

Troubleshoot Secure Access Roaming Module "Cloud Service Unavailable" or "Unprotected" Status

Contents

[Introduction](#)

[Problem](#)

[DNS Protection Status is Unprotected](#)

[Web Protection Status is Cloud Service Unavailable](#)

[Solution](#)

[Related Information](#)

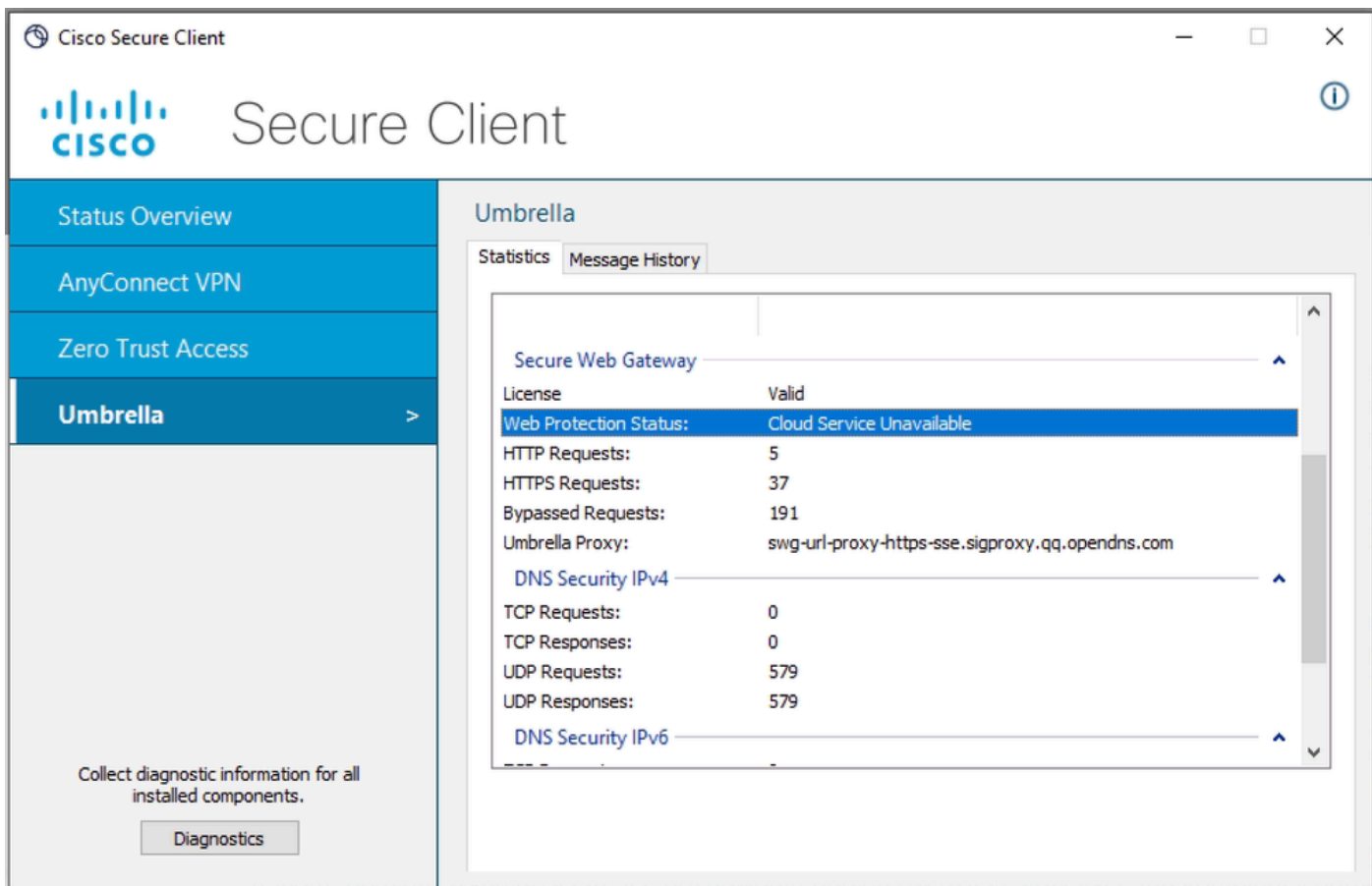
Introduction

This document describes a way to investigate root cause of status "Cloud Service Unavailable" or "Unprotected" in Roaming Module of Secure Client.

