Winevent 로그 이벤트 시간과 Splunk 인덱스 시간 사이에 인덱싱 지연이 보고되었습니다.

적용 가능성

Splunk Enterprise

설명

식별 방법

먼저 이벤트 시간(_time)과 Splunk indextime(_indextime) 사이의 델타를 확인하고 영향을 받는 인덱스와 소스 유형에 따라 다음을 수정합니다. 다음 예에서는 보안 채널 이벤트 로깅을 쿼리합니다.

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

조사

확인할 중요 사항:

- 인덱싱 지연은 언제 처음 나타납니까?
- 지연이 처음 나타날 무렵에 구축이 변경된 부분은 무엇입니까?
- 레이턴시는 시간이 지남에 따라 증가합니까(선형), 아니면 변동합니까?

최대 및 평균 eps(초당 이벤트 수)를 확인합니다.

```
index=_internal source=*metrics.log* security per_source_thruput host=<splunk_forwarder>
```

```
| timechart max(eps) max(max_age)
```

레이턴시가 시간에 따라 달라지는 경우, 높은 EPS 값은 데이터 처리량의 병목 현상을 나타낼 수 있습니다.

해결

빠른 수정

이 옵션은 현재 이벤트를 가져오고 누락된 기록 데이터를 신경 쓰지 않으려는 경우에만 적용됩니다. 각 문제 전달자에서 한 번에 하나씩 다음을 수행해야 합니다.

1. $SPLUNK_HOME\etc\system\local\inputs.conf에서 current_only = 1을 설정합니다.

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

2. 전달자 중지
3. 부실 검사점 파일의 이름을 바꿉니다.

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

4. 전달자 시작
5. 새 보안 검사점 파일이 생성되었는지 확인합니다.
6. 체크포인트 및 보안 로그가 서로 수백 개의 이벤트를 발생하는지 확인합니다.
7. $SPLUNK_HOME\etc\system\local\inputs.conf 파일의 이름을 변경하여 current_only=1 설정을 기본값으로 되돌립니다.
8. Splunk 다시 시작
9. 검색 - 데이터 수집 여부 확인

UF 정기 리부팅을 수행하는 것이 도움이 되지 않는 이유

UF를 재부팅하면 검사점을 다시 읽고 메모리 버퍼를 잠시 비우기만 하면 됩니다. 이벤트 생성 속도가 처리/처리량 제한을 초과하므로, 전달자는 다시 시작한 후 몇 분 내에 lag 누적을 즉시 다시 시작

합니다.

장기 솔루션

빠른 수정을 수행하려면 영향을 받는 각 UF에 대한 수동 개입이 필요하며, 이는 일반적으로 대규모 구축에서는 가능하지 않으며 일부 관리자는 UF에 대한 명령줄 액세스 권한이 없습니다. 여기에 UF가 또 뒤처질 가능성이 높다. 가장 좋은 방법은 장기적인 솔루션을 제공하는 것입니다. 모든 환경이 다르기 때문에 모든 시나리오에 적용할 수 있는 간단한 설정이 없습니다. 그러나 이 KB의 제안은 여러 활용 사례에 적용되어야 합니다. 이 문서의 모든 구성 제안을 적용하려면 UF에서 Splunk를 다시 시작해야 합니다. 따라서 배포 서버의 사용자 지정 앱에서 Splunkd를 다시 시작해야 합니다.

먼저 (limits.conf)를 사용해 보십시오.

최대 KBps 열기

아래 옵션을 변경하기 전에 limits.conf/[thruput]maxKBps를 살펴보십시오. UF의 기본 설정은 256KB/s이며, 이는 전달자를 최대 22GB/일 총 네트워크 전송으로 제한합니다. 보안 로그를 생성하는 사용량이 많은 서버에서 256KB/s 이상을 쉽게 생성하므로 전달자가 이를 따라잡을 수 없습니다.

그래서, 첫째로, limits.conf에서 이 값을 0(무제한)으로 설정하여 네트워크 대역폭의 캡처를 해제합니다.

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

대부분의 Universal Forwarder가 Deployment Server로 전화를 거는 경우 모범 사례는 모든 Forwarder가 한 번에 수신할 수 있도록 배포 앱을 통해 이 구성을 설정하는 것입니다(가능한 경우 먼저 테스트 환경에서, 그런 다음 프로덕션 환경에서). 테스트 환경을 사용할 수 없는 경우 다른 옵션은 UF 및 모니터링의 소규모 그룹인 "test" serverclass에 배포한 다음 모든 곳에 적용하는 것입니다. 이 참조 자료 문서에서 제공하는 모든 제안에도 동일하게 적용됩니다.

추가 옵션

전달자의 처리 과정을 열어도 문제가 완전히 해결되지 않을 경우 다음 사항을 고려하십시오.

inputs.conf/[WinEventLog://Security] 전용

옵션#1 - 배치 크기 변경

기본적으로 Splunk는 이벤트를 작은 배치(10개 이벤트)로 캡처하고 디스크의 체크포인트 파일을 자주 업데이트합니다. 사용 중인 호스트에서 체크포인트 파일의 디스크 I/O가 주요 초크 지점이 됩니다. batch_size를 500으로 설정하고, 디스크 동기화를 600s(checkpointInterval = 600)로 지연시키고, suppress_checkpoint = true로 설정하면 splunk-winevtlog.exe의 EPS(초당 이벤트) 처리량이 크게 증가합니다.

