Solucionar problemas de implantação e conectividade do conector de recursos de acesso seguro no Google Cloud Docker

Contents

Problema

As tentativas de implantar o Conector de Recursos de Acesso Seguro no Docker não tiveram êxito.

Embora o conector tenha sido instalado corretamente, não foi possível estabelecer a conectividade com o Cisco Secure Access.

As verificações de diagnóstico relataram erros de desconexão de túnel e de comunicação do servidor.

O ambiente usa máquinas virtuais Red Hat 9 hospedadas no Google Cloud, conectadas através de um firewall Fortinet com uma regra "any any".

A identificação e solução de problemas revelou possíveis incompatibilidades de MTU entre as interfaces de rede como um fator contribuinte.

Ambiente

- Tecnologia: suporte à solução (SSPT - contrato necessário)
- Subtecnologia: Acesso seguro - Conector de recursos (instalação, atualização, registro, conectividade, recurso privado)
- Plataforma: Red Hat 9 Máquinas Virtuais no Google Cloud
- Rede: firewall Fortinet entre acesso seguro e VM (qualquer regra "qualquer um" em vigor)
- Região do Conector: iuvz83r.mxc1.acgw.sse.cisco.com
- MTU padrão do VPC do Google Cloud: 1460 bytes
- MTU padrão da Docker Bridge (docker0): 1500 bytes (antes da alteração)
- Interface de rede única (eth0) por VM

Resolução

Siga estas etapas para diagnosticar e resolver problemas de conectividade do Secure Access Resource Connector em um ambiente de nuvem Docker/Google:

Verificar a Resolução DNS para a Região do Conector

Use `nslookup` para confirmar se a região de Acesso Seguro pode ser resolvida na VM.

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

Saída de exemplo:

```
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
```

Verificar a conectividade de rede para acesso seguro

Use `ping` e `telnet` para validar a conectividade para acesso seguro da VM.

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

Saída de exemplo:

```
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
```

Saída de exemplo:

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

Verifique a conectividade do túnel e execute o diagnóstico

Execute o utilitário de diagnóstico do conector para verificar o status do túnel.

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

Saída de exemplo:

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

Verifique a interface de rede e as configurações de MTU

Verifique os endereços IP e a MTU de todas as interfaces usando `ifconfig` e `ip a`.

```
ifconfig
ip a
```

Exemplo de saída para 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
```

Verificar se o tráfego TCP é capturado

Use `tcpdump` para capturar o tráfego entre a VM e a região Secure Access.

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

Exemplo de saída (não mostrando pacotes capturados):

```
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
```

Destrua e reinstale o conector, se necessário

Pare e destrua o conector se o diagnóstico e o suporte técnico não estiverem funcionando:

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

Reinstale o conector e gere saída de suporte técnico

Após a reinstalação, gere suporte técnico para capturar logs de erros:

```
/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
```

Ajustar o MTU do Docker para corresponder ao VPC da nuvem do Google e à interface da VM

Altere o MTU na interface da ponte do Docker para corresponder ao padrão do VPC do Google Cloud (1460 bytes):

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

Verificar alteração de MTU:

```
ip a
```

Saída de exemplo:

```
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
```

Persistir Alteração de MTU do Docker em **`/etc/docker/daemon.json`**

Edite `/etc/docker/daemon.json` e adicione ou atualize o valor `mtu`:

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

Reinicie a VM para aplicar a configuração de MTU

Reinicie a VM completa para garantir que as configurações de MTU sejam totalmente aplicadas. Isso é necessário, pois é possível que reiniciar apenas o serviço Docker não imponha a alteração de MTU para todos os componentes de rede.

Depois de seguir essas etapas, a conectividade com o Acesso seguro foi estabelecida com êxito e a configuração pôde ser concluída.

Causa

A causa raiz foi uma incompatibilidade de MTU entre a interface de ponte do Docker (docker0) e a interface de rede VPC/VM do Google Cloud (eth0). As interfaces VPC e VM do Google Cloud assumem como padrão um MTU de 1460 bytes, enquanto o MTU padrão do Docker é de 1500 bytes.

Essa incompatibilidade causou fragmentação ou pacotes descartados, impedindo que o Conector de Recursos de Acesso Seguro estabelecesse um túnel. O alinhamento dos valores de MTU resolveu o problema de conectividade.

Conteúdo relacionado

- 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
- Suporte técnico e downloads da Cisco

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.