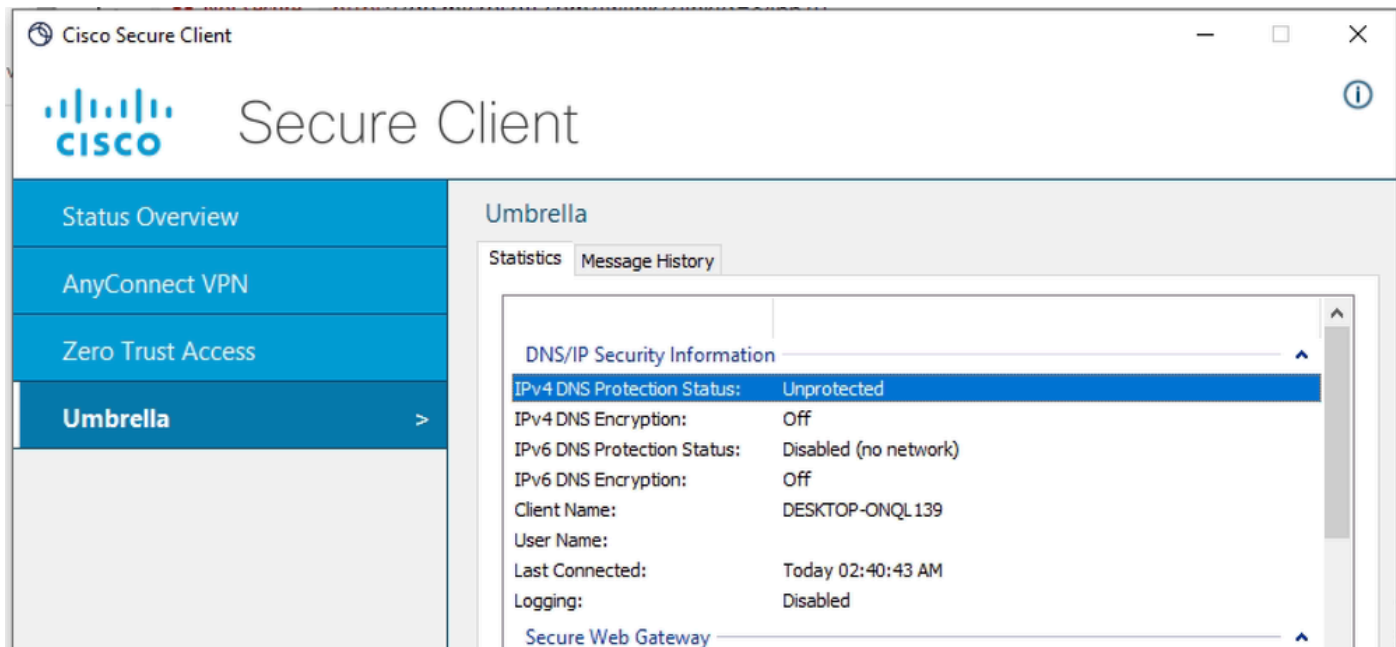
Problem

When a user launches Roaming Module of Secure Client and expects to use DNS and/or Web protection, erroneous states can be seen in Secure Client User Interface:

Cloud Service Unavailable for Web Protection Status



Unprotected for DNS Protection Status



The reason behind those errors is that Roaming Module cannot contact its cloud services due to network connectivity issues.

If this problem was not seen on affected client PC in the past, it means that most probably network that PC is connected to is restricted and does not meet requirements outlined in [SSE Documentation](#)

DNS Protection Status is Unprotected

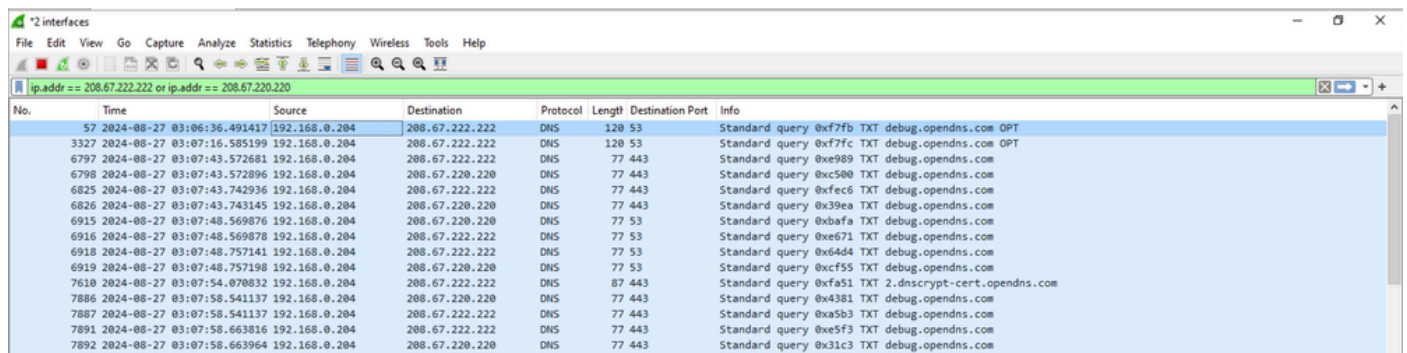
When you see Unprotected DNS state, then most probably Roaming Module does not have upstream connectivity to OpenDNS servers (**208.67.222.222** and **208.67.220.220**).