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

옵션#2 - renderXML을 사용하여 API 호출 수 감소

기본적으로 Splunk는 이벤트 내의 각 필드에 대해 API 호출을 실행합니다. renderXml을 활성화하면 대신 이벤트당 API 호출이 실행되어 성능이 향상될 수 있습니다. XML로 Windows 이벤트 로그 이벤트 표시

```
renderXml = true
```

옵션#3 - 데이터 페이로드를 줄이기 위해 표시하지 않는 필드를 필터링합니다.

이 옵션은 원시 속도에 효과적이지만 감사 요구 사항(해당되는 경우)을 충족해야 합니다.

이러한 설정은 바이트 크기 및 API 렌더링 호출을 줄이기 위해 전송하기 전에 원시 이벤트에서 헤더 필드를 삭제합니다. SIEM, CIM 모델 또는 규정 준수 대시보드가 키워드(예: Audit Success/Failure), TaskCategory 또는 Opcode를 사용하는 경우, 이러한 필드를 삭제하면 탐지가 중단됩니다.

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

옵션#4 - 입력 스탠자에서 **`FormatMessage()`** API를 활성화합니다.

`evtFormatMessage()`는 대부분의 Windows 배포에서 기본 API이며, 이전 FormatMessage() API와 비교했을 때 처리 오버헤드가 증가했습니다. 이전 API로 되돌리면 인덱싱 지연이 완화될 수 있습니다.

renderXml이 true인 경우 이 **값을 설정하지 마십시오**. 레거시 `FormatMessage() API`는 상위 evtFormatMessage() DLL 문자열 조회를 우회하므로 클래식 일반 텍스트 다중 라인 로그(renderXml `= false`)를 구문 분석할 때만 더 빠릅니다. `renderXml = true`를 사용할 때 Splunk는 메시지 형식 DLL을 호출하지 않고 구조화된 형식으로 최신 EVTX API(`EvtRender`)를 이미 직접 읽습니다.

```
use_old_eventlog_api = true
```

UF 수집 파이프라인 증가(server.conf)

기본적으로 전달자는 단일 구문 분석 및 수집 파이프라인을 실행합니다. 무거운 이벤트 볼륨을 수집하는 멀티코어 머신의 경우 파이프라인 세트를 늘려 병렬 처리를 허용할 수 있습니다. 이 명령은 전달자가 로컬 Windows API에서 데이터를 읽고 패키지한 다음 네트워크로 푸시하는 데 사용하는 내부 스레드의 수를 제어합니다.

UF에서는 `1`에서 `2`로 수집 파이프라인을 늘리는 리소스 공간이 인덱서 또는 헤비 포워더에 비해 매우 가볍습니다. UF는 경량형 읽기, 일괄 처리 및 네트워크 출력만 수행하기 때문입니다(regex 변환, 필드 추출 또는 디스크에 대한 인덱싱 없음).

Windows 서버에 CPU 코어가 4개 이상이고 RAM이 8GB 이상인 경우 parallelIngessionPipelines `=2`를 설정하면 대개 안전하며 과도한 이벤트 백로그를 지우고 향후 지연을 방지하는 데 크게 도움이 됩니다. UF가 가동되는 동안 CPU 사용량이 증가할 수 있지만, 섭취가 정상 수준으로 돌아가면 다시 감소합니다.

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

Outputs.conf 대기열 크기

입력 일괄 처리를 500으로 조정하는 경우 네트워크 레이어가 입력 작업자를 차단하지 않도록 TCP 출력 대기열에 충분한 메모리 버퍼가 있는지 확인하는 것이 좋습니다.

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

Windows Defender/Antivirus 제외 확인

보안 엔드포인트 에이전트(예: Defender, CrowdStrike, Carbon Black)는 `splunk-winevtlog.exe`를 적극적으로 검사하고 `$SPLUNK_HOME\var\lib\splunk\modinput\WinEventLog\` 디렉터리를 잠글 수 있습니다. 전체 `$SPLUNK_HOME` 디렉터리 및 `splunk-winevtlog.exe`가 실시간 파일 시스템 검사에서 제외되었는지 확인하십시오.

도메인 컨트롤러 선호도/네트워크 라우팅:

문제 UF가 도메인 컨트롤러에서 실행되거나 높은 Active Directory 트래픽을 처리하는 경우, 로컬 방화벽 또는 네트워크 속도 제한이 수신 계층(HF 또는 인덱서)에 대한 Splunk TCP 포트(기본값 `9997`)를 제한하지 않는지 확인합니다.

관련 링크

- Splunk 설명서:
  - Splunk Enterprise로 Windows 이벤트 로그 데이터 모니터링
  - 법용 전달자를 위한 고급 컨피그레이션
  - 사양 파일: server.conf, inputs.conf, outputs.conf
- Splunk KB 문서
  - UF 또는 HF에서 지연된 로그 트러블슈팅
  - UF가 Splunk에 보안 로그를 보내지 않음
- Splunk Lantern 기사: 포워딩 계층의 성능 튜닝

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.