Security Verification of a Virtual Private Network over MPLS

Cédric Llorens1, Ahmed Serhrouchni2
1 Equant, La Défense, France
cedric.llorens@equant.com
2 GET-Télécom Paris, LTCI-UMR 5141 CNRS, Paris, France
ahmed.serhrouchni@equant.com

Abstract. We present in this paper how to assess a VPN (Virtual Private Network) security implemented over the Multi Protocol Label Switching (MPLS) protocol. This assessment is based on the definition of a MPLS/VPN security policy and on a reverse-engineering process performed on the network routers configurations. This paper details the algorithms as well as their asymptotic time complexity required to assess this security policy. Moreover, this paper also suggests an approach to rank a VPN perimeter.

1 Introduction

With the deployment of the Multi Protocol Label Switching (MPLS) protocol used at the network core layer, network operators have developed the Virtual Private Network (VPN) service including Quality Of Service (QOS) and the usage of private IP ranges. A MPLS/VPN security is based on the following security mechanisms; the first one is the network routing mechanism, which isolates the VPNs and guarantees the VPNs integrity within the network; the second is the configuration of the VPNs in the network routers configurations in order to create the VPNs topologies.

To protect a MPLS/VPN, a network security policy must be defined, implemented and checked periodically. The assessment process, required to ensure the application of the security policy, consists of parsing the network routers configurations and verifying the security policy through a reverse-engineering process [1,2]. More precisely, this assessment approach is used to verify the security policy of our MPLS backbone composed of several thousands of routers, representing millions of configuration lines and implementing several thousands of VPNs. It allows to check

Please use the following format when citing this chapter:

Llorens, C. and Serhrouchni, A., 2007, in IFIP International Federation for Information Processing, Volume 229, Network Control and Engineering for QoS, Security, and Mobility, IV, ed. Gaïti, D., (Boston: Springer), pp. 339–353.

at the end the isolation and integrity of the VPNs defined in the routers configurations.

We start by presenting the MPLS and MP-BGP (Multi Protocol Border Gateway Protocol) protocols used to create the Virtual Private Networks. Then, we define a security policy targeting the MPLS/VPN security, its implementation in the network routers configurations (based on a CISCO implementation) and the algorithms (with their time complexity) required to assess this policy. Finally, we conclude by describing how this work will be part of a global "Framework for Network Risk Measurement".

The intended audience covers everyone interested in security assessment and configuration management in the context of network management, including network operators.

2 MPLS and MP-BGP protocols

MPLS is a packet-forwarding technology, which uses labels to make data forwarding decisions. With MPLS, the Layer 3-header analysis is done just once (when the packet enters the MPLS domain). Label inspection drives subsequent packet forwarding. The key idea behind MPLS is the use of a forwarding mechanism based on label swapping that can be combined with a set of different control modules. Each control module is responsible for assigning and distributing a set of labels, as well as for maintaining other relevant control information [3]. A MPLS network is composed of the CE (Customer Edge) devices located in the customer premises and used to interconnect the customer sites to the MPLS network, of the PE (Provider Edge) backbone devices used to implement the value-added services (VPN, QOS, etc.) and to interconnect the customer sites, and of the P (Provider) backbone devices used for switching as shown in figure 1.

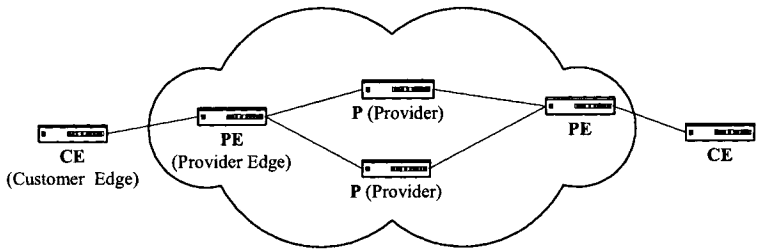

Fig. 1. A MPLS network implementing Virtual Private Networks

A MPLS/VPN is implemented over the MPLS protocol by creating dedicated routing tables per VPN. These routing tables are created thanks to an extension of the Border Gateway Protocol (BGP) protocol named MP-BGP. A VPN or a Virtual Routing and Forwarding instance (VRF) is a dedicated routing table used for connecting a set of sites to a VPN service. The VRFs definitions are defined in the PE routers configurations. Moreover, only the IP prefixes present within the VRF are

advertised through the MPLS backbone. So, each MP-BGP advertisement contains for instance a VPNV4 address (including a route distinguisher value (RD) and an ip prefix value) and the Route-target extended community (RT), which will determine into which VRFs the route should be installed.

