Atraso de indexação relatado entre a hora do evento do log Winevent e a hora do índice de Splunk

Aplicabilidade

Empresa Splunk

Descrição

Como identificar

Primeiro confirme o delta entre o tempo do evento (_time) e o tempo do índice de Splunk (_indextime), modifique o seguinte, dependendo do índice e do tipo de origem afetados. Este exemplo está consultando o registro de eventos do canal de Segurança:

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

Investigação

Itens importantes a serem verificados:

- Quando o atraso de indexação apareceu pela primeira vez?
- Quais alterações, se houver, foram feitas na implantação na hora em que o atraso apareceu pela primeira vez?
- A latência aumenta com o tempo (linear) ou flutua?

Confirme o máximo e a média de eventos por segundo (eps):

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

Se a latência variar ao longo do tempo, um alto valor de EPS pode indicar gargalos no throughput de dados.

Resolução

Correção rápida

Essa opção se aplica somente se você quiser obter eventos atuais e não se importar com dados históricos ausentes. O seguinte deve ser executado um de cada vez em cada encaminhador de problemas:

1. Defina `current_only = 1` em `$SPLUNK_HOME\etc\system\local\inputs.conf`:

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

2. Parar o encaminhador
3. Renomear o arquivo de ponto de verificação obsoleto:

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

4. Inicie o encaminhador
5. Certifique-se de que um novo arquivo de ponto de verificação de segurança seja gerado
6. Confirme se os registros de ponto de verificação e segurança estão entre algumas centenas de eventos
7. Renomeie o arquivo `$SPLUNK_HOME\etc\system\local\inputs.conf` para reverter a configuração `current_only=1` para o padrão.
8. Reiniciar Splunk
9. Pesquisar para confirmar se os dados estão sendo inseridos

Por que realizar reinicializações regulares de UF provavelmente não ajudará

Reinicializar a UF simplesmente força-a a reler os pontos de verificação e esvaziar brevemente os buffers de memória. Como a taxa de geração de eventos excede o limite de processamento/throughput, o encaminhador começará imediatamente a acumular o atraso novamente dentro de minutos após a reinicialização.

Soluções de longo prazo

A correção rápida requer intervenção manual em cada UF afetada, o que geralmente não é viável em uma implantação grande (e alguns administradores não têm acesso de linha de comando para UFs) aplicada. Além disso, é muito provável que as UFs fiquem para trás novamente. Sua melhor opção pode ser fornecer uma solução de longo prazo. Como cada ambiente é diferente, não há uma configuração simples que possa ser aplicada a cada cenário. No entanto, as sugestões neste KB devem funcionar para muitos casos de uso. Todas as sugestões de configuração neste artigo exigem que o Splunk seja reiniciado na UF para entrar em vigor, portanto, verifique se o aplicativo personalizado no Servidor de Implantação reiniciou o Splunkd.

Tente isso primeiro (limits.conf)

Abrir maxKBps

Antes de ajustar as opções abaixo, consulte `limits.conf/[throughput]maxKBps`. A configuração padrão em uma UF é 256KB/s, o que limita o encaminhador a ~22 GB/dia de transmissão total de rede. Um servidor ocupado que gera registros de segurança produz facilmente mais de 256 KB/s, tornando fisicamente impossível para o encaminhador manter-se atualizado.

Então, a primeira coisa: Defina esse valor como 0 (ilimitado) em `limits.conf` para desbloquear a largura de banda da rede:

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

Como a maioria dos Encaminhadores Universais está ligando para casa para um Servidor de Implantação, a prática recomendada é definir essa configuração por meio de um aplicativo de implantação para que todos os encaminhadores a recebam de uma só vez (primeiro em um

ambiente de teste, se possível, e depois em produção). Outra opção, se um ambiente de teste não estiver disponível, é implantar em uma classe de servidor de "teste" de um pequeno grupo de UFs e monitorar antes de aplicar em todos os lugares. O mesmo se aplica a todas as sugestões oferecidas neste artigo do KB.

Opções adicionais

Se a abertura do throughput dos encaminhadores não resolver o problema completamente, considere o seguinte:

Específico para `inputs.conf/[WinEventLog://Security]`

Opção 1 - Alterar o tamanho do lote

Por padrão, o Splunk captura eventos em pequenos lotes (10 eventos) e atualiza o arquivo de ponto de verificação no disco com frequência. Em hosts ocupados, a E/S de disco no arquivo de ponto de verificação se torna um ponto de estrangulamento principal. A definição de `batch_size = 500`, atrasando a sincronização do disco para 600s (`checkpointInterval = 600`) e a definição de `suppress_checkpoint = true` aumenta drasticamente o throughput de EPS (eventos por segundo) de `splunk-winevtlog.exe`.