You would see the log in **cscombrellaplugin.txt** file, which is part of **DART** bundle.

```
2024-08-27 03:07:43 [8880] [DEBUG] < 12> Dns Protection IPv4 State Machine: checking reachability of pr
2024-08-27 03:07:43 [8880] [DEBUG] < 12> Dns Protection IPv4 State Machine: probing for OpenDNS resolve
2024-08-27 03:07:43 [8880] [DEBUG] < 13> Dns Protection IPv6 State Machine: rejected all candidate reso
2024-08-27 03:07:48 [8880] [DEBUG] < 12> Dns Protection IPv4 State Machine: checking reachability of pr
2024-08-27 03:07:48 [8880] [DEBUG] < 12> Dns Protection IPv4 State Machine: probing for OpenDNS resolve
2024-08-27 03:07:53 [8880] [DEBUG] < 12> Dns Protection IPv4 State Machine: rejected all candidate reso
```

In order to double check and confirm connectivity issues you can collect wireshark capture on egress physical interface of the PC (WiFi or Ethernet), and use the display filter to look only for traffic destined to OpenDNS resolvers:

```
ip.addr == 208.67.222.222 or ip.addr == 208.67.220.220
```



The image shows a Wireshark capture window with the display filter 'ip.addr == 208.67.222.222 or ip.addr == 208.67.220.220'. The capture shows a series of DNS queries from source IP 192.168.0.204 to destination IP 208.67.222.222 and 208.67.220.220. The queries are for TXT records for 'debug.opendns.com' and '2.dnscrypt-cert.opendns.com'. The queries are sent to UDP ports 53 and 443. No responses are visible in the capture.

No.	Time	Source	Destination	Protocol	Length	Destination Port	Info
57	2024-08-27 03:06:36.491417	192.168.0.204	208.67.222.222	DNS	120	53	Standard query 0xf7fb TXT debug.opendns.com OPT
3327	2024-08-27 03:07:16.585199	192.168.0.204	208.67.222.222	DNS	120	53	Standard query 0xf7fc TXT debug.opendns.com OPT
6797	2024-08-27 03:07:43.572681	192.168.0.204	208.67.222.222	DNS	77	443	Standard query 0xe989 TXT debug.opendns.com
6798	2024-08-27 03:07:43.572896	192.168.0.204	208.67.220.220	DNS	77	443	Standard query 0xc508 TXT debug.opendns.com
6825	2024-08-27 03:07:43.742936	192.168.0.204	208.67.222.222	DNS	77	443	Standard query 0xfec6 TXT debug.opendns.com
6826	2024-08-27 03:07:43.743145	192.168.0.204	208.67.220.220	DNS	77	443	Standard query 0x39ea TXT debug.opendns.com
6915	2024-08-27 03:07:48.569876	192.168.0.204	208.67.220.220	DNS	77	53	Standard query 0xbafa TXT debug.opendns.com
6916	2024-08-27 03:07:48.569878	192.168.0.204	208.67.222.222	DNS	77	53	Standard query 0xe671 TXT debug.opendns.com
6918	2024-08-27 03:07:48.757141	192.168.0.204	208.67.222.222	DNS	77	53	Standard query 0x64d4 TXT debug.opendns.com
6919	2024-08-27 03:07:48.757198	192.168.0.204	208.67.220.220	DNS	77	53	Standard query 0xc555 TXT debug.opendns.com
7610	2024-08-27 03:07:54.070832	192.168.0.204	208.67.222.222	DNS	87	443	Standard query 0xfa51 TXT 2.dnscrypt-cert.opendns.com
7886	2024-08-27 03:07:58.541137	192.168.0.204	208.67.220.220	DNS	77	443	Standard query 0x4381 TXT debug.opendns.com
7887	2024-08-27 03:07:58.541137	192.168.0.204	208.67.222.222	DNS	77	443	Standard query 0xa5b3 TXT debug.opendns.com
7891	2024-08-27 03:07:58.663816	192.168.0.204	208.67.222.222	DNS	77	443	Standard query 0xe5f3 TXT debug.opendns.com
7892	2024-08-27 03:07:58.663964	192.168.0.204	208.67.220.220	DNS	77	443	Standard query 0x31c3 TXT debug.opendns.com

As you see in the snippet from Wireshark, it is clear that client keeps retransmitting DNS TXT queries destined to 208.67.222.222 and 208.67.220.220 on UDP port 443 and 53, but does not receive any response.

