Indexeringsvertraging gemeld tussen de tijd van het Winevent-logboek en de tijd van de Splunk-index

toepasbaarheid

Splunk Enterprise

Beschrijving

Hoe het te identificeren

Bevestig eerst de delta tussen event time (_time) en Splunk indextime (_indextime), wijzig het volgende afhankelijk van welke index en brontype wordt beïnvloed. In dit voorbeeld wordt de logboekregistratie van het beveiligingskanaal opgevraagd:

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

onderzoek

Belangrijke dingen om te verifiëren:

- Wanneer is de indexeringsvertraging voor het eerst opgetreden?
- Welke wijzigingen, indien van toepassing, zijn er aangebracht in de implementatie rond de tijd dat de vertraging voor het eerst optrad?
- Verhoogt de latentie in de loop van de tijd (lineair) of fluctueert deze?

Bevestig de maximale en gemiddelde gebeurtenis per seconde (eps):

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

Als de latentie in de loop van de tijd varieert, kan een hoge EPS-waarde indicatief zijn voor knelpunten in de gegevensdoorvoer.

resolutie

Snelle oplossing

Deze optie is alleen van toepassing als u actuele gebeurtenissen wilt ophalen en het u niet kan schelen dat historische gegevens ontbreken. De volgende handelingen moeten één voor één worden uitgevoerd op elke probleemforwarder:

1. Stel `current_only = 1` in `$SPLUNK_HOME\etc\system\local\inputs.conf in`:

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

2. Stop de expediteur
3. De naam van het oude controlepuntbestand wijzigen:

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

4. Start de expediteur
5. Zorg ervoor dat er een nieuw Security checkpoint-bestand wordt gegenereerd
6. Bevestig dat de checkpoint- en beveiligingslogs in een paar honderd gebeurtenissen van elkaar zijn opgeslagen
7. Wijzig de naam van het bestand `$SPLUNK_HOME\etc\system\local\inputs.conf` om de instelling `current_only=1` terug te zetten naar de standaard.
8. Splunk opnieuw starten
9. Zoeken om te bevestigen dat gegevens worden opgenomen

Waarom het uitvoeren van reguliere UF-reboots waarschijnlijk niet zal helpen

Het opnieuw opstarten van de UF dwingt hem eenvoudigweg om checkpoints en korte lege geheugenbuffers opnieuw te lezen. Omdat de gebeurtenisgeneratiesnelheid de verwerkings- / doorvoerlimiet overschrijdt, zal de forwarder onmiddellijk opnieuw vertraging beginnen op te bouwen binnen enkele minuten na de herstart.

Lange termijn oplossingen

De snelle oplossing vereist handmatige interventie op elke getroffen UF, wat over het algemeen niet haalbaar is in een grote implementatie (en sommige beheerders hebben geen opdrachtregeltoegang tot UF's) toegepast. Bovendien is het zeer waarschijnlijk dat de UF's weer achterop raken. De beste optie kan zijn om een oplossing op lange termijn te bieden. Omdat elke omgeving anders is, is er geen eenvoudige instelling die op elk scenario kan worden toegepast. De suggesties in deze KB moeten echter voor veel gebruiksgevallen werken. Alle configuratiesuggesties in dit artikel vereisen dat Splunk opnieuw wordt opgestart op de UF om van kracht te worden, dus zorg ervoor dat de aangepaste app op de implementatieserver Splunk opnieuw heeft opgestart.

Probeer dit eerst (limits.conf)

Open maxKBps

Voordat u de onderstaande opties aanpast, kijkt u naar de `limieten.conf/[thruput]maxKBps`. De standaardinstelling op een UF is 256KB/s, wat de doorzender beperkt tot ~22 GB/dag totale netwerktransmissie. Een drukke server die beveiligingslogs genereert, produceert gemakkelijk meer dan 256 KB / s, waardoor het fysiek onmogelijk is voor de expediteur om bij te blijven.

Dus, ten eerste: Stel deze waarde in op 0 (onbeperkt) in `limits.conf` om de netwerkbandbreedte te ontgrendelen:

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

Aangezien de meeste Universal Forwarders naar een Deployment Server bellen, is het de beste praktijk om deze configuratie via een implementatie-app in te stellen, zodat alle forwarders deze

tegelijk ontvangen (eerst op een testomgeving indien mogelijk, vervolgens in productie). Een andere optie als een testomgeving niet beschikbaar is, is om te implementeren naar een "test" serverklasse van een kleine groep UF's en monitor voordat u overal toepast. Hetzelfde geldt voor alle suggesties die in dit KB-artikel worden aangeboden.

Extra opties

Als het openen van de doorvoer van de expediteurs het probleem niet volledig oplost, overweeg dan het volgende:

Specifiek voor `inputs.conf/[WinEventLog://Security]`

Optie #1 - De batchgrootte wijzigen

Standaard grijpt Splunk gebeurtenissen in kleine batches (10 gebeurtenissen) en werkt het checkpoint-bestand regelmatig bij op de schijf. Op drukke hosts wordt schijf-I/O in het controlepuntbestand een belangrijk knelpunt. Door `batch_size = 500 in te stellen`, de schijfsynchronisatie uit te stellen tot 600 seconden (`checkpointInterval = 600`) en door `suppress_checkpoint = true in te stellen`, wordt de EPS-doorvoer van `splunk-winevtlog.exe` drastisch verhoogd.

