

Problemen oplossen met de implementatie en connectiviteit van de Secure Access Resource Connector op Google Cloud Docker

Inhoud

uitgeven

Pogingen om de Secure Access Resource Connector op Docker te implementeren zijn mislukt.

Hoewel de connector correct is geïnstalleerd, kon geen verbinding met Cisco Secure Access worden gemaakt.

Diagnostische controles meldden tunnelverbreking en fouten in de servercommunicatie.

De omgeving maakt gebruik van Red Hat 9 virtuele machines gehost in Google Cloud, verbonden via een Fortinet-firewall met een "any" -regel.

Problemen oplossen onthulde potentiële MTU-mismatches tussen netwerkinterfaces als een bijdragende factor.

milieu

- Technologie: oplossingsondersteuning (SSPT - contract vereist)
- Subtechnologie: Veilige toegang - Bronaansluiting (installeren, upgraden, registreren, connectiviteit, Privé-bron)
- Red Hat 9 Virtual Machines op Google Cloud
- Netwerk: Fortinet-firewall tussen Secure Access en VM ("elke" regel is van kracht)
- Connectorregio: iuvz83r.mxc1.acgw.sse.cisco.com
- Google Cloud VPC standaard MTU: 1460 bytes
- Docker bridge (docker0) standaard MTU: 1500 bytes (vóór wijziging)
- Eén netwerkinterface (eth0) per VM

resolutie

Volg deze stappen om connectiviteitsproblemen met Secure Access Resource Connector in een Docker/Google Cloud-omgeving te diagnosticeren en op te lossen:

Controleer de DNS-resolutie voor het connectorgebied

Gebruik `nslookup` om te bevestigen dat het beveiligde toegangsgebied kan worden opgelost vanaf

de VM.

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

Voorbeeld van uitvoer:

```
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
```

Controleer de netwerkconnectiviteit voor veilige toegang

Gebruik `ping` en `telnet` om de connectiviteit met Secure Access van de VM te valideren.

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

Voorbeeld van uitvoer:

```
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
```

Voorbeeld van uitvoer:

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

Controleren op tunnelconnectiviteit en diagnostiek uitvoeren

Voer het diagnosehulpprogramma van de connector uit om de tunnelstatus te controleren.

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

Voorbeeld van uitvoer:

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

Netwerkitinterface en MTU-instellingen controleren

Controleer IP-adressen en MTU van alle interfaces met behulp van `ifconfig` en een `ip`.

```
ifconfig  
ip a
```

Voorbeelduitvoer voor `eth0` en `docker0`:

```
[root@degcprcra02 ~]# 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
```

Controleer of TCP-verkeer is vastgelegd

Gebruik `tcpdump` om verkeer tussen de VM en de Secure Access-regio vast te leggen.

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

Voorbeeld uitvoer (geen vastgelegde pakketten weergeven):

```
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
```

Vernietig en installeer de connector indien nodig opnieuw

Stop en vernietig de connector als de diagnostiek en technische ondersteuning niet werken:

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

Installeer de connector opnieuw en genereer de uitvoer voor technische support

Genereer na het opnieuw installeren technische ondersteuning om foutlogboeken vast te leggen:

```
/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 error
2026-02-13 23:48:20.398887500 >> state: Disconnected
```

Docker MTU aanpassen aan Google Cloud VPC en VM-interface

Wijzig de MTU op de Docker Bridge-interface om overeen te komen met de Google Cloud VPC-standaard (1460 bytes):

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

MTU-wijziging controleren:

ip a

Voorbeeld van uitvoer:

```
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
```

Aanhoudende verandering van Docker MTU in `/etc/docker/daemon.json`

Bewerk `/etc/docker/daemon.json` en voeg de `mtu`-waarde toe of werk deze bij:

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

Start de VM opnieuw op om de MTU-configuratie toe te passen

Start de volledige VM opnieuw op om ervoor te zorgen dat de MTU-instellingen volledig worden toegepast. Dit is noodzakelijk, omdat het mogelijk is dat alleen het opnieuw opstarten van de Docker-service de MTU-wijziging niet afdwingt voor alle netwerkcomponenten.

Na het volgen van deze stappen kon de verbinding met Secure Access tot stand worden gebracht en kon de configuratie worden voltooid.

Oorzaak

De hoofdoorzaak was een MTU-mismatch tussen de Docker-bruginterface (`docker0`) en de Google Cloud VPC/VM-netwerkinterface (`eth0`). De Google Cloud VPC en VM interfaces standaard een MTU van 1460 bytes, terwijl de Docker standaard MTU is 1500 bytes.

Deze mismatch veroorzaakte fragmentatie of gedropte pakketten, waardoor de Secure Access Resource Connector geen tunnel kon opzetten. Het uitlijnen van de MTU-waarden loste het connectiviteitsprobleem op.

Verwante inhoud

- <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 Technical Support en downloads](#)

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.