

Troubleshoot Non-Browser Applications in Umbrella

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Overview](#)

[Compatibility Problems](#)

[Microsoft 365 Applications](#)

[Certificate Pinning Bypass](#)

[TLS Compatibility Bypass](#)

[Troubleshooting \(Advanced\)](#)

[Identify Exclusions for Certificate Pinning](#)

[Identify Exclusions for Incompatible TLS Versions](#)

Introduction

This document describes how to troubleshoot non-browser applications in Cisco Umbrella.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on Cisco Umbrella.

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, ensure that you understand the potential impact of any command.

Overview

This article explains best practices and troubleshooting steps for configuring Non-Browser applications to function with Umbrella Secure Web Gateway. In most cases, no configuration changes are required. However, certain applications do not work well with security/inspection functions (such as SSL Decryption) and exceptions must be added to make the application function with a web proxy. This applies to Umbrella SWG as well as other web proxy solutions.

This is useful in circumstances where the website/browser version of an application works, but the

desktop/mobile version of the application does not.

Compatibility Problems

Applications can be incompatible for these reasons:

Umbrella Root CA Installation	<p>The Cisco Umbrella Root CA must always be trusted for errorless TLS connections.</p> <ul style="list-style-type: none">• Solution: For Non-Web applications ensure the Cisco Umbrella Root CA is trusted in the System / Local Machine certificate store.
Certificate Pinning	<p>Certificate Pinning (PKP) is when the application expects to receive a precise leaf (or CA certificate) to validate the TLS handshake. The application cannot accept a certificate generated by a web proxy and is not compatible with SSL Decryption functions.</p> <ul style="list-style-type: none">• Solution: Bypass the Application or Domain from SSL Decryption using a Selective Decryption List (see Warning after table) <p>More details about Applications known to be affected by certificate pinning are available here: Public Key Pinning/Certificate Pinning</p>
TLS Version Support	<p>The application can use an older TLS Version / Cipher which is not supported by SWG for security reasons.</p> <ul style="list-style-type: none">• Solution: Bypass the traffic from being sent to Umbrella using the External Domains feature (PAC / AnyConnect) or VPN exclusions (Tunnel) (see Warning after table).
Non-Web Protocol	<p>Some applications use non-http(s) protocols but still send this data over common web ports that are intercepted by SWG. SWG cannot understand this traffic.</p> <ul style="list-style-type: none">• Solution: Consult the Application Vendor to determine the destination addresses / IP ranges used by the software. This software needs to be excluded from SWG using External Domains (PAC / AnyConnect) or VPN Exclusions (Tunnel) (see Warning after table).
SAML Authentication	<p>Most non-browser applications are unable to perform SAML authentication. Umbrella does not challenge non-browser applications for SAML and therefore User/Group based filtering policies cannot match.</p> <ul style="list-style-type: none">• Solution: Enable the IP Surrogates feature so user information can be cached for use with Non-Browser applications.• Alternative: Allow the Application/Domain in a web rule based on Network or Tunnel Identities (not users/groups).
HTTP Range Requests	<p>Some applications use HTTP “Byte-Range” requests when downloading data; meaning only a small chunk of the file is downloaded at a time. These requests are disabled for security reasons in SWG because this technique can also be used to bypass Anti-Virus detection.</p>

