Google Cloud Dockerでのセキュアアクセスリソースコネクタの導入と接続のトラブルシューティング

内容

お問い合わせ内容

DockerにSecure Access Resource Connectorを展開しようとしましたが、失敗しました。

コネクタは正しくインストールされましたが、Cisco Secure Accessへの接続を確立できませんでした。

診断チェックでは、トンネルの接続解除とサーバ通信エラーが報告されました。

この環境では、Google Cloudでホストされ、Fortinetファイアウォールを介して「any any」ルールで接続されたRed Hat 9仮想マシンを使用します。

トラブルシューティングの結果、ネットワークインターフェイス間の潜在的なMTUの不一致が原因であることが判明しました。

環境

- テクノロジー：ソリューションサポート（SSPT – 契約が必要）
- サブテクノロジー：セキュアアクセス – リソースコネクタ（インストール、アップグレード、登録、接続、プライベートリソース）
- プラットフォーム：Google Cloud上のRed Hat 9仮想マシン
- ネットワーク：Secure AccessとVM間のFortinetファイアウォール（「any any」ルールが適用）
- コネクタ領域： iuvz83r.mxc1.acgw.sse.cisco.com
- Google Cloud VPCデフォルトMTU:1460バイト
- Docker bridge(docker0)デフォルトMTU:1500バイト（変更前）
- VMごとに1つのネットワークインターフェイス(eth0)

解決策

次の手順に従って、Docker/Google Cloud環境でのSecure Access Resource Connectorの接続の問題を診断して解決します。

コネクタ領域のDNS解決の確認

nslookupを使用して、セキュアアクセス領域をVMから解決できることを確認します。

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

出力例：

```
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
```

セキュアなアクセスのためのネットワーク接続の確認

pingとtelnetを使用して、VMからセキュアアクセスへの接続を確認します。

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

出力例：

```
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
```

出力例：

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

トンネル接続の確認と診断の実行

コネクタ診断ユーティリティを実行して、トンネルステータスを確認します。

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

出力例：

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

ネットワークインターフェイスとMTU設定の確認

`ifconfig`と`ip a`を使用して、すべてのインターフェイスのIPアドレスとMTUを確認します。

```
ifconfig
ip a
```

eth0およびdocker0の出力例：

```
[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トラフィックがキャプチャされたかどうかの確認

`tcpdump`を使用して、VMとセキュアアクセスリージョン間のトラフィックをキャプチャします。

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

出力例 (キャプチャされたパケットがないことを示す) :

```
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
```

必要に応じてコネクタを破棄して再インストールする

DiagnosticsとtechsupportがDiagnosticsとtechsupportが機能しない場合は、コネクタを停止して破棄します。

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

コネクタを再インストールし、テクニカルサポートの出力を生成する

再インストール後、エラーログをキャプチャするテクニカルサポートを生成します。

```
/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およびVMインターフェイスに合わせたDocker MTUの調整

Google Cloud VPCのデフォルト (1460バイト) に一致するように、Dockerブリッジインターフェイスのを変更します。

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

MTUの変更を確認します。

```
ip a
```

出力例：

```
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.jsonでのDocker MTU変更の永続化

/etc/docker/daemon.jsonを編集し、mtu値を追加または更新します。

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

MTU設定を適用するためのVMの再起動

MTU設定が完全に適用されるように、VM全体を再起動します。 Dockerサービスだけを再起動しても、すべてのネットワークコンポーネントに対して最大伝送ユニットの変更が強制されるわけではないため、これが必要になります。

これらの手順を実行すると、セキュアアクセスへの接続が正常に確立され、設定を完了できます。

原因

根本的な原因は、Dockerブリッジインターフェイス(docker0)とGoogle Cloud VPC/VMネットワークインターフェイス(eth0)の間のMTUの不一致でした。 Google Cloud VPCおよびVMインターフェイスのデフォルトのMTUは1460バイトですが、DockerのデフォルトMTUは1500バイトです。

この不一致により、パケットの断片化または破棄が発生し、Secure Access Resource Connectorによるトンネルの確立が妨げられました。 MTU値を揃えることで、接続の問題は解決しました。

関連コンテンツ

- 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
- シスコのテクニカルサポートとダウンロード

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。