Mesure de la sécurité "logique" d'un réseau d'un opérateur de télécommunications

Cédric Llorens

Thèse

Présentée pour obtenir le grade de docteur
de l'Ecole Nationale Supérieure
des Télécommunications

Spécialité : **Informatique et Réseaux**

Cédric Llorens

Mesure de la sécurité "logique" d'un réseau d'un opérateur de télécommunications

Soutenu le 7 novembre 2005 devant le jury composé de:

Rapporteurs
- Guy Pujolle
- Omar Cherkaoui

Examinateurs
- Philippe Godlewski
- Samir Thomé
- Robert Erra
- Prosper Chemouil

Directeur de thèse
- Ahmed Serhrouchni

Remerciements

Je remercie tout d'abord mon directeur de thèse Ahmed Serhrouchni pour son soutien tout au long de ce travail. Je suis heureux de pouvoir lui exprimer mes plus vifs remerciements et ma très sincère reconnaissance.

Je remercie sincèrement Guy Pujolle, Omar Cherkaoui, Philippe Godlewski, Shamir Thomé, Robert Erra et Prosper Chemouil qui m'ont fait l'honneur de siéger dans ce jury et de juger mon travail.

Je remercie enfin Denis Valois, mon collègue et ami, pour tous les moments de réflexion que nous avons partagés.

Résumé

Cette thèse présente un prototype et une méthodologie pour mesurer la sécurité "logique" d'un réseau d'un opérateur de télécommunications. Cette méthode consiste à définir une politique de sécurité, à vérifier l'application de la politique de sécurité dans les configurations des équipements réseau, à définir des indicateurs de sécurité afin d'établir un tableau de bord de la sécurité réseau, à quantifier le risque associé à la non application de la politique de sécurité et enfin à définir des priorités pour corriger les faiblesses de sécurité les plus critiques.

Abstract

This thesis presents a prototype and a methodology for measuring the "logical" security level of a data provider network. This method consists of defining a network security policy, of checking the network security policy through a reverse engineering analysis performed on the network devices configurations, of defining security indicators in order to build a network security dashboard, of computing the resulting risk and of defining priorities in order to fix the most critical security weaknesses.

TABLE DES MATIERES

INDEX DES FIGURES

INDEX DES TABLEAUX

Abréviations

ACL : Access Control List
ATM : Asynchronous Transfer Mode
ARP : Address Resolution Protocol
BGP : Border Gateway Protocol
BLAS : Basic Linear Algebra Subprograms
CE : Customer Edge
CESTI : Centre d'Evaluation de la Sécurité des Technologies de l'Information.
CDP : Cisco Discovery Protocol
CTCPSEC : Canadian Trusted Computer Product Evaluation Criteria
DAG : Directed Acyclic Graph
DF : Don't Fragment
DOS : Denial Of Services Attacks
DDOS : Distributed Denial Of Services Attacks
DLCI : Data Link Connection Identifier
EGP : Exterior Gateway Protocol
IGMP : Internet Group Management Protocol
IGP : Interior Gateway Protocol
IGRP : Interior Gateway Routing Protocol
IOS : Internetworking Operating System
IP : Internet Protocol
ISO : International Standard Organisation
IETF : Internet Engineering Task Force
IPSEC : IP SECurity
ITSEC : Information Technology Systems Evaluation Criteria
IS_IS : Intermediate System to Intermediate Systems
LAN : Local Area Network
LDP : Label Distribution Protocol
LER : Label Edge Router
LSR : Label Switching Router
LSP : Label Switch Path
LTD : Label Tag Distribution
MP-BGP : Multi Protocol Border Gateway Protocol
MPLS : Multi Protocol Label Switching
MTU : Maximum Transfer Unit
NIST : National Institute Standard Technologies
NSA : National Security Agency

OSI : Open Systems Interconnection
OSPF : Open Shortest Path First
P : Provider
PE : Provide Edge
PPP : Point To Point Protocol
RIP : Routing Information Protocol
RFC : Request For Comments
RR : Route Reflector
RSVP : Ressource reSerVation Protocol
SNMP : Simple Network Management Protocol
TCP : Transport Control Protocol
TCSEC : Trusted Computer Systems Evaluation Criteria
TFN : Tribe Flood Network
TTL : Tive To Live
UDP : User Data Protocol
VLAN : Virtual Lan Access Network
VRF : Virtual Routing Forwarding
VTP : Virtual Trunking Protocol
VPI/VCI : Virtual Path Identifier/Virtual Circuit Identifier
VPN : Virtual Private Network

Introduction générale

La pérennité de toute Entreprise passe aujourd'hui par une disponibilité permanente de son système d'information. L'information nécessaire au bon fonctionnement de l'Entreprise englobe aussi bien les données stratégiques que les données de tous les jours. Le système d'information doit être vu comme un ensemble qui inclut aussi bien l'information elle-même que les systèmes nécessaires pour la mettre en œuvre.

La continuité de l'activité de l'Entreprise appelle à la continuité de son système d'information. Cette continuité ne peut être assurée que par la mise en place de moyens de protection permettant d'apporter un niveau de sécurité adapté aux enjeux spécifiques de l'Entreprise. Ces derniers peuvent varier d'une Entreprise à l'autre, mais la mise en place de la protection des systèmes d'information répond à des critères communs à adapter en conséquence.

Les objectifs de sécurité d'un système d'information doivent répondre à des critères de continuité d'activité et être orientés "valeur" plutôt que "sécurité". On comprend aisément qu'une information sans système d'information pour la mettre en œuvre est vaine, et qu'un système d'information coupé de ses utilisateurs est sans objet.

La sécurité des réseaux est donc devenue un des éléments-clés de la continuité des systèmes d'information de l'Entreprise quelles que soient son activité, sa taille et sa répartition géographique. Sachant de plus que la sécurité informatique au sens large devient une problématique planétaire avec l'avènement de l'Internet, la maîtrise et la mesure de la sécurité logique des réseaux, basées sur les configurations des équipements réseau, devient une priorité majeure pour les opérateurs de télécommunications et leurs clients.

L'objectif de cette thèse est de présenter une méthodologie, des métriques et des algorithmes pour mesurer la sécurité logique d'un réseau "multi-services IP" d'un opérateur de télécommunications.

Les réseaux "multi-services IP" (multi-services) des opérateurs de télécommunications offrent non seulement des services de connexion réseau à Internet basés sur le protocole IP (Internet Protocol), mais offrent aussi des services de connexion réseau à valeur ajoutée de type réseau privé virtuel (Virtual Private Network) basés sur le protocole MPLS (Multi Protocol Label Switching). Sachant que toutes ces connexions transitent sur une architecture réseau physique partagée, la garantie de l'intégrité et de la disponibilité du réseau ainsi que de ces services nécessitent de vérifier régulièrement le niveau de sécurité.

Les évaluations existantes de la sécurité sont soit basées sur des critères d'évaluation comme les critères communs [CC 2002], soit basées sur une

analyse probabiliste des risques [Bedford et al. 2001, Stamatelotos 2002]. Cependant, d'autres types d'évaluation de la sécurité ont été proposés [Philips et al. 1998, Swiler et al. 2001], on retiendra plus particulièrement le travail réalisé par le laboratoire LAAS [Ortalo 1998] qui définit une mesure de la sécurité comme l'effort pour un attaquant d'obtenir des privilèges sur des objectifs de sécurité. Par ailleurs, de nombreux travaux de recherche explorent le domaine des graphes d'attaques, afin de déterminer quel est le jeu minimal de règles de sécurité permettant d'assurer la sécurité d'un système [Somesh et al. 2002]. Bien que d'autres travaux couvrent aussi la gestion des risques pour un système d'information [Wulf et al. 1996, Williams et al. 1998, Bush et al. 2001], aucun papier n'évoque réellement la mesure de la sécurité d'un réseau de télécommunications.

Grâce à l'expérience acquise durant le développement de notre prototype de mesure de la sécurité d'un réseau d'un opérateur de télécommunications [Valois et Llorens 2002, Llorens et al. 2003], nous présentons dans ce papier une méthodologie pour mesurer la sécurité d'un réseau multi-services. Cette approche permet de prendre en compte à la fois la problématique relative au nombre d'équipements, mais aussi la problématique de la complexité en temps des algorithmes utilisés pour implémenter les tests de sécurité que nous décrirons par la suite. Le réseau de France Télécom/Equant comporte de l'ordre de 40.000 équipements réseau représentant près de 20 millions de lignes de configurations.

Nous définirons tout d'abord ce qu'est un réseau multi-services et les menaces qui pèsent sur ce réseau. Nous détaillerons ensuite la politique de sécurité d'un réseau multi-services en posant un certain nombre de règles de sécurité génériques. Nous commenterons alors les différents types de méthodes d'évaluation de la sécurité existantes et argumenterons sur le choix de la méthode retenue. Enfin, nous décrirons en détail le prototype d'évaluation de la sécurité que nous avons développé au sein de France Télécom/Equant.

Chapitre 1. Définition d'un réseau multi-services

1.1. Introduction

Dans les réseaux IP, le routage des paquets s'effectue sur les adresses IP, ce qui nécessite de lire les en-têtes IP à chaque passage sur un nœud réseau. Pour réduire ce temps de lecture, deux protocoles ont vu le jour afin d'améliorer le transit global par une commutation des paquets au niveau 2 et non plus 3, comme le fait IP. Ces protocoles sont ATM (Asynchronous Transfer Mode), sur une initiative de l'ATM Forum, et MPLS (MultiProtocol Label Switching), sur une initiative de CISCO et IBM. Un réseau multi-services se base sur le protocole MPLS, qui est devenu un standard IETF (Internet Engineering Task Force), et route des paquets dans le réseau à partir de labels et non à partir d'adresses IP. La commutation de paquets se réalise sur ces labels et ne consulte plus les informations relatives au niveau 3 incluant les adresses IP [RFC2917, RFC3031].

Même si l'amélioration des équipements hardware ne rend plus aussi nécessaire qu'auparavant la commutation au niveau 2 plutôt qu'au niveau 3, le protocole MPLS permet aussi la création de réseaux privés virtuels reposant sur des classes de services afin de garantir des délais d'acheminement.

Un réseau privé virtuel MPLS/VPN permet de connecter des sites distants sur un réseau partagé par tous les clients. Le trafic du réseau privé virtuel est isolé logiquement des autres trafics VPN. Cette isolation est réalisée par un mécanisme de routage fondé sur le protocole MP-BGP (Multi Protocol-Border Gateway Protocol), qui est une extension du protocole de routage BGP (Border Gateway Protocol) pour les réseaux MPLS. De plus, chaque VPN peut faire transiter les classes d'adresses IP qu'il désire sans qu'il y ait de conflit d'adresses IP avec d'autres VPN, puisque chaque VPN a sa propre table de routage et que, sur les réseaux MPLS, la commutation du trafic réseau est réalisée sur des labels uniques et non sur des adresses IP [RFC3032].

1.2. Les composants d'un réseau multi-services

Un réseau multi-services est composé :

- De routeurs P (Provider) ou LSR (Label Switch Router) dédiés à la commutation.
- De routeurs PE (Provider Edge) ou LER (Label Edge Router) dédiés à la création des MPLS/VPNs ainsi qu'à la connectivité réseau avec les équipements localisés chez les clients.

- De routeurs RR, Route Reflector, dédiés à la centralisation des tables de routage des MPLS/VPNs et de la table de routage globale du réseau multi-services.
- De routeurs CE (Customer Edge), installés chez les clients et connectés aux routeurs PE.

Deux types de services sont offerts sur ce réseau, le premier service permet de connecter un CE à Internet en exploitant la table de routage globale du réseau multi-services. Le second service permet de connecter un CE à un réseau MPLS/VPN en exploitant les tables de routage dédiées aux VPNs comme illustré à la figure suivante :

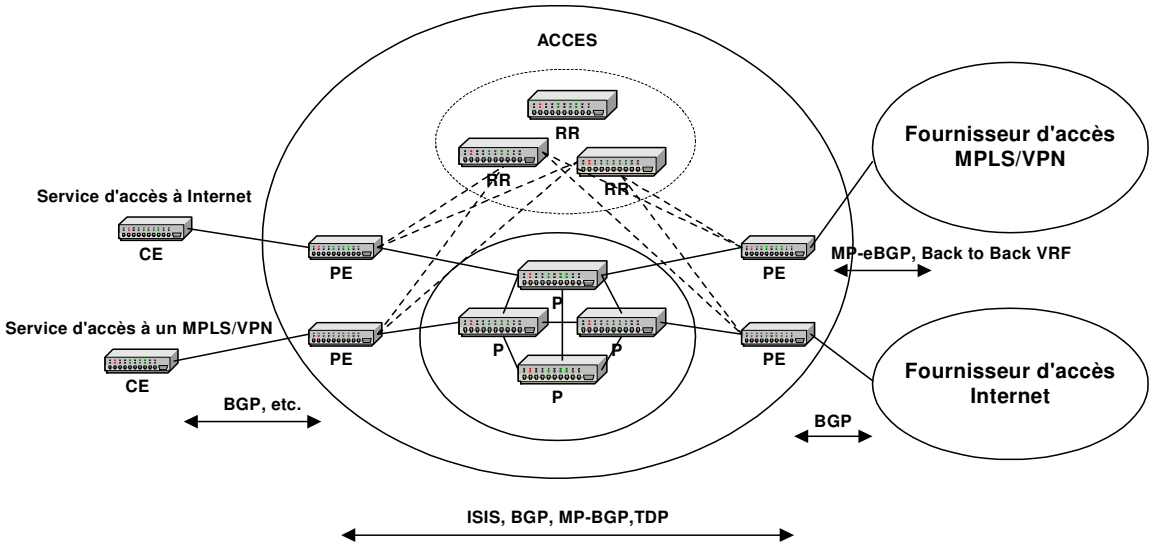


Figure 1: Réseau multi-services

Dans le cadre du service MPLS/VPN, seules les configurations des routeurs PE contiennent la définition effective des MPLS/VPN et des accès Internet, les configurations des routeurs P, RR et CE n'ayant aucune connaissance de la configuration des MPLS/VPN. Enfin, différents protocoles de routage (BGP, MP-BGP, IS_IS) s'exécutent au sein du réseau multi-services afin d'offrir les services réseaux décrits précédemment.