3 MPLS/VPN security policy

Based on these technical considerations, we present a MPLS/VPN security policy covering the following security sub-areas:

- The MP-BGP routing topology security policy ensures the MPLS/VPNs integrity and availability.
- The MPLS/VPN perimeter security policy ensures the MPLS/VPNs isolation and integrity.

3.1 MP-BGP routing topology security policy

The network routing used both internally and with external partners is critical in an MPLS networking area, and could impact directly the network availability and integrity in case of routing attacks. These attacks are generally based on spoofing, session hijacking, route flapping, route de-aggregation, un-authorized route injection, etc. These attacks are also based on routing architecture design weaknesses, routing bugs, routing attribute, routing cascade failures, etc. [4,5].

The MP-BGP routing topologies could be viewed at 2 levels. The first level considers the Autonomous Systems (AS) and permits to construct a graph of AS connectivity from which, routing loops may be pruned and some policy decisions at the AS level may be enforced. The second level considers the devices within an AS and permits to construct a graph of device connectivity which is generally more flexible, more scalable and provides more efficient ways of controlling the exchange of information.

These two topology levels are critical and must follow defined rules in order to ensure the network availability and integrity. Any violation of these routing configuration rules could generate serious routing instabilities and impacts for the network and its services. In this area, we define the following security rules:

- Security rule: The "MP-BGP AS" graph must be resilient for availability purpose in order to limit the impacts in case of an incident. Moreover, it should be noted that the routing topology generally follows physical/geographic topology mainly due to cost and technical reasons. So, it means that the "MP-BGP AS" graph must be connected and each vertex (AS) is at a minimum 2 edges connected to each of its backbone internal vertex connections. Regarding the other vertex connections, 2 edges connections are required for "partner" MP-BGP links, and 1 edge connection is at minimum required for each of the other links. We will say that the graph is "as-connected".

Moreover, despite all the security features that could be implemented, a model of trust based on common procedures must be defined between the operators in order to avoid any serious impact like the merging of the backbones routing tables.

- Security rule: The "MP-BGP AS router" graphs (one per AS) must be resilient for availability purpose. It means that the graph inside an AS must be complete for a "full meshing" model or biconnected for a "Route Reflector" model. Moreover, the "Route Reflector" model states that any router, which is not a "Route Reflector" client, must be fully meshed with all the similar routers within an AS ("Route Reflector" server). A consequence is that the graph composed of the "Route Reflector" servers must be complete.

Please note that a combination of both designs is obviously possible to avoid a global "full meshing" design, very consuming in terms of router memory and processing, but also to avoid a global "Route Reflector" model design bringing sub-optimal routing issues.

These security rules are checked in the assessment part by computing the routing graphs extracted from the routers configurations.

3.2 MPLS/VPN perimeter security policy

The configuration of a MPLS/VPN in the network routers configurations is also critical for the VPN security, because any configuration errors could impact the VPN integrity and isolation by connecting the VPN to others unwanted VPNs. The mechanism by which, a MPLS/VPN controls the distribution of VPN routing information, is the use of the MP-BGP route-target extended communities. The MP-BGP route-target extended community string follows a predefined format in order to create a VPN by importing or exporting this route-target community. Importing a route-target means that you learn the routes associated to this route-target, exporting a route-target means that you send your routes to this route-target. A VPN is generally configured on several PE routers.

Knowing that any configuration errors could impact a MPLS/VPN integrity and isolation, we define the following minimum set of security rules:

- Security rule: a VPN configuration must be compliant with the service provisioning requirements. For example, a VPN must be only connected to the authorized VPNs. Any error could bring security weaknesses and could impact the integrity and isolation of the VPN.

- Security rule: a VPN configuration must be consistent. Any inconsistency could bring security weaknesses and could impact the integrity and isolation of the VPN. The minimum set of integrity rules are:
  - Any route-target extended community (RT) export statement defined for a VPN must refer (within the MPLS/VPN network) to a minimum of one import statement related to this RT at a PE level (any connection must be bi-directional).
  - Any route-target extended community (RT) import statement defined for a VPN must refer to a minimum of one export statement related to this RT at a PE level (any connection must be bi-directional).

