GPE-VPN: Programmable LISP-based Virtual Private Networks
draft-maino-gpe-vpn-00

Abstract

GPE-VPN is an architecture for programmable SD-WAN solutions that
leverages the Generic Protocol Encapsulation (GPE) overlay.

GPE-VPN uses an extended LISP-based map-assisted control plane to
dynamically lookup forwarding policies on demand. A northbound
programmable mapping system is used to store and retrieve mappings
and forwarding policies.

The GPE-VPN data plane is secured with IPsec based encryption.

Overlay tunnels, as well as cryptographic parameters, are provisioned
on demand.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 22, 2016.
# 1. Introduction

GPE-VPN is an architecture for programmable Software Defined VPNs that leverages the Generic Protocol Encapsulation (GPE) overlay [I-D.ietf-nvo3-vxlan-gpe].

GPE is effectively merging VXLAN [RFC7348] and LISP [RFC6830] encapsulation in a single format with supports for multi-tenancy and multi-protocol payloads.

## Table of Contents

1. Introduction .............................................. 2
2. Definition of Terms ........................................ 3
3. GPE-VPN Overall Architecture ............................... 3
4. Data Plane Encapsulation .................................... 4
5. Data Plane Operations ....................................... 7
   5.1. Per Destination Mapping ................................. 8
   5.2. FlowMapping .......................................... 8
   5.3. Generic Mapping ....................................... 8
   5.4. Interworking ......................................... 9
6. Control Plane Operations .................................... 9
   6.1. Dynamic Policy Rendering .............................. 10
   6.1.1. In-Bound Load Balancing ............................ 10
   6.1.2. Overlay Re-encapsulation ........................... 11
   6.1.3. Group Based Access Control ......................... 12
   6.1.4. Service Chaining ................................ 12
   6.2. Key Management Services .............................. 12
7. Security Considerations .................................... 13
8. IANA Considerations ....................................... 13
9. Acknowledgements ......................................... 13
10. Normative References .................................... 13
Authors’ Addresses ........................................... 15
GPE-VPN uses an extended LISP-based map-assisted control plane to
dynamically lookup forwarding policies on demand. A controller-based
mapping system is used to store and retrieve the mapping and
forwarding policies. The mapping system is programmable via
northbound API.

GPE-VPN data plane is secured with IPsec based encryption.

Overlay tunnels, as well as cryptographic parameters, are provisioned
on demand.

2. Definition of Terms

   CPE: Customer-premises equipment or customer-provided equipment
   (CPE) is the VPN Tunnel Endpoint that enables access to the VPN.
   In this memo CPE and xTR are used interchangeably.

   GPE: Generic Protocol Encapsulation. In this memo is used to
   refer to both VXLAN-GPE and LISP-GPE frame formats.

For definitions of other terms, notably Map-Request, Map-Reply,
Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Map-Server
(MS), and Map-Resolver (MR) please consult the LISP specification
[RFC6830].

3. GPE-VPN Overall Architecture

A GPE-VPN is designed to enable VPN administrators to specify a high
level intent-based policy that shall be implemented by the VPN. This
includes policies such as connectivity, encryption, access control,
and service chaining.
As specified by the intent-based policy, the GPE-VPN auto-configures the CPEs that implement the data plane of the VPN, and orchestrates and provisions the infrastructure needed to operate the VPN. The intent-based policy is then resolved into network forwarding policy, and is stored in the GPE-VPN mapping infrastructure along with other provisioned network state. The network state is then made available, on demand, to the CPEs using the LISP control protocol. Similarly, the cryptographic state is provisioned on demand by the Key Management Server.

