Contents

Introduction

This document describes some of the current limitations and available workarounds to make AnyConnect and the OpenDNS Roaming Client work together.

Prerequisites

Working knowledge of the AnyConnect and OpenDNS Roaming client.

Familiarity with ASA or IOS/IOS-XE headend configuration (tunnel-group/group-policy) for AnyConnect VPN.

Requirements

Cisco recommends that you have knowledge of these topics:

- ASA or IOS/IOS-XE headend
- Endpoint running the AnyConnect VPN client and OpenDNS Roaming client

Components Used

The information in this document is based on these software and hardware versions:

- ASA headend running release 9.4
- Windows 7
- AnyConnect client 4.2.00096
- OpenDNS Roaming client 2.0.154

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

OpenDNS is developing an AnyConnect plugin with the Cisco AnyConnect team to be available in the future. While no dates have been set, this integration will allow the Roaming Client to work with the AnyConnect client without the workarounds addressed. This will also enable AnyConnect to be a delivery mechanism for the Roaming Client.

Functionality

AnyConnect DNS handling

The VPN headend can be configured in a couple different ways to handle traffic from the AnyConnect client.

1. Full tunnel configuration (tunnel-all): This forces all traffic from the endpoint to be sent across the VPN tunnel encrypted, and therefore traffic never leaves the public interface adapter in clear text
2. Split tunnel configuration:
   a. Split-include tunneling : Traffic destined only to specific subnets or hosts defined on the VPN headend is sent across the tunnel, all other traffic is sent outside the tunnel in clear text

   b. Split-exclude tunneling: Traffic destined only to specific subnets or hosts defined on the VPN headend is excluded from encryption and leaves the public interface in clear text, all other traffic is encrypted and only sent across the tunnel

Each of these configurations determine how DNS resolution is handled by the AnyConnect client, depending on the operating system on the endpoint. There has been a change in behavior in the DNS handling mechanism on AnyConnect for Windows, in release 4.2 after the fix for CSCuf07885.

Windows 7+

Tunnel-all configuration (and split-tunneling with tunnel-all DNS enabled)

Pre AnyConnect 4.2:

Only DNS requests to DNS servers configured under the group-policy (tunnel DNS servers) are allowed. The AnyConnect driver responds to all other requests with a 'no such name' response. As a result, DNS resolution can only be performed using the tunnel DNS servers.

AnyConnect 4.2 +

DNS requests to any DNS servers are allowed, as long as they are originated from the VPN adapter and are sent across the tunnel. All other requests are responded with 'no such name' response, and DNS resolution can only be performed via the VPN tunnel

Prior to CSCuf07885 fix, AC restricts the target DNS servers, however with the fix for CSCuf07885 , it restricts which network adapters can initiate DNS requests.

Split-include configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect driver does not interfere with the native DNS resolver. Therefore, DNS resolution is performed based on the order of network adapters, and AnyConnect is always the preferred adapter when VPN is connected. So a DNS query will be first sent via the tunnel and if it does not get resolved, the resolver will attempt to resolve it via the public interface. The split-include access-list will have to include the subnet covering the Tunnel DNS server(s). Starting with AnyConnect 4.2, host routes for the Tunnel DNS server(s) are automatically added as split-include networks (secure routes) by the AnyConnect client, and therefore the split-include access-list no longer requires explicit addition of the tunnel DNS server subnet.

Split-exclude configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect driver does not interfere with the native DNS resolver. Therefore, DNS resolution is performed based on the order of network adapters, and AnyConnect is always the preferred adapter when VPN is connected. So a DNS query will be first sent via the tunnel and if it does not get resolved, the resolver will attempt to resolve it via the public interface. The split-exclude access-list should not include the subnet covering the Tunnel DNS server(s). Starting with AnyConnect 4.2, host routes for the Tunnel DNS server(s) are automatically added as split-include networks (secure routes) by the AnyConnect client, and therefore that will prevent misconfiguration in the split-exclude access-list.

Split-DNS (tunnel-all DNS disabled, split-include configured)

Pre AnyConnect 4.2

DNS requests matching the split-dns domains are allowed to tunnel DNS servers, but are not allowed to other DNS servers. To prevent such internal DNS queries from leaking out the tunnel, the AnyConnect driver responds with 'no such name' if the query is sent to other DNS servers. So split-dns domains can only be resolved via the tunnel DNS servers.

DNS requests not matching the split-dns domains are allowed to other DNS servers, but not allowed to tunnel DNS servers. Even in this case, the AnyConnect driver responds with 'no such name' if a query for non split-dns domains is attempted via the tunnel. So non split-dns domains can only be resolved via public DNS servers outside the tunnel.

AnyConnect 4.2 +

DNS requests matching the split-dns domains are allowed to any DNS servers, as long as they originate from the VPN adapter. If the query is originated by the public interface, AnyConnect driver responds with a 'no such name' to force the resolver to always use the tunnel for name resolution. So split-dns domains can only be resolved via the tunnel.