- Unauthorized (in violation with the predefined format) RT import and export statements must not been configured for a VPN.
- Forbidden (used for administration purpose) RT import and export statements must not been configured for a VPN.

These security rules are checked in the assessment part by computing the MPLS/VPN graph extracted from the routers configurations. Effectively, we build from the network routers configurations a MPLS/VPN directed graph where each vertex represent a VPN, and where a directed edge from VPN(a) to VPN(b) is deduced by the MP-BGP route-target extended community import and export statements configured in an asymmetric manner to create a VPN.

In addition to these rules defined for ensuring a VPN isolation and integrity, the VPN perimeter also acts as an important element regarding the VPN security. To provide a first input, we suggest hereafter an approach to rank a VPN perimeter based on the level of potential threats (graph approach, graph and probabilistic approach). This approach is not dependent on the underlying protocol layer (like MPLS BGP, IPSEC, SSL, etc.) used to build the VPN.

3.2.1 Rank a VPN perimeter through a graph and probabilistic approach

It should be noted that this approach only suggests a possible way to rank a VPN perimeter. It does not provide any mathematical proof and accuracy of the measure, but only intends to open potential research areas regarding "network security" measurement.

A Bayesian network is the convergence of the graph and probability theories. In our framework, a VPN graph can be also viewed as Bayesian network under a certain transformation (the graph must be a directed acyclic graph). This approach could permit to quantify the probability that a VPN is penetrated by other VPNs, if we take into account the MPLS graph data and probabilistic distributions. A VPN graph is not by default a directed graph, so if want to compute the perimeter of a VPN_i, then we need to transform the MPLS/VPN graph to a $DAG(VPN_i)$ graph. At this level, the $DAG(VPN_i)$ graph is a tree computed from the node VPN_i and deduced from a spanning tree algorithm for example.

Please note that we voluntary limit our VPN perimeter in this paper to one DAG due to the exponential number of possible DAGs that can be deduced from the initial graph [6]. It should be considered as a major limitation of the current approach and must be generalized in further works.

The main objective consists to define an approach based on the probability to penetrate a VPN in order to rank a VPN perimeter. The basic idea is to sum all the probabilities that could permit to penetrate a VPN. So, more the probability is lower, more the VPN perimeter is better. At last, it will provide an approach to rank a VPN perimeter within different graph topologies

By definition, a Bayesian network is a directed acyclic graph where each node (i.e. a VPN) is a random variable. So, if a Bayesian network is composed of the following random variables (X_1, X_2 ... X_n), where X_i is "True" if the VPN_i has been penetrated and "False" otherwise, for $i \in [1..n]$, then the probability value that the VPN_i has been penetrated is [7].

For $i \in [1..n]$ and if k is the number of events for which X_i is in the state "True", then we have:

$$P(X_i = \text{True}) = \sum_{m=1}^{k} P(X_i = \text{True}, X_1(m), X_2(m) \dots X_n(m)) \quad (1)$$

If $(X_i = \text{True}, X_1(1), X_2(1) \dots X_n(1))$, …, $(X_i = \text{True}, X_1(k), X_2(k) \dots X_n(k))$ are mutually exclusive, then we have:

$$P(X_i = \text{True}) = \sum_{m=1}^{k} P(X_i = \text{True} \mid X_1(m), X_2(m) \dots X_n(m)) * P(X_1(m), X_2(m) \dots X_n(m)) \quad (2)$$

If we consider the joint probability distribution (chain rule), where $Pa(X_j)$ represents the parents set of X_j (or causes) in the Bayesian graph, then we have:

$$P(X_i = \text{True}) = \sum_{m=1}^{k} P(X_i = \text{True} \mid X_1(m), X_2(m) \dots X_n(m)) * \prod_{j=1, j\neq i}^{n} P(X_j(m) \mid Pa(X_j(m))) \quad (3)$$

For example, if we consider the following Bayesian graphs (A, B, C) in order to compute the perimeter of the VPN_4 as shown in the figure 2.

