクリックして詳細を表示します。

図 *8:* フォレンジックプロセスの詳細

フォレンジックイベントに表示されるフィールド

各フォレンジックイベントには、有用なデータを提供する多くのフィールドがあります。さまざまなタイプのフォレンジックイベントすべてに共通するフィールドがいくつかあります。また、特定のフォレンジックイベントに固有のフィールドもいくつかあります。

以下は、UI を構成するフィールドのリストです。最初の表では、すべてのフォレンジックイベントに共通するフィールドについて説明します。次に、各アラートとともに表示されるプロセス情報を説明する表、フォレンジックイベントごとに固有のフィールドを記載した表が続きます。データの保存方法やエクスポート方法が原因で、一部のフィールドは複数のテーブルに存在する場合があることに注意してください。

共通のフィールド

フィールド	説明
Bin attr ctime	Linux での変更時刻/Windows での作成時刻（バイナリ形式）
Bin attr hash	バイナリの SHA256 ハッシュ
Bin attr mtime	バイナリの変更時刻
Bin attr name	ファイルシステム上のバイナリの名前
Bin attr size	ファイルシステム上のバイナリのサイズ
Bin exec path	バイナリのフルパス
Cmdline	実行されるプロセスの完全なコマンドライン
Event time usec	このイベントが観測された時間（マイクロ秒単位）

Process Info

フィールド	説明
プロセス ID（Process ID）	プロセスのプロセス ID
[親プロセス ID（Parent Process ID）]	プロセスの親のプロセス ID
ユーザ	プロセスを実行したユーザー
実行パス（Execution path）	プロセスに対応するバイナリのフルパス。
開始時刻（Start time）	プロセスが開始された時刻。
フルコマンド（Full command）	実行されたプロセスの完全なコマンドライン

特権昇格

フィールド	説明
親コマンドライン（Parent cmdline）	プロセスの親の完全なコマンドライン
Parent exe	プロセスの親のフルパス
Parent Uptime (microseconds)	プロセスの親が実行されてからの時間

フィールド	説明
親ユーザー名（Parent Username）	プロセスの親を実行したユーザー
Types bitmap suid binary	バイナリに suid ビットが設定されているかどうかを表示

ユーザーログオン

フィールド	説明
認証タイプパスワード（Auth type password）	パスワード認証を示します
認証タイプの公開キー（Auth type pubkey）	キーベースの認証を示します
ログインタイプ-SSH（Type login ssh）	ユーザーが ssh 経由でログインしたことを示します
ログインタイプ-バッチ（Type login win batch）	Windows バッチログインを示します（タイプ 4、schtasks など）
ログインタイプ-キャッシュ（Type login win cached）	キャッシュされた資格情報によるログオンを示します（タイプ 11、CachedIntetractive）
ログインタイプ-インタラクティブ（Type login win interactive）	インタラクティブログオンを示します（タイプ 2、RDP など）
ログインタイプ-ネットワーククリアテキスト（Type login win network cleartext）	SSH 経由のログオンを示します（タイプ 8）
ログインタイプ-ネットワーク（Type login win network）	ネットワークログインを示します（タイプ3、Psexecなど）
ログインタイプ-新しい資格情報の使用（Type login win new cred）	新しいログイン情報の使用を示します（タイプ 9、Runas コマンドなど）
ログインタイプ-リモートインタラクティブ（Type login win remote interactive）	リモートログオンを示します(タイプ 10、RDP など)
ログインタイプ-サービス（Type login win service）	サービスが SCM によって開始されたことを示します（タイプ 5）
ログインタイプ-ロック解除（Type login win unlock）	ワークステーションがロック解除されたことを示します（タイプ 7）
Src IP	ログインイベントが生成されたソース IP
送信元ポート	ログインイベントが生成されたソースポート

フィールド	説明
ユーザー名	ログインイベントに関連付けられているユーザー名

ユーザーログオン失敗

フィールド	説明
[認証タイプ - パスワード（Auth type password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH（Type login ssh）]	ユーザーが SSH 経由でログインしたことを示します
[ログインタイプ - バッチ（Type login win batch）]	Windows バッチログインを示します（タイプ 4、たとえば schtasks）
[ログインタイプ - キャッシュ（Type login win cached）]	キャッシュされた資格情報によるログオンを示します（タイプ 11、CachedIntetractive）
[ログインタイプ - インタラクティブ（Type login win interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - ネットワーククリアテキスト（Type login win network cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - ネットワーク（Type login win network）]	ネットワークログインを示します（タイプ3、Psexec など）
[ログインタイプ-新しい資格情報の使用（Type login win new cred）]	新しい資格情報の使用を示します（タイプ9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブ（Type login win remote interactive）]	リモートログオンを示します（タイプ10、RDP など）
[ログインタイプ - サービス（Type login win service）]	サービスがSCMによって開始されたことを示します（タイプ 5）
[ログインタイプ - ロック解除（Type login win unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Src IP）]	ログインイベントが生成されたソース IP
[送信元ポート（Src Port）]	ログインイベントが生成されたソースポート

フィールド	説明
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

シェルコード

フィールド	説明
Signal sources bitmap cmd as sh no tty	シェルプロセスに関連付けられたttyがないことを示します。
Signal sources bitmap powershell	プロセスにpowershell dllがロードされていることを示します（System.Management.Automation）。

File Access

フィールド	説明
ファイル（File）	アクセスしたファイルのフルパス
Perm read perm	ファイルに読み取り権限があることを示します
Perm read write perm	ファイルに読み取りおよび書き込み権限があることを示します
Perm write perm	ファイルに書き込み権限があることを示します

ユーザーアカウント

フィールド	説明
[ユーザー名（Username）]	作成されたユーザーのユーザー名
[操作アカウントの追加（Ops acct add）]	新しいアカウントが追加されたことを示します

未確認コマンド

フィールド	説明
異常 - スコア（Anomaly - Score）	コマンドラインが以前に表示されていた頻度を示すスコア（0～1.0）。スコアが低いほど、コマンドがより異常であることを示します。
異常 - 類似性 - 高（Anomaly - Similarity - High）	異常スコアが 0.8 より大きく、1 より小さい場合は true
異常 - 類似性 - 中（Anomaly - Similarity - Medium）	異常スコアが 0.6 より大きく、0.8 以下の場合は true
異常 - 類似性 - 低（Anomaly - Similarity - Low）	異常スコアが 0 より大きく、0.6 以下の場合は true
異常 - 類似 - 確認済み（Anomaly - Similarity - Seen）	異常スコアが 1 の場合、つまり、同じコマンドが以前に確認されていた場合は true
異常 - 類似性 - ユニーク（Anomaly - Similarity - Unique）	異常スコアが 0 の場合、つまり、コマンドが未確認の場合は true
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent exepath）	親プロセスのバイナリパス
親の稼働時間（Parent uptime）	親プロセスが実行されてからの時間
親ユーザー名（Parent username）	親プロセスを実行したユーザーのユーザー名
センサー稼働時間（Sensor uptime）	センサーの稼働時間

未確認ライブラリ

フィールド	説明
ライブラリパス（Lib Path）	以前はプロセスに関連付けられていなかったライブラリファイルの完全パス

raw ソケットの作成

フィールド	説明
Exe Path	raw ソケットを作成したプロセスの完全パス

ライブラリ変更

フィールド	説明
ライブラリの変更された名前	変更されたライブラリのフルパス

サイドチャネル

フィールド	説明
信号ソースビットマップ Meltdown	Meltdown エクスプロイトの使用を示します。

ユーザーログオンの追跡

フィールド	説明
[ユーザー名（Username）]	プロセスを実行したユーザー名

移行手順

フィールド	説明
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent exepath）	親プロセスのバイナリパス
親の稼働時間 usec（Parent uptime usec）	親プロセスが実行されてからの時間
親ユーザー名（Parent username）	親プロセスを実行したユーザーのユーザー名
usec が最後に変更されてからの時間	プロセス開始時刻からバイナリファイル変更時刻までの経過時間（mtime）
ユーザー名	プロセスを実行したユーザーのユーザー名

ネットワーク異常

ネットワーク異常イベントに関連する属性のリストについては、ネットワーク異常イベントのフォレンジックルールを参照してください。

フォレンジック分析：検索可能なフィールド

以下の表は、[フォレンジック分析（Forensics Analysis）]ページの検索バーで検索可能なフィールドの説明を示しています。

その他のフィールド

フィールド	説明
フォレンジックルール名（Forensic Rule Name）	特定のフォレンジックルールによってラベル付けされたイベント
Hostname	特定のホスト名からのイベント
センサー ID（Sensor ID）	特定のセンサーからのイベント
重大度	特定の重大度のイベント