DNS requests not matching the split-dns domains are allowed to any DNS servers as long as they originate from the physical adapter. If the query is originated by the VPN adapter, AnyConnect responds with 'no such name' to force the resolver to always attempt name resolution via the public interface. So non split-dns domains can only be resolved via the public interface.

Mac OS X

Tunnel-all configuration (and split-tunneling with tunnel-all DNS enabled)

When AnyConnect is connected, only Tunnel DNS servers are maintained in the system DNS configuration, and therefore DNS requests can only be sent to the Tunnel DNS server(s).

Split-include configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect does not interfere with the native DNS resolver. The tunnel DNS servers are configured as preferred resolvers, taking precedence over public DNS servers, thus ensuring that the initial DNS request for a name resolution is sent over the tunnel. Since DNS settings are global on Mac OS X, it is not possible for DNS queries to use public DNS servers outside the tunnel as documented in CSCtf20226 . Starting with AnyConnect 4.2, host routes for the Tunnel DNS server(s) are automatically added as split-include networks (secure routes) by the AnyConnect client, and therefore the split-include access-list no longer requires explicit addition of the tunnel DNS server subnet.

Split-exclude configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect does not interfere with the native DNS resolver. The tunnel DNS servers are configured as preferred resolvers, taking precedence over public DNS servers, thus ensuring that the initial DNS request for a name resolution is sent over the tunnel. Since DNS settings are global on Mac OS X, it is not possible for DNS queries to use public DNS servers outside the tunnel as documented in CSCtf20226 . Starting with AnyConnect 4.2, host routes for the Tunnel DNS

server(s) are automatically added as split-include networks (secure routes) by the AnyConnect client, and therefore the split-include access-list no longer requires explicit addition of the tunnel DNS server subnet.

Split-DNS (tunnel-all DNS disabled, split-include configured)

If split-DNS is enabled for both IP protocols (IPv4 and IPv6) or it's only enabled for one protocol and there is no address pool configured for the other protocol:
True split-DNS, similar to Windows, is enforced. True split-DNS means that requests matching the split-DNS domains are only resolved via the tunnel, they are not leaked to DNS servers outside the tunnel.

If split-DNS is enabled for only one protocol and a client address is assigned for the other protocol, only "DNS fallback for split-tunneling" is enforced. This means AC only allows DNS requests matching the split-DNS domains via tunnel (other requests are replied by AC with "refused" response to force failover to public DNS servers), but cannot enforce that requests matching split-DNS domains are not sent in the clear, via the public adapter.

Linux

Tunnel-all configuration (and split-tunneling with tunnel-all DNS enabled)

When AnyConnect is connected, only Tunnel DNS servers are maintained in the system DNS configuration, and therefore DNS requests can only be sent to the Tunnel DNS server(s).

Split-include configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect does not interfere with the native DNS resolver. The tunnel DNS servers are configured as preferred resolvers, taking precedence over public DNS servers, thus ensuring that the initial DNS request for a name resolution is sent over the tunnel.

Split-exclude configuration (tunnel-all DNS disabled and no split-DNS)

AnyConnect does not interfere with the native DNS resolver. The tunnel DNS servers are configured as preferred resolvers, taking precedence over public DNS servers, thus ensuring that the initial DNS request for a name resolution is sent over the tunnel.

Split-DNS (tunnel-all DNS disabled, split-include configured)

If split-DNS is enabled, only "DNS fallback for split-tunneling" is enforced. This means AC only allows DNS requests matching the split-DNS domains via tunnel (other requests are replied by AC with "refused" response to force failover to public DNS servers), but cannot enforce that requests matching split-DNS domains are not sent in the clear, via the public adapter.

OpenDNS Roaming client

The Roaming client is a piece of software that manages DNS services on the endpoint, and utilizes the OpenDNS public DNS servers to secure and encrypt DNS traffic.

Ideally, the client should be in a protected and encrypted state. However, if the client is unable to establish a TLS session with the OpenDNS public resolver server (208.67.222.222), it attempts to send DNS traffic unencrypted on UDP port 53 to 208.67.222.222. The Roaming Client exclusively uses OpenDNS's public resolver IP address 208.67.222.222 (there are a few others such as 208.67.220.220, 208.67.222.220, and 208.67.220.222). The roaming client once installed, sets 127.0.0.1 (localhost) as the local DNS server and overrides the current per-interface DNS settings. Current DNS settings are stored in local resolv.conf files (even on Windows) within the Roaming Client configuration folder. OpenDNS will backup even those DNS servers which are learned via the AnyConnect adapter. For example, if 192.168.92.2 is the DNS server on the public adapter, OpenDNS will create the resolv.conf at the following location:

