SIEM 監視

SIEM 監視

Pandora FMS ドキュメント一覧に戻る

概要

SIEM は Security Information and Event Management の略です。SIEM は、セキュリティイベントの収集、フィルタリング、正規化、相関分析、可視化といった セキュリティ監視 機能を表します。SIEM は、セキュリティ情報管理 (SIM) とセキュリティイベント管理 (SEM) を組み合わせたものです。

SIEM は、通常のログ (Apache ログなど)、特定のセキュリティツールのログ (ファイアウォール、IDS、ハニーポット、EDR などのログ)、または別のツール (別の SIEM) からの強化されたイベントから取得されるセキュリティイベントを管理します。

すべての SIEM PFMS 機能を最大限に活用するには、エンドポイント に少なくともバージョン 783 をインストールすることを 推奨 します。

動作の仕組み

Pandora FMS SIEM は、コレクション から取得したログエントリを処理し、データを正規化し、これらのエントリに基づいてセキュリティイベントを生成する役割を担います。

ログ収集と同様に、生成された SIEM イベントは OpenSearch® に保存されるため、この監視を機能させるための最初の要件は OpenSearch インストール です。

Pandora FMS SIEM は、サーバ内の 2 つのコンポーネントを使用してセキュリティイベントを生成するため、イベントも 2 つのステップで生成されます。

- データ標準化: すべてのログ収集エントリがデコードされ、新しい正規化されたログエントリが生成され、一時的に保存されます。
- イベント生成: 正規化されたログは、一連のルールに準拠しているかどうかがチェックされます。準拠している場合は、すべてのルール情報とイベントを生成した正規化されたログを含む SIEM イベントが生成されます。

したがって、SIEM 監視の完全なフローは次のようになります。

- エージェント、プラグイン、その他の情報ソースは、ログを監視または生成し、「dataserver」または「syslogserver」に送信されます。
- `dataserver` と `syslogserver` はこれらのログエントリを処理し、ログ収集用に設定された

OpenSearch サーバに保存します。
- `siemserver` は、ログ収集で取得したすべてのログをデコードし、SIEM 監視用に設定された OpenSearch サーバ上に正規化されたログを生成します。これらの正規化されたログは一時的に保存されます。
- `siemevents` は、正規化された各ログを処理し、事前定義された一連のルールを満たす場合、SIEM イベントを生成して、SIEM 監視用に設定された OpenSearch サーバに保存します。

生成されたイベントは、Pandora FMS コンソールから操作を確認できます。

コンソール設定

コマンドセンター が有効になっている場合、Pandora SIEM はノードでのみ使用可能になります。

SIEM イベント監視を使用するには、まずメイン設定から有効化する必要があります。

メニュー 管理(Management) → セッティング(Settings) → システム設定(System Settings) → SIEM → SIEM の有効化(Activate SIEM) に移動し、フォームをすべて入力して 更新(Update) をクリックして変更を保存します。

これにより、ログの正規化と SIEM イベント生成用に指定された OpenSearch サーバに必要なテンプレートが作成されます。

また、`siemserver` および `siemevents` が開始された Pandora FMS サーバでもこのタイプの監視が有効になります。

サーバ設定

SIEM イベント監視を実行するには、サーバ設定 で `siemserver` および `siemevents` サーバを有効にします。

ログ収集エントリをデコードおよび正規化するには、各ケースで必要な情報を取得する方法を確立するデコード XML ファイル（以下、「デコーダー」）が必要です。これらの XML ファイルは、サーバ設定の `siem_decoders` パラメータで指定されたパスに配置されます。デフォルトでは以下のパスです。

```
/usr/share/pandora_server/util/siem/decoders
```

SIEM イベントを生成するには、正規化されたログから取得した情報に基づいてイベントを生成する条件を設定したルール XML ファイル（以下、「ルール」）が必要です。これらの XML ファイルは、サーバ設定の `siem_events_rules` パラメータで指定されたパスに配置されます。デフォルトでは

以下の場所にあります。

```
/usr/share/pandora_server/util/siem/rules
```

Wazuh® の デコーダー と ルール の XML ファイルは、Pandora FMS SIEM 監視でサポートされています。

エージェント設定

ログ収集の大部分は Pandora FMS エンドポイントを通じて行われます GNU/Linux® と MS Windows® の両方のシステムで動作するこれらのエージェントには、このタスクを実行するための特定の種類のモジュールが搭載されています。

すべての SIEM PFMS 機能を最大限に活用するには、エンドポイント に少なくともバージョン 783 をインストールすることを 推奨 します。

SIEM 監視は収集されたログの種類に大きく依存するため、その種類を示すにはログ収集モジュールで module_source_type を指定する必要があります。

タイプは デコーダー と ルール によって使用されるため、各ログに示されるタイプを確認するには、アクティブな デコーダー と ルール をチェックする必要があります。

最も一般的に使用されるログの種類は次のとおりです。

- syslog
- ids
- web-log
- squid
- windows
- host-information
- ossec

Linux® エージェント

Linux システムにおけるログ収集は、主にログファイルの読み取りによって行われます。これは、以下の最小限の構造を持つモジュール設定を使用することで実現できます。

```
module_begin
module_name <program_name>
module_type log
module_regexp <path_to_log_file>
```

```
module_pattern <capture_regexp>
module_source_type <log_type>
module_end
```

たとえば、Apache サーバからすべてのアクセスログエントリを収集するには、次のようにします。

```
module_begin
module_name apache
module_type log
module_regexp /var/log/httpd/access_log
module_pattern .*
module_source_type web-log
module_end
```

上記のようなログから収集されたエントリは、次のような デコーダー で正規化されます。

- web-accesslog
- web-accesslog-ip
- web-accesslog-domain

このようなログのデコードされたログでは、次のようなイベントが生成されます。

- Common web attack
- XSS (Cross Site Scripting) attempt
- SQL injection attempt

MS Windows® エージェント

MS Windows® でのログ収集は主にシステムイベントの監視によって行われますが、Linux システムのようにログファイルを読み取ることによっても行うことができます。

Windows イベントシステムを使用すると、これらの 2 つの最小構成のいずれかを含むモジュール設定を使用して、これらのログを収集できます。

イベントが Application、System、または Security のいずれかの場合:

```
module_begin
module_name <module_name>
module_type log
module_logchannel
module_source <Application|System|Security>
module_source_type <log_type>
module_end
```

または、別のチャネルに属するイベントの場合:

```
module_begin
```

```
module_name <module_name>
module_type log
module_logchannel
module_source <log_channel_path>
module_source_type <log_type>
module_end
```

たとえば、すべての Security および Windows Defender イベントエントリを収集するには、次のようにします。

```
module_begin
module_name Windows_LogEvents_System
module_type log
module_logchannel
module_source Security
module_source_type ossec
module_end

module_begin
module_name Windows_LogchannelEvents_WindowsDefender
module_type log
module_logchannel
module_source Microsoft-Windows-Windows Defender/Operational
module_source_type ossec
module_end
```