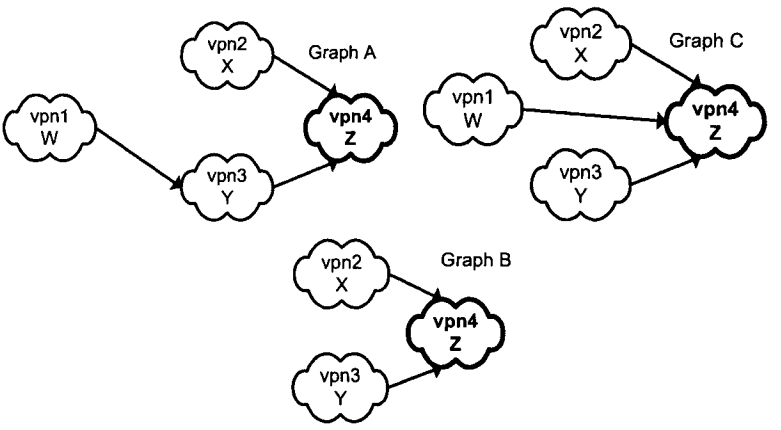

Fig. 2. VPN perimeter : examples of network graph topologies : graph A, graph B, graph C

The graph A shows four VPNs that we can associate to these discrete random variables (W,X,Y,Z). Each random variable has two states, the state "True" means that the VPN has been penetrated and a state "False" otherwise. If we want to rank the VPN_4 perimeter associated to the random variable Z, then we need to fix the following probabilities values:

P(W=True) = 0,5 and P(W=False) = 0,5, P(X=True) = 0,5 and P(X=False) = 0,5, P(Y=True) = 0,5 and P(Y=False) = 0,5, P(Z=True) = 0,5 and P(Z=False) = 0,5

Table 1. Basic VPN penetration probabilities

	W = T	W = F
Y = T	0,8	0,5
Y = F	0,2	0,5

Table 2. Conditional VPN penetration probabilities

	W = T X = T Y = T	T F T	T T F	T F F	F F F	F T F	F T T	F F T
Z= T	0,8	0,8	0,8	0,5	0,5	0,8	0,8	0,8
Z=F	0,2	0,2	0,2	0,5	0,5	0,2	0,2	0,2

So, if we compute the probability that the VPN_2, VPN_3, VPN_4 have been penetrated for the graph B, then we can obtain a VPN perimeter ranking within a network graph topology as described hereafter:

For the VPN_2 (for graph B):

$P(X=T) = 0,5$

For the VPN_3 (for graph B):

$P(Y=T) = 0,5$

For the VPN_4 (for graph B):

$$P(Z=T) = P(Z=T|X=T,Y=T) * P(X=T,Y=T) + \ldots + P(Z=T|X=F,Y=F) * P(X=F,Y=F)$$

$$P(Z=T) = P(Z=T|X=T,Y=T) * P(X=T) * P(Y=T) + P(Z=T|X=F,Y=F) * P(X=F) * P(Y=F) + P(Z=T|X=T,Y=F) * P(X=T) * P(Y=F) + P(Z=T|X=F,Y=T) * P(X=F) * P(Y=T)$$

$$P(Z=T) = 3*0.8*0.5*0.5+0.5*0.5*0.5$$

$$P(Z=T) = 0.725$$

In this example, the VPN_2 and the VPN_3 (0.5) have better VPN perimeter rankings (less exposed to security threats) than the VPN_4 (0.725) in the network graph topology B.

Now, if we compute the probability that the VPN_4 has been penetrated for the graph A, B, C. Then, we can obtain a network graph topology ranking for a dedicated VPN perimeter as described hereafter:

For the graph A (VPN_4):

$$P(Z=T) = P(Z=T|W=T,X=T,Y=T) * P(W=T,X=T,Y=T) + \ldots + P(Z=T|W=F,X=F,Y=F) * P(W=F,X=F,Y=F)$$

$$P(Z=T) = P(Z=T|W=T,X=T,Y=T) * P(Y=T|W=T) * P(X=T) * P(W=T) + P(Z=T|W=T,X=F,Y=T) * P(Y=F|W=T) * P(X=T) * P(W=T) + P(Z=T|W=T,X=T,Y=T) * P(Y=T|W=T) * P(X=T) * P(W=T) + P(Z=T|W=T,X=F,Y=F) * P(Y=F|W=T) * P(X=F) * P(W=T) + P(Z=T|W=F,X=F,Y=F) * P(Y=F|W=F) * P(X=F) * P(W=F) + P(Z=T|W=F,X=T,Y=F) * P(Y=T|W=F) * P(X=F) * P(W=F)$$

