Resolución de problemas de implementación y conectividad del conector de recursos de acceso seguro en Google Cloud Docker

Contenido

Problema

No se pudo implementar el conector de recursos de acceso seguro en Docker.

Aunque el conector se instaló correctamente, no se pudo establecer la conectividad con Cisco Secure Access.

Las comprobaciones de diagnóstico notificaron la desconexión del túnel y los errores de comunicación del servidor.

El entorno utiliza máquinas virtuales Red Hat 9 alojadas en Google Cloud, conectadas a través de un firewall Fortinet con una regla "any any".

La resolución de problemas reveló posibles discrepancias de MTU entre las interfaces de red como un factor que contribuye.

Entorno

- Tecnología: Asistencia para soluciones (SSPT, contrato necesario)
- Subtecnología: Acceso seguro - Conector de recursos (instalación, actualización, registro, conectividad, recurso privado)
- Plataforma: Red Hat 9 máquinas virtuales en Google Cloud
- Red: firewall Fortinet entre Secure Access y VM (regla "any any" en vigor)
- Región del conector: iuvz83r.mxc1.acgw.sse.cisco.com
- MTU predeterminada de VPC de Google Cloud: 1460 bytes
- Docker bridge (docker0) MTU predeterminada: 1500 bytes (antes del cambio)
- Interfaz de red única (eth0) por VM

Resolución

Siga estos pasos para diagnosticar y resolver problemas de conectividad del conector de recursos de acceso seguro en un entorno de Docker/Google Cloud:

Comprobar la resolución DNS para la región del conector

Utilice `nslookup` para confirmar que la región Secure Access se puede resolver desde la VM.

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

Ejemplo de salida:

```
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
```

Compruebe la conectividad de red para un acceso seguro

Utilice `ping` y `telnet` para validar la conectividad a Secure Access desde la máquina virtual.

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

Ejemplo de salida:

```
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
```

Ejemplo de salida:

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

Comprobar la conectividad del túnel y ejecutar diagnósticos

Ejecute la utilidad de diagnóstico del conector para comprobar el estado del túnel.

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

Ejemplo de salida:

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

Verificar la interfaz de red y la configuración de MTU

Verifique las direcciones IP y la MTU de todas las interfaces mediante `ifconfig` e `ip a`.

```
ifconfig
ip a
```

Ejemplo de salida para eth0 y 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
```

Comprobar si se captura el tráfico TCP

Utilice `tcpdump` para capturar el tráfico entre la VM y la región Secure Access.

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

Ejemplo de salida (que muestra que no se capturaron paquetes):

```
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
```

Destruya y vuelva a instalar el conector si es necesario

Detenga y destruya el conector si los diagnósticos y el soporte técnico no funcionan:

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

Vuelva a instalar el conector y genere los resultados de la asistencia técnica

Después de la reinstalación, genere soporte técnico para capturar los registros de errores:

```
/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
```

Ajuste la MTU de Docker para que coincida con la interfaz de VM y VPC de Google Cloud

Cambie la MTU en la interfaz de Docker Bridge para que coincida con el valor predeterminado de VPC de Google Cloud (1460 bytes):

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

Verificar cambio de MTU:

```
ip a
```

Ejemplo de salida:

```
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
```

Persist Docker MTU Change in **`/etc/docker/daemon.json`**

Edite `/etc/docker/daemon.json` y agregue o actualice el valor de `mtu`:

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

Reinicie la máquina virtual para aplicar la configuración de MTU

Reinicie la máquina virtual completa para asegurarse de que la configuración de MTU se aplica por completo. Esto es necesario, porque es posible que al reiniciar sólo el servicio Docker no se aplique el cambio de MTU para todos los componentes de la red.

Después de seguir estos pasos, la conectividad con Secure Access se estableció correctamente y la configuración pudo completarse.

Causa

La causa raíz fue una discordancia de MTU entre la interfaz de puente Docker (docker0) y la interfaz de red VPC/VM de Google Cloud (eth0). Las interfaces VPC y VM de Google Cloud tienen una MTU predeterminada de 1460 bytes, mientras que la MTU predeterminada de Docker es de 1500 bytes.

Esta discordancia causó fragmentación o paquetes perdidos, impidiendo que el conector de recursos de acceso seguro establezca un túnel. La alineación de los valores de MTU resolvió el problema de conectividad.

Contenido 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
- Soporte técnico y descargas de Cisco

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).