上記のようなイベントから収集された入力は、デコーダー によって windows_eventchannel として正規化されます。

上記のようなイベントのデコードされたログでは、次のようなイベントが生成されます。

- Windows error event
- Short-time multiple Windows Defender warning events
- Multiple Windows Defender error events

ログファイル監視を使用する場合、設定は Linux システムと同じです。次のような最小限の設定が必要です。

```
module_begin
module_name <program_name>
module_type log
module_regexp <path_to_log_file>
module_pattern <capture_regexp>
module_source_type <log_type>
module_end
```

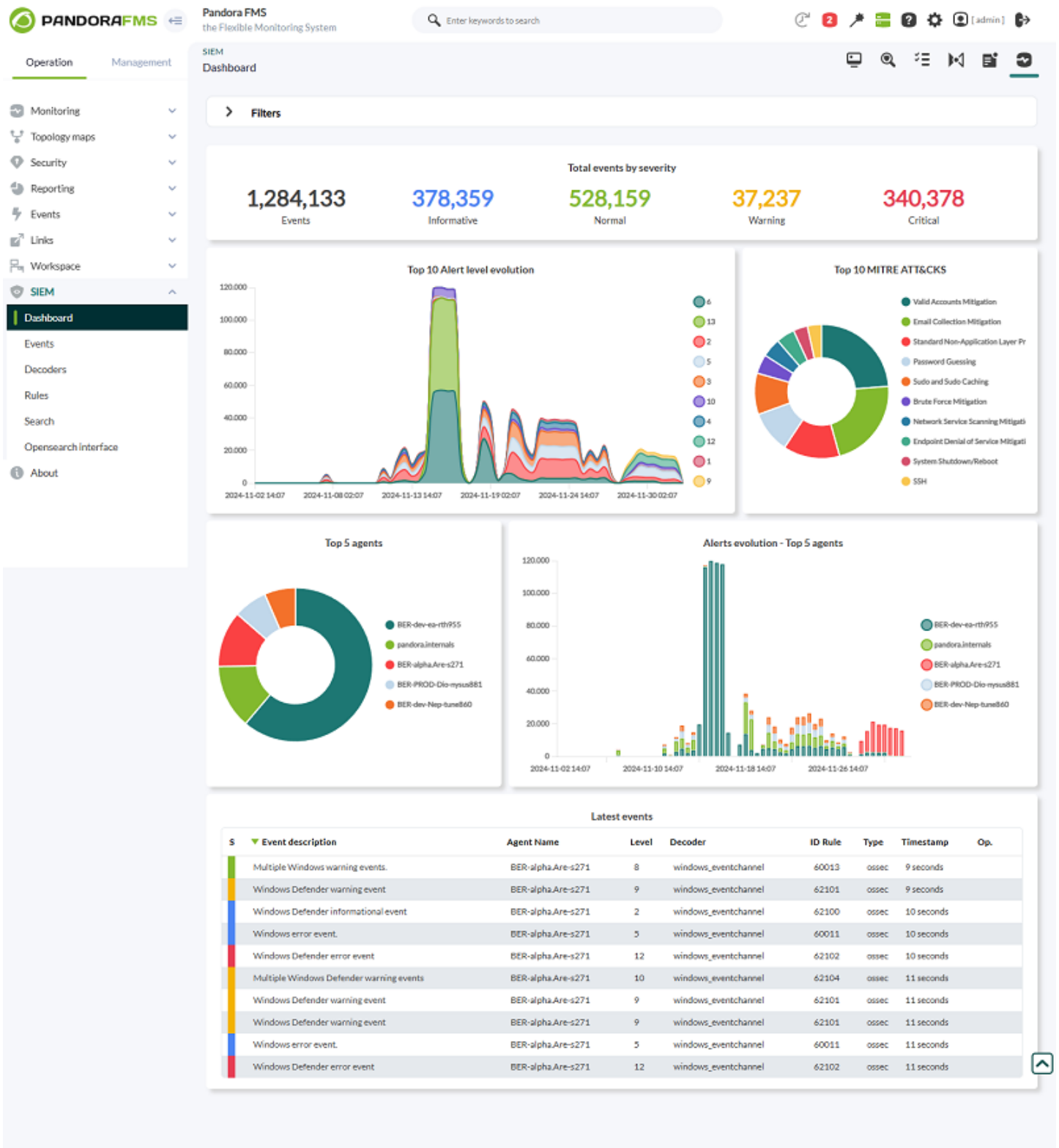
たとえば□X サーバのすべてのログ エントリを収集するには、次のようにします。

```
module_begin
module_name xserver
```

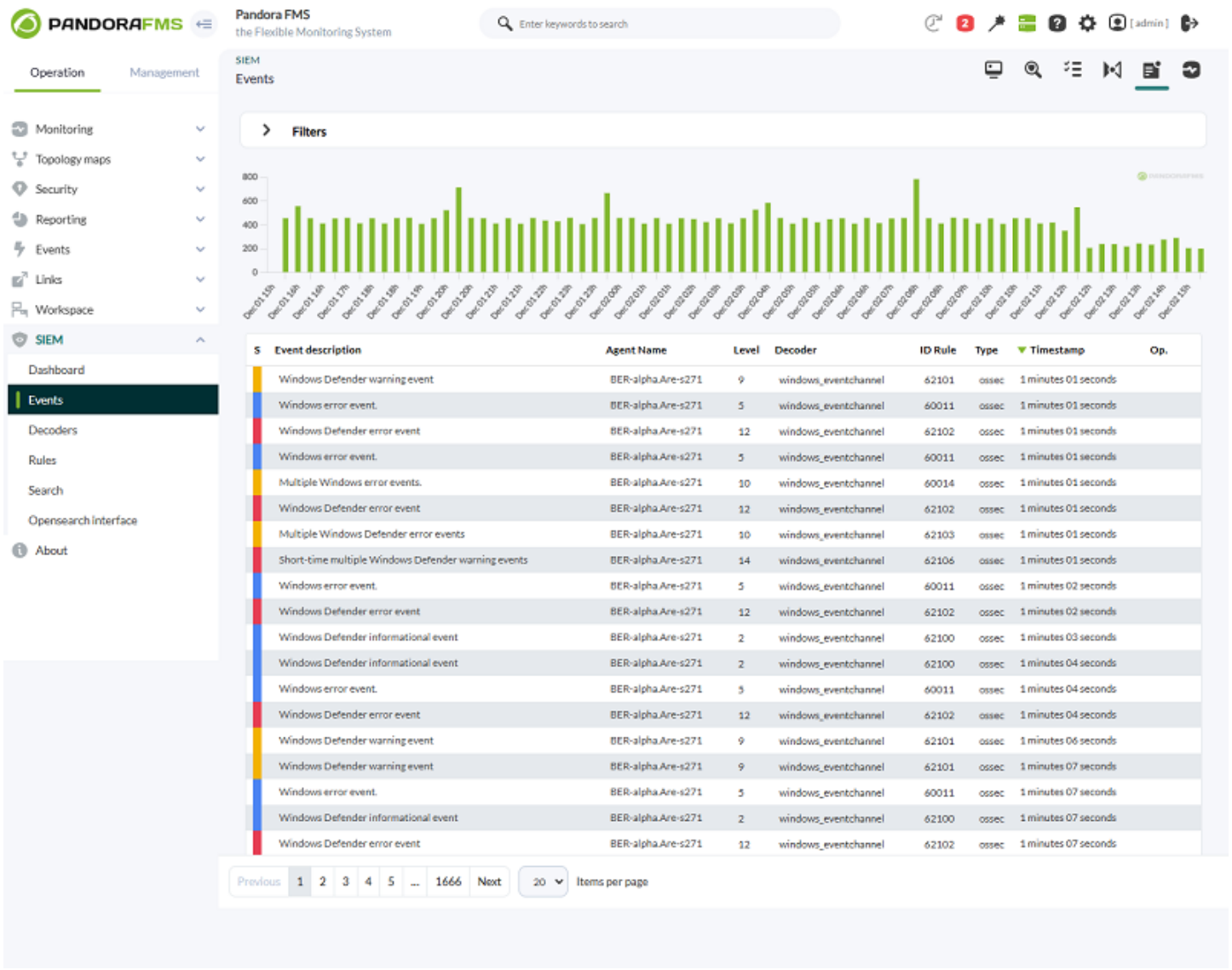
```
module_type log
module_regexp C:\server\logs\xserver.log
module_pattern .*
module_source_type xserver
module_end
```