1.3. Les protocoles réseau

1.3.1. Le protocole MPLS

Comme il l'a été brièvement expliqué en introduction, le principe de base de MPLS est la commutation de labels. Ces labels, simples nombres entiers, sont insérés entre les en-têtes de niveaux 2 et 3. Les routeurs commutent ces labels tout au long du réseau jusqu'à destination, sans avoir besoin de consulter l'en-tête IP et leur table de routage [RFC3031].

Cette technique, appelée Label Swapping, est similaire à la commutation de cellules sur ATM avec les informations de Virtual Path Identifier/Virtual Circuit Identifier ou à la commutation sur réseau Frame Relay avec les Data Link Connection Identifier [RFC3034]. Les routeurs MPLS situés à la périphérie du réseau, qui possèdent à la fois des interfaces IP traditionnelles et des interfaces connectées au backbone MPLS, sont chargés d'imposer ou de retirer les labels des paquets IP qui les traversent. Les routeurs d'entrées, qui commutent les labels, sont appelés Label Switch Router (LSR) ou Provider (P). Tandis que les routeurs d'entrée ou de sortie, qui ajoutent ou retirent les labels, sont appelés Label Edge Router (LER) ou PE (Provider Edge).

Un label MPLS occupe 4 octets (32-bits) et se présente sous la forme:

- Label (20 bits).
- Exp (3 bits) : Champ expérimental, utilisé pour la QoS. Equivalent au champ TOS de l'en-tête IP.
- S (1 bit) : Champ "bottom of stack". Lorsque ce bit est à 1, le bas de la pile est atteint, et l'en-tête de niveau 3 est placé juste après.
- TTL (8 bits) : Ce champ a le même rôle que le champ TTL de l'en-tête IP.

Le format des labels MPLS est générique et peut notamment être utilisé sur Ethernet, Frame-Relay, etc.

1.3.2. Le protocole IP

Le protocole IP est né avec les premiers réseaux de recherche américains dans le milieu des années 1970. Mais ce n'est qu'avec l'apparition en 1993 du premier navigateur MOSAIC, qu'il a été largement diffusé. La version actuelle d'IP (IPv4) a depuis été retenue par l'ensemble des constructeurs informatiques comme le protocole universel de niveau réseau.

Le protocole IP fait partie de la couche Internet de la suite de protocoles TCP/IP, et permet l'élaboration et le transport des datagrammes IP, sans toutefois en assurer la livraison. Concrètement, le protocole traite les datagrammes IP indépendamment les uns des autres en définissant leur représentation, leur routage et leur expédition [RFC791].

Le protocole IP détermine le destinataire du message grâce à 3 champs :

- Le champ adresse IP : Ce champ indique l'adresse de la machine.
- Le champ masque de sous-réseau : Un masque de sous-réseau permet au protocole IP de déterminer la partie de l'adresse IP qui concerne le réseau.
- Le champ passerelle par défaut : Ce champ permet de savoir à quelle machine remettre le datagramme si jamais la machine de destination n'est pas sur le réseau local.

Les données circulent sur Internet sous forme de datagrammes. Les datagrammes sont des données encapsulées, c'est-à-dire des données auxquelles on a ajouté des en-têtes correspondant à des informations sur leur transport. Les données contenues dans les datagrammes sont analysées et éventuellement modifiées par les routeurs.

Voici la signification des différents champs :

- Version (4 bits) : Ce champ indique la version du protocole IP que l'on utilise afin de vérifier la validité du datagramme.
- Longueur d'en-tête (4 bits) : Ce champ indique le nombre de mots de 32 bits constituant l'en-tête.
- Type de service (8 bits) : Ce champ indique la façon selon laquelle le datagramme doit être traité.
- Longueur totale (16 bits) : Ce champ indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données.
- Identification, drapeaux et déplacement de fragment sont des champs qui permettent la fragmentation des datagrammes.
- Durée de vie appelée aussi TTL (Time To Live) (8 bits) : Ce champ indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme.
- Protocole (8 bits) : Ce champ permet de savoir de quel protocole est issu le datagramme comme les protocoles suivants :
    - ICMP: 1
    - IGMP: 2
    - TCP: 6
    - UDP: 17
- Somme de contrôle de l'en-tête (16 bits) : Ce champ permet de contrôler l'intégrité de l'en-tête afin de déterminer si celui-ci n'a pas été altéré pendant la transmission. La somme de contrôle est le complément à un de tous les mots de 16 bits de l'en-tête.
- Adresse IP source (32 bits) : Ce champ représente l'adresse IP de la machine émettrice, il permet au destinataire de répondre.
- Adresse IP destination (32 bits) : Ce champ indique l'adresse IP du destinataire du message.

La fragmentation d'un datagramme se fait au niveau des routeurs, c'est-à-dire lors de la transition d'un réseau dont le MTU (Maximum Transfer Unit) est plus faible. Si le datagramme est trop grand pour passer sur le réseau, le routeur va le fragmenter, c'est-à-dire le découper en fragments de tailles inférieures au MTU du réseau. Le routeur va ensuite envoyer ces fragments de manière indépendante.

1.4. Les protocoles de routage

Tous les protocoles de routage ont pour objectif de maintenir les tables de routage du réseau dans un état intègre et cohérent. Pour y parvenir, les protocoles diffusent des informations de routage aux autres systèmes du réseau afin de transmettre les modifications des tables de routage. Ces protocoles réceptionnent en contrepartie les informations de routage d'autres systèmes du réseau afin de mettre à jour les tables de routage.

Dans les premiers grands réseaux, les tables de routage étaient statiques et donc maintenues à jour par des techniciens de bout en bout. De nos jours, les mises à jour des tables de routage et le calcul du meilleur chemin sont automatiquement propagés sur le réseau par les protocoles de routage.

Voici une liste non exhaustive des algorithmes qui peuvent être mis en œuvre lors du processus de routage. Les algorithmes de routage conçus doivent être les plus simples possibles afin d'être efficace pour le calcul et la propagation des mises à jour des tables de routage :

- Algorithmes de routage à trajet unique ou multiple : Ils permettent le multiplexage du trafic sur plusieurs lignes.
- Algorithmes de routage hiérarchique : Ils définissent des groupes logiques de nœuds, appelés domaines, systèmes autonomes ou zones. Certains routeurs peuvent communiquer avec les routeurs d'autres domaines, alors que d'autres routeurs ne peuvent communiquer qu'à l'intérieur de leur propre domaine, simplifiant ainsi les algorithmes en fonction des exigences de routage des routeurs appartenant à la hiérarchie.
- Algorithmes de routage absolu : Par opposition aux algorithmes hiérarchiques, ces algorithmes fonctionnent dans des réseaux homogènes, dans lesquels tous les routeurs sont sur un même pied d'égalité.
- Algorithmes de routage intra domaine : Ils ne fonctionnent que dans les limites d'un domaine.
- Algorithmes de routage inter domaine : Ils fonctionnent tant au sein d'un domaine qu'entre divers domaines.
- Algorithmes de routage d'état des liens : Ils testent régulièrement l'état des liens avec leurs voisins et diffusent périodiquement ces états à tous les autres routeurs du domaine. L'algorithme du plus court chemin est généralement fondé sur l'algorithme de Dijkstra, qui calcule le plus court chemin vers chaque destination. Les avantages de tels algorithmes sont

d'offrir une convergence rapide sans boucle et à chemins multiples. De plus, chaque passerelle calcule ses propres routes indépendamment des autres, et les métriques sont généralement précises et couvrent plusieurs besoins. En revanche, ces algorithmes sont souvent plus complexes à mettre en œuvre et consommateurs de ressources.

- Algorithmes de routage à vecteur de distance : Ils diffusent régulièrement aux voisins l'état des routes. En se fondant sur les routes reçues, chaque voisin met à jour sa propre table en fonction de l'adresse du réseau destination, de celle du routeur permettant d'atteindre le réseau destination et du nombre de sauts nécessaire pour l'atteindre. Le calcul de routes distribuées s'appuie le plus souvent sur l'algorithme de Bellman-Ford. Les avantages d'un tel algorithme sont une forte interopérabilité entre les systèmes du réseau et de faibles impacts sur les ressources système. La convergence des tables de routage se montre en revanche lente lorsque les réseaux deviennent grands, la taille des informations de routage étant proportionnelle au nombre de réseau. De plus, des phénomènes de bouclage peuvent intervenir.

Suivant l'algorithme utilisé, plusieurs paramètres peuvent intervenir lors d'une décision de routage. Les critères de routage s'appuient généralement sur les éléments suivants :

- Longueur du trajet : Définit un critère de décision à partir du nombre de liens qu'un paquet doit traverser pour se rendre du point d'origine au point de destination.
- Fiabilité : Définit un critère de décision fondé sur la fiabilité de chaque lien du réseau.
- Délai de transmission : Définit un critère de décision fondé sur le temps requis afin d'acheminer un paquet du point d'origine au point de destination.
- Largeur de bande : Définit un critère de décision fondé sur la capacité de transmission d'un lien.
- Charge : Définit un critère de décision fondé sur les ressources d'un routeur comme le nombre de paquets traités par seconde, ressource mémoire, etc.
- Coût de la communication : Définit un critère de décision fondé sur un coût appliqué à un lien.

IGP (Interior Gateway Protocol) et EGP (Exterior Gateway Protocol) sont les deux grandes familles de protocoles de routage dans les réseaux IP. Un réseau de routage est découpé généralement en systèmes autonomes, dits AS (Autonomous System), ou zones de responsabilité. Dans un système autonome, le protocole de routage utilisé est de type IGP. Pour les échanges de routage entre systèmes autonomes différents, le protocole de routage utilisé est de type EGP.

1.4.1. Le protocole de distribution des labels

Les P routeurs se basent sur l'information de label pour commuter les paquets au travers du backbone MPLS. Chaque routeur, lorsqu'il reçoit un paquet, utilise le label pour déterminer l'interface et le label de sortie. Il est donc nécessaire de propager les informations sur ces labels à tous les P routeurs [RFC3036].

Pour cela, des protocoles de distribution de labels sont utilisés. Différents protocoles sont employés pour l'échange de labels entre LSR:

- TDP/LDP (Tag/Label Distribution Protocol) : Utilisé pour faire correspondre des labels à des adresses IP unicast.
- RSVP (Resource Reservation Protocol) : Utilisé en "Traffic Engineering" pour établir des LSP en fonction de critères de ressources et d'utilisation des liens.
- MP-BGP (MultiProtocol Border Gateway Protocol) : Utilisé pour l'échange de routes VPNv4.

Pour échanger les labels correspondants aux routes IP unicast apprises par un protocole IGP, les routeurs CISCO emploient le protocole TDP, utilisant TCP sur le port 711. Ce protocole est un protocole propriétaire défini par CISCO. Le protocole défini par l'IETF est LDP, qui utilise TCP sur le port 646. Bien que ces deux protocoles soient fonctionnellement identiques, ils sont incompatibles entre eux, à cause de différences dans le format des paquets.

1.4.2. Les protocoles de routage interne IGP

1.4.2.1. Les caractéristiques fondamentales

Les protocoles IGP sont conçus pour gérer le routage interne d'un réseau avec des objectifs de forte convergence des nouvelles routes injectées dans les tables de routage. Les décisions de routage s'appuient sur une unique métrique afin de favoriser la fonction de convergence. Le nombre d'entrées dans les tables de routage doit aussi être limité afin de renforcer la convergence des tables de routage.

Parmi les protocoles les plus utilisés, citons principalement:

- OSPF (Open Shortest Path First) : Protocole de routage à état de liens fondé sur le calcul des chemins les plus courts et sur une architecture hiérarchique constituée de zones OSPF.
- IS_IS (Intermediate System to Intermediate Systems) : Protocole de routage à état de liens fondé sur le calcul des chemins les plus courts et sur une architecture hiérarchique constituée de domaines IS_IS.

D'autres protocoles, tels RIP (Routing Information Protocol) ou IGRP (Interior Gateway Routing Protocol), existent et font partie des protocoles de routage à vecteur de distance.

1.4.2.2. Le protocole de routage IS_IS

IS_IS est un protocole interne de routage. Issu de l'ensemble des protocoles OSI (Open Systems Interconnexion), il fournit un support pour la mise à jour d'informations de routage entre de multiples protocoles. Il s'agit d'un protocole par état des liaisons de type SPF (Shortest Path First) dont la dernière version est conforme à la norme ISO 10589 [IS_IS 1992, RFC1195].

Le routage IS_IS utilise deux niveaux hiérarchiques de routage. La topologie de routage IS_IS est découpée en aires de routage de niveaux 1 ou 2. Les routeurs de niveau 1 connaissent la topologie dans leur aire, incluant tous les routeurs de cette aire. Cependant, ces routeurs de niveau 1 ne connaissent ni l'identité des routeurs, ni les destinations à l'extérieur de leur aire. Ils routent tout le trafic vers les routeurs interconnectés au niveau 2 dans leur aire.

Les routeurs de niveau 2 connaissent la topologie réseau du niveau 2 et savent quelles adresses sont atteignables pour chaque routeur. Les routeurs de niveau 2 n'ont pas besoin de connaître la topologie à l'intérieur d'une aire de niveau 1. Seuls les routeurs de niveau 2 peuvent échanger les paquets de données ou les informations de routage directes avec les routeurs externes situés en dehors de leur aire de routage.

