Google Cloud Docker에서 Secure Access Resource Connector 구축 및 연결 문제 해결

목차

문제

Docker에서 Secure Access Resource Connector를 배포하지 못했습니다.

커넥터가 올바르게 설치되었지만 Cisco Secure Access에 연결할 수 없습니다.

진단 검사에서 보고된 터널 연결 끊기 및 서버 통신 오류를 확인합니다.

이 환경에서는 Google Cloud에서 호스팅되고 "any any" 규칙이 적용된 Fortinet 방화벽을 통해 연결된 Red Hat 9 가상 머신을 사용합니다.

트러블슈팅으로 인해 네트워크 인터페이스 간의 잠재적 MTU 불일치가 기여 요인으로 드러났습니다.

환경

- 기술: 솔루션 지원(SSPT - 계약 필요)
- 하위 기술: 보안 액세스 - 리소스 커넥터(설치, 업그레이드, 등록, 연결, 전용 리소스)
- 플랫폼: Google Cloud의 Red Hat 9 Virtual Machines
- 네트워크: Secure Access와 VM 간의 Fortinet 방화벽("any" 규칙 적용)
- 커넥터 영역: iuvz83r.mxc1.acgw.sse.cisco.com
- Google Cloud VPC 기본 MTU: 1460바이트
- Docker bridge(docker0) 기본 MTU: 1500바이트(변경 전)
- VM당 단일 네트워크 인터페이스(eth0)

해결

Docker/Google Cloud 환경에서 Secure Access Resource Connector 연결 문제를 진단하고 해결하려면 다음 단계를 수행합니다.

커넥터 영역에 대한 DNS 확인

nslookup을 사용하여 VM에서 보안 액세스 영역을 확인할 수 있는지 확인합니다.

```
nslookup iuvz83r.mxc1.acgw.sse.cisco.com
```

출력 예:

```
Server:         64.102.6.247
Address:        64.102.6.247#53
Non-authoritative answer:
Name:   iuvz83r.mxc1.acgw.sse.cisco.com
Address: 163.129.128.72
Name:   iuvz83r.mxc1.acgw.sse.cisco.com
Address: 163.129.128.70
Name:   iuvz83r.mxc1.acgw.sse.cisco.com
Address: 163.129.128.66
Name:   iuvz83r.mxc1.acgw.sse.cisco.com
Address: 163.129.128.68
```

보안 액세스에 대한 네트워크 연결 확인

ping 및 `telnet`을 사용하여 VM에서 보안 액세스에 대한 연결을 검증합니다.

```
ping iuvz83r.mxc1.acgw.sse.cisco.com
```

출력 예:

```
PING iuvz83r.mxc1.acgw.sse.cisco.com (163.129.128.66) 56(84) bytes of data.
64 bytes from 163.129.128.66: icmp_seq=1 ttl=57 time=44.7 ms
64 bytes from 163.129.128.66: icmp_seq=2 ttl=57 time=43.8 ms
...
telnet iuvz83r.mxc1.acgw.sse.cisco.com 443
```

출력 예:

```
Trying 163.129.128.66...
Connected to iuvz83r.mxc1.acgw.sse.cisco.com.
Escape character is '^]'.
```

터널 연결을 확인하고 진단 실행

커넥터 진단 유틸리티를 실행하여 터널 상태를 확인합니다.

```
/opt/connector/data/bin/diagnostic
```

출력 예:

```
###check tunnel connection:
error: tunnel is not connected
```

네트워크 인터페이스 및 MTU 설정 확인

ifconfig 및 ip a를 사용하여 모든 인터페이스의 IP 주소 및 MTU를 확인합니다.

```
ifconfig
ip a
```

eth0 및 docker0의 출력 예:

```
[root@degcpprcra02 ~]# ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet x.x.x.x netmask x.x.x.x broadcast x.x.x.x
inet6 fe80::1c66:46ff:fe1d:8bed prefixlen 64 scopeid 0x20<link>
ether 1e:66:46:1d:8b:ed txqueuelen 0 (Ethernet)
RX packets 974 bytes 119775 (116.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 848 bytes 161554 (157.7 KiB)
TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
inet x.x.x.x netmask x.x.x.x broadcast 0.0.0.0
ether 42:01:c0:a8:80:b0 txqueuelen 1000 (Ethernet)
RX packets 20175 bytes 7755728 (7.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 21550 bytes 31402300 (29.9 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

TCP 트래픽이 캡처되었는지 확인

tcpdump를 사용하여 VM과 보안 액세스 영역 간의 트래픽을 캡처합니다.