SIEM イベント

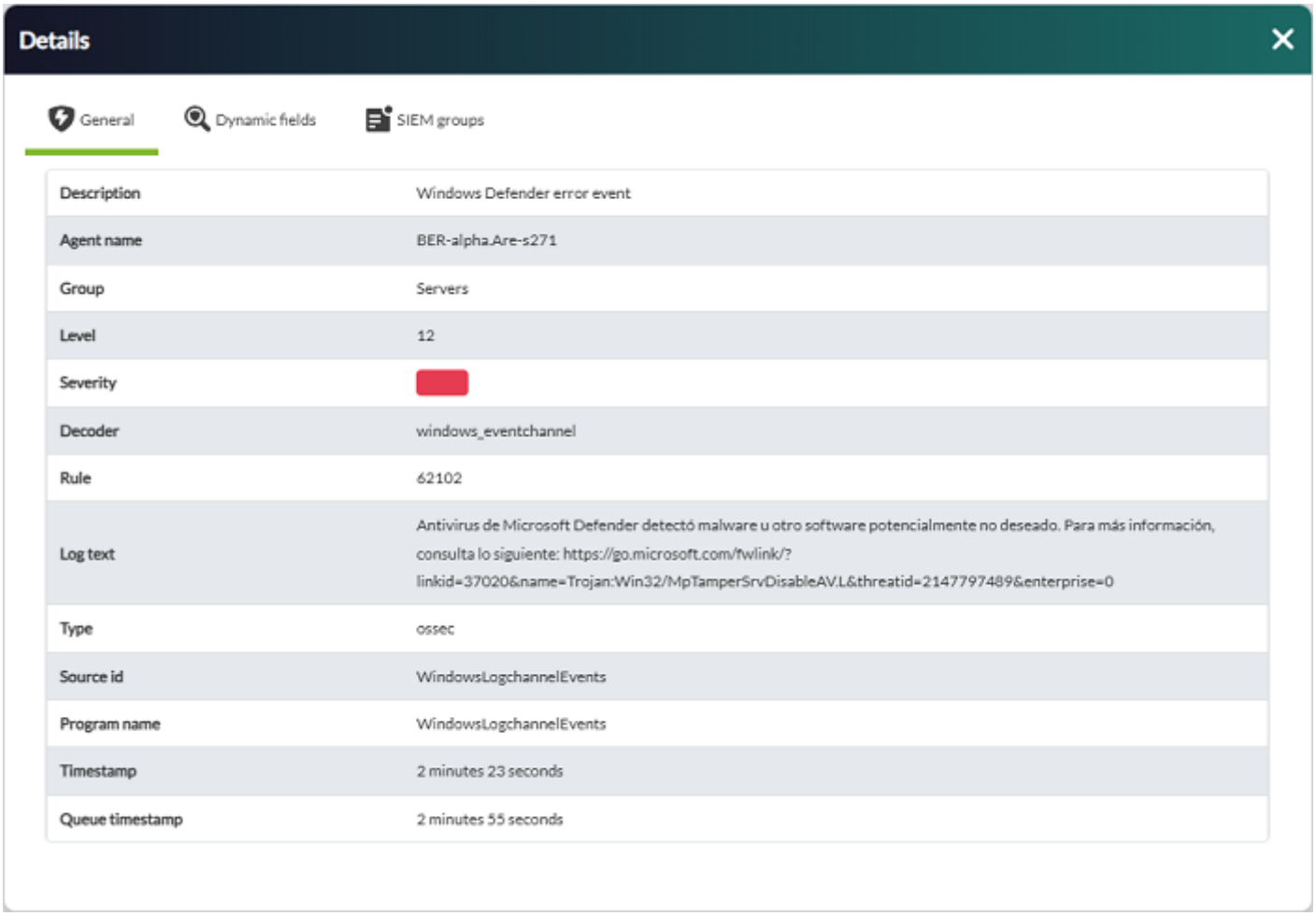
SIEM 監視を有効にして設定すると、操作(Operation) → SIEM → ダッシュボード(Dashboard) メニューで監視ステータスのプレビューにアクセスできるようになります。