+ P(Z=T|W=F,X=T,Y=T) * P(Y=T|W=F) * P(X=T) * P(W=F)
+ P(Z=T|W=F,X=F,Y=T) * P(Y=F|W=F) * P(X=T) * P(W=F)

P(Z=T) = 0.8*0.5*0.8*0.5 + 0.8*0.8*0.5*0.5 + 0.8*0.5*0.2*0.5 + 0.5*0.5*0.2*0.5 + 0.5*0.5*0.5*0.5 + 0.8*0.5*0.5*0.5 + 0.8*0.5*0.5*0.5 + 0.8*0.5*0.5*0.5

P(Z=T) = 0,7475

For the graph B (VPN_4):

P(Z=T) = P(Z=T|X=T,Y=T) * P(X=T,Y=T)
+ … + P(Z=T|X=F,Y=F) * P(X=F,Y=F)

P(Z=T) = P(Z=T|X=T,Y=T) * P(X=T) * P(Y=T)
+ P(Z=T|X=F,Y=F) * P(X=F) * P(Y=F)
+ P(Z=T|X=T,Y=F) * P(X=T) * P(Y=F)
+ P(Z=T|X=F,Y=T) * P(X=F) * P(Y=T)

P(Z=T) = 3*0.8*0.5*0.5+0.5*0.5*0.5

P(Z=T) = 0.725

For the graph C (VPN_4):

P(Z=T) = P(Z=T|W=T,X=T,Y=T) * P(W=T,X=T,Y=T)
+ … + P(Z=T|W=F,X=F,Y=F) * P(W=F,X=F,Y=F)

P(Z=T) = P(Z=T|W=T,X=T,Y=T) * P(X=T) * P(Y=T) * P(W=T)
+ P(Z=T|W=T,X=F,Y=T) * P(X=F) * P(Y=T) * P(W=T)
+ P(Z=T|W=T,X=T,Y=T) * P(X=T) * P(Y=T) * P(W=T)
+ P(Z=T|W=T,X=F,Y=F) * P(X=F) * P(Y=F) * P(W=T)
+ P(Z=T|W=F,X=F,Y=F) * P(X=F) * P(Y=F) * P(W=F)
+ P(Z=T|W=F,X=T,Y=F) * P(X=T) * P(Y=F) * P(W=F)
+ P(Z=T|W=F,X=T,Y=T) * P(X=T) * P(Y=T) * P(W=F)
+ P(Z=T|W=F,X=F,Y=T) * P(X=F) * P(Y=T) * P(W=F)

P(Z=T) = 7*0.8*0.5*0.5*0.5+0.5*0.5*0.5*0.5

P(Z=T) = 0.7625

In this example, the graph B (0.725) has a better ranking for the VPN_4 perimeter than the graph A (0,7475), and the graph A (0,7475) has a better ranking for the VPN_4 perimeter than the graph C (0.7625). The VPN_4 is less exposed to threats in graph B.

4 MPLS/VPN security policy assessment

This assessment approach is used to verify the security policy of our MPLS backbone composed of several thousands of routers, representing millions of configuration lines and implementing several thousands of VPNs. It allows to check at the end the isolation and integrity of the VPNs defined in the routers configurations.

4.1 MP-BGP routing topology security policy assessment

The MP-BGP topology information is directly implemented in the router configuration, so we can extract this information by parsing each router configuration belonging to the MPLS/VPN network. For a CISCO MP-BGP implementation, the following statements express it (Each VPN is associated with one or more VPN routing/forwarding instances (VRFs). A VRF defines the VPN membership of a customer site attached to a PE router):