```
tcpdump -i eth0 host iuvz83r.mxc1.acgw.sse.cisco.com
```

출력 예(캡처된 패킷 없음 표시):

```
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
6 packets received by filter
0 packets dropped by kernel
```

필요한 경우 커넥터 제거 및 재설치

진단 및 기술 지원이 작동하지 않을 경우 커넥터를 중지하고 제거하십시오.

```
/opt/connector/install/connector.sh stop --destroy
cd /opt
rm -rf connector
```

커넥터를 다시 설치하고 기술 지원 출력 생성

재설치 후 기술 지원을 생성하여 오류 로그를 캡처합니다.

```
/opt/connector/data/bin/techsupport > techsupport.txt
Sample output showing connection errors:
2026-02-13 23:48:20.398772500  >> warning: Connection attempt has failed.
2026-02-13 23:48:20.398775500  >> warning: Unable to contact iuvz83r.mxc1.acgw.sse.cisco.com.
2026-02-13 23:48:20.398775500  >> error: Connection attempt has failed due to server communication erro
2026-02-13 23:48:20.398887500  >> state: Disconnected
```

Google Cloud VPC 및 VM 인터페이스와 일치하도록 Docker MTU 조정

Docker 브리지 인터페이스의 MTU를 Google Cloud VPC 기본값(1460바이트)과 일치하도록 변경합니다.

```
ip link set dev docker0 mtu 1460
```

MTU 변경 확인:

```
ip a
```

출력 예:

```
docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc noqueue state UP group default
link/ether 1e:66:46:1d:8b:ed brd ff:ff:ff:ff:ff:ff
inet x.x.x.x brd x.x.x.x scope global docker0
    valid_lft forever preferred_lft forever
inet6 fe80::1c66:46ff:fe1d:8bed/64 scope link
    valid_lft forever preferred_lft forever
```

/etc/docker/daemon.json에서 Docker MTU 변경 유지

/etc/docker/daemon.json를 편집하고 mtu 값을 추가하거나 업데이트합니다.

```
{
  ...
  "mtu": 1460
}
```

VM을 다시 시작하여 MTU 컨피그레이션 적용

MTU 설정이 완전히 적용되도록 전체 VM을 다시 시작합니다. Docker 서비스만 다시 시작하면 모든 네트워킹 구성 요소에 대해 MTU 변경 사항이 적용되지 않을 수 있으므로 이 작업이 필요합니다.

이 단계를 수행한 후 Secure Access에 대한 연결이 성공적으로 설정되었으며 컨피그레이션을 완료할 수 있습니다.

원인

근본 원인은 Docker 브리지 인터페이스(docker0)와 Google Cloud VPC/VM 네트워크 인터페이스(eth0) 간의 MTU 불일치입니다. Google Cloud VPC 및 VM 인터페이스의 기본 MTU는 1460바이트인 반면, Docker 기본 MTU는 1500바이트입니다.

이 불일치로 인해 조각화되거나 삭제된 패킷이 발생하여 Secure Access Resource Connector에서 터널을 설정하지 못했습니다. MTU 값을 정렬하면 연결 문제가 해결되었습니다.

관련 콘텐츠

- https://securitydocs.cisco.com/docs/csa/olh/120695.dita
- https://securitydocs.cisco.com/docs/csa/olh/120776.dita
- https://securitydocs.cisco.com/docs/csa/olh/120727.dita
- https://securitydocs.cisco.com/docs/csa/olh/120772.dita
- https://securitydocs.cisco.com/docs/csa/olh/120762.dita
- https://securitydocs.cisco.com/docs/csa/olh/120685.dita
- Cisco 기술 지원 및 다운로드

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.