フォレンジック分析の検索用語

共通のフィールド

これらのフィールドは、さまざまなイベントタイプで共通しています。「Event name - Event」というプレフィックスが付いています。たとえば、「Binary Changed - Binary Attribute - CTime (epoch nanoseconds)」などです。

フィールド	説明
Binary Attribute - CTime (epoch nanoseconds)	Linux での変更時刻/Windows での作成時刻（バイナリ形式）
Binary Attribute - Hash	バイナリの SHA256 ハッシュ
Binary Attribute - MTime (epoch nanoseconds)	バイナリの変更時刻
Binary Attribute - Filename	ファイルシステム上のバイナリの名前
Binary Attribute - Size (bytes)	ファイルシステム上のバイナリのサイズ
Event Binary Path	バイナリのフルパス
コマンド ライン	実行されるプロセスの完全なコマンドライン

バイナリ変更

「共通フィールド」の表に記載されている以外に検索用語はありません。

File Access

ファイルアクセスの検索用語には、「File Access -」というプレフィックスが付いています（例：「File Access - Filename」）。

フィールド	説明
ファイル名	アクセスしたファイルのフルパス
Is = Permission - Read	ファイルに読み取り権限があることを示します
Is = Permission - ReadWrite	ファイルに読み取りおよび書き込み権限があることを示します
Is = Permission - Write	ファイルに書き込み権限があることを示します

移行手順

プロセス後検索語には、プレフィックス「Follow Process -」が付いています（例：「Follow Process - Parent Command Line」）。

フィールド	説明
親コマンドライン（Parent Command Line）	親プロセスの完全なコマンドライン
親 Exec パス（Parent Exec Path）	親プロセスのバイナリパス
親の稼働時間（Parent Uptime）（マイクロ秒）	親プロセスが実行されてからの時間
親ユーザー名（Parent Username）	親プロセスを実行したユーザーのユーザー名
最後にファイルが変更されてからのプロセス開始時間（Process Start Time Since Last File Changed）（マイクロ秒）	プロセスの開始から最新の（対応する）ファイル変更までの経過時間
ユーザー名	後続するプロセスに関連付けられたユーザー名

Follow User Logon

Follow User Logon 検索用語には、「Follow User Logon -」というプレフィックスが付いています（例：「Follow User Logon - Username」）。

フィールド	説明
[ユーザー名（Username）]	プロセスに関連付けられたユーザー名

Ldap

Ldap 検索語には、プレフィックス「Ldap -」が付いています（例：「Ldap - Department」）。

フィールド	説明
部署名（Department）	プロセスのユーザー名に関連付けられたAMS Ldap ユーザーの部門（利用可能な場合）
説明	プロセスのユーザー名に関連付けられたAMS Ldap ユーザーの説明（利用可能な場合）
ユーザー名	プロセスに関連付けられたAMS Ldap ユーザー名（利用可能な場合）

ライブラリ変更

Library Changed 検索用語には、「Library Changed -」というプレフィックスが付きます。例：「Library Changed - Department」

フィールド	説明
ライブラリファイル名	変更されたライブラリのフルパス

特権昇格

Privilege Escalation 検索用語には、プレフィックス「Privilege Escalation -」が付いています（例：「Privilege Escalation - Parent Command Line」）。

フィールド	説明
親コマンドライン	プロセスの親の完全なコマンドライン
親 Exec パス	プロセスの親のフルパス
親の稼働時間（マイクロ秒）	プロセスの親が実行されてからの時間
親ユーザー名	プロセスの親を実行したユーザー

フィールド	説明
タイプ：SUID バイナリ	バイナリに SUID ビットが設定されているかどうかを示します。

Process Info

Process Info 検索語には、接頭辞「Process Info - 」が付いています（例：「Process Info - Binary Hash」）。

フィールド	説明
バイナリハッシュ（Binary Hash）	プロセスに関連付けられたバイナリのハッシュ
トークン化されたコマンド文字列（Command String Tokenized）	プロセスのトークン化されたコマンドライン。
コマンド文字列	プロセスの完全なコマンドライン
Exec Path	プロセスに対応するバイナリのフルパス。

Raw Socket

raw ソケット検索語には、プレフィックス「Raw Socket -」が付いています （例：「Raw Socket - Exec Path」）。

フィールド	説明
Exec Path	raw ソケットを作成したプロセスの完全パス

シェルコード

シェルコードの検索用語には、「Shellcode -」のプレフィックスが付いています（例：「Shellcode - Source - Not From Login」）。

フィールド	説明
ソース - ログインからではない（Source - Not From Login）	シェルプロセスに関連付けられた tty がないことを示します。
ソース - Powershell（Source - Powershell）	プロセスに powershell dll がロードされていることを示します（System.Management.Automation）。

サイドチャネル

サイドチャネルの検索語には、プレフィックス「シェルコード -（Shellcode ）」が付いています。たとえば、[シェルコード - ソース - Meltdown（Shellcode - Source - Meltdown）] となります。

フィールド	説明
[ソース - Meltdown（Source - Meltdown）]	Meltdown エクスプロイトの使用を示します。

未確認コマンド

未確認コマンドの検索用語には、「Unseen Command -」というプレフィックスが付きます。
例：Unseen Command - Anomaly - Similarity - High

フィールド	説明
異常 - スコア（Anomaly - Score）	コマンドラインが以前に表示された頻度を示すスコア（0 ～ 1.0）。スコアが低いほど、コマンドがより異常であることを意味します
異常 - 類似性 - 高（Anomaly - Similarity - High）	異常スコアが 0.8 より大きく、1 より小さい場合は true
異常 - 類似性 - 中（Anomaly - Similarity - Medium）	異常スコアが 0.6 より大きく、0.8 以下の場合は true
異常 - 類似性 - 低（Anomaly - Similarity - Low）	異常スコアが 0 より大きく、0.6 以下の場合は true
異常 - 類似 - 確認済み（Anomaly - Similarity - Seen）	異常スコアが 1 の場合、つまり、同じコマンドが以前に見られた場合は true
異常 - 類似性 - ユニーク（Anomaly - Similarity - Unique）	異常スコアが 0 の場合、つまり、コマンドがこれまでに見られなかった場合は true
親コマンドライン（Parent cmdline）	親プロセスの完全なコマンドライン
親 ExePath（Parent Exepath）	親プロセスのバイナリパス
親の稼働時間（Parent uptime）	親プロセスが実行されてからの時間
親ユーザー名（Parent Username）	親プロセスを実行したユーザーのユーザー名
センサー稼働時間（Sensor Uptime）	センサーの稼働時間
異常 - 最新の類似コマンド（Anomaly - Latest Similar Commands）	イベントのコマンドに類似した、以前に観察された最新の 5 つのコマンド

未確認ライブラリ

未確認ライブラリの検索用語には、「Unseen Library -」というプレフィックスが付いています（例：「Unseen Library - Lib Filename」）。

フィールド	説明
ライブラリファイル名（Lib Filename）	以前はプロセスに関連付けられていなかったライブラリファイルの完全パス

ユーザ アカウント

ユーザーアカウントの検索語には、プレフィックス「ユーザーアカウント -」が付いています（例：「ユーザー アカウント - アカウント名」）。

フィールド	説明
アカウント名	作成されたユーザーのユーザー名
操作 - アカウントの追加	新しいアカウントが追加されたことを示します

ユーザーログオン

ユーザーログオンの検索用語には、プレフィックス「ユーザーログオン -（User Logon -）」が付いています。例：「ユーザーログオン - 認証タイプ - パスワード（User Logon - Auth Type - Password）」

フィールド	説明
[認証タイプ - パスワード（Auth Type - Password）]	パスワード認証を示します
[認証タイプ - 公開キー（Auth type - Pubkey）]	キーベースの認証を示します
[ログインタイプ - SSH経由のログイン（Login Type - Login Via SSH）]	ユーザーが SSH 経由でログインしたことを示します
[ログインタイプ - Windowsバッチログイン（Login Type - Windows Login Batch）]	Windows バッチログインを示します（タイプ 4、schtasks など）
[ログインタイプ - Windowsキャッシュログイン（Login Type - Windows Login Cached）]	キャッシュされたクレデンシャルによるログオンを示します (タイプ 11、CachedIntetractive）