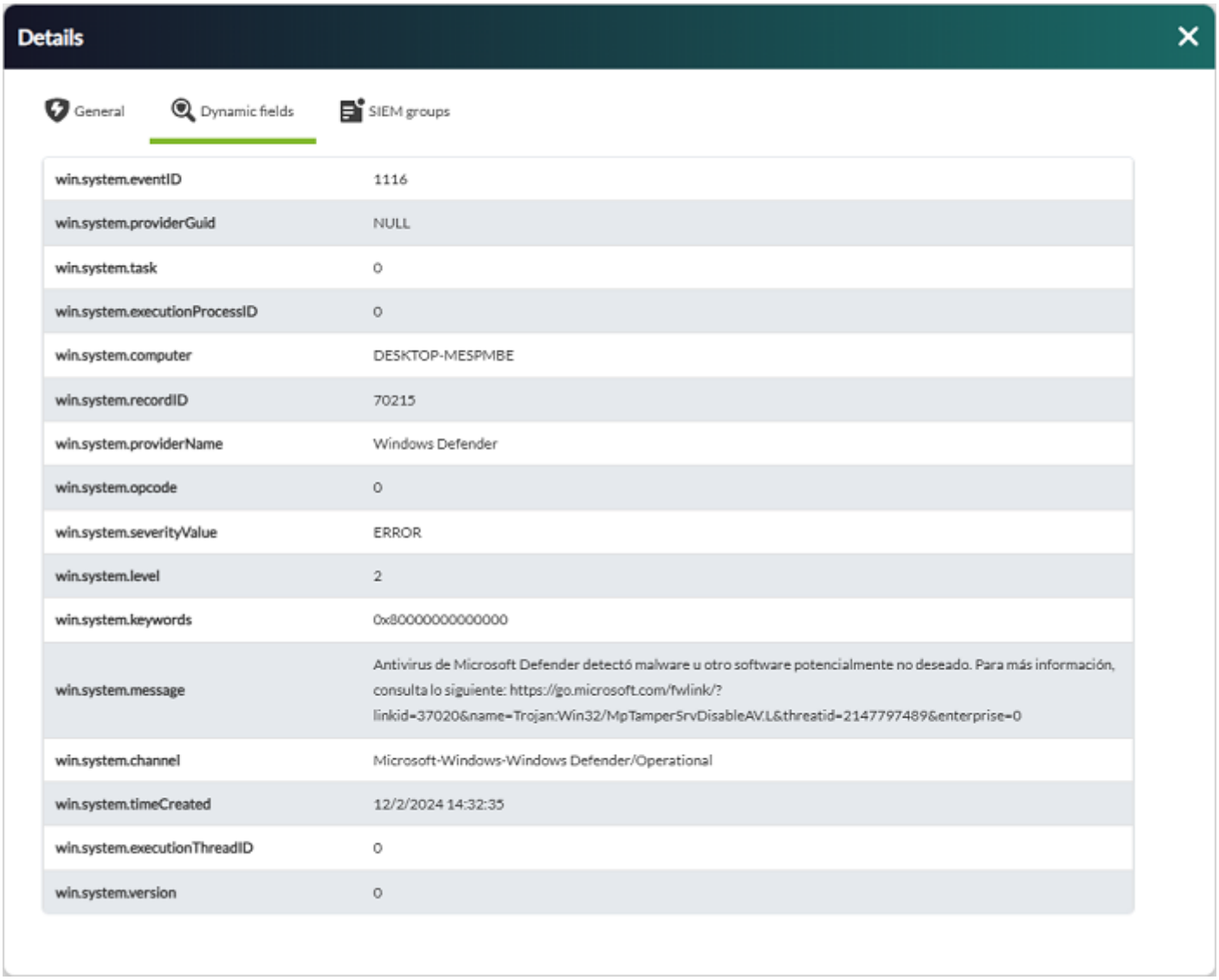
生成された SIEM イベントは、メニュー 操作(Operation) → SIEM → イベント(Events) ですべて表示できます。



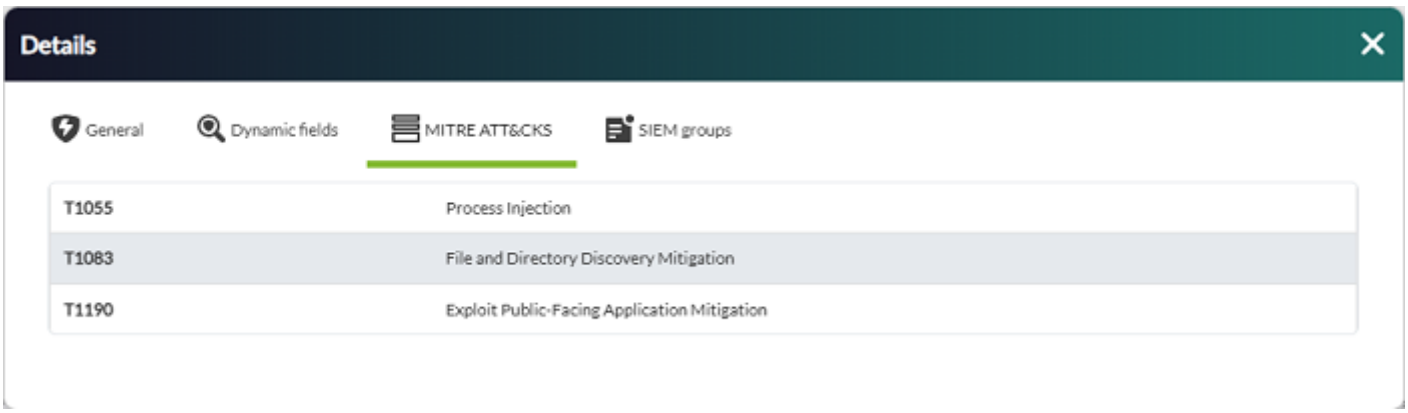
各 SIEM イベントにはイベントの詳細を示すウィンドウがあり、正規化されたログの情報とイベントを生成したルールが表示されます。