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

Opção 2 - Usar renderXML para reduzir o número de chamadas à API

Por padrão, o Splunk emite uma chamada de API para cada campo dentro do evento. Habilitar renderXml emitirá uma chamada de API por evento, potencialmente melhorando o desempenho: [Exibir eventos do Log de Eventos do Windows em XML]

```
renderXml = true
```

Opção 3 - Filtrar quais campos são suprimidos para reduzir o payload de dados

Essa opção é eficaz para velocidade bruta, mas certifique-se de que ela atenda aos requisitos de auditoria (se aplicável).

Essas configurações removem os campos do cabeçalho do evento bruto antes de enviá-los para reduzir o tamanho do byte e as chamadas de renderização de API. Se o SIEM, os modelos CIM ou os painéis de conformidade dependerem de `palavras-chave` (por exemplo, êxito/falha de auditoria), `TaskCategory` ou `Opcode`, a eliminação desses campos interromperá essas detecções.

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

Option#4 - Habilite a API **`FormatMessage()`** na estrofe de entrada

`evtFormatMessage()` é a API padrão na maioria das implantações do Windows, que tem uma sobrecarga de processamento maior quando comparada à API FormatMessage() anterior. Reverter para a API mais antiga pode aliviar o atraso de indexação.

Não defina isso se **`renderXml = true`**. A API `FormatMessage()` legada é mais rápida somente ao analisar logs clássicos de várias linhas de texto sem formatação (`renderXml = false`), porque ela ignora a pesquisa pesada da string `evtFormatMessage()` DLL. Ao usar `renderXml = true`, o Splunk já lê diretamente da API EVTX moderna (`EvtRender`) em formato estruturado sem chamar DLLs de formatação de mensagem.

```
use_old_eventlog_api = true
```

Aumentar Pipelines de Ingestão de UF (server.conf)

Por padrão, um encaminhador executa um único pipeline de análise e inclusão. Para máquinas multi-núcleo que ingerem um volume de eventos pesado, o conjunto de distribuição pode ser aumentado para permitir o processamento paralelo. Isso controla quantos threads internos o encaminhador usa para ler dados de APIs locais do Windows, empacotá-los e enviá-los para a rede.

Em uma UF, a pegada de recursos de aumento de pipelines de ingestão de `1` para `2` é muito leve

em comparação com um Indexador ou Forwarder Pesado, porque a UF só faz leitura leve, lotes e saída de rede (sem transformações regex, extração de campo ou indexação para disco).

Se o servidor Windows tiver pelo menos 4 núcleos de CPU e 8 GB de RAM, a configuração `parallelIngestionPipelines = 2` é geralmente segura e ajudará significativamente a limpar um acúmulo de eventos pesados e a evitar atrasos futuros. O uso da CPU pode aumentar enquanto a UF fica presa, mas cairá novamente quando a ingestão voltar aos níveis normais.

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

Tamanho da fila Outputs.conf

Se estiver ajustando o lote de entrada para 500, considere garantir que a fila de saída TCP tenha buffer de memória suficiente para que a camada de rede não bloqueie o operador de entrada:

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

Verificar Exclusões do Windows Defender/Antivírus

Os agentes de ponto de extremidade de segurança (por exemplo, Defender, CrowdStrike, Carbon Black) podem inspecionar agressivamente o `splunk-winevtlog.exe` e bloquear o diretório `$SPLUNK_HOME\var\lib\splunk\modinput\WinEventLog\`. Verifique se o diretório `$SPLUNK_HOME` e o `splunk-winevtlog.exe` inteiros estão excluídos da inspeção do sistema de arquivos em tempo real.

Afinidade do controlador de domínio/roteamento de rede:

Se as UFs com problema estiverem sendo executadas em Controladores de Domínio ou manipulando alto tráfego do Ative Diretory, verifique se nenhum firewall local ou limite de taxa de rede está limitando a porta TCP de Splunk (padrão `9997`) à camada de recebimento (HFs ou indexadores).

Links relacionados

- Documentação do Splunk:
  - Monitorando dados do Log de Eventos do Windows com o Splunk Enterprise
  - Configurações avançadas para o Encaminhador universal
  - Arquivos de especificação: server.conf, inputs.conf, outputs.conf
- Artigos do KB do Splunk
  - Troubleshooting de logs atrasados de UF ou HF
  - UF não enviando logs de segurança para Splunk
- Artigo da Lanterna Splunk: Ajuste de desempenho da camada de encaminhamento

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.