Le domaine de niveau 2 agit comme domaine d'échange entre les aires de niveau 1. La figure suivante illustre une topologie de routage des aires IS_IS de niveaux 1 et 2 [RFC1142] :

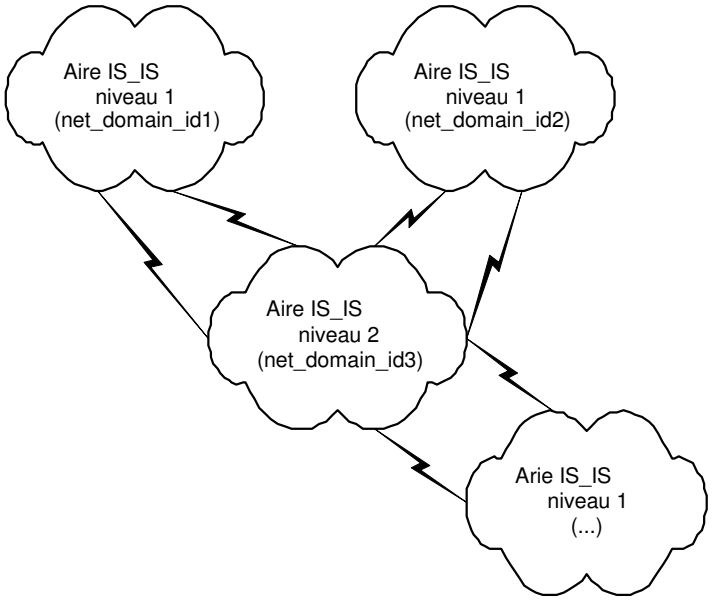


Figure 2 : Topologie de routage des aires IS_IS

Pour des raisons de disponibilité des sessions de routage entre les aires, le nombre de sessions de routage entre les aires 1 et 2 doit être supérieur au minimum à deux sessions. Le réseau de routage formé par les routeurs des aires 1 ou 2 doit définir un graphe connexe [Lacomme et al. 2003]. De plus, l'aire de niveau 2 est un point d'articulation du graphe.

La figure suivante illustre une topologie de routage des équipements réseau au sein d'une aire IS_IS de niveau 1 ou 2 [RFC1142] :

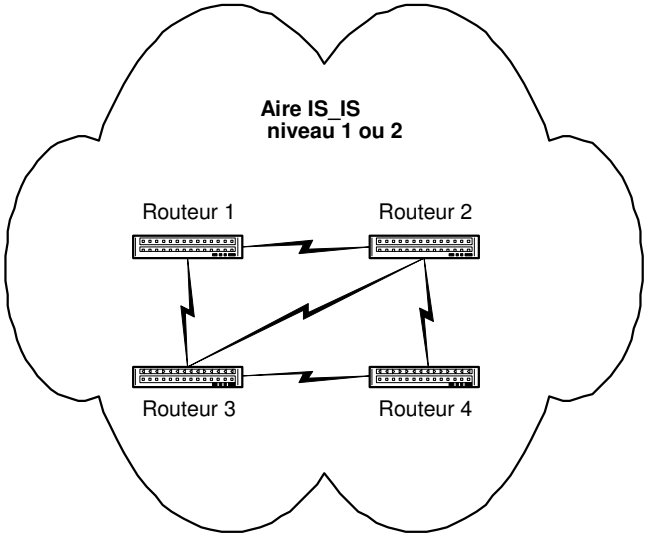


Figure 3 : Topologie de routage des équipements réseaux au sein d'une aire IS_IS

Le graphe doit être connexe et sans point d'articulation pour assurer la disponibilité du réseau [Lacomme et al. 2003].

1.4.3. Les protocoles de routage externe EGP

1.4.3.1. Les caractéristiques fondamentales

Les protocoles EGP sont conçus pour gérer le routage externe d'un réseau avec des objectifs de convergence et d'optimisation de nouvelles routes injectées dans les tables de routage du réseau. Généralement, le nombre d'entrées dans les tables de routage est souvent important. Par exemple, un routeur qui comporterait l'ensemble des routes actuelles d'Internet compterait près de 100 000 entrées.

Parmi les protocoles EGP utilisés de nos jours, c'est le protocole de routage BGPv4 qui s'est imposé.

1.4.3.2. Le protocole de routage BGP

BGP est un protocole de routage externe qui a remplacé EGP. Le protocole BGP permet d'échanger des informations d'accessibilité entre les AS. Il peut notamment fournir des informations détaillées concernant chaque route et les utiliser pour sélectionner le meilleur chemin [RFC1774].

Les sessions de routage au sein d'un AS sont appelées sessions iBGP. Les sessions de routage entre AS sont appelées sessions eBGP. BGP ne transmet pas de métrique dans les mises à jour de routes, mais transmet la meilleure route vers chaque système autonome susceptible d'être adoptée pour atteindre une destination donnée.

Lorsqu'un routeur BGP reçoit des mises à jour en provenance de plusieurs systèmes autonomes décrivant différents chemins vers une même destination, il choisit alors le meilleur itinéraire pour l'atteindre et le propage vers ses voisins.

Une décision de routage est fondée sur plusieurs attributs comme :

- AS-path : Cet attribut liste les numéros de tous les AS qu'une mise à jour a traversé pour atteindre une destination.
- Origin : Cet attribut donne des informations sur l'origine de la route. Ces informations peuvent être de type IGP (la route annoncée provient du même système autonome que l'annonceur), de type EGP (la route est apprise et ne provient pas du même système autonome) ou Incomplète (la route est apprise d'une autre manière).
- Next hop : Cet attribut contient l'adresse IP du routeur vers lequel l'information doit être émise pour atteindre le réseau.
- Weight : Cet attribut est utilisé dans le processus de sélection de chemin lorsqu'il existe plus d'une route vers une même destination. Cet attribut de poids est local et n'est pas propagé dans les mises à jour de routage.
- Local preference : Cet attribut a un rôle similaire à l'attribut de poids, si ce n'est qu'il fait partie des informations de mise à jour de routage.
- Multi-exit discriminator: Cet attribut indique aux routeurs voisins externes le chemin à privilégier vers un AS lorsque celui-ci possède plusieurs points d'entrée.
- Community : Cet attribut est utilisé pour grouper des destinations auxquelles des décisions de routage peuvent être appliquées.

Le premier niveau de topologie de routage définit les différentes sessions de routage entre les systèmes autonomes participant au routage BGP. Le découpage en différents AS dépend de nombreux paramètres, tels que le nombre d'équipements réseau, la localisation géographique, etc. Pour des raisons de disponibilité et de résilience des connexions entre les systèmes autonomes, le nombre de sessions de routage entre les systèmes autonomes de l'opérateur de télécommunications doit être supérieur au minimum à deux sessions.

La figure suivante illustre une topologie de routage d'un réseau d'un opérateur de télécommunications avec ses partenaires et ses clients au niveau des systèmes autonomes :

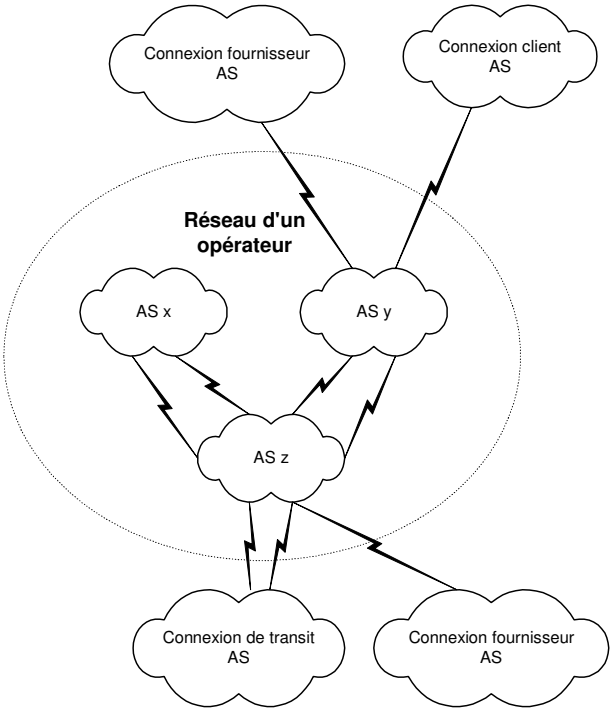


Figure 4 : Topologie de routage des systèmes autonomes BGP

Le graphe doit être connexe pour assurer la disponibilité du réseau [Lacomme et al. 2003]. Il doit être noté que le graphe peut avoir des points d'articulation.

Le second niveau de topologie de routage concerne le découpage d'un système autonome en sous-systèmes autonomes, ou SubAS. Ce type de topologie est un modèle de type confédération, qui n'est plus vraiment utilisé du fait de la difficulté de maintenir la cohérence des configurations.

La figure suivante illustre une topologie de routage des sous-systèmes autonomes au sein d'un système autonome [RFC3065] :

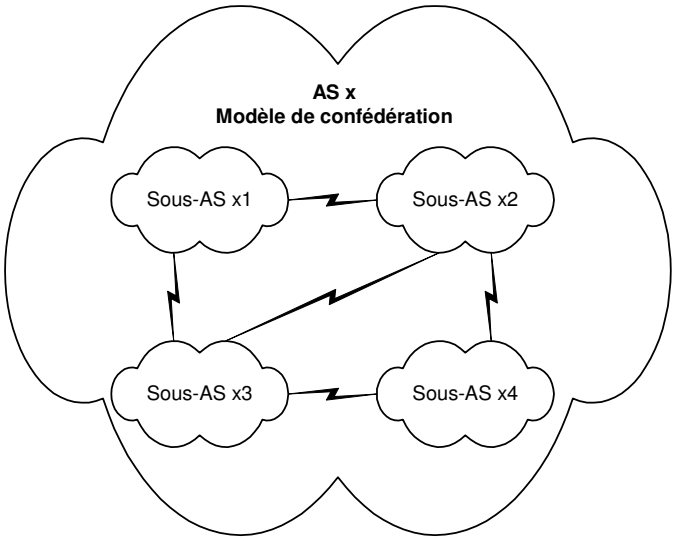


Figure 5: Topologie de routage des sous-systèmes autonomes BGP

Le dernier niveau de topologie de routage se situe au niveau d'un système autonome. Il est composé de plusieurs modèles possibles. Dans le modèle "full-meshing", tous les équipements ont une session de routage BGP avec les autres équipements. Dans le modèle de type "route reflector", tous les équipements "réflecteurs de routes" ont une session de routage BGP avec les autres équipements "réflecteurs de routes". De plus, tous les équipements non "réflecteurs de routes" ont au minimum deux sessions de routage BGP avec des équipements "réflecteurs de routes" comme l'illustrent les figures suivantes [RFC2796] :

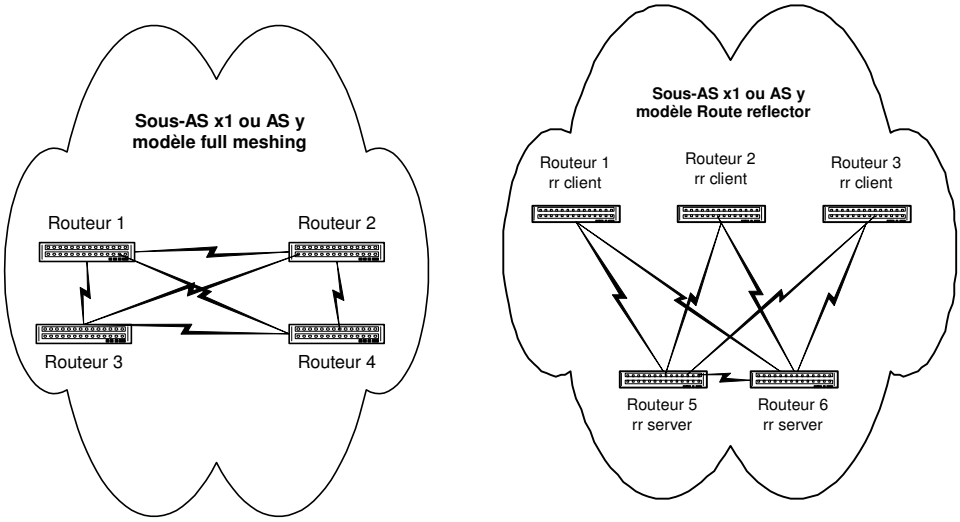


Figure 6 : Topologies de routage des équipements réseau BGP

Dans les deux cas, les graphes doivent être connexes et sans points d'articulation pour assurer la disponibilité du réseau de routage [Lacomme et al. 2003].

1.4.3.3. Le protocole de routage MP-BGP et les réseaux privés virtuels

Un réseau privé virtuel MPLS/VPN permet de connecter des sites distants sur un réseau partagé par tous les clients. Le trafic du réseau privé virtuel est isolé logiquement des autres trafics VPN. Cette isolation est réalisée par un mécanisme de routage fondé sur le protocole MP-BGP, qui est une extension du protocole de routage BGP pour les réseaux MPLS.

Le protocole MP-BGP fonctionne en collaboration avec un protocole de distribution de labels afin d'associer un label à une route externe. Dans ce cas, deux niveaux de labels sont utilisés, le premier label correspond à la route dans le VPN concerné et le second label correspond au PE permettant d'atteindre le prochain saut BGP [RFC2547].

De plus, chaque VPN peut faire transiter les classes d'adresses IP qu'il désire sans qu'il y ait de conflit d'adresses IP avec d'autres VPN. Chaque VPN a en effet sa propre table de routage et, sur les réseaux MPLS, la commutation du trafic réseau est réalisée sur des labels uniques et non sur des adresses IP. Pour cela, un identifiant appelé RD (Route Distinguisher) est accolé à chaque subnet

IPv4 afin de créer une route VPNv4. En revanche, dans le cas d'un Extranet ou d'un fournisseur de services, les adresses IP devront être uniques afin de partager les ressources communes.