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

Optie # 2 - Gebruik renderXML om het aantal API-oproepen te verminderen

Standaard geeft Splunk een API-aanroep uit voor elk veld binnen de gebeurtenis. Als u renderXML inschakelt, wordt in plaats daarvan een API-oproep per gebeurtenis uitgevoerd, waardoor de prestaties mogelijk worden verbeterd: [Windows Event Log-gebeurtenissen weergeven in XML]

```
renderXml = true
```

Optie #3 - Filter welke velden worden onderdrukt om de belasting van gegevens te verminderen

Deze optie is effectief voor ruwe snelheid, maar zorg ervoor dat deze voldoet aan de auditvereisten (indien van toepassing).

Met deze instellingen worden header-velden uit de raw-gebeurtenis verwijderd voordat ze worden verzonden om de bytegrootte en API-renderingsoproepen te verminderen. Als hun SIEM-, CIM-modellen of nalevingsdashboards afhankelijk zijn van `trefwoorden` (bijv. Audit Success / Failure), `TaskCategory` of `Opcode`, zal het laten vallen van deze velden deze detecties doorbreken.

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

Optie #4 - Schakel de **`FormatMessage()`** API in de invoerstanza in

`evtFormatMessage()` is de standaard API in de meeste Windows-implementaties, die een verhoogde verwerkingsoverhead heeft in vergelijking met de vorige FormatMessage() API. Terugkeren naar de oudere API kan de vertraging bij het indexeren verminderen.

Stel dit niet in als **`renderXML = true`**. De oudere `FormatMessage()` API is alleen sneller bij het parseren van klassieke platte tekst multi-line logs (`renderXML = false`), omdat het de zware `evtFormatMessage()` DLL string lookup omzeilt. Bij gebruik van `renderXml = true`, leest Splunk al rechtstreeks van de moderne EVTX API (`EvtRender`) in gestructureerde indeling zonder DLL's voor berichtopmaak aan te roepen.

```
use_old_eventlog_api = true
```

Verhoog UF Ingestion Pipelines (server.conf)

Standaard voert een forwarder een enkele parsing- en inslikpijpleiding uit. Voor multicore-machines die zwaar evenementenvolume opnemen, kan de pipelineset worden verhoogd om parallelle verwerking mogelijk te maken. Dit bepaalt hoeveel interne threads de forwarder gebruikt om gegevens van lokale Windows-API's te lezen, te verpakken en op het netwerk te duwen.

Op een UF is de voetafdruk van het verhogen van de inname pijpleidingen van `1` tot `2` zeer lichtgewicht in vergelijking met een Indexer of Heavy Forwarder, omdat de UF alleen lichtgewicht lezen, batchen en netwerk output doet (geen regex transformaties, veld extractie, of indexeren naar schijf).

Als de Windows-server ten minste 4 CPU-cores en 8 GB RAM heeft, is het instellen van `parallelle IngestionPipelines = 2` meestal veilig en helpt het aanzienlijk om een zware achterstand op te ruimen en toekomstige vertraging te voorkomen. Het CPU-gebruik kan toenemen terwijl de UF wordt ingehaald, maar zal opnieuw dalen zodra de inname terugkeert naar normale niveaus.

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

Uitgangen.conf wachtrijgrootte

Als u de invoerbatching aanpast tot 500, moet u ervoor zorgen dat de TCP-uitvoerwachtrij voldoende geheugenbuffer heeft, zodat de netwerklaag de invoerwerknemer niet blokkeert:

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

Controleer de uitsluitingen van Windows Defender/Antivirus

Security endpoint agents (bijvoorbeeld Defender, CrowdStrike, Carbon Black) kunnen `splunk-winevtlog.exe` agressief inspecteren en de directory `$SPLUNK_HOME\var\lib\splunk\modinput\WinEventLog\ vergrendelen`. Zorg ervoor dat de volledige `$SPLUNK_HOME` directory en `splunk-winevtlog.exe` zijn uitgesloten van real-time bestandssysteeminspectie.

Affiniteit met domeincontroller / netwerkroutering:

Als de UF's van het probleem worden uitgevoerd op domeincontrollers of een hoog Active Directory-verkeer verwerken, moet u ervoor zorgen dat de Splunk TCP-poort (standaard `9997`) niet wordt afgetopt op de ontvangende laag (HF's of indexers).

Gerelateerde links

- Splunk-documentatie:
  - Windows Eventlog-gegevens bewaken met Splunk Enterprise
  - Geavanceerde configuraties voor de Universal Forwarder
  - Specificatiebestanden: server.conf, inputs.conf, outputs.conf
- Splunk KB-artikelen
  - Problemen oplossen met vertraagde logbestanden van UF of HF
  - UF stuurt geen beveiligingslogboeken naar Splunk
- Splunk Lantern-artikel: Prestaties afstemmen van de forwarding tier

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.