SNI Routing with BIG-IP

Eric Chen, 2018-21-05

In the previous article, The Three HTTP Routing Patterns, Lori MacVittie covers 3 methods of routing. Today we will look at Server Name Indication (SNI) routing as an additional method of routing HTTPS or any protocol that uses TLS and SNI.

Using SNI we can route traffic to a destination without having to terminate the SSL connection. This enables several benefits including:

- Reduced number of Public IPs
- Simplified configuration
- More intelligent routing of TLS traffic

Terminating SSL Connections

When you have a SSL certificate and key you can perform the cryptographic actions required to encrypt traffic using TLS. This is what I refer to as “terminating the SSL connection” throughout this article. When you want to route traffic this is a chicken and an egg problem, because for TLS traffic you want to be able to route the traffic by being able to inspect the contents, but this normally requires being able to “terminate the SSL connection”. The goal of this article is to layer in traffic routing for TLS traffic without having to require having/knowing the original SSL certificate and key.

Server Name Indication (SNI)

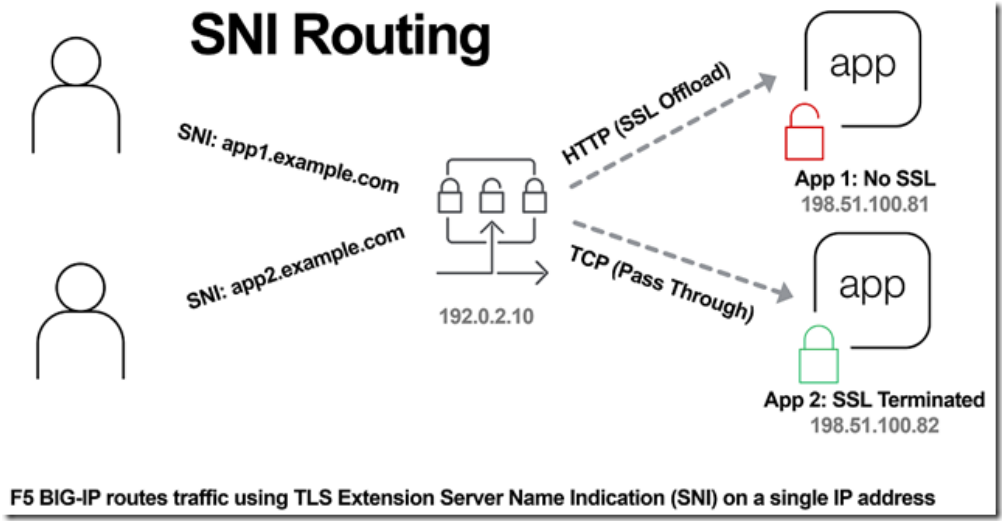
SNI is a TLS extension that makes it possible to "share" certificates on a single IP address. This is possible due to a client using a TLS extension that requests a specific name before the server responds with a SSL certificate.

Prior to SNI, the other options would be a wildcard certificate or Subject Alternative Name (SAN) that allows you to specify multiple names with a single certificate.

SNI with Virtual Servers

It has been possible to use SNI on F5 BIG-IP since TMOS 11.3.0. The following KB13452 outlines how it can be configured. In this scenario (from the KB article) the BIG-IP is terminating the SSL connection. Not all clients support SNI and you will always need to specify a “fallback” profile that will be used if a SNI name is not used or matched. The next example will look at how to use SNI without terminating the SSL connection.

SNI Routing



F5 BIG-IP routes traffic using TLS Extension Server Name Indication (SNI) on a single IP address

Occasionally you may have the need to have a hybrid configuration of terminating SSL connections on the BIG-IP and sending connections without terminating SSL.

One method is to create two separate virtual servers, one for SSL connections that the BIG-IP will handle (using clientssl profile) and one that it will not handle SSL (just TCP). This works OK for a small number of backends, but does not scale well if you have many backends (run out of Public IP addresses).

Using SNI Routing we can handle everything on a single virtual server / Public IP address. There are 3 methods for performing SNI Routing with BIG-IP

1. iRule with binary scan
   a. Article by Colin Walker code attribute to Joel Moses
   b. Code Share by Stanislas Piron
2. iRule with X509::extensions
3. Local Traffic Policy

Option #1 is for folks that prefer complete control of the TLS protocol. It only requires the use of a TCP profile. Options #2 and #3 only require the use of a SSL persistence profile and TCP profile.