4. Data Plane Encapsulation

In a GPE-VPN frames are encapsulated accordingly to the VXLAN-GPE specification.
Confidentiality and integrity protection are afforded by the use of ESP (Encapsulating Security Payload) in transport mode. ESP is used with an authenticated encryption with associated data (AEAD) cipher such as GCM [RFC4106] that provides confidentiality, integrity, and anti-replay protection to the payload. The use of an AEAD cipher provides integrity and anti-replay protection to the GPE header as well as protection for the Virtual Network Identifier (VNI) and the other GPE fields from being hijacked over the network.
GPE with ESP-GCM

GPE can also be used in combination with the Network Service Header (NSH), as defined in [I-D.ietf-sfc-nsh], to provide application-level service chaining. Encryption and NSH are combined thanks to GPE multiprotocol support.
Data Plane Operations

In a GPE-VPN the CPE performs two main data plane functions:

1. GPE encapsulate un-encapsulated packets that are routed in the overlay network

2. De-capsulate GPE encapsulated packets that are routed in the underlay network

In order to perform the encapsulation function, the CPE uses a map-cache that maps the flow in the overlay to the location(s) (IP address in the underlay network) of the next hop or the destination
CPE depending on the mapping/forwarding policy defined in the mapping system.

The map cache is populated on demand using the LISP [RFC6830] map-request/map-reply protocol. In order to support multi-tenancy, each entry in the map-cache is associated with a VNI that identifies the overlay network of a specific tenant.

The map-cache supports multi-homing and load balancing by supporting mapping a single overlay end point to multiple locations in the underlay along with their priority and weight.

5.1. Per Destination Mapping

The simplest form of mapping supported by a map-cache is the mapping between the Endpoint Identifier (EID, the destination IP or MAC address of the endpoint in the overlay space) and the locator IP address of the destination CPE (its IP address in the underlay space).

EIDs can be either an IP address (L3 overlay) or a MAC address (L2 overlay).

This is the basic function that allows to interconnect VPN sites creating a virtual overlay network that constitutes the VPN. Any endpoint identified by a source EID in a VPN, can reach any other endpoint identified by a destination EID as long as the CPE has an entry in its map-cache for that destination EID.

5.2. FlowMapping

Some CPEs have the capability to map a generic n-tuple (typically a subset of the <source EID, destination EID, source Port, destination Port, Protocol> as defined in [I-D.rodrigueznatal-lisp-multi-tuple-eids]) onto the next hop or destination CPE location.

This allows a much finer granularity in applying the connectivity policy of the VPN. A different connectivity policy can be applied to each pair of endpoints in the overlay, or even per each protocol.

Per Destination Mapping is, in fact, a subset of FlowMapping.

5.3. Generic Mapping

Some CPEs have the capability to map and encap based on a generic "tag" (metadata contained in the packet).
This enables GPE-VPN to offer overlay services to various protocols and applications, as a specific flow is tunneled to a given destination CPE based on the value of the metadata.

As an example a packet may include an NSH header [I-D.ietf-sfc-nsh] that contains metadata used to identify an application-level service function chain that should be applied to that packet before reaching destination. The map-cache can be programmed [I-D.ermagan-lisp-nsh] to tunnel that packet to the service node that implements the next hop in that service function chain and, eventually, to its destination.

5.4. Interworking

Interworking between the VPN and outside networks (e.g. the Internet) is provided by special gateways (LISP PxTRs) that support the encap and decap function for incoming (to the VPN) and outgoing packets.

Gateways also provide reachability to the VPN endpoints from the outside network by advertising highly aggregated EID-Prefix space on behalf of the GPE-VPN.

6. Control Plane Operations

The rendering of the connectivity policy between GPE-VPN sites is based on map-and-encap. When an un-encapsulated flow reaches a CPE, the CPE uses the LISP map-request/map-reply protocol to query the mapping system for the location of the next hop associated with this flow.

The CPE then caches the map-reply that contains the mapping associated with the given flow in its map-cache, so that subsequent packets in this flow hitting that CPE can be encapsulated right away. The map-cache entries are aged and refreshed accordingly to their utilization.

The map-request is encoded using an extensible format [I-D.ietf-lisp-lcaf] that can include a rich set of information not only about the specific flow that has generated the map-request, but also about the CPE sending the request.

As an example, the location of the source CPE can be included in the map-request, or its unique ID, providing the mapping server with information about the current location of the source endpoint that initiated that flow.
6.1. Dynamic Policy Rendering

The map-request is sent to a logically centralized map server that is programmed, via northbound API, with the rendering of the high level policies of the GPE-VPN.