La sécurité logique d'un MPLS/VPN repose sur la configuration logique du VPN dans les configurations des routeurs PE. Pour mieux comprendre les enjeux de configuration des MPLS/VPNs, prenons l'exemple de deux VPNs (rouge, bleu), que nous allons définir afin de relier deux sites différents pour chacun des VPNs, comme illustré à la figure suivante :

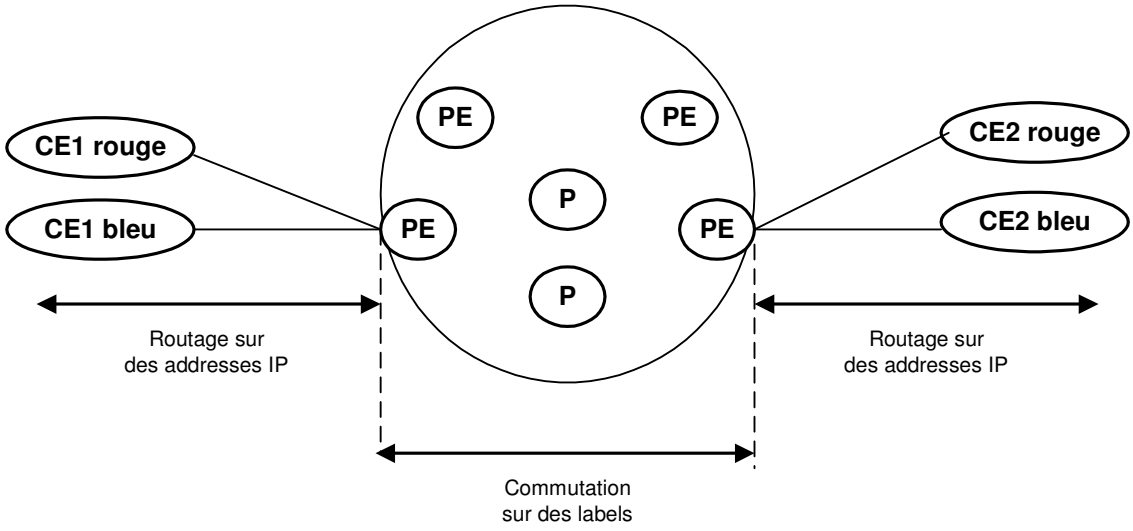


Figure 7 : Exemple de réseaux privés virtuels MPLS/VPNs

Nous avons vu que le RD permet de garantir l'unicité des routes VPNv4 échangées entre les PEs, mais ne définit pas la manière dont les routes vont être insérées dans les VPNs. Pour y parvenir, l'import et l'export de routes sont réalisés à l'aide d'une communauté étendue BGP appelée Route-Target (RT). Les routes-targets doivent être vues comme des filtres appliqués sur les routes VPNv4.

Dans notre exemple, les routeurs "CE1 A" et "CE2 A" appartiennent au "MPLS/VPN A" et les routeurs "CE1 B" et "CE2 B" appartiennent au "MPLS/VPN B". La configuration des routeurs PE permet de créer ces VPNs sur le réseau par les configurations décrites ci-dessous des deux PE (implémentation CISCO). Nous utiliserons le terme VRF (Virtual Routing Forwarding) par la suite pour désigner un VPN.

Configuration du routeur PE, connecté à CE1 A et CE1 B :

```
# Définition du MPLS/VPN A
ip vrf A

  # La valeur du rd (route distinguisher) permet d'identifier les routes échangées
  # entre les routeurs PE pour chaque MPLS/VPN
```

```
   rd 1

   # Les valeurs des route-targets RT permettent de définir un MPLS/VPN.
   # Le MPLS/VPN A n'accepte que les routes  reçues véhiculant le RT 1 et
   # exporte les routes apprises de son côté au réseau MPLS en insérant le RT 1
   route-target import 1
   route-target export 1
!
# Définition du MPLS/VPN B
ip vrf B
  rd 2
  route-target import 2
  route-target export 2
!
# Connexion de CE1 A au PE : Cette connexion appartient au MPLS/VPN A
interface …
  ip vrf forwarding A
  …
!
# Connexion de CE1 B au PE : Cette connexion appartient au MPLS/VPN B
interface …
  ip vrf forwarding B
  …
!
# Description de la session de routage avec le MPLS/VPN A
address-family ipv4 vrf A
 neighbor 10.10.10.102 activate
 neighbor 10.10.10.102 send-community
 neighbor 10.10.10.102 as-override
!
# Description de la session de routage avec le MPLS/VPN B
address-family ipv4 vrf B
 neighbor 192.10.10.102 activate
 neighbor 192.10.10.102 send-community
 neighbor 192.10.10.102 as-override
!
```

Configuration du routeur PE, connecté à CE2 A et CE2 B :

```
# Définition du MPLS/VPN A
ip vrf A

  # La valeur du rd (route distinguisher) permet d'identifier les routes échangées
  # entre les routeurs PE pour chaque MPLS/VPN
  rd 3
```

```
  # Les valeurs des route-targets RT permettent de définir un MPLS/VPN.
  # Le MPLS/VPN A n'accepte que les routes reçues véhiculant le RT 1
  # et exporte les routes apprises de son côté au réseau MPLS en insérant le RT 1
  route-target import 1
  route-target export 1
!
# Définition du MPLS/VPN B
ip vrf B
  rd 4
  route-target import 2
  route-target export 2
!
# Connexion de CE2 A au PE : Cette connexion appartient au MPLS/VPN A
interface …
  ip vrf forwarding A
  …
!
# Connexion de CE2 B au PE : Cette connexion appartient au MPLS/VPN B
interface …
   ip vrf forwarding B
   …
!
# Description de la session de routage avec le MPLS/VPN A
address-family ipv4 vrf A
 neighbor 10.10.10.104 activate
 neighbor 10.10.10.104 send-community
 neighbor 10.10.10.104 as-override
!
# Description de la session de routage avec le MPLS/VPN B
address-family ipv4 vrf B
 neighbor 192.10.10.104 activate
 neighbor 192.10.10.104 send-community
 neighbor 192.10.10.104 as-override
!
```

L'isolation d'un MPLS/VPN repose donc sur la configuration logique des PE routeurs. Le périmètre d'un MPLS/VPN peut être déterminé uniquement à partir de toutes les configurations des PE routeurs constituant le réseau MPLS/VPNs.

1.4.3.4. Les interconnexions des réseaux MPLS/VPNs

Les opérateurs de télécommunications développent de plus en plus les services réseaux MPLS/VPNs et ont besoin de s'interconnecter pour étendre un

MPLS/VPN sur un autre réseau. Cette interconnexion entre deux réseaux MPLS/VPNs se réalise de deux manières possibles, soit sur le modèle "VRF-To-VRF, soit sur le modèle "MP-eBGP" que nous allons décrire ci-après.

A) Le modèle "VRF-To-VRF"

Comme l'illustre la figure suivante, le modèle "VRF-To-VRF" permet d'interconnecter en point à point un VPN entre les réseaux des opérateurs de télécommunications MPLS/VPN:

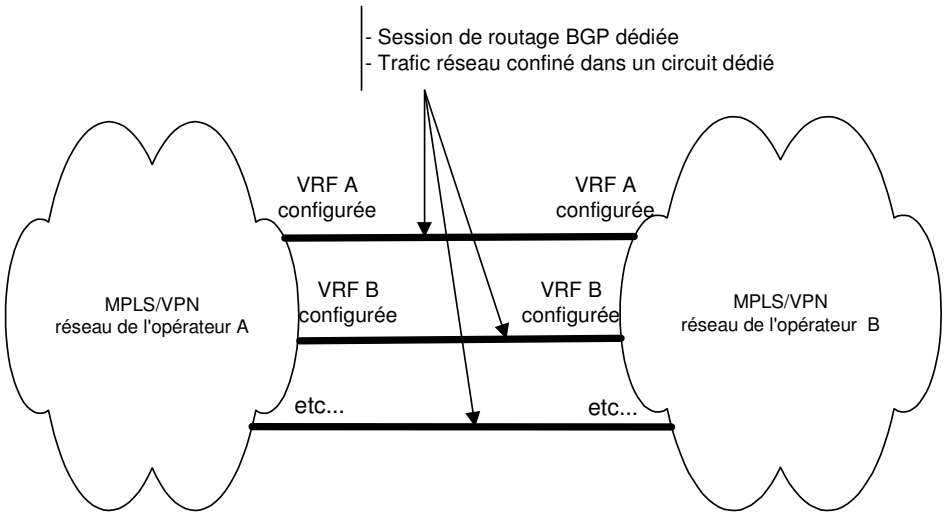


Figure 8 : Modèle d'interconnexion "VRF-To-VRF"

Les avantages de ce modèle sont les suivants :

- Le confinement des VPNs dans des tunnels dédiés en point à point. L'isolation entre les VPNs interconnectés est ainsi renforcée.
- La granularité d'analyse en cas de problème réseau est fine par l'isolation de configuration des VPNs.
- Il est possible de filtrer le trafic IP par VPN. On peut alors effectivement filtrer le trafic du VPN sur les adresses IP, et/ou sur les ports TCP/UDP par les mécanismes classiques de filtrage des données (i.e. Access Control List).
- Il est possible de filtrer le routage par VPN. On peut effectivement contrôler les adresses IP routées sur le VPN par les mécanismes classiques de filtrage de route (i.e. Prefix-list). Notons qu'il est aussi possible de mettre en œuvre des mécanismes de contrôle de l'instabilité des mises à jour des routes (i.e. Dampening).

Les désavantages de ce modèle sont les suivants :

- La configuration des VPNs devient consommatrice en terme de ressources mémoire si le nombre de VPNs augmente considérablement. Notons alors que le nombre des équipements d'interconnexion devra augmenter entre les deux réseaux MPLS.

- L'architecture d'interconnexion devient complexe si le nombre des équipements d'interconnexion entre les deux réseaux MPLS augmente considérablement. Rappelons que cette architecture doit assurer la disponibilité réseau de l'interconnexion MPLS/VPN.

B) Le modèle "MP-eBGP"

Comme l'illustre la figure suivante, le modèle "MP-eBGP" permet d'interconnecter de manière globale deux réseaux d'opérateurs de télécommunications MPLS/VPN:

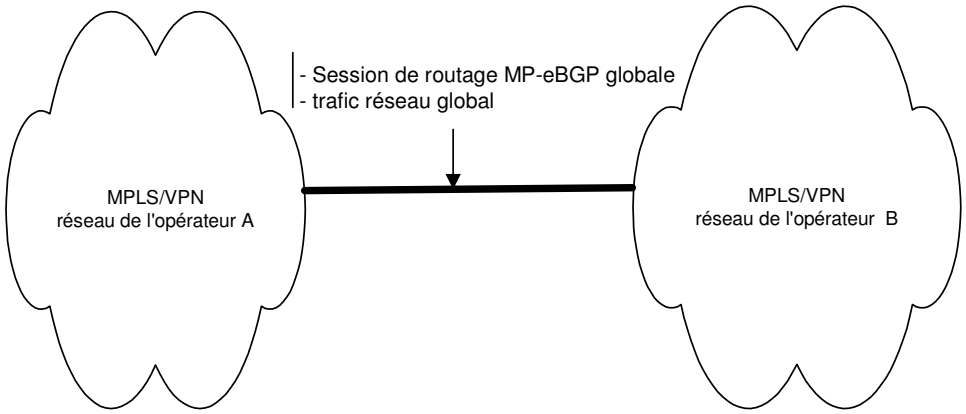


Figure 9 : Modèle d'interconnexion "MP-eBGP"

Les avantages de ce modèle sont les suivants :

- La configuration des VPNs est simplifiée puisqu'on ne définit plus les connections point à point, ni même les VPNs explicitement. Seuls des filtrages, configurés de manière symétrique, des routes-targets entre les deux réseaux des opérateurs de télécommunications permettent de contrôler les interconnexions des VPNs.
- Le modèle est extensible même si le nombre de VPNs à interconnecter devient important. Rappelons encore qu'il n'y a pas de configuration explicite de VPNs à créer.
- L'architecture réseau est simplifiée et extensible même si le nombre de VPNs à interconnecter devient important. Seules les capacités de commutation des équipements d'interconnexion seront impactées.

Les désavantages de ce modèle sont les suivants :

- La granularité d'analyse en cas de problème réseau n'est plus fine, et nécessite en revanche d'analyser les sessions de routage MP-eBGP.
- Il n'y a pas de possibilité de filtrer le trafic IP par VPN (dans la limite des technologies existantes). Cependant, un filtrage est possible au niveau du trafic des données de l'ensemble des VPNs.
- Il n'y a pas de possibilité de filtrer le routage des adresses IP par VPN (dans la limite des technologies existantes). Cependant, un filtrage est

possible au niveau des routes-targets échangées entre les deux réseaux permettant de définir les interconnexions des VPNs.

Les deux modèles (VRF-To-VRF, MP-eBGP) ont leurs avantages et leurs inconvénients. Il doit être cependant noter que non seulement les technologies et les fonctions de contrôle évoluent, mais aussi qu'une interconnexion entre deux réseaux MPLS/VPN peut être un mixte de ces deux modèles.

1.5. Conclusion

Un réseau multi-services est un réseau complexe qui met en œuvre différents protocoles de routage afin d'offrir ses services réseau. De plus, Les différentes topologies de routage définies sur le réseau doivent tenir compte de l'architecture physique du réseau sur lesquelles elles reposent.

Le chapitre suivant décrit les menaces qui pèsent sur un réseau multi-services. Une classification des attaques est aussi détaillée afin de mieux comprendre les éléments à protéger.

