Troubleshoot Windows 10 Native VPN API (Modern/Metro apps)

Contents

Introduction

Impact

Root Cause

Resolution

Introduction

This document describes the utilization of VPN applications built on Microsoft's Universal Windows Platform (UWP). These applications typically appear as apps in the Metro/Modern GUI of Windows 8 or higher. Desktop applications and traditional Windows style programs do not use the UWP and are not impacted.

If you are seeing issues with the non-App style windows VPN or always on VPN, see our note on Windows VPN at Umbrella Roaming Client: Compatibility Guide for Software and VPNs .

The AnyConnect Umbrella roaming module does not result in the same interoperability issue.

Impact

When the roaming client is active, UWP VPN application users receive a "No such host is known" or similar error message when attempting to initiate a VPN session. This prevents the user from connecting successfully to the VPN. This error does not appear if the roaming client is stopped or if the desktop edition of the VPN client is utilized.

For example, these do not work:

- Pulse Secure App + roaming client
- SonicWall MobileConnect + roaming client

For these, connecting to the same VPN head end does work:

- Pulse Secure desktop application + roaming client
- SonicWall NetExtender or SonicWall Desktop application + roaming client
- Pulse Secure App + AnyConnect Umbrella roaming module*
- SonicWall MobileConnect + AnyConnect Umbrella roaming module*

*AnyConnect Umbrella module does not set 127.0.0.1 (another interface) as the DNS setting, but rather redirects DNS to 127.0.0.1 using a kernel driver. The UWP apps are not aware they are using a different adapter for DNS and are not impacted at this time.

Root Cause

By design, Microsoft constructed "modern" apps in Windows 8+ to be more sandboxed. One of these limitations applies to VPN applications. VPNs built on the UWP (apps) are restricted to use the interface generating the query. Since the Umbrella roaming client is listening on 127.0.0.1 - a different interface l0 - the query therefore never hits the roaming client (or any other DNS forwarding run on 127.0.0.1).

The following call allows for a socket creation to be restricted to just one interface. This is the case for known UWP VPN apps which constrain DNS to their interface:

ConnectAsync(HostName, String, SocketProtectionLevel, NetworkAdapter)

“The name resolution mechanism used by the ConnectAsync(Hostname, String, SocketProtectionLevel, NetworkAdapter) method is limited to the specified interface for the domain name system (DNS) namespace.” https://learn.microsoft.com/en-us/uwp/api/windows.networking.sockets.streamsocket

The Windows VPN API is thought to utilize this or a similar connection mechanism which results in UWP "app" style VPNs to not function with the roaming client due to it using 127.0.0.1 for DNS. As a result, VPN connections fail as DNS is sent to 127.0.0.1; however, due to the limitations of the call it is never received nor answered by the roaming client. As a result, the VPN connection fails due to DNS failure.

The root cause is similar in its origin as a design limitation to the Windows NCSI connectivity indicator where 127.0.0.1 is not ever queried to confirm DNS connectivity (resolved by Microsoft).

Resolution

At this time for the standalone roaming client, there is no update to the roaming client that can change this Windows behavior resulting from 127.0.0.1 being set as the local DNS server. This is a core equirement for the roaming client to function. Until Microsoft allows for 127.0.0.1 to be used for DNS by a UWP VPN app, the only option is to either **switch to the desktop edition of the VPN client** or **utilize the AnyConnect roaming module** which utilizes a kernel driver to direct DNS.