フィールド	説明
[ログインタイプ - Windowsインタラクティブログイン（Login Type - Windows Login Interactive）]	インタラクティブログオンを示します（タイプ 2、RDP など）
[ログインタイプ - Windowsネットワーククリアテキスト（Login Type - Windows Network Cleartext）]	SSH 経由のログオンを示します（タイプ 8）
[ログインタイプ - Windowsネットワーク（Login Type - Windows Network）]	ネットワークログインを示します（タイプ3、Psexec など）
[ログインタイプ - 新しいクレデンシャルによるWindowsログイン（Login Type - Windows Login New Credential）]	新しいクレデンシャルの使用を示します（タイプ 9、Runas コマンドなど）
[ログインタイプ - リモートインタラクティブWindowsログイン（Login Type - Windows Login Remote Interactive）]	リモートログオンを示します（タイプ10、RDP など）
[ログインタイプ - Windowsログインサービス（Login Type - Windows Login Service）]	サービスが SCM（タイプ 5）によって開始されたことを示します
[ログインタイプ - Windowsログインのロック解除（Login Type - Windows Login Unlock）]	ワークステーションがロック解除されたことを示します（タイプ 7）
[送信元IP（Source IP）]	ログインイベントが生成された送信元 IP
[送信元ポート（Source Port）]	ログインイベントが生成された送信元ポート
[ユーザー名（Username）]	ログインイベントに関連付けられているユーザー名

ユーザーログオン失敗

User Logon Failed 検索語には、プレフィックス「User Logon Failed -」が付いています。例：「User Logon Failed - Auth Type - Password」

フィールド	説明
認証タイプ - パスワード（Auth Type - Password）	パスワード認証を示します
認証タイプ - 公開キー（Auth type - Pubkey）	キーベースの認証を示します
ログイン タイプ - SSH 経由のログイン（Login Type - Login Via SSH）	ユーザーが ssh 経由でログインしたことを示します

フィールド	説明
ログインタイプ - Windows バッチログイン（Login Type - Windows Login Batch）	Windows バッチログインを示します（タイプ 4、schtasks など）
ログインタイプ - Windows キャッシュログイン（Login Type - Windows Login Cached）	キャッシュされたログイン情報によるログオンを示します（タイプ 11、CachedIntetractive）
ログインタイプ - Windows インタラクティブログイン（Login Type - Windows Login Interactive）	対話型ログオンを示します（タイプ 2、RDP など）
ログインタイプ - Windows ネットワーククリアテキスト（Login Type - Windows Network Cleartext）	ssh 経由のログオンを示します（タイプ 8）
ログインタイプ - Windows ネットワーク（Login Type - Windows Network）	ネットワークログインを示します（タイプ3、Psexecなど）
ログインタイプ - Windows 新しいログイン情報によるログイン（Login Type - Windows Login New Credential）	新しいログイン情報の使用を示します（タイプ 9、Runas コマンドなど）
ログインタイプ - Windows リモート インタラクティブ ログイン（Login Type - Windows Login Remote Interactive）	リモート ログオンを示します（タイプ 10、RDP など）
ログインタイプ - Windows ログインサービス（Login Type - Windows Login Service）	サービスが SCM によって開始されたことを示します（タイプ 5）
ログインタイプ - Windows ログインロック解除（Login Type - Windows Login Unlock）	ワークステーションがロック解除されたことを示します（タイプ 7）
ソース IP（Source IP）	ログインイベントが生成されたソース IP
送信元ポート（Source Port）	ログインイベントが生成されたソースポート
ユーザー名	ログインイベントに関連付けられているユーザー名

フォレンジックアラート

フォレンジックイベントは、一致するルールに**アラート**アクションが含まれている場合、Secure Workload アラートシステムで見つけることができます。

フォレンジックアラートへのアクセス

このセクションでは、フォレンジックアラートにアクセスする方法について説明します。

はじめる前に

- [サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。
- [フォレンジック（Forensics）] アラート送信元のアラートをオンにする必要があります。

ステップ 1 左側のツールバーの [アラート（Alerts）] をクリックします。

ステップ 2 [アラート（Alerts）] ページが表示されます。

アラート詳細の確認

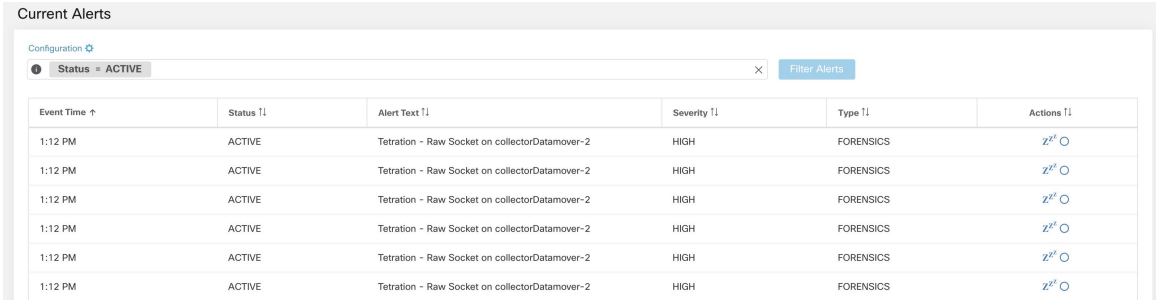
はじめる前に

[サイト管理者（Site Admin）]、[カスタマーサポート（Customer Support）]、または [範囲所有者（Scope Owner）] としてシステムにログインする必要があります。

ステップ 1 アラートページから、確認するアラートをクリックします。

ステップ 2 プロファイル/ルールをクリックして、一致するフォレンジックプロファイル/ルールの詳細を表示します。アラートが発生した後に一致するプロファイル/ルールが更新されると、警告インジケータが表示されることに注意してください。

図 *9*: フォレンジックアラートのページ



さらに、アラートをスヌーズしたり、含めたり除外したりすることができます。詳細については、「現在のアラート」セクションを参照してください。

外部との統合

フォレンジックアラートは、syslog などの外部監視ツールに送信できます。フォレンジックアラートは JSON 形式で送信されます。JSON フィールドの定義は、上記の「フォレンジックイベントに表示されるフィールド」セクションで定義されています。

JSON Kafka 出力の例は次のとおりです。


```
{
"severity": "HIGH",
"tenant_id": 0,
"alert_time": 1595573847156,
"alert_text": "Tetration - Anomalous Unseen Command on collectorDatamover-1",
"key_id":
→"d89f926cddc7577553eb8954e492528433b2d08e:5efcfdf5497d4f474f1707c2:5efcfdf6497d4f474f1707d6:20196:CMD_
→NOT_SEEN",
"alert_id": "/Alerts/5efcfdf5497d4f474f1707c2/DataSource{location_type='TETRATION',
→location_name='forensics', location_grain='MIN', root_scope_id=
→'5efcfdf5497d4f474f1707c2'}/
→db10d21631eebefc3b8d3aeaba5a0b1b45f4259e85b591763d7eaee9161ca076",
"root_scope_id": "5efcfdf5497d4f474f1707c2",
"type": "FORENSICS",
"event_time": 1595573795135,
"alert_details": "{\"Sensor Id\":\"d89f926cddc7577553eb8954e492528433b2d08e\",\
→"Hostname\":\"collectorDatamover-1\",\"Process Id\":20196,\"scope_id\":\
→"5efcfdf5497d4f474f1707c2\",\"forensic\":{\"Unseen Command\":\"true\",\"Unseen
→Command - Sensor Uptime (microseconds)\":\"34441125356\",\"Unseen Command - Parent
→Uptime (microseconds)\":\"35968418683\",\"Unseen Command - Parent Username\":\"root\
→",\"Unseen Command - Parent Command Line\":\"svlogd -tt /local/logs/tetration/efe/ \
→",\"Unseen Command - Parent Exec Path\":\"/sbin/svlogd\",\"Unseen Command - Anomaly
→- Score\":\"0\",\"Unseen Command - Anomaly - Similarity - Unique\":\"true\",\
→"Process Info - Command String\":\"gzip \",\"Process Info - Exec Path\":\"/bin/gzip\
→"},\"profile\":{\"id\":\"5efcfdf6497d4f474f1707e4\",\"name\":\"Tetration Profile\",\
→"created_at\":1593638390,\"updated_at\":1593638390,\"root_app_scope_id\":\
→"5efcfdf5497d4f474f1707c2\"},\"rule\":{\"id\":\"5efcfdf6497d4f474f1707d6\",\"name\
→":\"Tetration - Anomalous Unseen Command\",\"clause_chips\":\"[{\\\"type\\\":\\\
→"filter\\\",\\\"facet\\\":{\\\"field\\\":\\\"event_type\\\",\\\"title\\\":\\\"Event
→type\\\",\\\"type\\\":\\\"STRING\\\"},\\\"operator\\\":{\\\"label\\\":\\\"\\u003d\\\
→",\\\"type\\\":\\\"eq\\\"},\\\"displayValue\\\":\\\"Unseen Command\\\",\\\"value\\\
→":\\\"Unseen Command\\\"},{\\\"type\\\":\\\"filter\\\",\\\"facet\\\":{\\\"field\\\
→":\\\"forensic_event__cmd_not_seen_data__cmdline_anomaly_info__score\\\",\\\
→"title\\\":\\\"Unseen Command - Anomaly - Score\\\",\\\"type\\\":\\\"NUMBER\\\"},\\\
→"operator\\\":{\\\"label\\\":\\\"\\u003c\\\",\\\"type\\\":\\\"lt\\\"},\\\
→"displayValue\\\":\\\"0.6\\\",\\\"value\\\":\\\"0.6\\\"}]\",\"created_at\
→":1593638390,\"updated_at\":1595539498,\"root_app_scope_id\":\
→"5efcfdf5497d4f474f1707c2\"}}"
}
```