There can be multiple reasons behind such behavior, most probably perimeter firewall device is blocking egress DNS traffic to OpenDNS servers, or only allowing traffic to a specific DNS servers.

Web Protection Status is Cloud Service Unavailable

When you see Service Unavailable Web protection state, then most probably Roaming Module does not have upstream connectivity to Secure Web Gateway servers.

If PC does not have IP connectivity to SWG servers, you would see the log in **Umbrella.txt** file, which is part of **DART** bundle.

Date : 08/27/2024
Time : 06:41:22

Type : Warning
Source : csc_swgagent

Description : WARN | Thread 27cc | TCP handshake to SWG Proxy URL was not successful. Since fail open p

In order to investigate further, collect packet capture to prove that PC does not have connectivity with SWG server.

Issue the command in terminal to get SWG IP address:

```
<#root>
```

```
C:\Users\admin>
```

```
nslookup swg-url-proxy-https-sse.sigproxy.qq.opendns.com
```

```
Server: ad.lab.local  
Address: 192.168.0.65
```

```
Non-authoritative answer:
```

```
Name: k8s-sigproxy-sigproxy-c8f482b42a-ddf1929ae349b3e5.elb.eu-west-2.amazonaws.com  
Address:
```

```
18.135.112.200
```

```
Aliases: swg-url-proxy-https-sse.sigproxy.qq.opendns.com  
swg-proxy_eu-west-2_1_1n.sigproxy.aws.umbrella.com
```

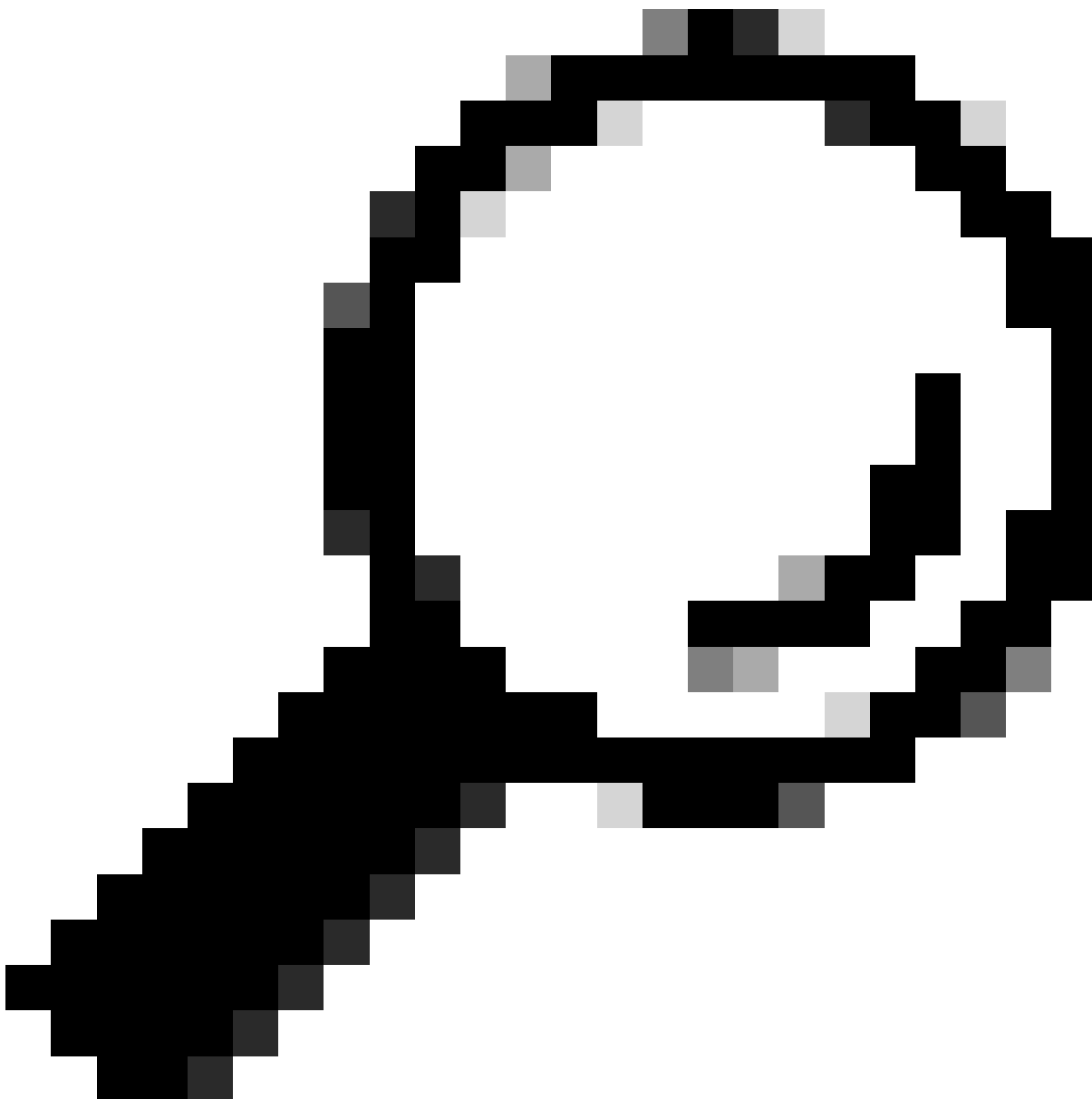
In order to double check and confirm connectivity issues, you can collect wireshark capture on egress physical interface of the PC (WiFi or Ethernet), and use display filter to look only for traffic destined to SWG server (use IP address obtained in previous step)

```
ip.addr == 18.135.112.200
```