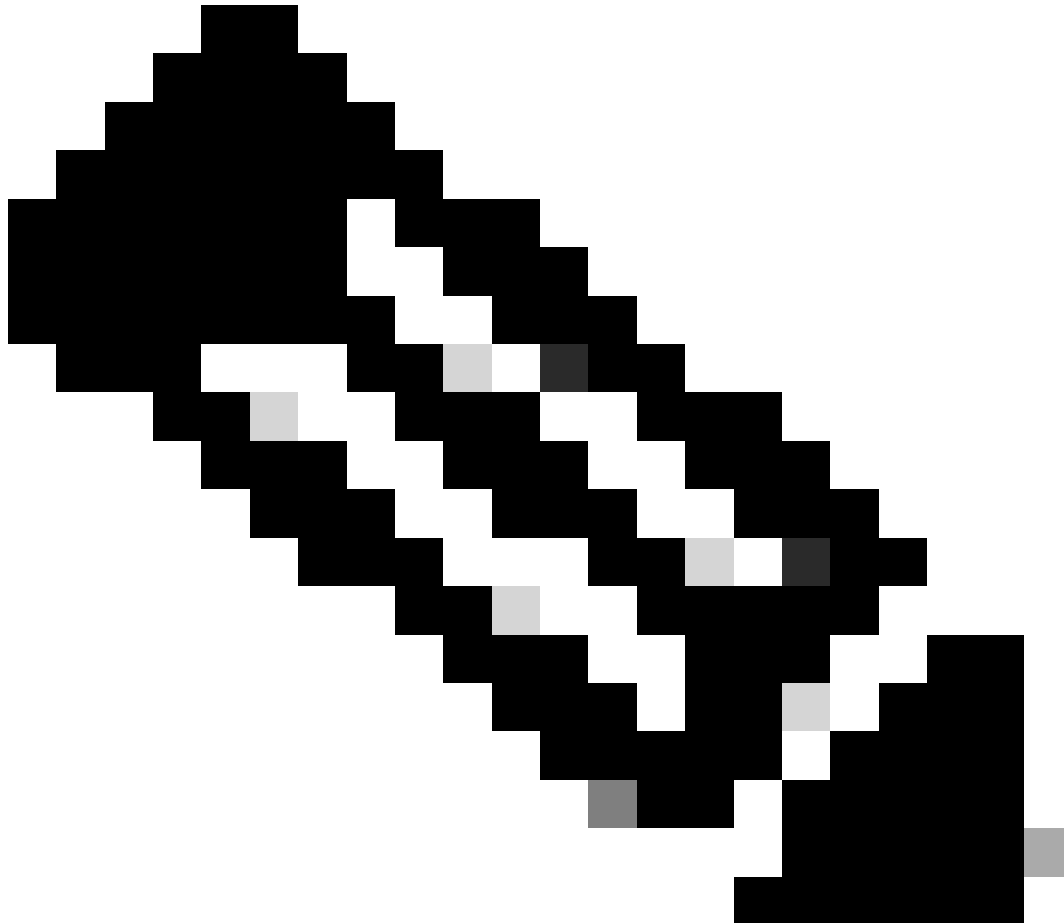
	<ul style="list-style-type: none"> • Solution (HTTPS): Bypass the Application or Domain from SSL Decryption* in Umbrella using Selective Decryption Lists. • Solution (HTTP): Bypass the Application or Domain from Anti-Virus scanning* using a web rule with the Override Security option. • Alternative: Contact Umbrella support if you wish to have Range requests enabled by default* for your organization.
Explicit Proxy Compatibility	<p>Some applications do not respect system proxy settings (eg. PAC files) and are generally not compatible with explicit web proxies. These applications do not route through Umbrella SWG in a PAC file deployment.</p> <ul style="list-style-type: none"> • Solution: The application must be allowed via the local network firewall. Consult the application vendor for details on destinations/ports to be allowed.



Warning: Creating these exceptions can disable security inspection functions including Anti-Virus scanning, DLP scanning, Tenant Controls, File Type Control and URL inspection. Only do this if you are happy to trust the source of these files. The business need for the application must be weighed against the security impact of disabling these features.

Microsoft 365 Applications

The Microsoft 365 Compatibility feature automatically excludes a number of Microsoft domains from SSL Decryption and policy enforcement functions. This feature can be enabled to resolve problems with the Desktop version of Microsoft apps. For more information see [Manage Global Settings](#).



Note: The Microsoft 365 Compatibility feature does not exclude all Microsoft domains. Umbrella uses Microsoft's recommendations for the list of domains which must be excluded from filtering. For more information see [New Office365 Endpoint Categories](#).

Certificate Pinning Bypass

Certificate Pinning (PKP) is a common cause of app compatibility issues. Cisco provide a comprehensive list of named Applications that can be configured to bypass SSL Decryption to workaround. The selective decryption can be configured in **Policies > Selective Decryption Lists**.

In most cases, the administrator can resolve certificate pinning issues simply by excluding the application by its name. This means that these issues can be resolved without having to learn or maintain lists of domains.

Application Testing

Applied To
Web Policy

Categories

Applications
1

Domains
0

Nov 24, 2022

^

List Name

Application Testing

0 Categories Selected

ADD

No Categories Selected

1 Applications Selected

ADD

Dropbox

No Applications Selected

0 Domains

ADD

No Domains

DELETE

CANCEL

SAVE

Alternatively, applications can be bypassed based on destination domain/IP address. Contact the application vendor to determine the applicable list of domains/IPs or see [Identify exclusions for certificate pinning](#).

TLS Compatibility Bypass

Legacy or custom TLS versions are a common cause of app compatibility issues. These problems can be solved by excluding the traffic from Umbrella in **Deployments > Domain Management > External Domains & IPs**. In a tunnel deployment the traffic can only be excluded by adding exceptions in your VPN configuration.

Add New Bypass Domain or Server

When you add a domain, all of its subdomains will inherit the setting. If 'example.com' is on the internal domains list, 'www.example.com' will also be treated as an internal domain.

Domain Type

☐ Internal Domains ☒ External Domains & IPs

Entity

whatsapp.net

Description

Applies To

Domain: Hosted PAC, AnyConnect, SWG Umbrella Chromebook Client

IP: AnyConnect, SWG Umbrella Chromebook Client

CANCEL

SAVE

Contact the application vendor to determine the applicable list of domains/IPs to exclude or see "Identify Exclusions for Incompatible TLS Versions" (later in this article).

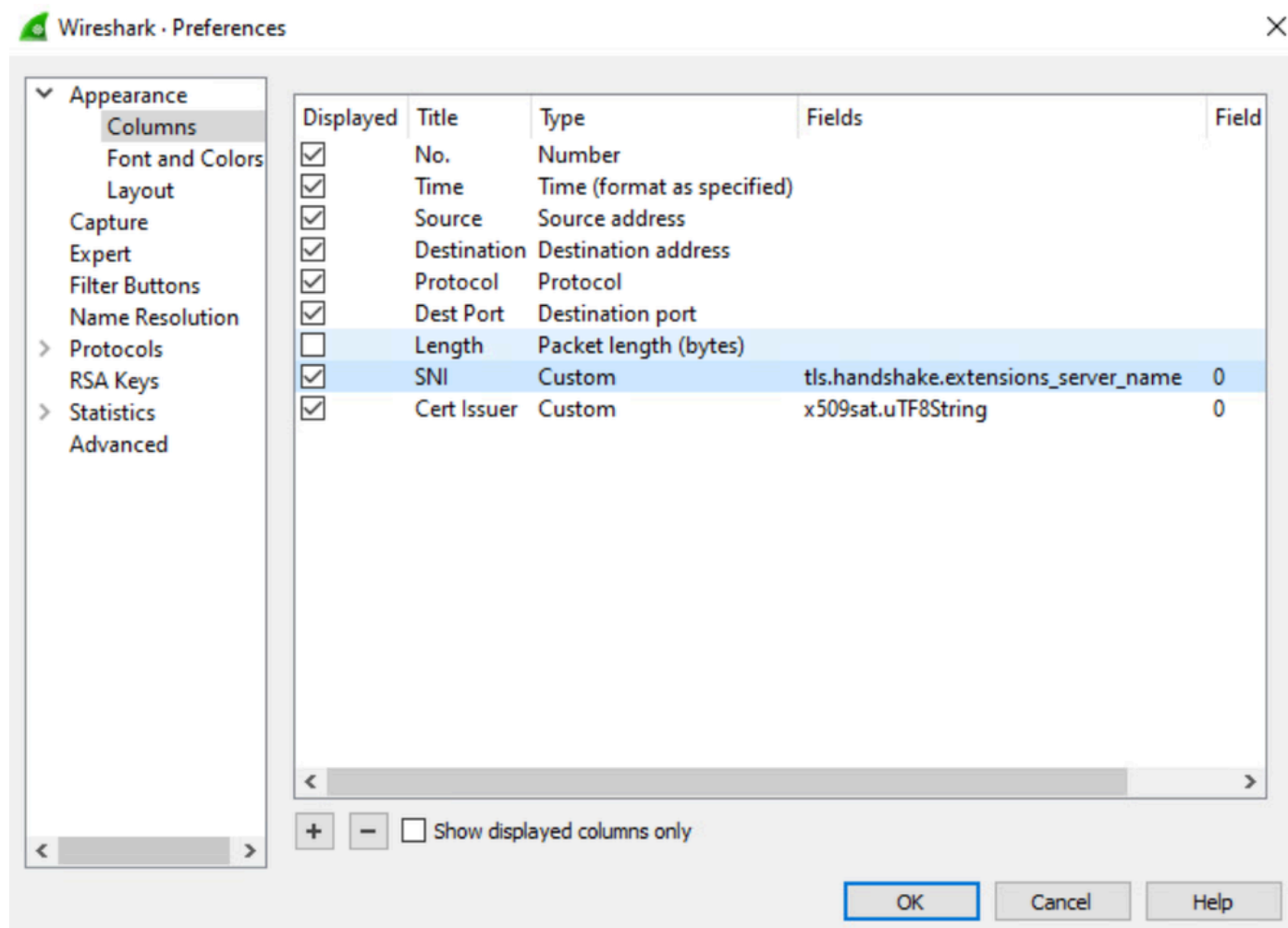
Troubleshooting (Advanced)

The remaining instructions in this article use Wireshark (www.wireshark.org) packet captures for troubleshooting purposes. Wireshark can help to identify which domains are used by applications for help with implementing custom exclusions. Before starting, add these custom columns in Wireshark:

1. Download Wireshark from www.wireshark.org.
2. Go to **Edit > Preferences > Columns**.

3. Create columns of type **Custom** with these fields:

```
http.host  
tls.handshake.extensions_server_name  
x509sat.UTF8String
```



To perform a packet capture, complete these instructions or see Capture Network Traffic with Wireshark.

1. Run Wireshark as an Administrator.
2. Select the relevant network interfaces in Capture > Options.
 - For PAC / Tunnel Deployments, capture on your normal LAN network interface.
 - For AnyConnect Deployments, capture on your LAN network interface and the loopback interface.
3. Close down all other applications except the problem application.
4. Flush your DNS cache: `ipconfig /flushdns`
5. Start the Wireshark capture.
6. Quickly replicate the issue and stop the Wireshark capture.

Identify Exclusions for Certificate Pinning

Certificate pinning is enforced on the client, meaning that the exact behavior and resolution steps differs for every application. In the capture output, look for telltale signs that a TLS connection is failing:

- A TLS connection is being quickly closed or reset (RST or FIN).
- A TLS connection is being repeatedly retried.
- The certificate for the TLS connection is being issued by Cisco Umbrella and is therefore being decrypted.

These example Wireshark filters can help to view the important details of TLS connections.

Tunnel / AnyConnec

```
tcp.port eq 443 && (tls.handshake.extensions_server_name || tls.handshake.certificate || tcp.flags.reset eq 1 || tcp.flags.fin eq 1)
```

PAC / Proxy Chaining

```
tcp.port eq 443 && (http.request.method eq CONNECT || tcp.flags.reset eq 1)
```

In this example, the DropBox desktop application is affected by certificate pinning when attempting to connect to *client.dropbox.com*.

(tls.handshake.extensions_server_name tls.handshake.certificate tcp.flags.reset eq 1 tcp.flags.fin eq 1)							
No.	Time	Source	Destination	Protocol	Dest Port	SNI	Info
281	43.038669	10.10.199.101	162.125.6.13	TCP	443		65148 → 443 [FIN, ACK] Seq=297 Ack=3804 Win=261120 Len=0
283	43.073849	162.125.6.13	10.10.199.101	TCP	65148		443 → 65148 [FIN, ACK] Seq=3804 Ack=298 Win=43008 Len=0
287	43.083933	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
292	43.141656	162.125.6.13	10.10.199.101	TLSv1.2	65149		Certificate, Server Key Exchange, Server Hello Done
296	43.175867	10.10.199.101	162.125.6.13	TCP	443		65149 → 443 [FIN, ACK] Seq=3804 Ack=298 Win=43008 Len=0
297	43.211415	162.125.6.13	10.10.199.101	TCP	65149		443 → 65149 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0
306	46.361407	13.107.21.200	10.10.199.101	TCP	65123		443 → 65123 [FIN, ACK] Seq=32 Ack=1 Win=83 Len=0
309	46.458616	13.107.21.200	10.10.199.101	TCP	65125		443 → 65125 [FIN, ACK] Seq=32 Ack=1 Win=83 Len=0
315	48.228572	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
320	48.272897	162.125.6.13	10.10.199.101	TLSv1.2	65151		Certificate, Server Key Exchange, Server Hello Done
324	48.315138	10.10.199.101	162.125.6.13	TCP	443		65151 → 443 [FIN, ACK] Seq=473 Ack=3804 Win=261888 Len=0
326	48.346412	162.125.6.13	10.10.199.101	TCP	65151		443 → 65151 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0
330	48.357435	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
335	48.408976	162.125.6.13	10.10.199.101	TLSv1.2	65152		Certificate, Server Key Exchange, Server Hello Done
339	48.449284	10.10.199.101	162.125.6.13	TCP	443		65152 → 443 [FIN, ACK] Seq=473 Ack=3804 Win=261888 Len=0
341	48.483947	162.125.6.13	10.10.199.101	TCP	65152		443 → 65152 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0
345	48.514224	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
350	48.555627	162.125.6.13	10.10.199.101	TLSv1.2	65153		Certificate, Server Key Exchange, Server Hello Done
354	48.595411	10.10.199.101	162.125.6.13	TCP	443		65153 → 443 [FIN, ACK] Seq=297 Ack=3804 Win=261888 Len=0
356	48.631537	162.125.6.13	10.10.199.101	TCP	65153		443 → 65153 [FIN, ACK] Seq=3804 Ack=298 Win=43008 Len=0
360	48.641737	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
365	48.685384	162.125.6.13	10.10.199.101	TLSv1.2	65154		Certificate, Server Key Exchange, Server Hello Done
369	48.742518	10.10.199.101	162.125.6.13	TCP	443		65154 → 443 [FIN, ACK] Seq=473 Ack=3804 Win=261888 Len=0
370	48.779104	162.125.6.13	10.10.199.101	TCP	65154		443 → 65154 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0
375	50.854534	10.10.199.101	172.217.15.110	TCP	443		64903 → 443 [FIN, ACK] Seq=2 Ack=74 Win=1020 Len=0
376	50.888092	172.217.15.110	10.10.199.101	TCP	64903		443 → 64903 [FIN, ACK] Seq=74 Ack=3 Win=83 Len=0
381	53.801686	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
387	53.845602	162.125.6.13	10.10.199.101	TLSv1.2	65156		Certificate, Server Key Exchange, Server Hello Done
390	53.888995	10.10.199.101	162.125.6.13	TCP	443		65156 → 443 [FIN, ACK] Seq=473 Ack=3804 Win=261120 Len=0
392	53.919108	162.125.6.13	10.10.199.101	TCP	65156		443 → 65156 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0
396	53.929107	10.10.199.101	162.125.6.13	TLSv1.2	44	client.dropbox.com	Client Hello
402	53.972689	162.125.6.13	10.10.199.101	TLSv1.2	65157		Certificate, Server Key Exchange, Server Hello Done
405	54.011019	10.10.199.101	162.125.6.13	TCP	443		65157 → 443 [FIN, ACK] Seq=473 Ack=3804 Win=261120 Len=0
406	54.047260	162.125.6.13	10.10.199.101	TCP	65157		443 → 65157 [FIN, ACK] Seq=3804 Ack=474 Win=43008 Len=0



Note: After adding the necessary exclusion(s), you can repeat these steps multiple times to identify all the destinations used by the application.

Identify Exclusions for Incompatible TLS Versions

Look for SSL/TLS connections that do not use the mandatory TLS1.2+ protocols supported by Umbrella SWG. This can include legacy protocols (TLS1.0 or earlier) or bespoke custom-protocols implemented by an application.

This example filter shows you initial TLS handshake packets along with DNS queries.

Tunnel / AnyConnect

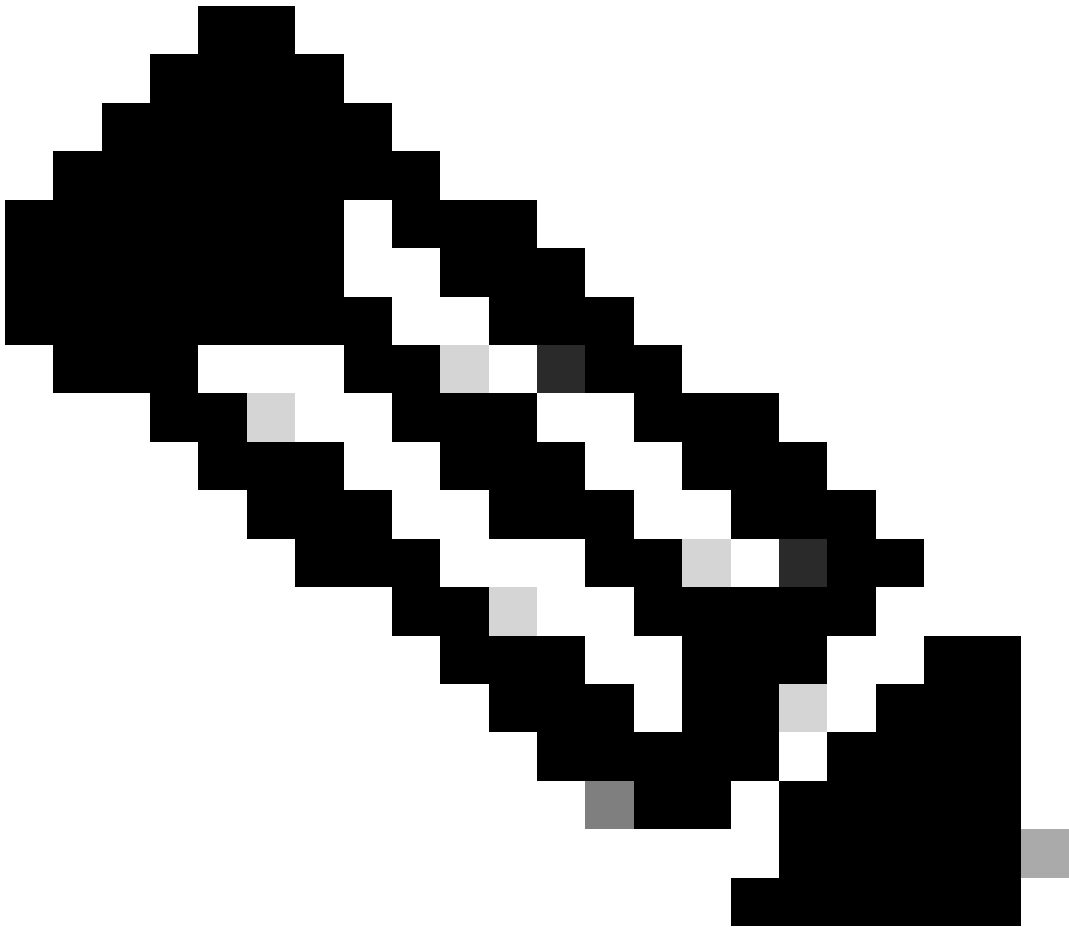
```
dns || (tls && tcp.seq eq 1 && tcp.ack eq 1)
```

PAC / Proxy Chaining

dns || http.request.method eq CONNECT

In this example, the Spotify desktop application is attempting to connect to *ap-gew4.spotify.com* using a non-standard or legacy "SSL" protocol which cannot be sent via SWG.

dns (ts && tcp.seq eq 1 && tcp.ack eq 1)						
No.	Time	Source	Destination	Protocol	Dest Port	SNI
374	62.554832	10.10.199.101	10.10.199.254	DNS	53	
375	62.589486	10.10.199.254	10.10.199.101	DNS	Legacy "SSL" protocol	
379	62.631391	10.10.199.101	34.158.0.131	SSL	443	
				Info		
				Standard query 0x3070 A ap-gew4.spotify.com DNS Information		
				Standard query response 0x3070 A ap-gew4.spotify.com A 34.158.0.13		
				Continuation Data		



Note: After adding the necessary exclusion(s), you can repeat these steps multiple times to identify all the destinations used by the application.