- hostname name : name of the router.
- ip address ip-address [subnet_mask] : define an IP address of a router link. It will be used to find the MP-BGP neighbors.
- router BGP autonomous-system : define the Autonomous System of the BGP (and MP-BGP) process.
  - neighbor ip-address ... : define the BGP neighbors if used in the framework of a multi-services backbone (offering Internet and MPLS services for example)
- address-family ipv4 vrf : configure sessions that carry IPv4 prefixes. This is used to establish the VPN routing/forwarding (VRF) table, it refers to the customers BGP sessions where a VRF defines the VPN membership of a customer site attached to a PE router.
  - neighbor ip-address ... : define the BGP neighbors
- address-family vpnv4 : configures sessions that carry VPN-IPv4 prefixes, each of which has been made globally unique by adding an 8-byte route distinguisher. This is used to establish the MP-iBGP sessions within the MPLS backbone and the MP-eBGP sessions.
  - neighbor ip-address ... : define the MP-eBGP and MP-iBGP neighbors

It should be noted that we will extract all the "ip address" for all the interfaces in order to find the MP-BGP interconnections. This is mainly due to that we have no "configuration" knowledge of the MP-BGP connections such as this router "number 1" is MP-BGP connected with the router "number 2", so we have to deduce it from all the network routers configurations.

The assessment process is composed of the following steps:

1) Extract and validate the MP-BGP information in the network routers configurations (PE and P routers) in order to create the "topology" file with the following format:
   - <router_name> : name : extracted from "hostname name" configuration
   - <BGP_as_id> : autonomous-system : extracted from "router BGP autonomous-system" configuration
   - <BGP_ip_address> : ip-address : extracted from "router BGP autonomous-system" and "neighbor ip-address" configuration
   - <MP-BGP-ipv4> : ip-address : extracted from "address-family ipv4 vrf" and "neighbor ip-address" configuration
   - <MP-BGP-vpnv4> : ip-address : extracted from "address-family vpnv4" and "neighbor ip-address" configuration

- <type_router> : extracted from "hostname" configuration, it states if the router is a P, PE or CE router

and the "repository" file (P, PE and CE routers) with the following format:

- <router_name> : name : extracted from "hostname name" configuration
- <ip_address> : ip-address : extracted from "ip address ip-address [subnet_mask] " configuration
- <type_router> extracted from "hostname" configuration, it states if the router is a P, PE or CE router

This information will be used to build the MP-BGP matrix and to deduce the MP-BGP graphs through a "jointure" operation performed between the "topology" file and "repository" file. The "repository" file is effectively required to determine from all the routers configurations the useful MP-BGP neighbors information. Obviously, it will be applicable for the internal connections knowing that we have the routers configurations. It should be noted that an unsolved "join" line means that the router points to an external connection such as partners. It will also permit to validate any simple inconsistency of the MP-BGP statements (AS definition, Peer-group definition, etc.).

2) Build the undirected graphs:

- Build the "MP-BGP AS" graph: if we consider the topology and repository files as set of data, we can deduce by the following relational algebra query the graph vertices and edges (2 routers are MP-BGP connected if a router MP-BGP neighbor ip address points to a router interface ip address):

```
/* Determine the MPLS graph vertices */
Distinct topology[BGP_as_id]
/* Determine the MPLS/VPN graph edges for each BGP AS areas */
For each value in topology[BGP_as_id] do
/* List the BGP AS interconnections equivalent to MP-eBGP sessions */
topology[BGP_as_id] as a join repository as b join topology[BGP_as_id] as
c
on
  a[MP-BGP-vpnv4] = b[ip_address]
  and b[router_name] =c[router_name]
where
   a[BGP_as_id] = value and
   a[BGP_as_id] != c[BGP_as_id];
Endfor
```

- Build the "MP-BGP AS router" graphs: if we consider the topology and repository files as set of data, we can deduce by the following relational algebra query the graph vertices and edges (one graph per AS):

```
/* List the BGP AS areas */
For each value in topology[BGP_as_id] do
/* Determine the MPLS graph vertices */
Distinct topology[router_name] where BGP_as_id = value
```

```
/* Determine the MPLS/VPN graph edges where the routers interconnections
are MP-iBGP sessions */
topology[router_name] as a join repository as b join topology[router_name]
as c
on
   a[MP-BGP-vpnv4] = b[ip_address] and
   b[router_name] =c[router_name]
where
   a[BGP_as_id] = value and
   a[BGP_as_id] = c[BGP_as_id] and
   a[type_router] != "CE" and
   b[type_router] != "CE"
/* Determine the MPLS/VPN graph edges where the routers interconnections
are BGP sessions */
topology[router_name] as a join repository as b join topology[router_name]
as c
on
   a[MP-BGP-ipv4] = b[ip_address] and
   b[router_name] =c[router_name]
where
   a[BGP_as_id] = value and
   a[BGP_as_id] = c[BGP_as_id] and
   a[type_router] = "PE" and
   b[type_router] = "CE"
Endfor
```

As the routing graphs are sparse graphs, we prefer to take an adjacency-list structure in order to have a better time complexity of the various graph operations [9].