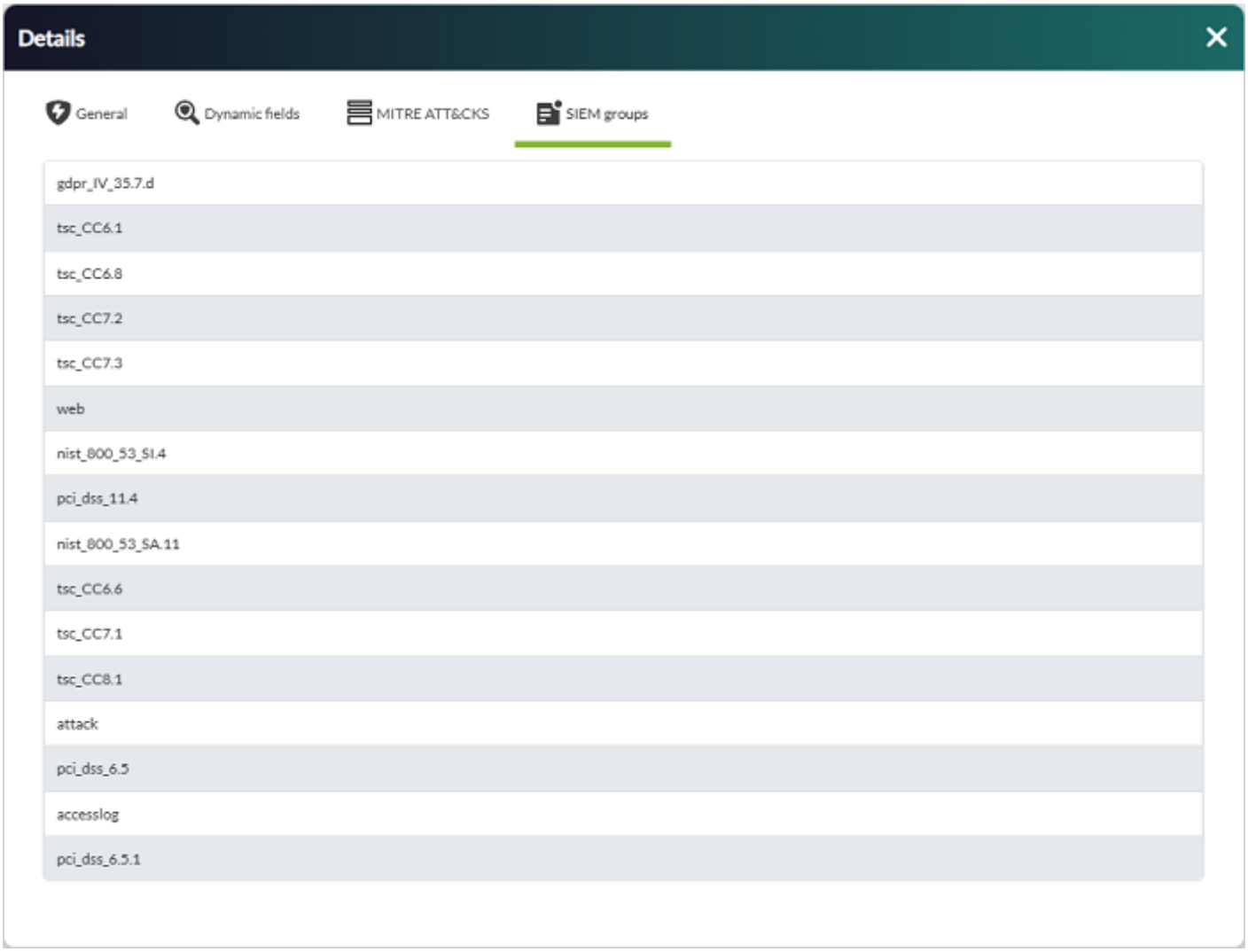


ログを正規化した デコーダー に応じて、イベントには、役立つログ情報を含む 動的フィールド(Dynamic fields) タブが表示されます。



イベントを生成したルールに応じて、MITRE ATT&CKS® および SIEM グループ(SIEM groups) タブが表示され、イベントの影響に関する有用な情報が表示されます。





一般的なダッシュボードとイベントテーブルの両方に表示される情報は、付属の SIEM ウィジェットを使用して Pandora FMS ダッシュボード に含めることができます。

デコーダー

Pandora FMS には□SIEM 監視用の一連の **デコーダー** がデフォルトで含まれていますが、管理者は独自のデコーダーを追加できます。

管理

新しい **デコーダー** を追加するには、まず Pandora FMS サーバ の設定パラメータ `siem_decoders` に指定されたパスにある XML ファイルを追加または編集します。

デコーダーは、Pandora FMS マスター サーバが各サービスの起動時に読み取る XML ファイルを介して環境にロードされます。

デコーダーがロードされると、その設定がデータベースに保存され、各 `siemserver` はログ収集を

通じて取得されたエントリに対してそれらを処理します。

Pandora FMS コンソールでは、メニュー 操作(Operation) → SIEM → デコーダ(Decoders) から、ロードされた デコーダー とその完全な設定を表示できます。

このビューから デコーダー を無効にすることもできます。これにより、siemserver はログ エントリを正規化するときにデコーダーを考慮しなくなります。

すべてのデコーダーは、再起動のたびに完全に読み込まれます。つまりXML ファイルから読み込めなかったデコーダーは（たとえある時点で読み込めていたとしても）使用できなくなります。また、コンソールから無効化されたデコーダーは（存在する場合）再び有効化されます。

書式

デコーダーは、Pandora FMS で XML ファイルを使用して設定およびロードされます。これらのファイルの有効な書式は次のとおりです。

```
<var name="VarName">VarValue</var>

<decoder name="DecoderName" discard="yes|no">
    <parent>DecoderName</parent>
    <program_name>REGEXP</program_name>
    <type>EventType</type>
    <prematch type="pcre2">REGEXP</prematch>
    <prematch offset="after_parent">REGEXP</prematch>
    <prematch offset="after_prematch">REGEXP</prematch>
    <regex type="pcre2">REGEXP</regex>
    <regex offset="after_parent">REGEXP</regex>
    <regex offset="after_regex">REGEXP</regex>
    <order>Field1, Field2.Sub1, Field2.Sub2</order>
    <json_null_field>string|discard</json_null_field>
</decoder>
```

var

↑書式

変数とその値を示すために使用され、後で XML で使用されます ($VarName)

```
<var name="VarName">VarValue</var>
```

decoder

↑書式

これはデコーダー情報とその名前です。XMLファイル自体に複数のデコーダーが含まれる場合があります。

- name: デコーダー の名前です。複数の デコーダー が同じ名前を持つ場合、それらはすべて評価されます。
- discard: yes または no の値を指定すると、この値に一致するログを デコーダー の評価から破棄するかどうかを指定します。discard="yes" が指定された デコーダー は、他のものより先に評価されます (親 デコーダー がない限り)。

```
<decoder name="DecoderName" discard="yes|no">
  …
  …
  …
</decoder>
```

parent

↑書式

階層構造を生成するために親 デコーダー の名前を指定します。

```
    <parent>DecoderName</parent>
```

program_name

↑書式

ログヘッダーまたはログの source_id で見つかるプログラムの名前。

- type: 正規表現の種類を指定します。指定がない場合は、OS Regex が使用されます。

```
    <program_name>REGEXP</program_name>
```

場合によっては、PCRE2 で次のように指定されます。

```
    <program_name type="pcre2">REGEXP</program_name>
```

type

↑書式

デコーダーが比較するログの種類。ログの type と一致する必要があります。

```
    <type>EventType</type>
```

prematch

↑書式

ログの内容がこれに一致する場合、正規化されたログが生成されます。

- type: 正規表現の種類を指定します。指定がない場合は、OS Regex が使用されます。
- offset を参照してください。

```
    <prematch type="pcre2">REGEXP</prematch>
    <prematch offset="after_parent">REGEXP</prematch>
    <prematch offset="after_prematch">REGEXP</prematch>
```

regex

↑書式

この正規表現でキャプチャされたグループは、ログの正規化された情報です。

- type: 正規表現の種類を指定します。指定がない場合は、OS Regex が使用されます。
- offset を参照してください。

```
    <regex type="pcre2">REGEXP</regex>
    <regex offset="after_parent">REGEXP</regex>
    <regex offset="after_regex">REGEXP</regex>
```

order

↑書式

regex によってキャプチャされた値は、キャプチャグループによってソートされてこれらのフィールド名に格納されます。

```
    <order>Field1, Field2.Sub1, Field2.Sub2</order>
```

json_null_field

↑書式

キャプチャグループの null 値は空の文字列として保存されるか、破棄されます。

```
    <json_null_field>string|discard</json_null_field>
```

offset

↑書式

これは、チェックの実行順序を示すために使用され、以下のチェックにおいて正規表現 after_parent□after_regex□after_prematch に一致するテキストをログコンテンツから削除します。例：

```
<decoder name="my_decoder">
  <prematch type="pcre2">^\d\d\d\d/\d\d/\d\d \d\d:\d\d:\d\d </prematch>
  <regex type="pcre2" offset="after_prematch">(\w):(\d+)</regex>
  <order>srcip,srcport</order>
</decoder>
```