alert_details の値自体がエスケープされた JSON 文字列であり、上記のアラートの内容を以下に示します。


```
{
"Sensor Id" : "d89f926cddc7577553eb8954e492528433b2d08e",
"Hostname" : "collectorDatamover-1",
```

```
"Process Id" : 20196,
"scope_id" : "5efcfdf5497d4f474f1707c2",
"forensic" : {
"Unseen Command" : "true",
"Unseen Command - Sensor Uptime (microseconds)" : "34441125356",
"Unseen Command - Parent Uptime (microseconds)" : "35968418683",
"Unseen Command - Parent Username" :"root",
"Unseen Command - Parent Command Line" : "svlogd -tt /local/logs/tetration/efe/ ",
"Unseen Command - Parent Exec Path" : "/sbin/svlogd",
"Unseen Command - Anomaly - Score" : "0",
"Unseen Command - Anomaly - Similarity - Unique" : "true",
"Process Info - Command String" : "gzip ",
"Process Info - Exec Path" : "/bin/gzip"
},
"profile" : {
"id" : "5efcfdf6497d4f474f1707e4", "name" :"Tetration Profile", "created_at" :1593638390,
"updated_at" : 1593638390,
"root_app_scope_id" : "5efcfdf5497d4f474f1707c2"
},
"rule" : {
"id" : "5efcfdf6497d4f474f1707d6",
"name" : "Tetration - Anomalous Unseen Command",
"clause_chips" : "[{\"type\":\"filter\",\"facet\":{\"field\":\"event_type\",\
"title\":\"Event type\",\"type\":\"STRING\"},\"operator\":{\"label\":\"=\",\"type\
":\"eq\"},\"displayValue\":\"Unseen Command\",\"value\":\"Unseen Command\"},{\"type\
":\"filter\",\"facet\":{\"field\":\"forensic_event  cmd_not_seen_data  cmdline_
anomaly_info  score\",\"title\":\"Unseen Command - Anomaly - Score\",\"type\":\
"NUMBER\"},\"operator\":{\"label\":\"<\",\"type\":\"lt\"},\"displayValue\":\"0.6\",\
"value\":\"0.6\"}]",
"created_at" : 1593638390,
"updated_at" : 1595539498,
"root_app_scope_id" : "5efcfdf5497d4f474f1707c2"
}
}
```


フォレンジックイベントの詳細は、フィールドフォレンジックに含まれます。フォレンジックイベントの属性のリストについては、「フォレンジックイベントに表示されるフィールド」を参照してください。フォレンジックイベントの属性は、UI のアラートの詳細にも表示されます。

フォレンジックスコア

フォレンジックスコアが表示される場所

セキュリティダッシュボード：

図 *10* : セキュリティダッシュボードの *[*フォレンジックスコア（***Forensics Score***）*]* セクション

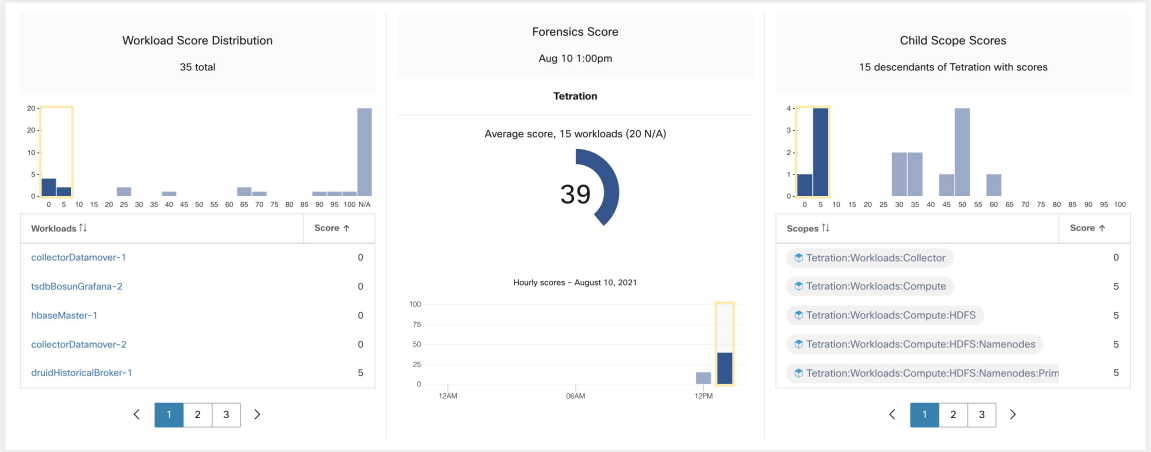
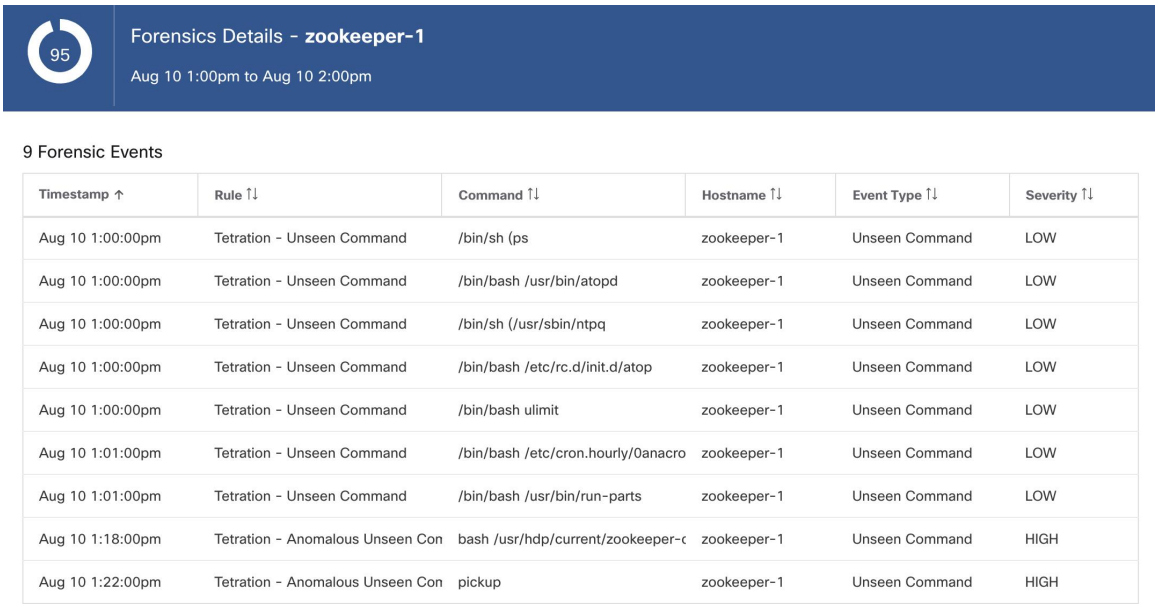


図 *11* : セキュリティダッシュボードの *[*フォレンジックスコアの詳細（***Forensics Score Details***）*]* セクション



フォレンジックスコアの計算方法

フォレンジックスコアはワークロードごとに計算されます。ワークロードのフォレンジックスコアは、該当範囲で有効になっているプロファイルに基づいて、そのワークロードで観察されたフォレンジックイベントから得られます。スコア 100 は、有効なプロファイルの設定されたルールを介してフォレンジックイベントが観察されなかったことを意味し、スコア 0 は、即時対応が必要なフォレンジックイベントが検出されたことを意味します。ある範囲のフォレンジックスコアは、その範囲内の平均ワークロードスコアです。特定の時間のフォレンジックスコアは、その時間内のすべてのスコアの最小値です。

- 重大度が REQUIRES IMMEDIATE ACTION のフォレンジックイベントは、範囲全体のスコアをゼロに減らします。
- 重大度が CRITICAL のフォレンジックイベントは、ワークロードのスコアを 10 の重みで減らします。
- 重大度が HIGH のフォレンジックイベントは、ワークロードのスコアを重み 5 で減らします。
- 重大度が MEDIUM のフォレンジックイベントは、ワークロードのスコアを重み 3 で減らします。
- 重大度が LOW のフォレンジックイベントは、フォレンジックスコアに寄与しません。これは、信号の品質がまだ調整中であり、ノイズが多い可能性がある新しいルールに推奨されます。