3) Check the "MP-BGP AS" graph: we check if the graph is as-connected. If $|V|$ is the number of vertices and $|E|$ is the number of edges in the graph, then the asymptotic time complexity (based on the Depth-First Search algorithm to check connectivity and compute articulation point) is $O(|V|+|E|)$ [8].

4) Check the "MP-BGP AS router" graph: we check if the graph is complete for the MP-iBGP full meshing model and is biconnected for the "Route Reflector" meshing model. Same algorithm and asymptotic time complexity detailed previously.

4.2 MPLS/VPN Perimeter topology security policy assessment

The MPLS/VPN topology information is directly implemented in the router configuration, so we can extract this information by parsing each router configuration belonging to the MPLS/VPN network. For a CISCO MPLS/VPN implementation, the following statements express it:

- ip vrf vrf_name : creates a VRF routing table and a CEF (forwarding) table, both named vrf_name.
- route-target {import | export | both} *route-target-ext-community:* To create a route-target extended community for a VRF.

The assessment process is composed of the following steps.

1) Extract and validate the MPLS/VPN information in the network routers configurations in order to create the "topology" file with the following fields (PE routers):

- <router_name> : name : extracted from the "hostname name" configuration
- <vrf_name> : name : extracted from "ip vrf vrf_name" configuration
- *<rt>* : *route-target-ext-community* : extracted from "route-target {import | export} *route-target-ext-community*" configuration
- <im_ex> : "export" |"import" : extracted from "route-target {import | export} *route-target-ext-community*" configuration

This information will be used to build the MPLS/VPN graph based on the "topology" file.

2) Build the MPLS/VPN graph:

The first step consists in determining the MPLS/VPN graph vertices. If we consider the topology as a set of data, we can deduce directly the vertices by the field "vrf".

Regarding the MPLS/VPN graph edges, we need to determine the interconnections between the VPNs thanks to the MP-BGP route-target extended community import and export statements.

For consistency and coherency of the network routers configurations, we will use a unique vrf_name within the MPLS/VPN network for a specific VPN.

Due to the asymmetric configuration of the RTs, the "export rule" and "import rule" will determine all the MPLS/VPN graph edges knowing that if a VRF(a) has an export statement with a RT(x) and that the VRF(b) has an import statement with RT(x), then there is a directed edge between VRF(a) and VRF(b). Moreover, it implies that there is also a directed edge between VRF(b) and VRF(a) in order to create a network connection (VRF(b) should have an export statement with RT(x) and VRF(a) should have an import statement with the RT(x)).

So, for each route-target RTx, we must build the lists of the VRFs exporting this RTx and the list of the VRFs importing this RTx as shown in the figure 3.

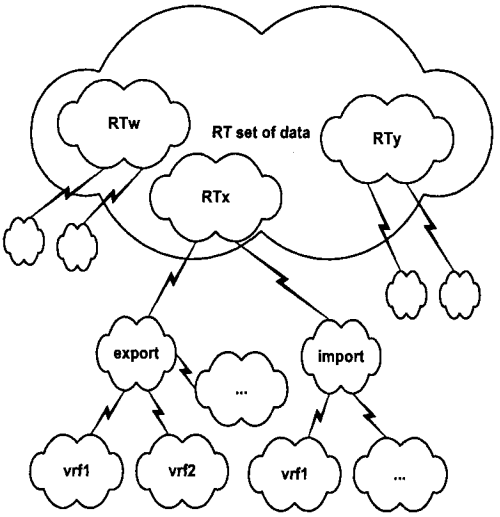

Fig. 3. The RT Trees hierarchy

Now, if we consider the topology as a set of data, we can deduce by the following relational algebra query the graph vertices and edges:

```
/* Determine the MPLS/VPN graph nodes */
Distinct topology[vrf_name]
/* Determine the MPLS/VPN graph edges */
For each value in topology[rt] do
    /* List the VRF interconnections */
    topology[vrf_name] as a join
    topology[vrf_name] as b
      on a[rt] = b [rt] = value
    where
    /* Directed edges statements */
    a[im_ex] = "export" and b[im_ex] = "import"
    and a[vrf_name]! = b[vrf_name]
Endfor
```

As the MPLS/VPN graph is a sparse graph by nature, we prefer to take an adjacency-list structure in order to have a better time complexity of the various graph operations [9].