ログ内のテキストが正規表現 ^\d\d\d\d/\d\d/\d\d \d\d□□:\d\d:\d\d に一致するかどうかを確認し、テキストの該当部分を破棄して、regex を評価する際にそれに応じたキャプチャ処理を行います。この例では、キャプチャグループを簡素化するために、評価時にテキストから日付を削除します。

ルール

Pandora FMS には SIEM 監視用のデフォルトの ルール セットが含まれていますが、管理者は独自のルールを追加できます。

ルール管理

新しい ルール を含めるには、まず、設定パラメータ siem_rules で Pandora FMS サーバに指定されたパスにある XML ファイルを追加または編集します。

ルールは、Pandora FMS マスター サーバが各サービスの起動時に読み込む XML ファイルによって環境にロードされます。

ルールがロードされると、その設定がデータベースに保存され、各 siemevents サーバはそれを SIEM 監視の正規化されたエントリとして処理します。

Pandora FMS コンソールでは、メニュー 操作(Operation) → SIEM → ルール(Rules) から、読み込まれた ルール とその完全な設定を確認できます。

このビューから ルール を無効にして、正規化されたログを処理するときに siemevents がそれらを考慮しないようにすることもできます。

すべてのルールは、再起動のたびにすべて読み込まれます。つまりXMLファイルから読み込めなかったルールは、たとえある時点で読み込めたとしても、利用できなくなります。デコーダーとは異なり、ルールには ID があり、これにより再起動のたびに無効にすることができます。

XMLファイルから読み取れなかったルールは、コンソールで「無効」とマークされSIEM イベントの生成時に考慮されません。管理者によって手動で無効化されたルールも考慮されません。したがって、ルールを評価するには、ルールがアクティブかつ有効化されている必要があります。

ルールを有効にし、あるルールが別のルールの評価と結果に依存する必要がある場合、最初のルールの数値識別子は 2 番目のルールの数値識別子よりも小さくする必要があります (つまり、数値の昇順で実行されます。対応するケーススタディ を参照してください)。

ソート

ルールは、最低レベル（0）から最高レベル（15）まで、複数のレベルに分類されます。以下の表は各レベルについて説明し、SIEM 監視によって生成される各イベントの重要度に関する情報を提供します。

レベル	タイトル	説明
0	Ignored.	アクションは実行されませんでした。誤検知を回避するために使用されます。これらのルールは、セキュリティとの関連性がないイベントを含む他のすべてのルールよりも先にスキャンされ、セキュリティイベントパネルには表示されません。
2	Notification of low system priority.	システム通知またはステータスメッセージ。セキュリティとの関連性がないため、セキュリティイベントパネルには表示されません。
3	Successful/authorized events.	これには、成功したログイン試行、ファイアウォールによって許可されたイベントなどが含まれます。

レベル	タイトル	説明
4	Low system priority error.	不適切な構成または未使用のデバイス/アプリケーションに関連するエラー。これらはセキュリティとの関連性がなく、通常はデフォルトのインストールまたはソフトウェアテストによって発生します。
5	User-generated error.	これには、パスワードの忘れ、アクションの拒否などが含まれます。これら自体にはセキュリティとの関連性はありません。
6	Low relevance attack.	これらは、システムに影響を与えないワームまたはウイルス（Apacheサーバーのコードレッドなど）を示します。また、頻繁な侵入検知システム（IDS）イベントや頻繁なエラーも含まれます。
7	Coincidence of “bad words”.	これらには（bad（error）などの単語が含まれます。これらのイベントのほとんどは分類されておらず、セキュリティの観点から何らかの関連性がある可能性があります。
8	First time seen.	初めて発生したイベントが含まれます。IDSイベントが初めてトリガーされたとき、またはユーザーが初めてログインしたときです。スニファーの起動などのセキュリティ関連のアクションも含まれます。
9	Invalid source error.	未知のユーザーまたは無効なソースからのログイン試行が含まれます。セキュリティに関連する可能性があります（特に繰り返し発生する場合）。これには（admin）アカウント（root）に関連するエラーも含まれます。
10	Multiple user-generated errors.	複数の間違ったパスワード、複数のログイン失敗などが含まれます。これらは攻撃を示している場合もあれば、ユーザーが認証情報を忘れただけの場合もあります。
11	Integrity check warning.	バイナリの変更やルートキットの存在（Rootcheckによる）に関するメッセージが含まれます。これらは攻撃が成功したことを示している可能性があります。また、無視されるIDSイベント（繰り返し回数が多い）も含まれます。
12	Event of high importance.	システム、カーネルなどからのエラーまたは警告メッセージが含まれます。特定のアプリケーションに対する攻撃を示している可能性があります。
13	Unusual error (high importance).	ほとんどの場合、一般的な攻撃パターンと一致します。
14	Security event of high importance.	ほとんどの場合、相関関係によってトリガーされ、攻撃を示しています。
15	Severe attack.	誤検知の可能性はありません。早急な対応が必要です。

これらのレベルに基づいて、イベントには特定の重要度が与えられ、コンソールに表示されます。

- 情報(Informational): レベル 0 から 6。
- 正常(Normal): レベル 7 から 8。
- 警告(Warning): レベル 9 から 11。
- 障害(Critical): レベル 12 から 15。

書式

詳細書式要素

```
<var name="VarName">VarValue</var>
```

```
<group name="GROUP1,GROUP2,">
    <rule id="N" level="N" frequency="N" timeframe="N" ignore="N"
overwrite="yes|no">
        <if_matched_sid>N</if_matched_sid>
        <if_matched_group>GROUP</if_matched_group>
        <same_id />
        <different_id />
        <same_field>Field1</same_field>
        <same_field>Field2.Sub1</same_field>
        <different_field>Field1</different_field>
        <different_field>Field2.Sub1</different_field>
        <description>TEXT</description>
        <match type="pcre2">RREGEXP</match>
        <match negate="yes|no">RREGEXP</match>
        <regex type="pcre2">RREGEXP</regex>
        <regex negate="yes|no">RREGEXP</regex>
        <decoded_as>DecoderName</decoded_as>
        <category>EventType</category>
        <field name="Field1">REGEXP</field>
        <field name="Field2.Sub1" negate="yes|no">REGEXP</field>
        <program_name negate="yes|no">REGEXP</program_name>
        <time>TIME-RANGE</time>
        <weekday>DAYS</weekday>
        <if_sid>PARENT1, PARENT2</if_sid>
        <if_group>GROUP</if_group>
        <if_level>N</if_level>
        <info type="text|link|cve">TEXT|LINK|CVE</info>
        <group>GROUP1,GROUP2,</group>
        <mitre>
            <id>MITRE_ID</id>
            <id>MITRE_ID</id>
        </mitre>
    </rule>
</group>
```

ケーススタディも参照してください。

var

↑完全な書式

```
<var name="VarName">VarValue</var>
```

変数とその値を示すために使用され、後で XML で使用されます ($VarName)

group

↑完全な書式.

```
<group name="GROUP1,GROUP2,">
…
</group>
```

rule によるグループ化が可能です。また、同じルールの条件にも使用されます。

rule

↑完全な書式

```
    <rule id="N" level="N" frequency="N" timeframe="N" ignore="N"
overwrite="yes|no">
    …
    </rule>
```

ルール内の情報は次のとおりです。

1. id: ルール識別子。他のルールが上書きされない限り、一意である必要があります。
2. level: イベント生成時のレベル（0□15）。レベルが 0 のルールはイベントを生成しません。
3. frequenity: イベントを生成するために同時発生する必要がある回数。ルールの頻度は、エージェント自身のログに対してチェックされ、他のログに対してはチェックされません。
4. timeframe: 同時発生が許容される時間枠（秒）。
5. ignore: このルールは、ここで指定した秒数後に frequenity カウンターをリスタートします。
6. overwrite: yes または no の値を指定することで、同じ ID を持つルールの設定を上書きできます。overwrite="yes" と同時に level="0" のルールが設定されている場合、他のルールよりも先に評価され、ログが一致した場合は、そのログの残りのルールの評価は破棄されます。

if_matched_sid

↑完全な書式

```
        <if_matched_sid>N</if_matched_sid>
```

指定された ID を持つ別の rule が timeframe 時間内に frequency で指定された回数アラートを発報した場合、ルールは満たされます。

if_matched_group

↑完全な書式

```
        <if_matched_group>GROUP</if_matched_group>
```

if_matched_sid と似ていますが、グループ用です。

same_id

↑完全な書式

```
        <same_id />
```

同じ ID による同意を与える必要があります。

different_id

↑完全な書式

```
        <different_id />
```

異なる ID による一致を指定する必要があります。

same_field

↑完全な書式

```
        <same_field>Field1</same_field>
        <same_field>Field2.Sub1</same_field>
```

フィールド内が同じ値で一致する必要があります。

different_field

↑完全な書式

```
        <different_field>Field1</different_field>
        <different_field>Field2.Sub1</different_field>
```

フィールド内の異なる値との一致が行われる必要があります。

description

↑完全な書式

```
        <description>TEXT</description>
```

生成されるイベントのテキスト。(変数は、説明と info でカスタムフィールドの値とともに使用できます。例: $(Field1)$(Field2.Sub1)$(Field2.Sub2))

match

↑完全な書式

```
        <match type="pcre2">RREGEXP</match>
        <match negate="yes|no">RREGEXP</match>
```

ログの内容がこれに一致する場合、SIEM イベントが生成されます。

1. type: 正規表現の種類を指定できます。指定しない場合は、OS Regex が使用されます。
2. negate: yes を指定すると、一致を否定できます。

regex

↑完全な書式

```
        <regex type="pcre2">RREGEXP</regex>
        <regex negate="yes|no">RREGEXP</regex>
```

“match ” と同じです。

1. type: 正規表現の種類を指定できます。指定しない場合は、OS Regex が使用されます。
2. negate: yes を指定すると、正規表現を拒否できます。

decoded_as

↑完全な書式

```
        <decoded_as>DecoderName</decoded_as>
```

指定された デコーダー によってログがデコードされた場合、ルールは満たされます。

category

↑完全な書式

```
        <category>EventType</category>
```

デコーダー のタイプが一致する場合、ルールは満たされます。

field

↑完全な書式

```
        <field name="Field1">REGEXP</field>
        <field name="Field2.Sub1" negate="yes|no">REGEXP</field>
```

指定されたフィールドが値と一致する場合、ルールに準拠します。

1. negate: yes 値を指定すると、フィールドとの一致を拒否できます。

program_name

↑完全な書式

```
        <program_name negate="yes|no">REGEXP</program_name>
```

ログのソースが一致する場合、ルールは満たされます。

1. negate: yes の値を指定すると、ソースログとの一致を拒否できます。

time

↑完全な書式

```
        <time>TIME-RANGE</time>
```

指定された時間範囲内にイベントが生成された場合、ルールは一致します。

weekday

↑完全な書式

```
        <weekday>DAYS</weekday>
```

イベントが生成された場合、ルールが日 (monday - sunday, weekdays, weekends) に準拠します。

if_sid

↑完全な書式

```
        <if_sid>PARENT1, PARENT2</if_sid>
```

親ルールのいずれかが満たされると、ルールは満たされます。

if_group

↑完全な書式

```
        <if_group>GROUP</if_group>
```

ログが指定されたグループ内の他のルールを満たす場合、ルールは満たされます。

if_level

↑完全な書式

```
        <if_level>N</if_level>
```

同じレベルの別のルールが満たされている場合、そのルールは満たされます。

info

↑完全な書式

```
        <info type="text|link|cve">TEXT|LINK|CVE</info>
```

生成されたイベントの追加情報。[1)]

group

↑完全な書式

```
        <group>GROUP1,GROUP2,</group>
```

ルールグループのリスト。

mitre

↑完全な書式

```
        <mitre>
            <id>MITRE_ID</id>
            <id>MITRE_ID</id>
        </mitre>
```

ルールの MITRE ID のリスト。

ケーススタディ

以下のコードは、PHP 関連の SELinux ログのルールを記述しています。このルールは、Web サーバ上の通常のポート以外のポートへの接続試行に対してイベントを作成します。

```
<rule id="100201" level="6">
        <decoded_as>setroubleshoot_program</decoded_as>
        <match>/usr/sbin/php-fpm</match>
        <field name="object_target"
type="pcre2">^(?!.*\b(9200|3306|80|443)$).*$</field>
        <field name="object_target" type="pcre2">^(?!.*(directory|file)
(conf|data_in|cron.lock)).*$</field>
        <description>SELinux prevented /usr/sbin/php-fpm execution on: $(action)
access on the $(object_target)</description>
        <mitre>
            <id>T1071.002</id>
        </mitre>
        <group>exec,threat,PHP</group>
</rule>
```