Chapitre 2. Les menaces réseau

2.1. Introduction

Les différentes catégories de menaces qui pèsent sur un réseau peuvent être classées comme illustré à la figure suivante :

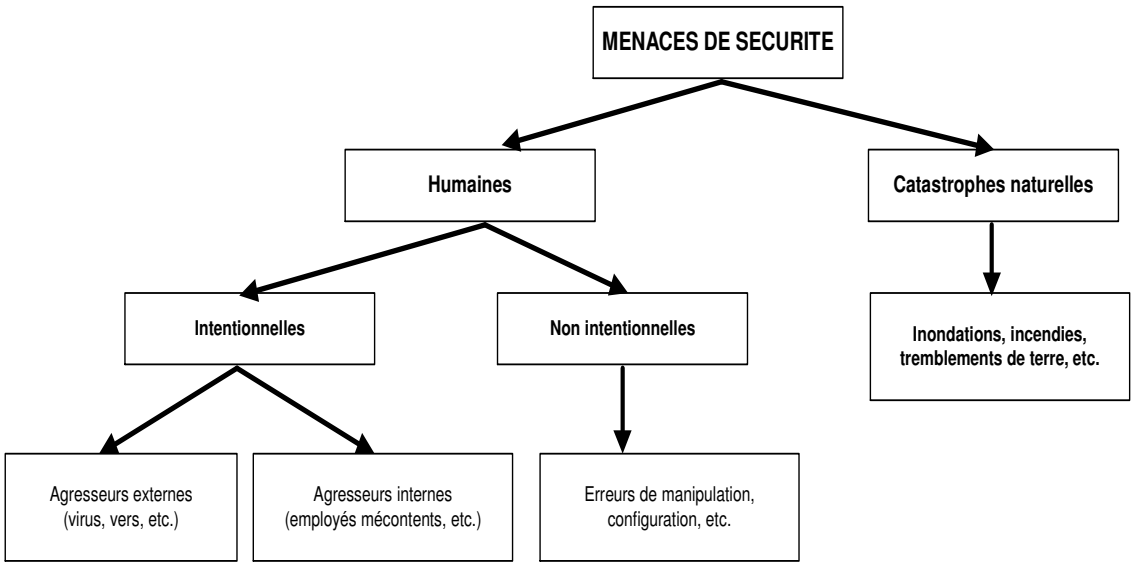


Figure 10: Typologie des menaces

Les menaces non intentionnelles ou imprévisibles, comme les catastrophes naturelles, ne mettent pas en œuvre d'outils ou de techniques particulières et n'ont évidemment pas d'objectif déterminé. À l'inverse, les menaces intentionnelles mettent généralement en œuvre des outils et des techniques d'attaques très variés.

Avec l'avènement d'Internet et des moyens de communication modernes, une nouvelle forme d'insécurité est apparue, qui s'appuie sur l'utilisation de code informatique pour perturber ou pénétrer les systèmes informatiques.

Les attaques touchent généralement les trois composantes suivantes d'un système. La couche réseau, en charge de connecter le système au réseau. Le système d'exploitation, en charge d'offrir un noyau de fonctions au système, et la couche application, en charge d'offrir des services spécifiques. Toutes ces composantes d'un système réseau constituent autant de moyens de pénétration pour des attaques de toute nature comme l'illustre la figure suivante :

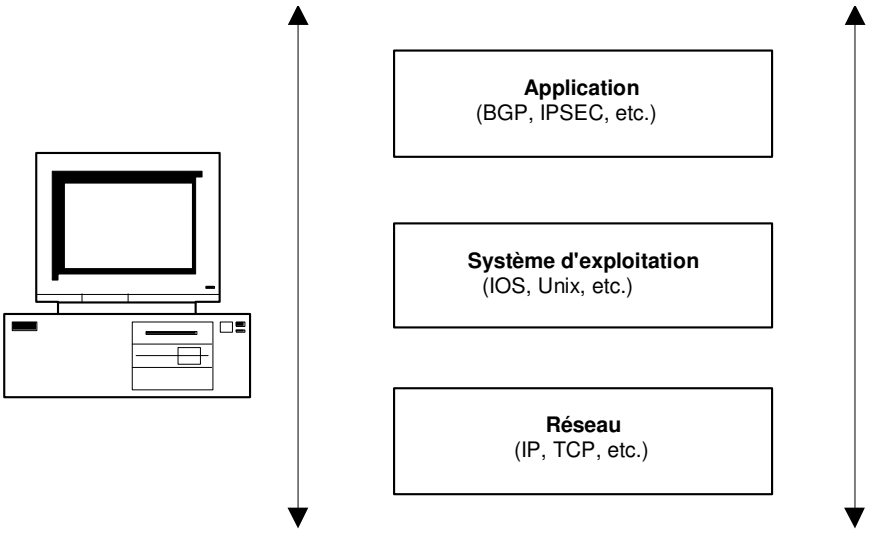


Figure 11 : Composantes d'un système susceptibles d'être attaquées

Les attaques réseau sont aujourd'hui si nombreuses qu'il serait illusoire de prétendre les décrire toutes. Il est cependant important de dresser une typologie des faiblesses de sécurité afin de mieux appréhender ces attaques, qui ont pour point commun d'exploiter des faiblesses de sécurité.

L'objectif de ce chapitre est de cerner les faiblesses les plus couramment exploitées par les attaques et de détailler les mécanismes de ces attaques. Comme tout effet a une cause, les attaques réseau s'appuient sur divers types de faiblesses, que l'on peut classifier par catégorie comme illustré à la figure suivante :

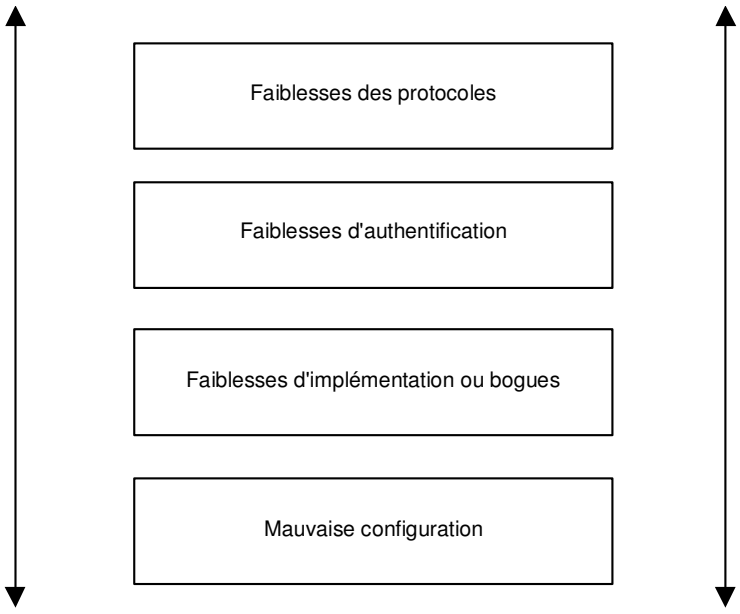


Figure 12 : Typologie des faiblesses de sécurité

Les protocoles réseau sont encore jeunes, et aucun d'eux n'a été conçu pour tenir compte des problèmes de sécurité à l'origine. Le protocole IP, par

exemple, ne comporte pas de couche sécurité. La plupart des protocoles utilisés dans un réseau, tels SNMP (Simple Network Management Protocol) pour la supervision ou BGP pour le routage, n'implémentent pas de couche de sécurité et s'exposent à diverses attaques, comme les attaques par fragmentation, déni de service, etc.

De même, les protocoles réseau n'ont prévu aucun mécanisme d'authentification véritable et subissent des attaques qui s'appuient sur ces faiblesses d'authentification, comme les attaques de type spoofing, man-in-the-middle, etc.

Les faiblesses d'implémentation ou bogues des programmes (système d'exploitation, application de routage, etc.) exposent aussi le réseau à d'autres attaques. La raison en est que les développements de logiciels et de piles réseau se font de plus en plus rapidement et sans règles de codage strictes. Parmi les innombrables attaques qui utilisent de mauvaises implémentations ou des erreurs de programmation, citons les attaques de type SYN flooding et ping-of-death.

Les faiblesses de configuration des équipements réseau peuvent provenir d'une mauvaise configuration d'un pare-feu, laissant passer du trafic non autorisé par la politique de sécurité.

En s'appuyant sur ces faiblesses, le pirate peut alors lancer un ensemble d'attaques afin de récolter des informations, mais aussi de pénétrer un réseau.

2.2. Les faiblesses des protocoles réseau

Les attaques sur les protocoles réseau sont assez fréquentes. Elles sont souvent utilisées pour repérer la topologie réseau et les services disponibles.

D'autres formes d'attaques utilisent des bogues ou de mauvaises implémentations des piles IP/TCP dans les systèmes réseau, par exemple par l'interprétation erronée du codage des RFC (Request For Comments).

La panoplie des attaques réseau est particulièrement large, comme nous le verrons avec les attaques par déni de services distribuées.

A) Attaques permettant d'établir la cartographie du réseau

Les attaques visant à établir la cartographie d'un réseau ont pour but de dresser les artères de communication des futurs systèmes cibles. Elles ont recours pour cela à des outils de diagnostic tel que Traceroute, qui permet de visualiser le chemin suivi par un paquet IP d'un hôte à un autre.

Traceroute utilise l'option durée de vie, ou TTL (Time To Live) du paquet IP afin d'émettre un message ICMP TIME_EXCEEDED (temps dépassé) pour chaque routeur qu'il traverse. Sachant que chaque routeur qui manipule un paquet décrémente le champ TTL, ce champ devient un véritable compteur de tronçon et permet de déterminer l'itinéraire précis suivi par les paquets IP vers un système cible.

B) Attaques permettant d'identifier les systèmes réseau (scanning)

Certaines attaques visent à identifier un système dans le but de dresser les futurs moyens de pénétration de ce système.

Il existe différentes techniques de balayage des systèmes comme illustré à la figure suivante :

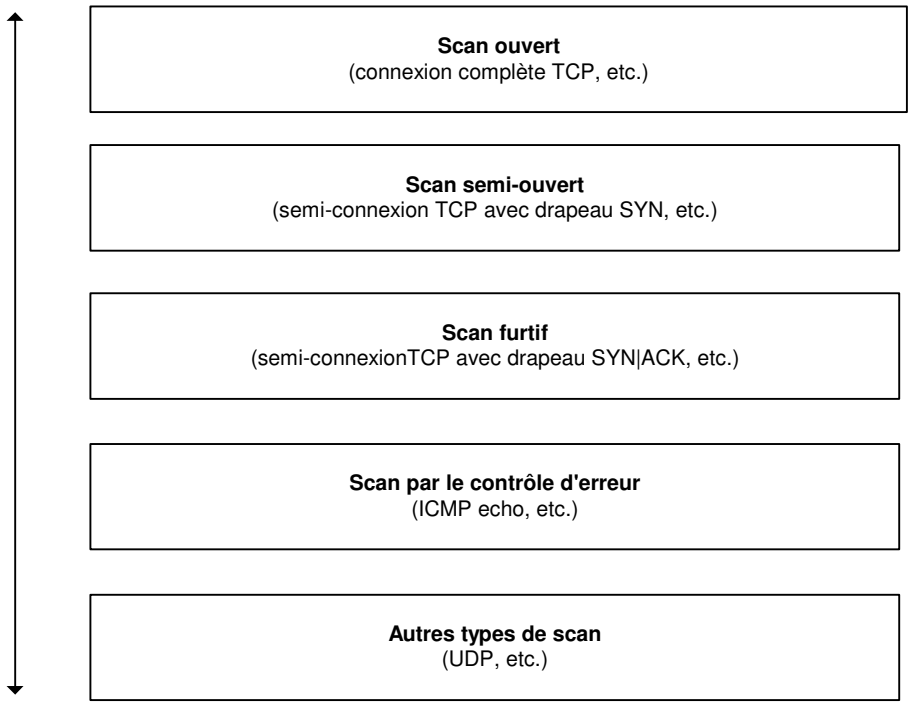


Figure 13 : Les différents types de scanning

En dehors des outils classiques de découverte de services, les outils de scanning permettent de réaliser des prises d'empreinte des systèmes cibles.

La prise d'empreinte d'une pile réseau permet d'identifier très rapidement le système d'exploitation d'un système cible donné. Sachant que chaque éditeur de piles réseau interprète généralement différemment les RFC des protocoles, l'implémentation de la pile TCP/IP est différente d'un système d'exploitation à un autre. En recherchant ces différences, il est possible de reconnaître les différents systèmes d'exploitation.

Les principales techniques de prise d'empreinte active sont les suivantes :

- Sondage par le bit TCP FIN : Bien que la RFC indique qu'il ne faut pas répondre à un tel paquet, certains systèmes d'exploitation ont tendance à répondre par des paquets avec les bits FIN et ACK [RFC793].
- Sondage par balise factice : Si l'on place une balise TCP non définie dans l'en-tête TCP d'un paquet SYN, certains systèmes d'exploitation retournent cette balise dans le paquet émis en réponse.
- Surveillance du bit DF (Don't Fragment) : Certains systèmes d'exploitation définissent ce bit afin d'augmenter les performances.

- Surveillance de la taille initiale de la fenêtre TCP : Dans certaines mises en œuvre de piles, cette taille est unique.
- Analyse des numéros d'acquittement des paquets TCP : Certaines piles implémentent des numéros de séquences spécifiques.
- Analyse du comportement face aux messages ICMP : Les systèmes d'exploitation implémentent des comportements et des réponses parfois différents face à des messages ICMP non conformes [RFC792].

La prise d'empreinte passive consiste à analyser le trafic du système cible afin d'identifier le système d'exploitation. Les données comme la durée de vie TTL du paquet IP, la taille de la fenêtre du paquet TCP ou le bit DF du paquet IP permettent de déterminer un comportement particulier, et donc un système d'exploitation.

C) Attaques permettant d'écouter le trafic réseau (sniffing)

L'attaque par sniffing est généralement utilisée par les pirates pour capturer les mots de passe. Lorsqu'on se connecte à un réseau qui utilise le mode broadcast, toutes les données en transit arrivent à toutes les cartes réseau connectées à ce réseau. En temps normal, seules les trames destinées à la machine sont lues, les autres étant ignorées.

Grâce à un sniffer, il est possible d'intercepter les trames reçues par la carte réseau d'un système pirate et qui ne lui sont pas destinées. Le système pirate se situe donc sur le réseau local et capture tous les paquets réseau transitant sur ce réseau afin d'obtenir des mots de passe.

D) Attaques sur la fragmentation des paquets IP

Les attaques par fragmentation ont été les premières attaques à passer au travers des éléments de filtrage IP réalisés par les pare-feu.

L'attaque par Tiny Fragments consiste à fragmenter sur deux paquets IP une demande de connexion TCP ou d'autres demandes sur une machine cible tout en traversant et en déjouant, par le mécanisme de fragmentation, un filtrage IP.