At this level, the VPN security configuration consistency rules are checked during the graph building. The service provisioning rules and the VPNs connections are checked by computing the MPLS/VPN graph connected components. If $|V|$ is the number of vertices and $|E|$ is the number of edges in the graph, then the asymptotic time complexity (based on the Depth-First Search algorithm to check connectivity and compute articulation point) is $O(|V|+|E|)$ [8].

At last, articulation points, a VPN direct and indirect connections, etc. can be deduced by using well-known standard graph algorithms in order to compute the VPN perimeter ranking.

5 CONCLUSION

A MPLS/VPN security policy could be assessed efficiently knowing that the algorithms used are polynomial-time with a degree <= 2. This assessment approach is used to verify the security policy of our MPLS backbone composed of several thousands of routers, representing millions of configuration lines and implementing several thousands of VPNs. It allows to check at the end the isolation and integrity of the VPNs defined in the routers configurations.

Considerable researches have been conducted to define the security metrics (measures) for a system (Computer System) such as [10,11]. Moreover, other publications also introduce the security metrics (measures) to assess risk management capabilities (Information Systems) [12,13] or to provide a general framework for security measurement and assessment, however a global definition of the security metrics (measures) for a network is missing.

Thanks to the experience gained during the development of the router configuration validation tool [1,2], we will engage our efforts to define a detailed framework for security measurement of a network and its services. It will include the definition of the network security attributes, the definition of the security metrics (measures) and the definition of a global "Framework for Network Risk Measurement".

REFERENCES

1. D. Valois, C. Llorens, Network Device Configuration Validation, Proceedings of 14th annual FIRST conference, Hawaii, 2002.

2. C. Llorens, D. Valois, Y. Le Teigner, A. Gibouin, Computational complexity of the network routing logical security, Proceedings of the IEEE international Information Assurance Workshop, Darmstadt, Germany, pp. 37-49, 2003.

3. E. Rosen, A. Viswanathan, R. Callon, Multiprotocol Label Switching Architecture, Internet Engineering Task Force, www.ietf.org, Proposed standard, 2001.

4. N. Feamster, Practical verification techniques for wide-area routing, ACM SIGCOMM Computer Communication Review, Volume 34, Issue 1, pp. 87-92, 2004.

5. N. Feamster, H. Balakrishnan, Towards a logic for wide-area Internet routing, Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, pp. 289-300, 2003.

6. R. Robinson, Counting unlabeled acyclic digraphs, in C.Little editor, Combinatorial Mathematics V, volume 622 of Lecture Notes in Mathematics, Springer, pp. 28-43, 1977.

7. Finn V.Jensen, Bayesian Networks and Decision Graphs, Springer, ISBN 0-387-95259-4, pp. 1-30, 2001.

8. R.E. Tarjan, Depth First Search and Linear Graph Algorithms, conference record of Twelfth Annual IEEE symposium on Switching and Automata theory, New York, pp. 114-121, 1971.

9. G. Brassard, P. Bratley, Fundamentals Of algorithmics, Prentice-Hall, ISBN 0-13-335068-1, pp. 219-258, 1996.

10. Common Criteria, the common criteria represents the outcome of a series of efforts to develop criteria for evaluation of IT security that are broadly useful within the international community, for more information see: http://www.commoncriteria.org

11. R. ORTALO, Évaluation quantitative de la sécurité des systèmes d'information, Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, Rapport LAAS 98164, 19 mai 1998.

12. J.R. Williams, G.F. Jelen, A framework for reasoning about assurance, Project report supported by National Securiy Agency, contract number MDA904-97-C-0223, 1998.

13. W.A. Wulf, D.M. Kienzle, A practical approach to security assessment, MOAT project report supported by DARPA, contract number N66001-96-C-8527, 1996.