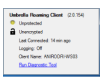
C:\ProgramData\OpenDNS\ERC\Resolver1-LocalAreaConnection-resolv.conf
nameserver 192.168.92.2

The roaming client will encrypt each packet set to OpenDNS; however, it does not start or use an encryption tunnel to 208.67.222.222. The Roaming Client does have an optional IP Layer Enforcement feature which will open up an IPSec connection for non-DNS purposes to block IP addresses. This will automatically disable in the presence of an active AnyConnect connection. It also binds to 127.0.0.1:53 to receive queries locally generated on the computer. When the endpoint needs to resolve a name, the local queries are directed to 127.0.0.1 due to the override, and then the Roaming Client's underlying dnscrypt-proxy process forwards them to the OpenDNS public servers over the encrypted channel.

If DNS is not permitted to flow to 127.0.0.1:53, then the Roaming Client will not be able to function and the following will occur. If the client is unable to reach the public DNS servers or the 127.0.0.1:53 bound address, it will transition to a fail-open state and restore the DNS settings on the local adapters. In the background, it continues to send probes to 208.67.222.222 and can transition to active mode if the secure connection is reestablished.

Limitations

Having looked at the high level functionality of both clients, it is evident that the roaming client needs to have the ability to change the local DNS settings and bind to 127.0.0.1:53 to forward queries across the secure channel. When VPN is connected, the only configurations where AnyConnect does not interfere with the native DNS resolver are the split-include and split-exclude (with split-tunnel-all DNS disabled). Therefore, it is currently recommended to use one of those configurations, when the roaming client is also in use. The Roaming client will remain in an unprotected/unencrypted state if tunnel-all configuration is used, or split-tunnel-all DNS is enabled, as shown in the image.



Workaround

If the intent is to protect communication between the roaming client and OpenDNS servers using the VPN tunnel, then a dummy split-exclude access-list can be used on the VPN headend. This will be the closest thing to a full tunnel configuration. If there is no such requirement, then split-include can be used where the access-list does not include the OpenDNS public servers, or split-exclude can be used where the access-list includes the OpenDNS public servers.

Additionally, when using the Roaming Client, split-DNS modes cannot be used as this will result in a loss of local DNS resolution. Split-tunnel-all DNS should also remain disabled; however, it is partially supported and should allow the Roaming Client to become encrypted post-failover.

Configurations

Tunnel OpenDNS traffic

This example uses a dummy IP address in the split-exclude access-list. With this configuration, all communication with 208.67.222.222 happens across the VPN tunnel, and the roaming client operates in an encrypted and protected state.

```
ciscoasa# sh run access-li split
access-list split standard permit host 2.2.2.2

ciscoasa# sh run group-policy
group-policy GroupPolicy-OpenDNS internal
group-policy GroupPolicy-OpenDNS attributes
 wins-server none
 dns-server value 1.1.1.1
 vpn-tunnel-protocol ssl-client
 split-tunnel-policy excludespecified
 split-tunnel-network-list value split
 default-domain value cisco.com
 address-pools value acpool
 webvpn
 anyconnect profiles value AnyConnect type user
ciscoasa#
```

Exclude OpenDNS traffic from VPN tunnel

This example uses the OpenDNS resolver address in the split-exclude access-list. With this configuration, all communication with 208.67.222.222 happens outside the VPN tunnel, and the roaming client operates in an encrypted and protected state.

```
ciscoasa# sh run access-li split
access-list split standard permit host 208.67.222.222

ciscoasa# sh run group-policy
group-policy GroupPolicy-OpenDNS internal
group-policy GroupPolicy-OpenDNS attributes
 wins-server none
 dns-server value 1.1.1.1
 vpn-tunnel-protocol ssl-client
 split-tunnel-policy excludespecified
```

```
 split-tunnel-network-list value split
 default-domain value cisco.com
 address-pools value acpool
 webvpn
 anyconnect profiles value AnyConnect type user
ciscoasa#
```

This example shows a split-include configuration for an internal 192.168.1.0/24 subnet . With this configuration, the roaming client will still operate in an encrypted and protected state since traffic to 208.67.222.222 is not sent via the tunnel.

```
ciscoasa# sh run access-li split
access-list split standard permit 192.168.1.0 255.255.255.0

ciscoasa# sh run group-policy
group-policy GroupPolicy-OpenDNS internal
group-policy GroupPolicy-OpenDNS attributes
 wins-server none
 dns-server value 1.1.1.1
 vpn-tunnel-protocol ssl-client
 split-tunnel-policy tunnelspecified
 split-tunnel-network-list value split
 default-domain value cisco.com
 address-pools value acpool
 webvpn
 anyconnect profiles value AnyConnect type user
ciscoasa# Note: Split-tunnel-all-dns must be disabled in all of the  scenarios
```

Verify

When VPN is connected, the Roaming client should show protected and encrypted as shown in this image: