Winevent口グイベント時間とSplunkインデックス時間の間に報告されたインデックス遅延

適用性

Splunkエンタープライズ

説明

識別する方法

まず、イベント時間(_time)とSplunkインデックス時間(_indextime)の差異を確認し、影響を受けるインデックスとソースタイプに応じて次の値を変更します。次の例では、セキュリティチャネルのイベントログを照会しています。

```
index=<splunk_index> sourcetype="WinEventLog:Security" host=<splunk_forwarder>
| fields _time, _indextime, host
| eval delay=(_indextime - _time)/60
| timechart avg(delay) by host
```

調査

確認すべき重要事項：

- インデックス作成の遅延が最初に発生したのはいつですか。
- 遅延が最初に発生した時期に、導入に対してどのような変更が行われたか（ある場合）。
- 遅延は時間の経過とともに増加するか（リニア）、または変動するか。

最大および平均イベント/秒(eps)を確認します。

```
index=_internal source=*metrics.log* security per_source_thruput host=<splunk_forwarder>
| timechart max(eps) max(max_age)
```

遅延が時間の経過とともに変化する場合、高いEPS値はデータスループットのボトルネックを示す可能性があります。

解決方法

クイック修正

このオプションは、現在のイベントを取得し、履歴データが失われても問題ない場合にのみ適用されます。各問題フォワーダで、次の操作を1つずつ実行する必要があります。

1. `$SPLUNK_HOME\etc\system\local\inputs.conf`で`current_only = 1`を設定します。

```
[WinEventLog://Security]
current_only = 1
```

2. フォワーダの停止
3. 古いチェックポイントファイルの名前を変更します。

```
ren "C:\Program Files\SplunkUniversalForwarder\var\lib\splunk\modinput\WinEventLog\Securit
```

4. フォワーダを開始する
5. 新しいセキュリティチェックポイントファイルが生成されていることを確認します
6. チェックポイントとセキュリティログが数百のイベントで相互にあることを確認
7. `$SPLUNK_HOME\etc\system\local\inputs.conf`ファイルの名前を変更し、`current_only=1`の設定をデフォルトに戻します。
8. Splunkの再起動
9. データが取り込まれていることを確認するための検索

通常のUF再起動を実行しても問題が解決しない理由

UFを再起動すると、チェックポイントの再読み込みと一時的な空のメモリバッファが強制的に実

行されます。イベント生成レートが処理/スループットの制限を超えているため、フォワーダは再起動後数分以内に再度ラグの蓄積を開始します。

長期にわたる解決策

この迅速な修正には、影響を受ける各UFに対する手動の介入が必要です。大規模な導入では一般的に実行可能ではありません（また、一部の管理者はUFに対するコマンドラインアクセスを持っていません）。それに加えて、UFは再び遅れをとる可能性が高いです。長期的なソリューションを提供することが最善の選択肢です。環境はそれぞれ異なるため、すべてのシナリオに適用できる簡単な設定はありません。ただし、このKBの提案は、多くの使用例で有効です。この記事のすべての設定の提案では、UF上でSplunkを再起動して有効にする必要があります。そのため、Deployment Server上のカスタムアプリケーションでSplunkdの再起動がチェックされていることを確認してください。

最初に試す(limits.conf)

maxKBpsを開く

次のオプションを調整する前に、`limits.conf/[thruput]maxKBps`を確認してください。UFのデフォルト設定は256 KB/sで、フォワーダのネットワーク伝送合計は最大22 GB/日に制限されます。セキュリティログを生成するビジー状態のサーバは、256 KB/sを超えるメモリを容易に生成するため、フォワーダが処理を継続することは物理的に不可能です。

したがって、まず最初に、`limits.conf`でこの値を0（無制限）に設定して、ネットワーク帯域幅の上限を解除します。

```
[thruput]
maxKBps = 0
```

ほとんどのユニバーサルフォワーダはデプロイメントサーバに電話をかけるため、ベストプラクティスは、すべてのフォワーダが一度にこの設定を受け取るように、デプロイメントアプリを介してこの設定を行うことです（最初は可能であればテスト環境で、その後は実稼働環境で）。 テスト環境が利用できない場合のもう1つの選択肢は、UFの小さなグループの「テスト」サーバクラスに展開し、あらゆる場所に適用する前に監視することです。同じことが、このサポート技術情報の記事で提供されるすべての提案にも当てはまります。

追加オプション

フォワーダのスループットを開いても問題が完全に解決しない場合は、次の点を考慮してください。

`inputs.conf/[WinEventLog://Security]`に固有

オプション#1 – バッチサイズの変更

デフォルトでは、Splunkは小さなバッチ（10個のイベント）でイベントを取得し、ディスク上のチェックポイントファイルを頻繁に更新します。ビジー状態のホストでは、チェックポイント・ファイル上のディスクI/Oが主要なチョークポイントになります。`batch_size = 500`を設定し、ディスク同期を600秒に遅延させ(`checkpointInterval = 600`)、`suppress_checkpoint = true`を設定すると、`splunk-winevtlog.exe`のEPS（イベント/秒）スループットが大幅に向上します。

```
batch_size = 500
suppress_checkpoint = true
checkpointInterval = 600
```

オプション#2 - renderXMLを使用してAPI呼び出しの数を減らす

デフォルトでは、Splunkはイベント内の各フィールドに対してAPIコールを発行します。renderXmlを有効にすると、代わりにイベントごとにAPI呼び出しが発行され、パフォーマンスが向上する可能性があります： [Windowsイベントログのイベントを XML で表示する]

```
renderXml = true
```

オプション#3：データペイロードを減らすために抑制するフィールドをフィルタリングする

このオプションはraw速度に対して有効ですが、監査要件を満たしていることを確認してください（該当する場合）。

これらの設定は、バイトサイズとAPIレンダリングコールを減らすために、送信前に生のイベントからヘッダーフィールドを取り除きます。SIEM、CIMモデル、またはコンプライアンスダッシ

ユボードがキーワード（例：監査成功/失敗）、TaskCategory、またはOpcodeに依存している場合、これらのフィールドをドロップすると、検出が中断されます。

```
suppress_sourcename = true
suppress_keywords = true
suppress_type = true
suppress_task = true
suppress_opcode = true
```

オプション#4 – 入力スタンザで**FormatMessage()** APIを有効にする

evtFormatMessage()は、ほとんどのWindows展開でデフォルトのAPIであり、以前のFormatMessage() APIと比較すると処理オーバーヘッドが増加します。古いAPIに戻すと、インデックス作成の遅延が軽減される場合があります。

renderXml = trueの場合は、これを設定しないでください。従来のFormatMessage() APIは、複雑なevtFormatMessage() DLL文字列のルックアップをバイパスするため、従来のプレーンテキストマルチラインログ(renderXml = false)を解析する場合にのみ高速です。renderXml = trueを使用すると、Splunkはすでに最新のEVTX API(EvtRender)から構造化された形式で直接読み取っており、メッセージ形式のDLLは呼び出されません。

```
use_old_eventlog_api = true
```

UFインジェストパイプラインの増加(server.conf)

デフォルトでは、フォワーダは単一の解析およびインジェストパイプラインを実行します。大量のイベントを処理するマルチコアマシンでは、パイプラインセットを増やして並列処理を可能にできます。これは、フォワーダがローカルWindows APIからデータを読み取り、パッケージ化し、ネットワークにプッシュするために使用する内部スレッドの数を制御します。

UF上では、取り込みパイプラインの1から2への増加によるリソースフットプリントは、インデクサまたはヘビーフォワーダと比較すると非常に軽量です。これは、UFでは軽量の読み取り、バッチ処理、およびネットワーク出力のみが行われるためです（正規表現の変換、フィールド抽出、またはディスクへのインデックス作成は行われません）。

Windowsサーバに少なくとも4つのCPUコアと8 GBのRAMがある場合、parallelIngrestionPipelines = 2を設定することは通常は安全であり、大量のイベントバックログ

のクリアと将来の遅延の防止に大きく役立ちます。CPU使用率はUFが追いつく間に増加する可能性がありますが、取り込みが通常のレベルに戻ると再び低下します。

```
[general]
parallelIngestionPipelines = 2
```

Outputs.confキューサイズ

入力バッチを500に調整する場合は、ネットワーク層が入力ワーカーをブロックしないように、TCP出力キューに十分なメモリバッファがあることを確認することを検討してください。

```
[tcpout:myindexers]
maxQueueSize = 50MB
```

Windows Defender/Antivirusの除外の確認

セキュリティエンドポイントエージェント（Defender、CrowdStrike、Carbon Blackなど）は、積極的に`splunk-winevtlog.exe`を検査し、`$SPLUNK_HOME\var\lib\splunk\modinput\WinEventLog\`ディレクトリをロックできます。 `$SPLUNK_HOME`ディレクトリ全体と`splunk-winevtlog.exe`がリアルタイムファイルシステムインスペクションから除外されていることを確認します。

ドメインコントローラのアフィニティ/ネットワークルーティング：

問題のUFがドメインコントローラで動作しているか、大量のActive Directoryトラフィックを処理している場合、ローカルファイアウォールまたはネットワークのレート制限によりSplunk TCPポート（デフォルトは`9997`）が受信側レイヤ（HFまたはインデクサ）にキャップされていないことを確認します。

関連リンク

- Splunkのドキュメント：
  - Splunk Enterpriseを使用したWindowsイベントログデータのモニタリング
  - ユニバーサルフォワーダの高度な設定
  - スペックファイル：server.conf、inputs.conf、outputs.conf

- Splunk KB記事
  - UFまたはHFからの遅延ログのトラブルシューティング
  - UFがSplunkにセキュリティログを送信しない
- Splunk Lanternの記事：転送層のパフォーマンスチューニング

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。