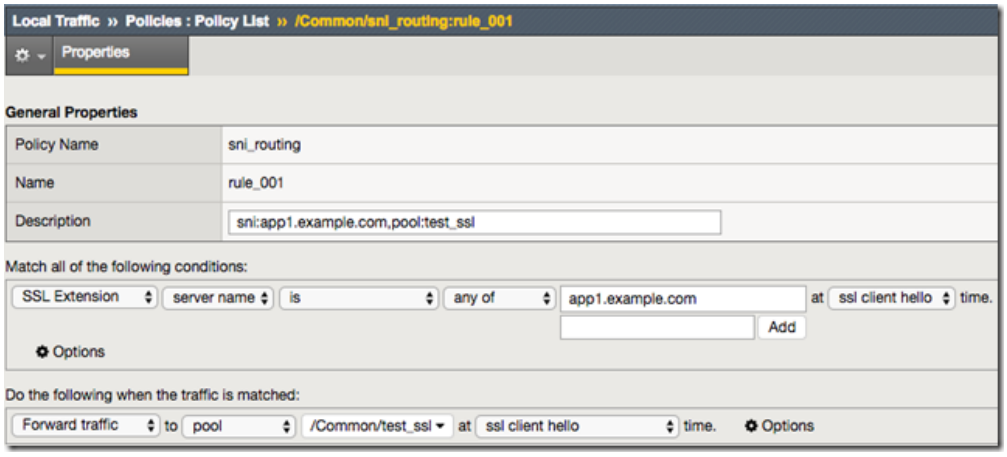
SNI Routing with Local Traffic Policy

We will skip option #1 and #2 in this article and look at using a Local Traffic Policy for SNI Routing. For a review of Local Traffic Policies check out the following DevCentral articles:

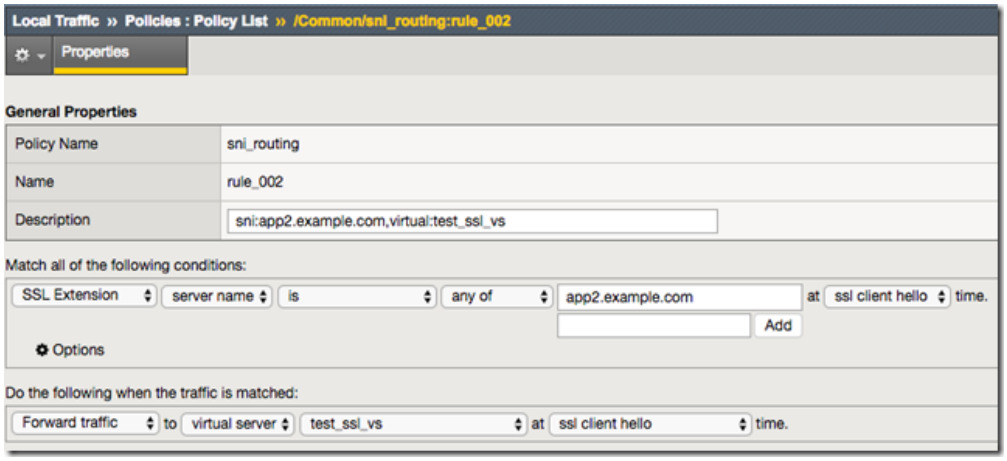
- LTM Policy Jan 2015
- Simplifying Local Traffic Policies in BIG-IP 12.1 June 2016

In previous articles about Local Traffic Policies the focus was on routing HTTP traffic, but today we will use it to route SSL connections using SNI.

In the following example, using a Local Traffic Policy named “sni_routing”, we are setting a condition on the SSL Extension “servername” and sending the traffic to a pool without terminating the SSL connection. The pool member could be another server or another BIG-IP device.



The next example will forward the traffic to another virtual server that is configured with a clientssl profile. This uses VIP targeting to send traffic to another virtual server on the same device.



In both examples it is important to note that the “condition”/“action” has been changed from occurring on “request” (that maps to a HTTP L7 request) to “ssl client hello”. By performing the action prior to any L7 functions occurring, we can forward the traffic without terminating the SSL connection. The previous example policy, “sni_routing”, can be attached to a Virtual Server that only has a TCP profile and SSL persistence profile. No HTTP or clientssl profile is required!

This method can also be used to solve the issue of how to consolidate multiple SSL virtual servers behind a single virtual server that have different APM and/or ASM policies. This is similar to the architecture that is used by the Container Connector for Cloud Foundry; in creating a two-tier load balancing solution on a single device.

Routed Correctly?

TLS 1.3 has interesting proposals on how to obscure the servername (TLS in TLS?), but for now this is a useful and practical method of handling multiple SSL certs on a single IP. In the future this may still be possible as well with TLS 1.3. For example the use of a HTTP Fronting service could be a tier 1 virtual server (this is just my personal speculation, I have not tried, at the time of publishing this was still a draft proposal).

In other news it has been demonstrated that a combination of using SNI and a different host header can be used for “domain fronting”. A method to enforce consistent policy (prevent domain fronting) would be to layer in additional conditions that match requested SNI servername (TLS extension) with requested HOST header (L7 HTTP header). This would help enforce that a tenant is using a certificate that is associated with their application and not “borrowing” the name and certificate that is being used by an adjacent service.

We don’t think of a TLS extension as an attribute that can be used to route application traffic, but it is useful and possible on BIG-IP.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com