No.	Time	Source	Destination	Protocol	Length	Destination Port	Info
7071	2024-08-27 06:41:19.812444	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56287 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7072	2024-08-27 06:41:19.812972	192.168.0.204	18.135.112.200	TCP	66		443 → 56287 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7128	2024-08-27 06:41:20.091970	192.168.0.204	18.135.112.200	TCP	66		56288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7129	2024-08-27 06:41:20.092096	192.168.0.204	18.135.112.200	TCP	66		56289 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7130	2024-08-27 06:41:20.092255	18.135.112.200	192.168.0.204	TCP	60		443 → 56288 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7131	2024-08-27 06:41:20.092255	18.135.112.200	192.168.0.204	TCP	60		443 → 56289 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7205	2024-08-27 06:41:20.314423	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56287 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7206	2024-08-27 06:41:20.314819	18.135.112.200	192.168.0.204	TCP	60		443 → 56287 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7289	2024-08-27 06:41:20.603627	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7290	2024-08-27 06:41:20.603645	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56289 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7291	2024-08-27 06:41:20.604033	18.135.112.200	192.168.0.204	TCP	60		443 → 56288 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7292	2024-08-27 06:41:20.604033	18.135.112.200	192.168.0.204	TCP	60		443 → 56289 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
7434	2024-08-27 06:41:21.110571	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56288 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7435	2024-08-27 06:41:21.110582	192.168.0.204	18.135.112.200	TCP	66		[TCP Port numbers reused] 56289 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

As you see in the snippet from Wireshark, it is clear that client keeps retransmitting TCP SYN packets destined to **18.135.112.200**, but receives TCP RST as response.

In this specific lab scenario, the perimeter firewall was blocking traffic to SWG IP address. In real-life scenario, you can see only TCP SYN retransmissions, not TCP RST.



Tip: If client cannot reach SWG servers, it by default enter **fail open** state where Web traffic is leaving through Direct Internet Access (WiFi or Ethernet). Web protection is not applied in fail open mode.

Solution

In order to quickly identify that underlying network is causing issues, user can connect to any other open network (hotspot, home WiFi) which does not have any perimeter firewall.

To fix described connection error, please make sure that the PC has unrestricted upstream connectivity as outlined in [SSE Documentation](#).

DNS Protection Status issues:

- 208.67.222.222 TCP/UDP port 53
- 208.67.220.220 TCP/UDP port 53

For Web Protection Status issues make sure that traffic to Ingress IP Addresses is allowed on perimeter firewall - [SSE Documentation](#)

Specific range of Ingress IP addresses depends on your location.

Related Information

- [Secure Access User Guide](#)
- [How to collect DART bundle from Cisco Secure Client](#)
- [Technical Support & Documentation - Cisco Systems](#)