Risoluzione dei problemi relativi alla distribuzione e alla connettività di Secure Access Resource Connector in Google Cloud Docker

Sommario

Problema

Tentativi di distribuzione di Secure Access Resource Connector nel Docker non riusciti.

Sebbene il connettore sia installato correttamente, non è stato possibile stabilire la connettività a Cisco Secure Access.

I controlli di diagnostica hanno segnalato la disconnessione del tunnel e errori di comunicazione del server.

L'ambiente utilizza macchine virtuali Red Hat 9 ospitate in Google Cloud, connesse tramite un firewall Fortinet con una regola "any any".

La risoluzione dei problemi ha rivelato potenziali mancata corrispondenza dell'MTU tra le interfacce di rete come fattore che contribuisce.

Ambiente

- Tecnologia: supporto per la soluzione (SPPT - contratto obbligatorio)
- Sottotecnologia: accesso sicuro - connettore risorse (installazione, aggiornamento, registrazione, connettività, risorse private)
- Piattaforma: Red Hat 9 Virtual Machines su Google Cloud
- Rete: firewall tra Secure Access e la VM (regola "any any" in place)
- Area connettore: iuvz83r.mxc1.acgw.sse.cisco.com
- MTU predefinita Google Cloud VPC: 1460 byte
- MTU predefinita Docker Bridge (docker0): 1500 byte (prima della modifica)
- Interfaccia di rete singola (eth0) per VM

Risoluzione

Per diagnosticare e risolvere i problemi di connettività di Secure Access Resource Connector in un ambiente Docker/Google Cloud, attenersi alla procedura seguente:

Verifica risoluzione DNS per l'area del connettore

Utilizzare `nslookup` per confermare che l'area Accesso sicuro può essere risolta dalla macchina virtuale.

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

Output di esempio:

```
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
```

Verifica della connettività di rete per l'accesso protetto

Utilizzare `ping` e `telnet` per convalidare la connettività per l'accesso sicuro dalla macchina virtuale.

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

Output di esempio:

```
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
```

Output di esempio:

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

Verifica connettività tunnel ed esegui diagnostica

Eseguire l'utilità di diagnostica del connettore per controllare lo stato del tunnel.

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

Output di esempio:

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

Verifica delle impostazioni dell'interfaccia di rete e dell'MTU

Controllare gli indirizzi IP e la MTU di tutte le interfacce con `ifconfig` e `ip a`.

```
ifconfig
ip a
```

Output di esempio per eth0 e docker0:

```
[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
```

Controlla se il traffico TCP è stato acquisito

Utilizzare `tcpdump` per acquisire il traffico tra la VM e l'area Accesso sicuro.

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

Output di esempio (nessun pacchetto acquisito):

```
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
```

Eliminare e reinstallare il connettore, se necessario

Arrestare e distruggere il connettore se la diagnostica e il supporto tecnico non funzionano:

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

Reinstallare il connettore e generare l'output del supporto tecnico

Dopo la reinstallazione, generare il supporto tecnico per acquisire i log degli errori:

```
/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
```

Regolazione dell'MTU del docker in base all'interfaccia VPC e VM di Google Cloud

Modificare l'MTU sull'interfaccia del bridge Docker in modo che corrisponda all'impostazione predefinita di Google Cloud VPC (1460 byte):

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

Verificare la modifica dell'MTU:

```
ip a
```

Output di esempio:

```
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
```

Rendi persistente la modifica dell'MTU del Docker in **`/etc/docker/daemon.json`**

Modificare `/etc/docker/daemon.json` e aggiungere o aggiornare il valore `mtu`:

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

Riavviare la VM per applicare la configurazione MTU

Riavviare la macchina virtuale completa per verificare che le impostazioni MTU siano state applicate completamente. Questa operazione è necessaria perché è possibile che il riavvio solo del servizio Docker non applichi la modifica MTU per tutti i componenti di rete.

Dopo aver eseguito questi passaggi, è stata stabilita la connettività ad Accesso sicuro ed è stato possibile completare la configurazione.

Causa

La causa principale è una mancata corrispondenza della MTU tra l'interfaccia del bridge Docker (docker0) e l'interfaccia di rete VPC/VM di Google Cloud (eth0). L'impostazione predefinita delle interfacce VPC e VM di Google Cloud è una MTU di 1460 byte, mentre l'MTU predefinita del Docker è di 1500 byte.

Questa mancata corrispondenza ha causato la frammentazione o l'eliminazione di pacchetti, impedendo al connettore delle risorse di accesso sicuro di stabilire un tunnel. L'allineamento dei valori MTU ha risolto il problema di connettività.

Contenuto correlato

- 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
- Supporto tecnico Cisco e download

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).