Le premier paquet IP contient des données comme les huit premiers octets de l'en-tête TCP, c'est-à-dire les ports source et destination et le numéro de séquence. Le second paquet contient la demande de connexion TCP effective.

Les premiers filtres IP appliquaient la même règle de filtrage à tous les fragments d'un paquet. Le premier fragment n'indiquant aucune demande de connexion explicite, le filtrage le laissait passer, de même que tous les fragments associés, sans davantage de contrôle sur les autres fragments. Lors de la défragmentation au niveau IP de la machine cible, le paquet de demande de connexion était reconstitué et passé à la couche TCP. La connexion s'établissait alors malgré le filtre IP.

L'attaque par Fragment Overlapping consiste à fragmenter deux paquets IP au moyen de l'option Overlapping pour faire une demande de connexion TCP ou une autre demande sur une machine cible tout en traversant un filtrage IP.

Le premier paquet IP contient les données de l'en-tête TCP avec les indicateurs à 0. Le second paquet contient les données de l'en-tête TCP avec la demande de connexion TCP. La demande de connexion est fragmentée en deux paquets IP contenant les fragments 0 et 1, chacun d'eux passant le système de filtrage et étant réassemblé par le système cible reconstituant un paquet TCP dû à l'overlapping des fragments 0 et 1.

E) Attaques par déni de service et par inondation (DoS)

Le déni de service est une attaque qui vise à rendre indisponibilité un service, un système ou un réseau. Ces attaques se basent généralement soit sur une faiblesse d'implémentation ou bogue, soit sur une faiblesse d'un protocole. Les premières attaques par déni de service sont apparues entre 1998 et l'an 2000 et visaient de grands sites Internet (Yahoo, Ebay, eTrade, etc.). Concernant le site Yahoo, ce site a été attaqué en février 2000 et a été "noyé" (flood) sous un gigabyte de données en quelques secondes pendant plus de 3 heures d'au moins 50 points réseau différents.

L'inondation est généralement la méthode la plus classique pour empêcher un réseau d'assurer sa mission. Son principe de fonctionnement est simple, une ou plusieurs machines inondent le réseau avec des paquets réseau afin de saturer la bande passante de celui-ci. Une fois que toute la bande est occupée, les autres machines ne peuvent plus travailler, ce qui génère une situation de refus de service.

L'inondation peut recourir à différentes méthodes. La plus classique est le Ping Flooding, où une machine envoie des paquets de ping ICMP REQUEST et attend en réponse un paquet ICMP REPLY. Sans mention d'un délai pour l'obtention de la réponse, la machine envoie ses paquets aussi vite qu'elle le peut, saturant ainsi le réseau.

L'attaque par smurf-and-fraggle est une variante de la précédente qui s'appuie sur une faiblesse de configuration des routeurs. Cette technique consiste à inonder le réseau avec des ping qui n'utilisent que des adresses de broadcast. Pour un paquet envoyé, toutes les machines d'un réseau répondent, ce qui augmente la saturation du réseau. Du fait de l'envoi des paquets ICMP avec une fausse adresse source vers une adresse de broadcast, chaque machine appartenant au réseau couvert par le broadcast répond aux systèmes victimes. Comme le pirate n'attend pas de trafic retour, il peut bombarder un ensemble d'adresses de broadcast et générer un trafic important par ce phénomène d'amplification.

L'attaque DDoS (Distributed Denial Of Services) est un dérivé de la précédente sous une forme distribuée comme illustré à la figure suivante :

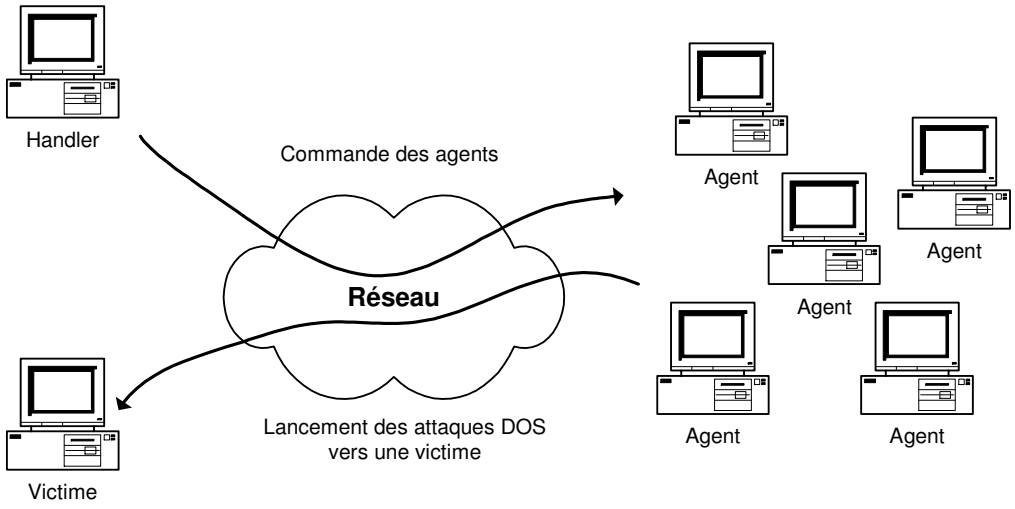


Figure 14: Attaque par déni de service distribué

La première étape consiste à pénétrer par diverses méthodes des systèmes dits handlers, ou maîtres (masters), et agents, ou esclaves (slaves). Le pirate contrôle ensuite dans une deuxième étape directement un ensemble de systèmes handlers, qui contrôlent eux-mêmes un ensemble de systèmes agents. La troisième étape consiste pour le pirate à déclencher son attaque vers un ou plusieurs systèmes cibles donnés. Cet ordre d'attaque aura été donné par les systèmes handlers, qui eux-mêmes auront reçu cet ordre du pirate.

Parmi les nombreuses attaques DDoS, citons TFN (Tribe Flood Network), historiquement la première, et Stacheldraht, qui chiffre les ordres de commandes échangés entre les handlers et les agents dans le champ données des paquets ICMP. Ces attaques ont fait des émules, et d'autres attaques sont apparues, comme Trinoo, qui s'appuie sur UDP pour les communications des ordres entre handlers et agents. Mais aussi TFN2K, une version entièrement revue de TFN, qui introduit des phénomènes de randomisation des ports utilisés pour les communications des ordres entre les handlers et les agents, ainsi qu'un phénomène aléatoire dans le lancement des attaques vers les systèmes cibles.

2.3. Les faiblesses d'authentification

La plupart des protocoles réseau n'ont prévu aucun mécanisme d'authentification véritable et subissent des attaques qui s'appuient sur ces faiblesses d'authentification, comme les attaques de type spoofing, man-in-the-middle.

A) L'attaque IP spoofing

L'attaque IP spoofing consiste à se faire passer pour un autre système en falsifiant son adresse IP. Le pirate commence par choisir le système qu'il veut attaquer. Après avoir obtenu le maximum de détails sur ce système cible, il détermine les systèmes ou adresses IP autorisés à se connecter au système

cible. Le pirate essaye de prévoir le numéro de séquence des paquets du serveur cible en envoyant plusieurs paquets et en analysant l'algorithme d'incrémentation de ce numéro. Le pirate rend inopérante la machine autorisée à accéder au serveur cible, de façon à s'assurer qu'elle ne répond pas au serveur cible. Le pirate falsifie son adresse IP en la remplaçant par celle de la machine invalidée et envoie une demande de connexion au serveur cible. Cette attaque est assez difficile à effectuer car elle se réalise en aveugle, le pirate ne recevant pas les données transmises par le serveur. Il doit maîtriser parfaitement les protocoles de manière à savoir ce qu'attend le serveur à tout moment. D'autres techniques plus évoluées permettent de contourner ce problème, comme les attaques dites man-in-the-middle ou les attaques de routage.

B) L'attaque man-in-the-middle

L'attaque man-in-the-middle consiste à faire passer les échanges réseau entre deux systèmes par le biais d'un troisième, sous le contrôle du pirate. Ce dernier peut transformer à sa guise les données à la volée, tout en masquant parfaitement à chaque acteur de l'échange la réalité de son interlocuteur.

Pour mettre en œuvre l'échange réseau approprié, il faut soit que la machine du pirate se trouve physiquement dans le chemin réseau emprunté par les flux de données, soit que le pirate réussisse à modifier le chemin réseau afin que sa machine devienne un des points de passage.

Au final, l'échange se présente sous l'une des trois formes suivantes :

- Relais transparent : La machine du pirate transforme les données à la volée. Elle veut rester la plus transparente possible et se comporte comme un routeur, conservant toutes les caractéristiques des paquets dont elle assure le transit, à l'exception du contenu.
- Relais applicatif : La machine du pirate assure l'échange entre deux machines A et B. A parle avec la machine du pirate, laquelle parle avec B. A et B n'échangent jamais réellement de données directement.
- Hijacking : La machine du pirate utilise la session engagée entre les deux machines A et B afin que ce soit la machine du pirate qui soit en session avec la machine B. A perd la session avec B, et la machine du pirate continue la session engagée par A sur B. Le hijacking des sessions TCP permet de rediriger un flux TCP en outrepassant les authentifications nécessaires à l'établissement des sessions. Cette attaque porte de manière plus spécifique sur l'analyse des numéros de séquences et des numéros d'acquittements relatifs aux paquets TCP.

C) Attaques de déchiffrage et de pénétration des systèmes par mots de passe

La plupart des protocoles et services réseau associés utilisent une procédure d'authentification fondée sur un couple (compte, mot de passe). Des attaques itératives de pénétration, dites Brute Force Attack, par le biais de séquences de tentatives d'authentification sur des comptes et des mots de passe différents

peuvent se révéler redoutables pour peu qu'elles s'étalent dans le temps afin de laisser le moins de trace possible.

Le déchiffrage des mots de passe implique que le système a déjà été pénétré. Par l'utilisation de ses droits ou d'attaques en escalade de privilèges, l'agresseur obtient la base de données des (comptes, mots de passe). Ces derniers étant généralement codés dans des fichiers, l'agresseur utilise un programme de décodage ou d'attaques itératives pour obtenir les versions non codées.

2.4. Les faiblesses d'implémentation, ou bogues

Les faiblesses d'implémentation, ou bogues, des programmes exposent le réseau à des attaques. Les développements des logiciels et des piles réseau s'effectuant de plus en plus rapidement et sans imposer de règles strictes, les attaques qui réalisent de mauvaises implémentations ou font des erreurs de programmation sont innombrables.

A) L'attaque TCP SYN

La technique d'inondation SYN, ou SYN flooding, n'est pas une inondation simple. Elle s'appuie sur une demande de connexion qui n'aboutit pas et qui sature les ressources du système visé.

Pour gérer les états de connexion entre deux parties, le protocole TCP recourt aux drapeaux URG, ACK, PUSH, RST, SYN et FIN présents dans l'en-tête TCP [RFC793].

La signification des drapeaux est la suivante :

- URG : Positionné à 1 si le champ Pointeur urgent du paquet TCP est utilisé.
- ACK : Positionné à 1 si le champ Numéro d'acquittement du paquet TCP est significatif. Sert à l'établissement de la connexion.
- PSH (PUSH) : Positionné à 1 indique que le paquet fonctionne suivant la méthode PUSH. Lorsqu'un paquet reçu au niveau TCP porte le flag PUSH, TCP le transmet immédiatement à la couche supérieure sans attendre d'autres segments. Cela permet un fonctionnement correct de l'écho lorsque des consoles sont connectées sur des systèmes informatiques. En l'absence de ce drapeau, TCP attend de rassembler plusieurs segments pour les transmettre à la couche supérieure, pour des raisons d'efficacité.
- RST : Sert à réinitialiser la connexion.
- SYN : Sert à synchroniser les numéros de séquence et à établir des connexions.
- FIN. Utilisé pour indiquer que l'émetteur n'a plus de données à émettre et que la connexion peut être libérée.

Les états d'une connexion TCP sont illustrés par les états pris par l'automate illustré à la figure suivante :

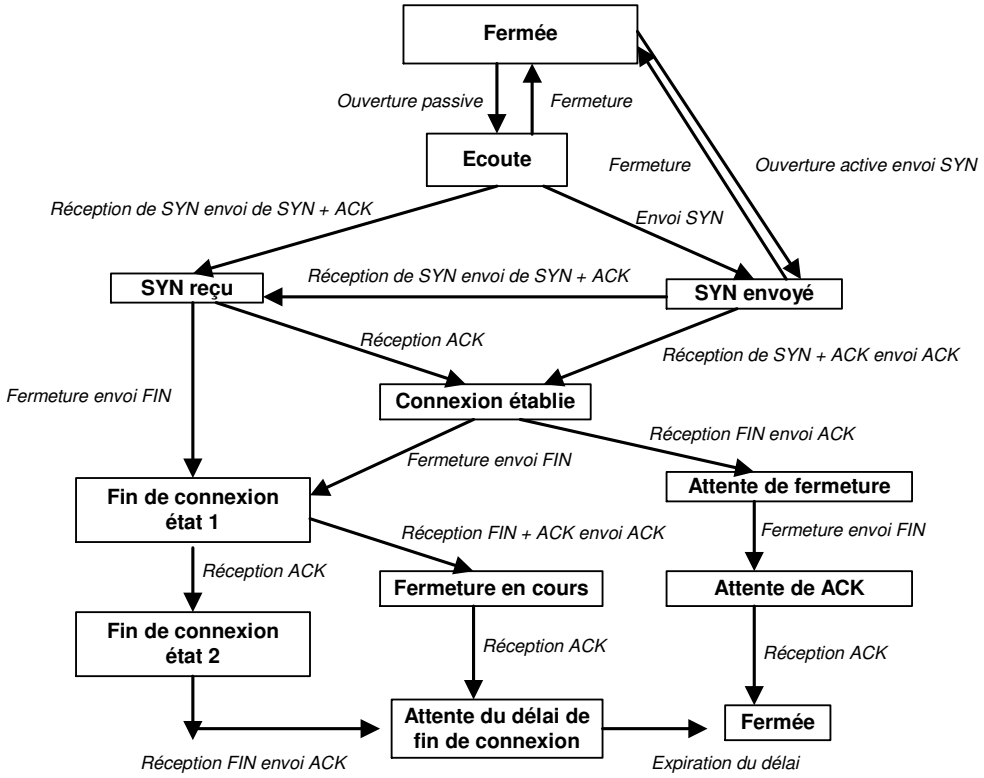


Figure 15 : Les états d'une session TCP

Les attaques de type TCP SYN se fondent sur un comportement prévisible de la pile IP/TCP. D'autres attaques sont imaginables sur tous les états possibles que peut prendre une session TCP.