たとえば、重大度が CRITICAL の 2 つのルール、重大度が HIGH の 1 つのルール、重大度が LOW の 1 つのルールにそれぞれ一致する 3 つのフォレンジックイベントがワークロードにあるとします。そのワークロードのフォレンジックスコアは、100 - 1 * 10 - 1 * 5 - 1 * 0 = 85 です。

フォレンジックスコアは、フォレンジック機能が有効になっていないワークロードの場合は該当しません。

フォレンジックスコアの改善方法

フォレンジックスコアの調整は、有効なフォレンジックルールを調整して行います。ノイズの少ないルールを作成すると、より正確なスコアが得られます。正当なフォレンジックイベント（侵入またはその他の悪質なアクティビティの証拠となるイベント）に対応して防止するのも、フォレンジックスコアを改善するための良い方法です。

警告

- フォレンジックスコアの詳細には、その時間内のすべてのフォレンジックイベントが表示されます。つまり、フォレンジックスコアの詳細には、フォレンジックスコアの計算に使用されるもの以外のフォレンジックイベントが表示される場合があります。
- フォレンジックスコアは現在、優れた可視性センサーと適用センサーで利用できます。

PCR ベースのネットワーク異常検出

ネットワークの異常機能は、Producer Consumer Ratio（PCR）の概念に基づいて、ワークロードに出入りする異常に大量のデータを検出します。PCR は次のように定義されます。

```
Egress app byte count - Ingress app byte count
  PCR = ------------------------------------------
        Egress app byte count + Ingress app byte count
```

PCR の値は [-1.0, 1.0] の範囲にあります。ここで、

- PCR = 1.0 は、ワークロードが純粋にデータを送信することを意味します
- PCR = -1.0 は、ワークロードが純粋にデータを受信することを意味します
- PCR = 0.0 は、ワークロードでデータの送受信のバランスが取れていることを意味します

他のフォレンジック機能と同様に、インテントベースの設定を使用して、記録したり警告したりするネットワーク異常イベントを設定できます。ワークロードから検出されたネットワーク異常イベントは 5 分ごとにエクスポートされ、設定されたルールと 5 分後に照合されます。その結果、新しいネットワーク異常イベントは、イベントの時刻から最大 10 分の遅延で、5 分ごとに UI でのみ観測されます。

（注） Secure Workload ソフトウェアの 3.2 および 3.1 バージョンでは、ネットワーク異常検出はデータリーク検出と呼ばれていました。

ネットワーク異常イベントのフォレンジックルール

フォレンジックルールを追加する方法については、「フォレンジック設定」を参照してください。

ルールの属性

このセクションでは、ネットワーク異常関連のルールを定義する属性の詳細について説明します。最も単純なネットワーク異常ルールは次のとおりです。

Event Type = Network Anomaly

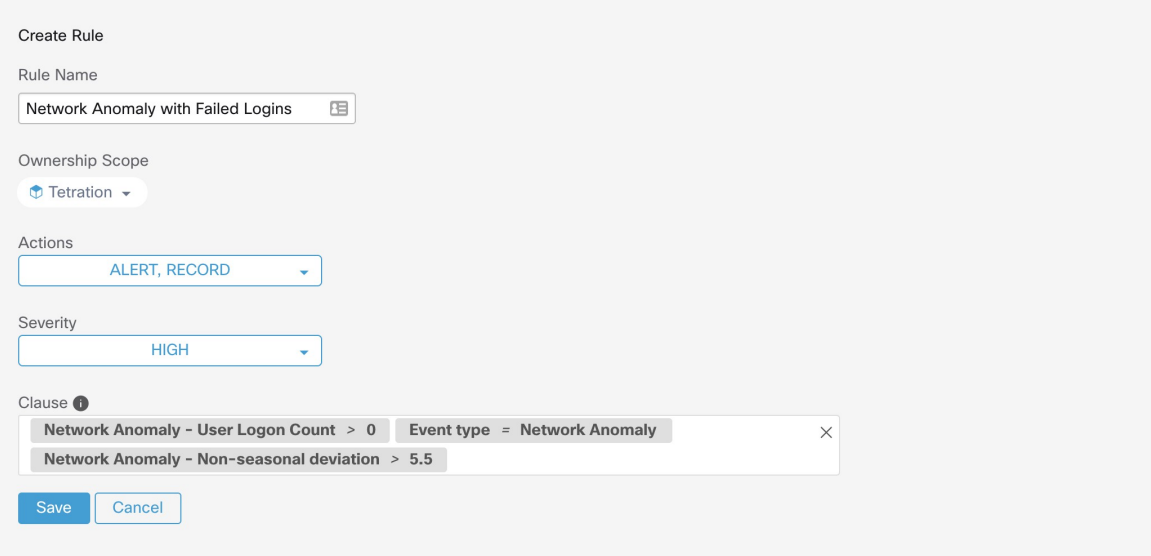
以下は、データセンター用のルールを調整するためのネットワーク異常イベントの他の属性です。

属性	説明
ホスト名	このイベントを発行するワークロードのホスト名。
タイムスタンプ（エポックミリ秒単位）	イベントのタイムスタンプ（ミリ秒単位）。
PCR 偏差	過去の複数の標準偏差としてのイベント時間の平均に基づく PCR の偏差。
非季節的偏差	これは、季節性パターンを（cron ジョブなどによって）削除した後の PCR 偏差です。非季節的偏差の値は常に 6.0 以上です。
PCR	生産者コンシューマ比率（Producer Consumer Ratio）。

属性	説明
EIR	送受信比（Egress Ingress Ratio）は、Egress App Byte Count の合計と Ingress App Byte Count の比率です。
送信側アプリバイト数	ワークロードから流出するパケットコンテンツ（ヘッダーを除く）の合計バイト数である、送信側アプリケーションのバイト数。
受信側アプリバイト数	ワークロードに流入するパケットコンテンツ（ヘッダーを除く）の合計バイト数である、受信側アプリケーションのバイト数。
プロトコル	PCR 時系列が計算されるプロトコル。現在、サポートされているプロトコルはTCP、UDP、および Aggregate です。Aggregate PCR は、TCP、UDP、およびICMPバイト数の合計に基づいて計算されます。
ユーザーログオン数	過去約15分間のワークロードでのユーザーログオンイベントの数。一致するルールがあるかどうかを考慮に入れない、ユーザーログオンイベントの数です。ユーザーログオンイベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。
ユーザーログオン失敗数	過去約15分間のワークロードでのユーザーログオン失敗イベントの数。一致するルールがあるかどうかを考慮に入れない、ユーザーログオン失敗イベントの数です。ユーザーログオン失敗イベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。
未確認コマンド数	過去約15分間のワークロードでの未確認コマンドイベントの数。一致するルールがあるかどうかを考慮に入れない、未確認コマンドイベントの数です。未確認コマンドイベントの詳細を知るには、関心のあるワークロードのイベントを記録し、[フォレンジック分析（Forensics Analysis）] ページでそれらを表示するルールを定義する必要があります。

属性	説明
日時（UTC）- 年	イベント時間の年。
日時（UTC）- 月	イベント時間の月（1、2、...。
日時（UTC）- 日	イベント時間の日（1、2、...。
日時（UTC）- 時間	イベント時間の時間（1、2、...、24）。
日時（UTC）- 分	イベント時間の分（1、2、...、60）。
日時（UTC）- 秒	イベント時間の秒（1、2、...、60）。
日時（UTC）- 曜日	イベント時間の曜日（0～7、月曜日から日曜日に対応）。

図 *12*：ネットワーク異常イベントのフォレンジックルールの定義



いくつかのサンプルルールを以下に示します。

リスト 7.10.1.1.1：UDP のみのネットワーク異常を検出します。

```
Event Type = Network Anomaly AND Network Anomaly Is = Protocol - UDP
```

リスト 7.10.1.1.2：名前に *sensitiveDataServer* が含まれているワークロードのサブセットの送信側アプリにおけるバイト数にしきい値を設定して、季節的なパターンを削除すると、非常に大きな偏差が検出されます（検出された場合）。

```
Event Type = Network Anomaly AND Network Anomaly - Non-seasonal Deviation > 10.0)
AND Network Anomaly - Egress App Byte Count > 1000000
AND Network Anomaly - Host Name CONTAINS sensitiveDataServer
```

リスト7.10.1.1.3：未確認コマンドイベントを伴うワークロードのネットワーク異常イベントが検出されます（ネットワーク異常イベントが毎日7.30AM UTC ～ 7.35AM UTC に発生することを除く）。

```
Event Type = Network Anomaly AND Network Anomaly - Unseen Command Count > 0
AND ( Network Anomaly - Date Time (UTC) - Hour != 7
OR Network Anomaly - Date Time (UTC) - Minute < 30 OR Network Anomaly - Date Time
‹→(UTC) - Minute > 35 )
```