したがって、ポート 9200、3306、80、および 443 への接続ではイベントは作成されません。*directory* または *file* タイプのイベントも作成されません。

SELinux で使用される一般的な デコーダー :

```
<decoder name="setroubleshoot">
    <program_name>^setroubleshoot$</program_name>
</decoder>

<decoder name="setroubleshoot_program">
    <parent>setroubleshoot</parent>
    <prematch type="pcre2">(SELinux is preventing|SELinux está negando
a)</prematch>
    <regex type="pcre2">(\S+) (?:from|de) (\S+) (?:access on the|el acceso a)
(.+?)\. (?:For complete SELinux messages run: sealert -l|Si quiere los mensajes
de SELinux completos, ejecute sealert -l) (\S+)$</regex>
    <order>binary,action,object_target,sealert_id</order>
</decoder>

<decoder name="setroubleshoot_program">
    <parent>setroubleshoot</parent>
    <prematch type="pcre2">failed to retrieve rpm info for path</prematch>
    <regex type="pcre2">failed to retrieve rpm info for path (\S+)</regex>
    <order>path</order>
</decoder>
```

ルール評価順序

次のケーススタディ (否定的な結果) は、ルールが他のルールを「照会」して一致が見つかったかどうかを確認し、一致自体を評価してイベントを生成する場合を示しています。

ハードウェアファイアウォールで失敗したログインを検出し、対応するアラートを作成するために使用される一連のルールがあります。

```
<rule id="81641" level="1">
    <decoded_as>fortigate-firewall-v6</decoded_as>
    <description>Fortigate v6 messages grouped.</description>
</rule>
```

- ルール 81641 はデコードされたログと一致し、データの取得を開始します。

```
<rule id="81603" level="0">
    <if_sid>81600,81601,81602,81641</if_sid>
    <description>Fortigate messages grouped.</description>
</rule>
```

- ルール ID 81603 は、前の 81641 を含む複数のルールに依存します。後者の 81641 は 81603 より大きいため、81603 は実行されません

```
<rule id="81614" level="4">
    <if_sid>81603</if_sid>
```

```
    <match>ssl-login-fail</match>
    <description>Fortigate: SSL VPN user failed login attempt.</description>
    <group>authentication_failed,
gdpr_IV_32.2,
gdpr_IV_35.7.d,
gpg13_7.1,
hipaa_164.312.b,
invalid_login,
nist_800_53_AC.7,
nist_800_53_AU.14,
pci_dss_10.2.4,
pci_dss_10.2.5,
tsc_CC6.1,
tsc_CC6.8,
tsc_CC7.2,
tsc_CC7.3,</group>
</rule>
```

- □ ID 100700 のルールは、ID 81641 のルールとの一致をチェックします。このルールは、ルール ID 81603 に依存します (81614 は 81603 より大きいため、実行されます)。

このケーススタディの問題は、ID による番号順で 81603 が実行されなかったため、81614 が 100700 に結果を返さないことです。

正規表現

正規表現はパターンを定義する文字のシーケンスです。

デコーダーと ルールに有効な正規表現には、OS Regex と PCRE2 の 2 種類があります。

OS Regex

C言語で作成されたライブラリをベースにしたシンプルな正規表現です。シンプルでありながら、最も一般的な正規表現をサポートするように設計されています。

許容される表現

式	有効な文字
\w	A-Z□a-z□0-9□'-'□'@'□'_'
\d	0-9
\s	スペース“ “
\t	タブ

式	有効な文字
\p	()*+,-.:;<=>?[]!"'#$%&|{}
\W	\w 以外の文字
\D	\d 以外の文字
\S	\s 以外の文字
\.	その他

修飾子

式	動作
+	1回以上の繰り返しに一致
*	0回以上の繰り返しに一致

特殊文字

式	動作
^	テキストの先頭を指定する
$	テキストの末尾を指定する
|	複数のパターン間で論理パターン or を作成する

エスケープ文字

次の文字を使用するには、\ でエスケープする必要があります。

$	(	)	\	|	<
\$	\(	\)	\	|	\<

制限事項

- 修飾子 * および + はバックスラッシュ式にのみ適用でき、単一文字には適用できません（例：\d+ はサポートされていますが、0+ はサポートされていません）。
- 選択はグループ内では使用できません。例：(foo|bar) は許可されていません。
- 複雑なバックトラックはサポートされていません。例：\p*\d*\s*\w*: はコロンのみに一致しません。\p* がコロンを包含するためです。
- . はピリオドに一致しますが、\. は任意の文字に一致します。
- \s はASCIIスペース（32）にのみ一致し、タブなどの他の空白には一致しません。
- キャレット、アスタリスク、またはそれ以上のリテラルに一致する構文はありません（ただし、\p は1つ以上のアスタリスクと他のいくつかの文字に一致します）。

PCRE2

Perl 互換正規表現 (PCRE2) は、再帰パターン、先読みアサーションと後読みアサーション、非キャプチャグループ、非貪欲な量指定子、文字と文字クラスの拡張書式などの機能を提供します。

詳細については、PCRE2 書式のドキュメント を参照してください。

許容される表現

式	有効な文字
.	改行以外の任意の文字
\d	任意の10進数。[0-9] に相当します。
\D	10進数以外の任意の文字。[^0-9] に相当します。
\h	任意の水平空白文字
\H	水平空白以外の任意の文字
\s	任意の空白文字。[\t\r\n\f] に相当します。
\S	空白以外の任意の文字。[^\t\r\n\f] に相当します。
\w	任意の単語文字
\W	任意の非単語文字

修飾子

式	動作
?	0 または 1、greedy
?+	0 または 1、possessive
??	0 または 1、lazy
*	0 以上、greedy
*+	0 以上、possessive
*?	0 以上、lazy
+	1 以上、greedy
++	1 以上、possessive
+?	1 以上、lazy
{n}	ちょうど n
{n,m}	n 以上、m 以下、greedy
{n,m}+	n 以上、m 以下、possessive
{n,m}?	n 以上、m 以下、lazy
{n,}	n 以上、greedy
{n,}+	n 以上、possessive
{n,}?	n 以上、lazy

エスケープ文字

式	動作
\f	次のページ (16進数 0C)
\n	改行 (16進数 0A)
\r	キャリッジリターン (16進数 0D)
\t	タブ (16進数 09)
\0dd	8進コード 0dd の文字
\o{ddd..}	8進コード ddd.. の文字
\xhh	16進コード hh の文字
v\x{hhh…}	16進コード hh.. の文字

SIEM アラート

トピックSIEM アラートシステムを参照してください。

SIEM レポート

トピックSIEM イベントレポートを参照してください。

Pandora FMS ドキュメント一覧に戻る

1)
変数は、説明と info でカスタムフィールドの値とともに使用できます。例:
$(Field1)$(Field2.Sub1)$(Field2.Sub2)