L'établissement d'une connexion TCP s'appuie sur les indicateurs du paquet TCP. Le client initie une connexion avec un serveur et envoie un paquet TCP avec le bit SYN activé dans les indicateurs TCP (le numéro de séquence initial est renseigné). Le serveur répond au client avec un paquet TCP dont les bits SYN et ACK sont activés (le numéro de séquence est incrémenté). Le client acquitte alors un paquet TCP avec le bit ACK activé (le numéro de séquence est incrémenté).

Le but de cette attaque n'est pas la saturation de la bande passante mais celle de la pile TCP/IP de la machine cible. Comme nous venons de le voir, TCP déclare une connexion ouverte après un double acquittement.

Dans son principe, une telle attaque consiste à initier une connexion TCP vers la machine cible mais de ne pas la terminer. Sachant que chaque connexion qui reste semi-ouverte consomme de la mémoire dans la pile TCP/IP de la machine cible et que cette mémoire n'est libérée qu'après un délai, il y a possibilité d'impacter le système cible si celui-ci ne gère pas correctement la file des connexions. Quand le noyau ne peut plus allouer de ressource, le serveur devient incapable de satisfaire les nouvelles demandes de connexion.

En pratique, on envoie un paquet SYN sur un port en écoute du serveur. Celui-ci répond par un SYN|ACK. La machine attaquante ignorant cette réponse, le serveur doit répéter ses SYN|ACK jusqu'à ce qu'il ait atteint le délai d'expiration. Il détruit alors l'entrée dans le noyau.

B) Les faiblesses du code

Les piles IP/TCP développées par différents constructeurs ou fournisseurs de services manifestent des différences de comportement malgré les définitions des RFC et contiennent de multiples faiblesses, qui peuvent être exploitées par des attaques bien ciblées.

Comme il est théoriquement impossible de vérifier l'absence de bogues dans un programme conçu avec les langages de programmation modernes, il existe une forte probabilité que des bogues permettent à des pirates de gagner des privilèges.

Les principales attaques qui s'appuient sur les erreurs de programmation associées aux piles TCP/IP sont :

- L'attaque "ping de la mort" : Cette attaque consiste à envoyer une suite de fragments d'une requête de type écho ICMP. Une fois réassemblés par la pile IP/TCP du système cible, ces fragments forment un paquet d'une taille supérieure à la taille maximale autorisée (65 507 octets) et peuvent faire déborder les variables internes, provoquant un comportement anormal du système.
- L'attaque "baiser de la mort" : Cette attaque consiste à envoyer un paquet IGMP (Internet Group Management Protocol) mal construit, mettant certains systèmes d'exploitation en refus de service.
- L'attaque "win nuke" : Cette attaque envoie un paquet TCP mal construit avec des données OOB (Out Of Band), mettant certains systèmes d'exploitation en refus de service.
- L'attaque "land" : Cette attaque demande une ouverture de session TCP avec l'adresse source du paquet égale à l'adresse destination et le port source égal au port destination. L'impact de l'attaque peut provoquer des comportements indésirables du système cible.
- L'attaque "teardrop" : Cette attaque envoie un paquet fragmenté de telle façon que les en-têtes du second paquet écrasent ceux du premier, affolant la pile TCP/IP. Cette attaque a été conçue initialement pour les paquets fragmentés ICMP, mais de nombreuses variantes ont été développées depuis pour fonctionner avec n'importe quel type de protocole IP. L'impact de l'attaque peut provoquer des comportements indésirables du système cible.

C) Attaques sur les bogues des systèmes d'exploitation

Les systèmes d'exploitation et les produits ou services additionnels qui y sont greffés contiennent de multiples faiblesses susceptibles d'être exploitées par des attaques ciblées.

La principale de ces attaques est le buffer overflow, ou débordement de tampon, qui consiste à copier plus de données dans un tampon que celui-ci ne peut en contenir. Si les contrôles ne sont pas suffisants, le débordement du tampon permet d'écrire dans la pile d'exécution du programme.

Le débordement de tampon peut permettre à la fois de stocker du code malicieux exécutable et de faire pointer le pointeur d'instruction de retour sur le code exécutable préalablement stocké. Même si le tampon utilisé pour le débordement est trop petit pour insérer du code exécutable malicieux, il reste toujours possible de l'insérer dans une variable d'environnement et de pointer en dernier lieu sur l'adresse de cette variable pour exécuter le code.

D'autres méthodes plus complexes sont utilisées pour mener à bien des attaques par débordement de tampon, qui restent les attaques les plus fréquentes. Tous les systèmes d'exploitation peuvent être visés.

D) Les faiblesses de configuration

Les faiblesses de configuration des équipements réseau, pare-feu, etc., sont également souvent utilisées pour mener à bien des attaques. Ces dernières peuvent provenir de :

- L'exploitation d'erreurs de configuration du système.
- Des configurations des équipements réseau qui doivent suivre des règles strictes afin d'éviter que le réseau ne joue un rôle de rebond dans des attaques éventuelles, notamment l'attaque smurf.
- D'une politique d'accès ou de mots de passe trop laxiste.
- De comptes utilisateurs génériques, standardisés avec des mots de passe triviaux et associés à des droits d'accès permissifs. Dès lors, un pirate peut commencer son intrusion non pas par la recherche de failles exploitables, mais simplement par des tentatives itératives de pénétration. Celles-ci peuvent commencer par les comptes classiques comme oracle, admin, toor, sybase, solaris, linux, etc. et avec des mots de passe identiques aux noms des comptes.

2.5. Les autres formes d'attaques

L'accès physique aux équipements réseau permet de prendre la main en tant qu'administrateur sur pratiquement tous les systèmes actuels.

La copie des configurations des équipements réseau est aussi une attaque redoutable, qui permet au pirate de reconstituer tout le réseau logique ainsi que les protections mises en place. La configuration des équipements réseau est par nature une information confidentielle du réseau.

L'écoute électronique pour récolter des informations peut aussi permettre de mener des attaques bien ciblées. Les diverses techniques d'écoute disponibles de nos jours permettent d'écouter n'importe quel type de média.

Enfin, le vol de secret se rencontre plus fréquemment dans l'ingénierie sociale. Par exemple, l'agresseur entre en contact avec la personne qu'il veut usurper en se faisant passer pour un technicien en intervention bloqué dans son travail par une demande d'authentification ou une permission trop forte. Pour peu qu'il soit convaincant, l'agresseur peut obtenir les comptes-mots de passe ou permissions qu'il désire, voire directement ceux de l'administrateur système.

2.6. Conclusion

Les attaques réseau reposent sur un ensemble de faiblesses de sécurité touchant différents domaines comme les protocoles réseau, les implémentations des piles réseau et les systèmes d'exploitation des systèmes réseau.

De plus, beaucoup d'attaques peuvent impacter le réseau de manière directe ou indirecte. Les virus informatiques sont susceptibles d'impacter le réseau de manière indirecte en provoquant des phénomènes de saturation ou de congestion du réseau. Ces autres formes d'attaques réseau s'appuient principalement sur les faiblesses des applications.

Les attaques détectées sont de plus en plus fréquentes et touchent de plus en plus les infrastructures réseau à la fois des opérateurs de télécommunications et des Entreprises. Par ailleurs, les attaques distribuées à l'aide d'outils automatisés sont faciles d'utilisation. Pour finir, les techniques d'attaques évoluent vers des techniques complexes à programmer mais aussi à détecter.

Après avoir présenté les menaces qui pèsent sur un réseau, nous détaillons dans le chapitre suivant comment définir une politique de sécurité réseau.

Chapitre 3. La politique de sécurité d'un réseau multi-services

3.1. Introduction

La sûreté de fonctionnement d'un système informatique ou réseau est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans les services que ce système lui délivre. Les notions relatives à la sûreté de fonctionnement peuvent être réparties en trois classes [Laprie 1995] :

- Les entraves : elles correspondent aux événements indésirables comme les fautes, les défaillances, les attaques, etc. qui entravent à la sûreté de fonctionnement.
- Les moyens : Ils correspondent aux méthodes et techniques comme la prévention des fautes, l'élimination des fautes, etc. permettant d'assurer un service conforme à la fonction attendue.
- Les attributs : Ils correspondent aux propriétés attendues d'un système comme la disponibilité, l'intégrité, la confidentialité, etc. permettant de définir les fonctions attendues d'un service.

La politique de sécurité d'un réseau se fonde avant tout sur une analyse des risques décrivant les ressources critiques du réseau, ses vulnérabilités, les probabilités d'occurrence des menaces sur ces ressources vitales, ainsi que leurs conséquences.

À partir de cette politique de sécurité, une architecture, des outils et des procédures sont définis et déployés afin de protéger les ressources critiques et de répondre aux objectifs de sécurité.

Les mesures de sécurité à mettre place peuvent être d'ordre divers, demander des ressources plus ou moins importantes et être implémentées dans des délais plus ou moins réalistes.

3.2. Les recommandations générales

Afin d'éviter un certain nombre d'écueils classiques, une politique de sécurité réseau doit respecter un ensemble de principes. Ces principes permettent notamment de bien cerner les enjeux de la rédaction d'un document de politique de sécurité, qui n'est pas un document comme les autres.

Un document de politique de sécurité peut être écrit de plusieurs manières, allant d'un texte unique à un ensemble de politiques de sécurité. Le choix d'écrire un ou plusieurs documents est le plus souvent dicté par la taille du

réseau. Plus le réseau est important, plus il est intéressant de créer des documents séparés, chaque niveau faisant référence au niveau supérieur.

Une politique de sécurité est l'expression du besoin de sécurité. La procédure, ou recommandation technique, est l'implémentation du besoin. Il est donc impératif de distinguer les deux. Lorsque certains produits pare-feu présentent les règles de filtrage comme une politique de sécurité, c'est le concept même de politique de sécurité qui est dévoyé. L'objectif d'une politique de sécurité est d'énoncer des résultats attendus, et non les moyens par lesquels les obtenir.

Les principes énoncés par une politique de sécurité assurent à cette dernière une pérennité beaucoup plus longue que les procédures de sécurité, qui sont appelées à être modifiées fréquemment pour tenir compte des avancées technologiques, des modifications d'architecture, etc.

On peut énoncer les principes suivants :

- Le principe de propriété : Le principe de propriété exige qu'une politique de sécurité décrive, pour chaque ressource, quels en sont les propriétaires. On doit entendre par propriété, non pas l'aspect légal de la propriété d'un bien, mais son aspect fonctionnel, qui consiste à en assurer la pérennité et la protection. Les propriétaires d'une ressource en ont la responsabilité et dictent les règles d'accès à cette ressource. Un schéma classique établit une distinction entre le propriétaire, l'administrateur et l'utilisateur d'une ressource. Le propriétaire définit les règles d'utilisation de ses ressources et les donne à l'administrateur, lequel a pour rôle de les appliquer aux demandes d'un utilisateur. En cas de problème, l'administrateur demande au propriétaire une dérogation aux droits d'accès. L'utilisateur n'est jamais en contact direct avec le propriétaire. Ce mode de fonctionnement garantit une certaine indépendance de l'administrateur face à l'utilisateur.
- L'autorité : La direction générale a autorité sur toutes les ressources du réseau. Elle délègue généralement cette autorité aux responsables de départements, qui peuvent à leur tour mandater un groupe au sein de leur département. Dans tous les cas, l'équipe sécurité, mandatée par la direction générale, dispose de l'autorité de vérifier l'application de la politique de sécurité sur toutes les ressources du réseau. Un comité de sécurité, constitué des responsables du réseau, doit être constitué afin de définir la stratégie sécurité du réseau et de trancher les problèmes de sécurité remontés par l'équipe sécurité ou d'autres départements.
- L'universalité : Le principe d'universalité veut qu'une politique de sécurité dicte des règles qui doivent être non seulement validées, quels que soient les aspects techniques mis en jeu, mais aussi appliquées. L'idée sous-jacente est que la conception initiale d'une politique de sécurité se détache au maximum des aspects technologiques et énonce des règles et principes. Seuls les guides, recommandations ou procédures impliquent des aspects techniques.

- L'orthogonalité : Le principe d'orthogonalité précise qu'une politique de sécurité peut être découpée en sous-parties distinctes, sous la condition que ces sous-parties forment un ensemble cohérent. L'idée sous-jacente est que la conception initiale d'une politique de sécurité et de ses domaines d'application doit être essentielle et fondamentale, de sorte à éviter une évolution inconsistante de la politique de sécurité, de ses guides et de ses recommandations.
- La simplicité : Une politique de sécurité est simple dans sa structure et claire dans les règles qu'elle énonce. Toute mauvaise compréhension d'une règle de la politique de sécurité conduit à ce qu'elle ne soit pas appliquée ou, pire, qu'elle le soit mal.
- L'auditabilité : Une politique de sécurité est auditable. Cela demande que les règles qu'elle énonce puissent être vérifiées dans les faits. Bien qu'il soit difficile de mesurer toute chose, la politique de sécurité est écrite dans cet objectif. L'idée sous-jacente est qu'une politique de sécurité constitue le référentiel ou la pierre angulaire de tout audit ou contrôle de sécurité. Les règles qu'elle énonce doivent pour cela être claires, précises et mesurables.
- La hiérarchie : Une politique de sécurité est structurée en une politique de sécurité de haut niveau, qui englobe les politiques de sécurité couvrant des domaines précis. Ces mêmes politiques de sécurité pointent sur des procédures qui détaillent des aspects techniques du domaine visé. L'idée sous-jacente est qu'une politique de sécurité doit être structurée en sous-politiques de sécurité, dans une approche allant du plus général au plus spécifique. Il est admis que deux à trois niveaux de politiques de sécurité conviennent dans la plupart des cas. Il convient toutefois de prendre garde au piège de l'arborescence des politiques de sécurité, qui pourrait contredire les principes de simplicité et d'orthogonalité.
- L'approbation : Une politique de sécurité est approuvée par la direction générale, et ce de manière officielle. De plus, la direction générale et les ressources humaines s'engagent à réprimer toute violation de la politique de sécurité qui pourrait mettre en péril la survie du réseau. Les cadres juridique et réglementaire couvrant la politique de sécurité et les actes de malveillance doivent être connus de tout le personnel du réseau.