ルール アクション

アクション	説明
RECORD	一致するイベントは、ネットワーク異常スコアに影響を与えます。該当するイベントをセキュリティダッシュボードまたは[ワークロードプロファイル（Workload Profile）] ページ/[ネットワーク異常（Network Anomaly Tab）] タブで見つけることができます。
ALERT	一致するイベントは、[アラート（Alerts）] ページと選択したアラートパブリッシャに表示されます。

次のセクションでは、UI で検出されたネットワーク異常イベントが表示される場所について詳しく説明します。

ネットワーク異常イベントが表示される場所

（注） ネットワーク異常イベントは、現在、[フォレンジック分析（Forensics Analysis）] ページには表示されていません。ネットワーク異常イベントは、次のページで確認できます。

- **セキュリティダッシュボード**：**RECORD** アクションを持つルールに一致するネットワーク異常イベントは、セキュリティダッシュボードの [ネットワーク異常スコア（Network Anomaly Score）] セクションで見つけることができます。スコアが最高ではない（100 未満の）ワークロードがある場合、ワークロード名をクリックすると、そのワークロードのPCR時系列とネットワーク異常イベントを表示できます。ネットワーク異常イベントテーブルの各行の右端には、対応するネットワーク異常イベントの前後におけるフローやその他のフォレンジックイベントを検索するのに役立つアクションリンクが表示されます。ネットワーク異常スコアレポートの既知の遅延については、「ネットワーク異常の遅延」を参照してください。

図 *13*：セキュリティダッシュボードのネットワーク異常スコア

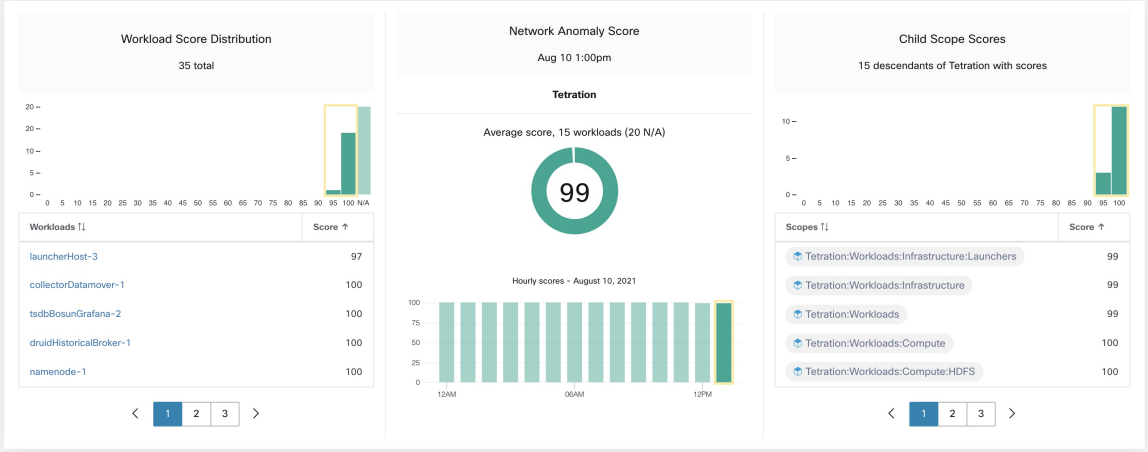
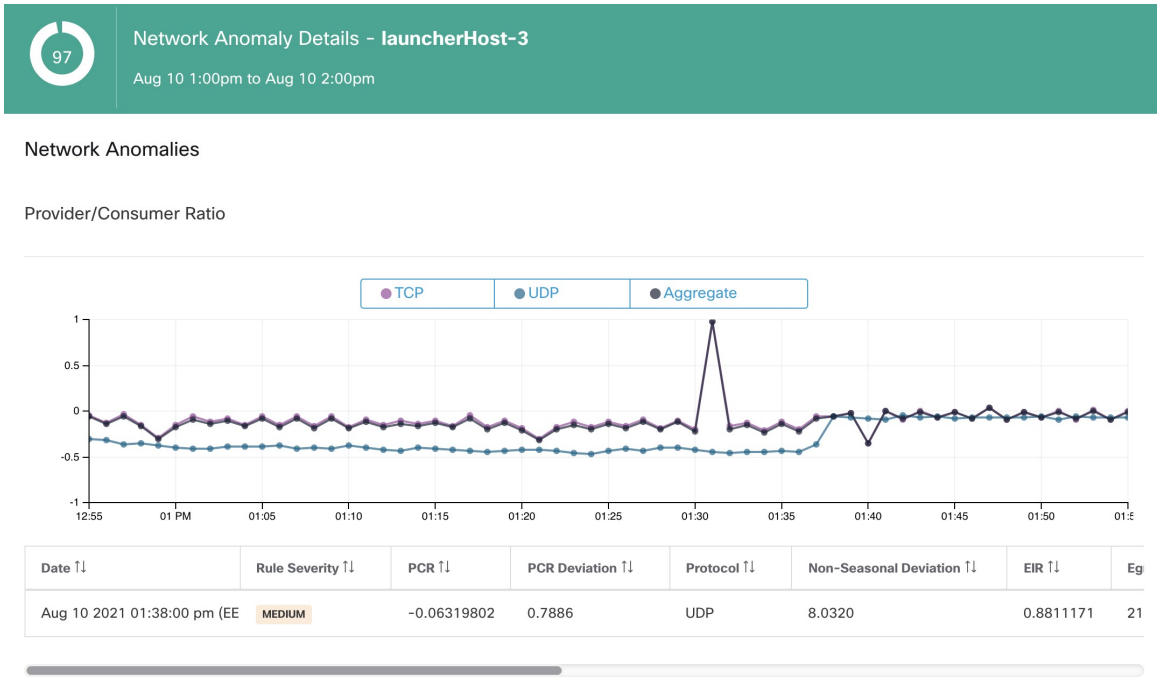


図 *14*：セキュリティダッシュボードのワークロード別にドリルダウンされたネットワーク異常スコア

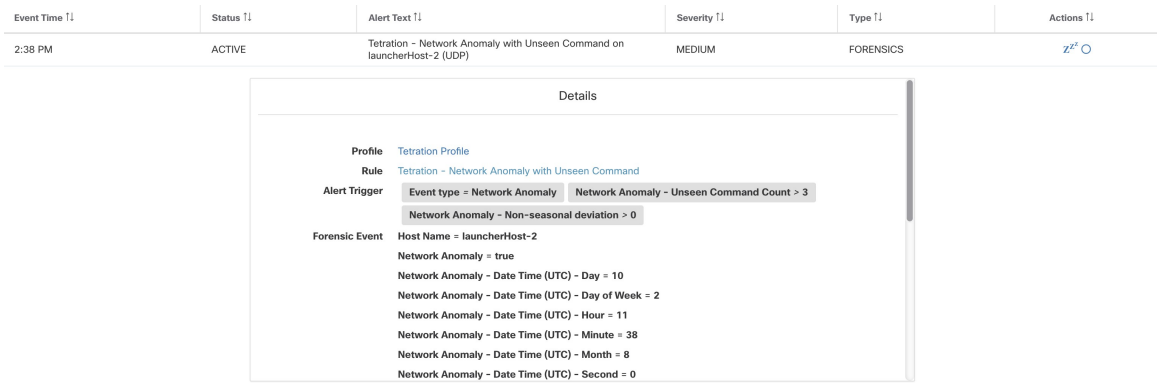


- [ワークロードプロファイル（Workload Profile）] ページ/[ネットワーク異常（Network Anomaly）] タブ：このページでは、PCR 時系列グラフと、**RECORD** アクションを持つルールに一致するネットワーク異常イベントを確認できます。このページに表示される内容は、セキュリティダッシュボードでワークロード名をクリックして表示される内容と非常によく似ています。

図 *15* : [ワークロードプロファイル（*Workload Profile*）] ページの [ネットワーク異常（*Network Anomaly*）] タブ

- **アラート**：ネットワーク異常ルールが **ALERT** アクションを指定して設定されている場合、一致したイベントは [アラート（Alerts）] ページに表示され、アラートパブリッシャからも利用できます。現在のアラート

図 *16* : ネットワーク異常アラート



ルールの重大度とネットワーク異常スコア

ネットワーク異常スコアは、フォレンジックスコアと同様に計算されます。ネットワーク異常スコアはワークロードごとに計算されます。ワークロードのネットワーク異常スコアは、該当範囲で有効になっているプロファイルに基づいて、そのワークロードで観察されたネットワーク異常イベントから得られます。スコアが 100 の場合、有効なプロファイルで設定されたルール経由でネットワーク異常イベントが観察されなかったことを意味します。スコアが 0 の場合、即時のアクションが必要なネットワーク異常イベントが検出されたことを意味します。