The mapping is dynamically updated to reflect both high level policy changes, as well as changes to the state of the network (as measured by various metrics, including probing sent in the data plane, or reachability information registered by the CPEs). In general telemetric infrastructures can be used to provide the mapping server with a very detailed monitoring of both the underlay and overlay network. In a typical GPE-VPN implementation a telemetry server will collect telemetry data from the network via southbound API. Data is fed to an analytics engine that will update the mapping server via northbound API.

The mapping server can then leverage this amount of information to provide a map-reply that is not only the rendering of the connectivity policy, but also the rendering of a number of other policies applied to the VPN (overlay re-encapsulation, in-bound load balancing, virtual topologies, group based access control, service chaining, ...).

This is one of the most powerful characteristics of a GPE-VPN that makes the mapping server a logically centralized policy enforcement point.

To allow for a more dynamic policy change, the LISP protocol is being extended [I-D.rodrigueznatal-lisp-ms-smr] to support map-server notifications that are used to implement publish/subscribe mechanisms.

Below is a list of the most common policy renderings that may be implemented in a GPE-VPN.

6.1.1. In-Bound Load Balancing

For each overlay end point, the mapping system can specify a list of the locator IP addresses associated with the destination CPE. Each locator is associated with a priority and weight that will determine the in-bound load balancing at the destination CPE.

The sending CPE, upon receiving a map-reply, will encapsulate the outgoing traffic according to the priority and weight associated with the locators in the mapping. This is used to implement active/active or active/stand-by inbound load balancing.
6.1.2. Overlay Re-encapsulation

The high level policy of a GPE-VPN may include overlay re-encapsulation at Re-encapsulating Tunnel Routers (RTRs). The re-encapsulation network function is rendered by the mapping system by providing a different next hop to mapping requests coming from different CPEs/routers.

This section provides a few examples of the use of the re-encapsulation network function.

6.1.2.1. Virtual Topologies

While a GPE-VPN supports any-to-any connectivity, it is possible to implement virtual topologies by manipulating the mappings and using overlay re-encapsulation.

For example to implement an hub-and-spoke topology, the mapping system will be programmed in such a way that map-requests coming from a spoke will always generate map-replays containing the address of the hub as destination locator. In this way all the traffic generated at a spoke will be directed to the hub first, de-capsulated and then re-encapsulated to the destination spoke.

6.1.2.2. Hierarchical VPNs

Some GPE-VPNs may have an hierarchical structure with CPEs grouped in regions that convey traffic to a core via a set of data centers positioned at the edge of the core. Depending on its geographical location, a CPE will send traffic to the RTR located at the edge data center that oversees that region.

Re-encapsulation may be required for a number of reasons, including traffic inspection.

Once a flow from a VPN site A directed to VPN site B hits CPE A, it will generate a map-request that will contain not only the destination EID, but also information about the requesting CPE (e.g. its location, or a site ID). The mapping system, in order to render the hierarchical VPN policy, will return a map-replay to CPE A specifying the location of the re-encapsulating router R as tunnel destination. Once the flow gets to the re-encapsulating router R, it will generate another map-request, for the same flow, but with attributes associated with router R. The mapping system will then render the hierarchical VPN policy by returning the locations of CPE B.
The dynamic manipulation of the mapping is what allows the mapping system to render complex re-encapsulation policies.

6.1.3. Group Based Access Control

Mapping manipulation can also be used to render Group Based Policies (GBP). The intent-based GBP will define groups of endpoints, and the ACLs that shall apply to each group. This is resolved into mapping state so that when the mapping system is resolving a map-request to connect endpoint A to endpoint B, the mapping will reflect the GBP policy, and will not provide connectivity if there’s an ACL preventing communication between the groups to which A and B belongs.