- 重大度が REQUIRES IMMEDIATE ACTION のネットワーク異常イベントは、範囲全体のスコアを 0 に減らします。
- 重大度が CRITICAL のネットワーク異常イベントは、ワークロードの影響スコアを 10 減らします。

- 重大度が HIGH のネットワーク異常イベントは、ワークロードの影響スコアを 5 減らします。
- 重大度が MEDIUM のネットワーク異常イベントは、ワークロードの影響スコアを 3 減らします。
- 重大度が LOW のネットワーク異常イベントは、ネットワーク異常スコアに影響を与えません。これは、信号の品質がまだ調整中であり、ノイズが多い可能性がある新しいルールに推奨されます。

ワークロードごとに、合計影響スコアが 5 分ごとに集計され、その 5 分以内の該当するワークロードのスコアが計算されます。

ネットワーク異常機能が有効になっているセンサータイプがないワークロードの場合、ネットワーク異常のスコアは N/A となります。

PCR データとネットワーク異常イベントの保持

PCR データとネットワーク異常イベントは 7 日間保持されます。

ネットワーク異常の遅延

セキュリティダッシュボードで報告されるネットワーク異常スコアには、5 分の遅延があります。たとえば、午前 10 時から午前 10 時 59 分までのワークロードのスコアは、午前 9 時 55 分から午前 10 時 54 分に発生するネットワーク異常イベントに基づいています。

警告

- 古いデータリークイベントは、ネットワーク異常イベントではなくデータリークイベントとして残ります。
- プロトコルごとのネットワーク異常検出は 3.3 の新機能であり、古いデータリークイベントではプロトコルは設定されません。

プロセスハッシュの異常検出

名前が示すように、この機能は、システム全体のプロセスバイナリハッシュの一貫性を評価することで、プロセスハッシュの異常を検出します。この機能には次のような開発背景があります。同じセットアップ構成から複製された Apache Web サーバーのファームがあるとします（これらのサーバーが同じ自動化スクリプトから展開されるなど）。その場合、すべてのサーバー上の httpd バイナリのハッシュは同じであると予想されます。不一致がある場合は異常であり、さらに調査が必要であると考えられます

正式には、同じルート範囲内のワークロード全体で、実行可能ファイルのバイナリパス、OS バージョン、パッケージ情報（該当する場合）1 の組み合わせが同一である一連のプロセスとしてプロセスグループを定義します。

（注） パッケージ情報は 3.4 リリース以降で含まれます。それ以前のリリースでは、プロセスグループは、実行可能ファイルのバイナリパスと OS バージョンの組み合わせのみに基づいて定義されていました。

上記の例では、すべての Apache Web サーバーが CentOS 7.7 および同じルート範囲で httpd 2.4.43 を実行していると仮定した場合、対応するプロセスグループは（すべてのサーバーにわたって）同じ組み合わせを持つプロセスのセットとなります（バイナリパスが `/usr/sbin/ httpd` で、OS バージョンが `CentOS-7.7` で、パッケージバージョンが `httpd-2.4.43`）。同じプロセスグループ内のすべてのバイナリのハッシュは同一であると予想されるため、不一致が検出されると異常が表示されます。

この機能は、異常なプロセスハッシュを検出するだけでなく、ユーザーがアップロードしたフラグ付きリストに表示されるプロセスハッシュも検出します。既知のマルウェアハッシュのリストがあり、それらのハッシュに関連付けられたプロセスが実行されているかどうかを知りたいということが開発の背景にあります。

誤報を減らすために、NIST が提供する National Software Reference Library の Reference Data Set（RDS）（NIST RDS データセットとも呼ばれます）を良性リストとして使用します。良性ハッシュは「安全」と見なされます（NIST RDS データセットを有効にする方法については、脅威インテリジェンスを参照してください）。独自の良性ハッシュリストをアップロードすることもできます。

NIST RDS データセットに加えて、**Cisco Secure Workload Hash Verdict** サービスもキュレートしています。このサービスを有効にすると、既知のマルウェアハッシュが発生した場合に、悪意のあるハッシュとして検出されます。一方、ハッシュが既知で正当な場合は、異常分析で良性とマークされることもあります。非常に大規模なデータセットと迅速な更新により、ワークロードで実行されているプロセスの承認または危険信号の付与に使用できるすべての既知の正当なプロセスハッシュをカバーしているため、Secure Workload Hash Verdict は Secure Workload クラウド経由でのみ利用できます。「脅威インテリジェンスの自動更新」を参照して、アプライアンスから Secure Workload Hash Verdict サービスにアクセスできることを確認してください。

この機能では、**プロセスハッシュスコア**と呼ばれるセキュリティスコアが出力されます。このスコアは 1 時間ごとに計算され、出力されます。他のすべてのセキュリティスコアと同様に、プロセスハッシュスコアは高いほど優れています。プロセスハッシュに固有のポイント:

- ハッシュスコアが 0 の場合、ハッシュにフラグが設定されているか、または悪意のあることを意味します。
- ハッシュスコアが 100 の場合、ハッシュが良性であるか、ワークロード全体で一貫している（不一致がない）ことを意味します。

- ハッシュスコアが 1 から 99 までの場合、ハッシュが異常であると見なされることを意味します（つまり、いくつかの不一致があります）。

ワークロードのプロセスハッシュスコアは、そのワークロードで観察されたすべてのハッシュの最小プロセスハッシュスコアであり、0 はシステム内にフラグ付きまたは悪意のあるプロセスハッシュがあることを意味し、100 はシステムでハッシュ異常が観察されていないことを意味します。

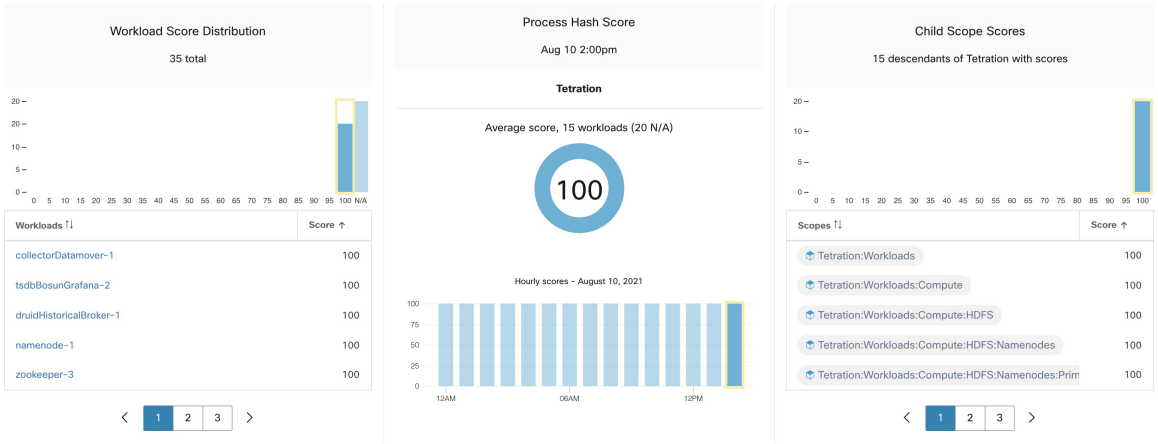
プロセスハッシュ機能を有効にする方法

プロセスハッシュ機能は、優れた可視性エージェントと適用エージェントでデフォルトで有効になっています。フォレンジック設定は必要ありません。システムにそのようなエージェントがある場合は、システムが起動してから 2 時間以内にスコア表示が開始されます。

プロセスハッシュスコアが表示される場所

- セキュリティダッシュボード：

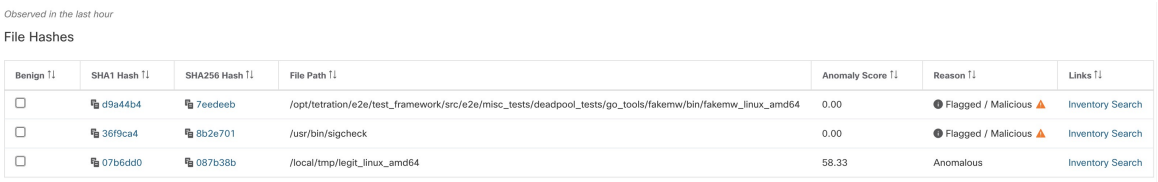
図 *17 :* セキュリティダッシュボードの *[*プロセスハッシュスコア（*Process Hash Score*）*]* セクション



セキュリティダッシュボードの [プロセスハッシュスコア（Process Hash Score）] セクションセキュリティ ダッシュボード

- [ワークロードプロファイル（Workload Profile）] ページ/[ファイルハッシュ（Workload Profile ）] タブ：

図 *18 : [*ワークロードプロファイル（*Workload Profile*）*]* ページの *[*ファイルハッシュ（*File Hashes*）*]* タブ



[ワークロードプロファイル（Workload Profile）]ページの[ファイルハッシュ（File Hashes）]タブ ワークロード プロファイル

プロセスハッシュスコアの計算方法

各プロセスハッシュについて、次のようにスコアを計算します。

1. ハッシュにフラグが付けられているか、悪意がある場合：`score = 0`
2. 上記とは異なり、ハッシュがクリーンな場合：`score = 100`
3. 上記とは異なり、ハッシュに何らかの異常がある場合：`score` が `[1, 99]` の範囲内。高いほど良い
4. 上記とは異なる場合：`score = 100`

（3）のスコアを計算するロジックは、まずハッシュのマイノリティスコア（同じルート範囲に属するワークロードの母集団における該当するハッシュの母集団の比率を1から引いた値）を計算し、次にそれを範囲 `[0.0, 1.0]` にマッピングするというものです。ハッシュのマイノリティスコアが 0.5 を超える場合、情報関数 `-log2(x)` を使用して、スコアを範囲 `[1.0, 99.0]` に再度マッピングします。上記の Apache Web サーバーファームの例を取り上げ、`httpd` のハッシュについて考えてみましょう。以下にいくつかのシナリオを示します。

- `httpd` がファーム内の 1000 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。1 台のサーバーで `h1`、残りの 999 台のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = 0.001, population_ratio(h2) = 0.999`. 実行されるアクション
  - `minority_score(h1) = 0.999, minority_score(h2) = 0.001`. 実行されるアクション
  - `score(h1) = -log2(0.999) * 98 + 1 = 1.14;`
  - `minority_score(h2) < 0.5, h2` は異常とは見なされないため、`score(h2) = 100` となります。
- `httpd` がファーム内の 10 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。1 台のサーバーで `h1`、残りの 9 台のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = 0.1, population_ratio(h2) = 0.9`. 実行されるアクション
  - `minority_score(h1) = 0.9, minority_score(h2) = 0.1`. 実行されるアクション
  - `score(h1) = -log2(0.9) * 98 + 1 = 15.90;`
  - `minority_score(h2) < 0.5, h2` は異常とは見なされないため、`score(h2) = 100` となります。

- `httpd` がファーム内の 2 台のサーバーに 2 つのハッシュ値（`h1` と `h2`）を持っているとします。一方のサーバーで `h1`、他方のサーバーでは `h2` です。この場合、次のように計算します。
  - `population_ratio(h1) = population_ratio(h2) = 0.5`. 実行されるアクション
  - `minority_score(h1) = minority_score(h2) = 0.5`. 実行されるアクション
  - `score(h1) = score(h2) = -log2(0.5) * 98 + 1 = 99.0`. これは、異常と見なされるハッシュに付けられる最高のスコアです。
- `httpd` がすべてのサーバーに 1 つのハッシュ値（`h1`）のみを持っているとします。この場合、`minority_score(h1) = 0.0 < 0.5` となります。したがって、異常とは見なされず、`score(h1) = 100` となります。

最終的に、ワークロードのプロセスハッシュスコアは、そのワークロードで観察されたすべてのハッシュの最小プロセスハッシュスコアになります。

`-log2(x)` 情報関数に関する追加情報は、こちらで見つけることができます。

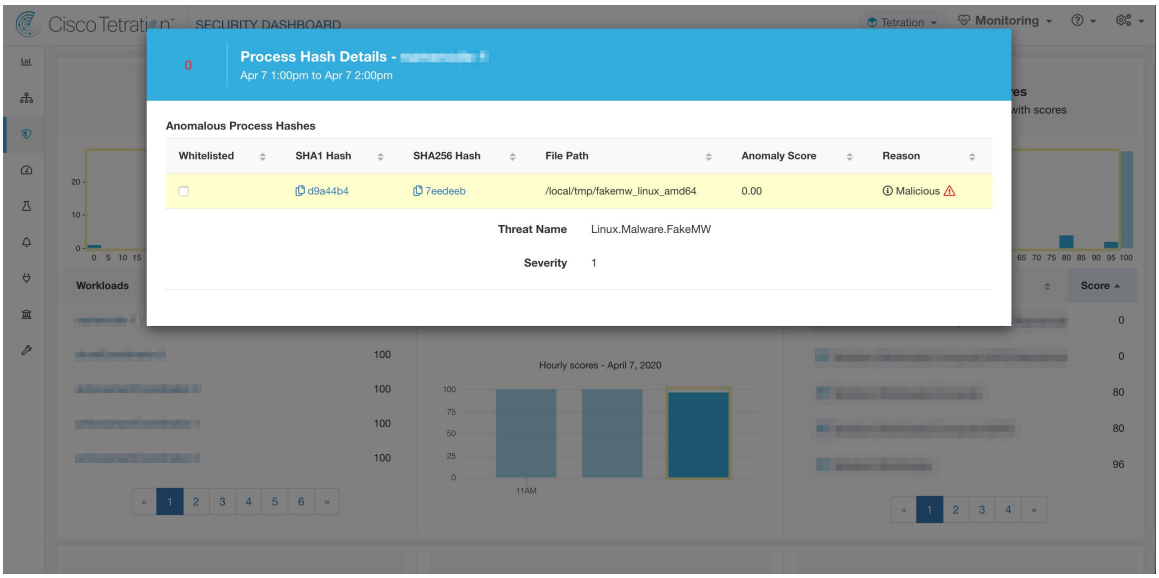
プロセスハッシュスコアの改善方法

ワークロードでのプロセスハッシュスコア 0 は、フラグ付きのプロセスハッシュまたは悪意のあるプロセスハッシュがそのワークロードに現れたことを意味します。このようなプロセスが再度実行されないようにすると、スコアが改善します。100 未満の正のプロセスハッシュスコアは、システム全体にプロセスハッシュの異常があることを意味します。悪意がある場合とそうでない場合がありますが、さらに調査が必要です。慎重に調査した結果、ハッシュが安全であると結論付けられた場合、それを良性リストに追加すると、スコアも改善します。ユーザーは [ファイルハッシュ（File Hashes）]/[プロセスハッシュの詳細（Process Hash Details）] ページで [良性（benign）] のチェックボックスをオンにするか、OpenAPI を介して良性のリストをアップロードすることで、異常なハッシュを「良性」として設定できます。

脅威情報の詳細

前述のように、Secure Workload Hash Verdict サービスが有効になっている場合、既知のマルウェアハッシュが発生すると、悪意があるとフラグが付けされます。その場合、悪意のあるハッシュの追加の脅威情報（脅威インテリジェンスプラットフォームを介して収集される）が提供されます。現在、追加の脅威データには、脅威の名前と重大度が含まれています。脅威名は脅威の名前で、重大度は脅威の重大度を示す 1 から 5 の値です。1 は重大度が最も低く、5 は重大度が最も高いことを意味します。

図 *19* : ユーザーは悪意のあるハッシュの行をクリックして、その脅威情報の詳細を表示できます



警告

- プロセスハッシュ分析タスクは1時間ごとに実行されますが、アクションによっては、予想されるスコアや結果がセキュリティダッシュボードに表示されるまでに最大2時間かかる場合があります。以下に例を示します。
  - ハッシュフラグ付きリストをアップロードし、そのリストにプロセスハッシュが表示された場合、スコアがセキュリティダッシュボードに反映されるまでに最大1時間かかる場合があります。
  - フラグ付きリストからハッシュを削除した場合、セキュリティダッシュボードでハッシュが完全に消去される（およびスコアが反映される）までに最大2時間かかる場合があります。
- 保持：
  - プロセスハッシュ分析の詳細な結果は、少なくとも 7 日間保持されます。
- [ワークロードプロファイル（Workload Profile）]ページの[ファイルハッシュ（File Hashes）]タブには、過去 1 時間に分析されたプロセスハッシュの詳細のみが表示されます。
- 以前のバージョンの優れた可視化エージェントと適用エージェント、および AnyConnect エンドポイントは、SHA256 ハッシュ値のみを報告します。したがって、SHA1 ハッシュのフラグ付きリストや良性リストとの照合は、これらのエージェントではサポートされていません。
- プロセスハッシュスコアは、特定のルート範囲について計算されます。ワークロードが複数のルート範囲に属している場合、そのワークロードが属するすべてのルート範囲の最小スコアがそのワークロードのプロセスハッシュスコアになります。

- プロセスハッシュの異常分析での誤認アラームをさらに減らすために、すべての Secure Workloadエージェントのバイナリをそれらのファイルパスに従って良性としてマークします。これらのハッシュがユーザー定義のハッシュリストに表示されない場合、またはSecure Workload Hash Verdict サービスによってフラグ付けされない場合にのみこのメカニズムが発生します。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。