6.1.4. Service Chaining

A GPE-VPN renders application-level Service Chaining Policies by using the mapping system to support the NSH protocol, as described in [I-D.ermagan-lisp-nsh]. NSH uses classification engines to classify flows that should be forced through a given service chain and tags them with an NSH header that contains a certain Service Path Identifier (SPI). Once classified and tagged, the packet needs to be routed to the associated next hop service node(s) that will implement the service chain. The mapping system in this case is programmed to support an SPI to Service Node Location lookup, so that a CPE receiving a packet with a given SPI and Service Index can lookup the address(es) of the next hop service nodes for the specified service chain.

This is an example of a generic mapping service provided by the GPE-VPN mapping system to support a specific protocol other than IP or Ethernet.

6.2. Key Management Services

Provisioning of Security Associations (SAs) for a GPE-VPN is a trade-off between the time needed to set up on demand the security association and the overall security afforded by the GPE-VPN security infrastructure.

There is a continuum of solutions that can be listed in (approximate) increasing order of security afforded:

1. Leverage the LISP map-request/map-reply protocol to provision on demand uni-directional security associations. The LISP crypto draft [I-D.ietf-lisp-crypto], is an example where the LISP map-request/map-reply protocol is augmented with an un-authenticated Diffie-Hellman exchange that provisions encryption keys to the CPEs tunnel endpoints. These mechanisms provide the most
efficient setup time of the SAs (that in [I-D.ietf-lisp-crypto]
requires just an additional round-trip between the CPEs) as a
trade-off with the security afforded, that relies on the security
of the LISP map-request/map-replay protocol and on the relatively
simple structure of the security messages exchanged between the
CPEs.

2. Leverage a Group Domain of Interpretation (GDOI) crypto protocol,
such as the GDOI IPsec extension [RFC6407], for key management.
These protocols use group key distribution mechanisms to securely
provision IPsec encryption keys (and SA parameters) to the CPEs
that belong to the same GPE-VPN. This typically requires
slightly longer SA setup times, compared with the previous
solutions, but the key management infrastructure is independent
from the LISP mapping infrastructure. This, typically, offers a
higher level of security compared to the previous solutions.

3. Leverage the traditional IKEv2 [RFC5996] protocol to negotiate
pairwise SAs between CPEs. While this typically offers the
highest level of security, the setup of SAs is quite time
consuming, and it has a significant impact on the advantages
introduced by creating tunnels on demand using a map-and-encap
protocol.

7. Security Considerations

Security considerations that applies to a GPE-VPN are discuss through
this memo.

8. IANA Considerations

No IANA considerations.

9. Acknowledgements

10. Normative References

[I-D.ermagan-lisp-nsh]
  Ermagan, V., Quinn, P., Lewis, D., Maino, F., and F.
  Coras, "LISP Control Plane integration with NSH", draft-
  ernelagan-lisp-nsh-00 (work in progress), October 2015.

[I-D.ietf-lisp-crypto]
  Farinacci, D. and B. Weis, "LISP Data-Plane
  Confidentiality", draft-ietf-lisp-crypto-03 (work in
  progress), December 2015.
[I-D.ietf-lisp-lcaf]

[I-D.ietf-nvo3-vxlan-gpe]

[I-D.ietf-sfc-nsh]

[I-D.rodrigueznatal-lisp-ms-smr]

[I-D.rodrigueznatal-lisp-multi-tuple-eids]


Internet-Draft  draft-maino-gpe-vpn  March 2016

[RFC6830]  Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The
Locator/ID Separation Protocol (LISP)", RFC 6830,
DOI 10.17487/RFC6830, January 2013,

[RFC7348]  Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger,
L., Sridhar, T., Bursell, M., and C. Wright, "Virtual
eXtensible Local Area Network (VXLAN): A Framework for
Overlaying Virtualized Layer 2 Networks over Layer 3
Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014,
<http://www.rfc-editor.org/info/rfc7348>.

Authors’ Addresses

Fabio Maino
Cisco Systems
Email: fmaino@cisco.com

Vina Ermagan
Cisco Systems
Email: vermagan@cisco.com

John Evans
Cisco Systems
Email: joevans@cisco.com

Horia Miclea
Cisco Systems
Email: hmiclea@cisco.com