Une politique de sécurité est moins touchée par l'évolution technologique, car elle décrit des besoins et non des moyens. Malgré tout, une politique de sécurité doit être revue régulièrement afin de tenir compte des modifications organisationnelles.

Enfin, une politique de sécurité est réaliste et tient compte à la fois des contraintes du réseau et des coûts générés par la sécurité comparés aux gains de sécurité engendrés.

3.3. Les guides de sécurité réseau des équipementiers

Les nombreux problèmes ou faiblesses de sécurité des équipements réseau ont forcé les équipementiers à considérer la sécurité comme une composante du développement des produits. CISCO, qui est le principal fournisseur mondial d'équipement réseau, met à la disposition sur son site Internet de nombreux guides sur les mécanismes de sécurité proposés dans ces équipements.

Un de ces documents traite de manière plus précise des problématiques de sécurité d'un opérateur de télécommunications [CISCO 2001]. Il couvre :

- La sécurité de la gestion de l'administration des équipements réseau, des protocoles de routage interne et externe,
- des problématiques des sessions de routage avec des tiers-parties (gestion des instabilités des routes, contrôle des annonces de routes, etc.), du service réseau de résolution des noms de domaine (Domain Name Service), du service de distribution du temps (Network Time Protocol),
- du service de création de tunnels chiffrés (IP SECurity), etc. Enfin, ce guide donne des exemples concrets de configuration réseau types.

De plus, toute faiblesse de sécurité détectée donne lieu à une alerte de sécurité. Cette dernière contient le(s) produit(s) et le(s) version(s) impactés, des informations sur les correctifs, mais aussi des recommandations de configuration en attendant les correctifs de sécurité.

Ces guides et alertes permettent de définir les mécanismes de sécurité à mettre en place afin de satisfaire les objectifs d'une politique de sécurité. Ils ne définissent cependant pas les besoins et les objectifs de sécurité d'un réseau d'un opérateur de télécommunications.

3.4. Les guides de sécurité réseau de la National Security Agency

Dans le cadre de la publication de documents de l'agence de la sécurité nationale américaine (National Security Agency), des recommandations de sécurité à la fois au niveau des équipements réseau et des systèmes d'exploitation sont disponibles sur Internet [NSA 2003]. Au niveau réseau, ces guides décrivent de manière précise les configurations ainsi que les mécanismes de sécurité qui doivent être mises en place afin de garantir un niveau de sécurité minimum. Deux documents existent :

- "Switch Security Configuration Guide" : Ce guide décrit les configurations assurant un niveau minimum de sécurité des équipements de niveau 2. Il couvre à la fois la sécurité de la gestion de l'administration, des services réseau offrant des réseaux locaux virtuels (VLAN: Virtual Local Area Network), mais aussi des protocoles réseau permettant de gérer des domaines d'équipements de niveau 2 comme le protocole Virtual Trunking Protocol (gestion de la politique des VLANs dans un domaine réseau de niveau 2), ou encore le protocole Spanning Tree Protocol (prévention des

boucles dans un domaine réseau de niveau 2). Enfin, ce guide donne à la fois des exemples types de configuration, mais aborde aussi la vérification des configurations.

- "Router Security Configuration Guide" : Ce guide décrit les configurations assurant un niveau minimum de sécurité des équipements de niveau 3. Il couvre à la fois la sécurité de la gestion de l'administration, des protocoles de routage interne et externe, du service réseau de résolution des noms de domaine DNS, du service de distribution du temps NTP, du service de création de tunnels chiffrés IPSEC, etc. Enfin, ce guide donne à la fois des exemples types de configuration, mais aborde aussi la vérification des configurations.

Ces guides de sécurité permettent de définir les mécanismes de sécurité à mettre en place afin de satisfaire les objectifs d'une politique de sécurité. Ils ne définissent cependant pas les besoins et les objectifs de sécurité d'un réseau d'un opérateur de télécommunications.

3.5. La politique de sécurité d'un réseau multi-services

Une politique de sécurité est l'ensemble des lois, des règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique [CC 2002].

Les propriétés principales attendues d'un réseau sont :

- La disponibilité : Il s'agit d'assurer que le réseau rende le service pour lequel il a été conçu sans interruption ou déni de services. La disponibilité d'un réseau est généralement assurée par son architecture ou topologie.
- L'intégrité : Il s'agit d'empêcher la corruption des données par des fautes accidentelles ou intentionnelles, et à garantir leur mise à jour. L'intégrité est généralement assurée par les protocoles réseaux employés afin de gérer les flux de données, le routage, etc.
- La confidentialité : La confidentialité peut être définie comme la prévention de la divulgation non-autorisée de l'information, nous considérons que les données associées aux flux réseaux sont protégées en amont avant de transiter par le réseau. L'utilisation d'algorithmes de chiffrement assurera la confidentialité des données. En revanche, nous devons absolument considérer l'étanchéité des services réseau, par exemple un VPN doit être isolé des autres VPNs du réseau par défaut, afin de garantir le niveau de confidentialité attendu du service (il s'agit ici de garantir l'isolation logique des trafics réseau).

Une politique de sécurité réseau peut se développer dans les trois directions suivantes :

- Une politique de sécurité physique : Elle détaille par exemple les objectifs et règles de sécurité des équipements réseau afin de faire face aux menaces comme le feu, les catastrophes naturelles, etc.,
- Une politique de sécurité administrative : Elle détaille par exemple les objectifs et règles de sécurité des procédures de gestion du réseau afin de faire face aux événements de congestion du réseau, etc.,
- Une politique de sécurité logique : Elle détaille par exemple les objectifs et règles de sécurité de configuration des accès des équipements réseau afin de faire face aux accès non autorisés, attaques, etc.

Nous détaillons brièvement ces trois politiques de sécurité dans les paragraphes suivants.

3.5.1. La politique de sécurité physique

Elle consiste essentiellement à se protéger contre les vols, fuites d'eau, incendies, coupures d'électricité, etc. Les règles génériques à considérer sont les suivantes :

- Une salle contenant des équipements réseau ne doit pas être vue de l'extérieur afin de ne pas attirer ou susciter des idées de vol ou de vandalisme.
- Une salle d'équipements réseau ne doit jamais être installée au rez-de-chaussée d'un bâtiment afin de ne pas être vulnérable à une inondation.
- Des périmètres de sécurité physique à accès restreint doivent être définis, et équipés de caméras de surveillance.
- Les ressources critiques doivent être placées dans le périmètre le plus sécurisé.
- Toute modification physique d'infrastructure doit être identifiée, reportée et validée.
- Des procédures doivent autoriser et révoquer l'accès aux périmètres de sécurité.
- Le site ne doit pas se trouver pas sur un lieu connu pour des catastrophes naturelles comme la foudre, tremblement de terre, inondation, etc.
- Des équipements de protection contre le feu, l'eau, l'humidité, les pannes de courant, le survoltage, etc., doivent être installés.
- Des procédures de supervision des éléments de protection doivent être mises en place.

3.5.2. La politique de sécurité administrative

L'exploitation du réseau et de ses services associés doit suivre un ensemble de procédures dites opérationnelles afin d'en assurer l'intégrité et la sécurité à moyen terme. Les règles génériques à considérer sont les suivantes :

- Les procédures opérationnelles doivent être définies et mises à jour.
- Des procédures opérationnelles doivent exister pour la supervision des éléments critiques.
- Des procédures de maintenance préventive doivent exister pour les éléments critiques de telle sorte que toute anomalie soit vérifiée et corrigée.
- Des sauvegardes des informations critiques doivent être effectuées dans un lieu physique distinct de la source. Cela couvre en premier lieu les configurations des équipements.
- Tout problème détecté doit être identifié et résolu.
- Des contre-mesures doivent permettre de vérifier que des problèmes ne restent pas sans solution.
- Tout problème ou incident de sécurité doit être remonté par les procédures opérationnelles aux responsables des domaines visés.
- Les procédures d'incident de sécurité doivent être connues de tout le personnel.

3.5.3. La politique de sécurité logique

La politique de sécurité réseau logique porte sur les configurations des équipements réseau. Les configurations détiennent toute l'information permettant de construire le réseau et ses services. La politique de sécurité réseau logique peut se décliner en un ensemble de règles de sécurité génériques suivantes garantissant la disponibilité et l'intégrité du réseau et de ses services [Valois et Llorens 2002, Llorens et al. 2003, NSA 2003, CISCO 2001] :

Règles de sécurité génériques	**Description**
Consistance du plan d'adressage	Il s'agit des règles qui garantissent la consistance du plan d'adressage des équipements réseau. De manière générique, il ne doit exister de doublons ni dans le plan d'adressage global du réseau, ni dans le plan d'adressage d'un VPN donné.
Consistance des	Il s'agit des règles qui garantissent la consistance des

configurations	configurations des équipements réseau. De manière générique, tout élément de configuration défini doit être appliqué, et tout élément de configuration appliqué doit être défini. Ces règles peuvent être complexes comme la vérification de la grammaire associée au langage de configuration.
Consistance des filtrages	Il s'agit des règles qui garantissent la consistance des filtrages utilisés pour contrôler par exemple les flux de données ou de routage. De manière générique, les éléments constituant un filtrage ne doivent être ni redondants, ni contradictoires entre eux. Ces règles peuvent être complexes comme la vérification des règles inutiles.
Routage	Il s'agit des règles de configuration relatives à la protection du routage réseau. Ces règles s'appliquent à la fois au routage interne du réseau ainsi qu'aux interconnexions de routage du réseau avec l'extérieur. Ces règles peuvent être complexes comme la vérification de la topologie du routage interne et externe du réseau, la consistance de la politique de routage, etc.
Service	Il s'agit des règles de configuration relatives à la protection des services du réseau. Ces règles peuvent être complexes comme la vérification des périmètres de sécurité d'un VPN.
Partenaires	Il s'agit des règles de configuration relatives à la protection des interconnexions avec les services réseau d'un partenaire.
Administration	Il s'agit des règles de configuration relatives à la protection des équipements réseau.

Table 3-1: Règles de sécurité génériques

3.6. Conclusion

La définition d'une politique de sécurité réseau vise à définir les besoins de sécurité, à élaborer des stratégies de sécurité afin de protéger les biens les plus critiques, et à définir le référentiel des contrôles de sécurité.

Après avoir défini les objectifs et le contenu d'une politique de sécurité réseau, nous détaillerons au chapitre suivant les méthodes d'évaluation existantes de la sécurité qui peuvent s'appliquer à un réseau.

Chapitre 4. Les méthodes d'évaluation de la sécurité

4.1. Introduction

Depuis l'après guerre, de nombreuses méthodes d'évaluation de la sécurité ont été développées. Dans le cadre de ce travail de recherche, plusieurs méthodes ont retenu notre attention. Nous les détaillons ci-après et précisons notre choix pour l'évaluation de la sécurité d'un réseau multi-services.

4.2. Les critères communs

4.2.1. L'historique

C'est en 1985, que la NSA et le NIST ont rédigé le document intitulé "Orange Book". Ce document traite essentiellement de la capacité d'un système à être résistant à des attaques. L'objectif d'un tel document était d'évaluer la sécurité d'un système en proposant des critères afin de définir un niveau de sécurité.

Transposer en Europe sous le nom des critères Trusted Computer Systems Evaluation Criteria [TCSEC 1985], la communauté européenne propose en 1991, les critères Information Technology Systems Evaluation Criteria [ITSEC 1991] comblant certaines lacunes des critères TCSEC notamment dans le domaine de l'analyse de risques.

Le Canada propose en 1993 les critères [CTCPEC 1993] (Canadian Trusted Computer Product Evaluation Criteria), qui est une combinaison des critères TCSEC et ITSEC.

Sous l'impulsion de l'ISO (International Standard Organisation), les auteurs des différents critères ont décidé d'aligner leurs critères pour définir les critères communs [CC 2002].

L'objectif des critères communs reste avant tout d'offrir un référentiel commun et reconnu de tous les Etats afin d'évaluer la sécurité d'un système.

4.2.2. Les concepts généraux

Les critères communs sont un guide pour le développement et le contrôle des produits commerciaux ayant des fonctions de sécurité attendues et attestées. Le produit (ou le système) comprend le système d'exploitation, les réseaux, les systèmes distribués et les applications utilisées. De plus, un système est évalué en fonction de l'usage pour lequel il a été prévu et non pour les qualités intrinsèques du système.

L'évaluation de la sécurité d'un système par les critères communs est toujours réalisée par un tiers afin d'assurer l'indépendance de l'évaluation. Plusieurs centres en France, appelé CESTI (Centre d'Evaluation de la Sécurité des Technologies de l'Information), sont habilités à délivrer des certifications.

Les critères communs adoptent une approche décrites par les étapes ci-après :

- Etape 1 : Elle permet de définir le contexte de l'évaluation considéré.
- Etape 2 : Elle définit les exigences de sécurité attendues.
- Etape 3 : Elle définit le niveau de garantie attendu.
- Etape 4 : Elle consiste à formuler les exigences de sécurité en fonction du niveau de sécurité espéré.
- Etape 5 : Elle définit ce que l'on souhaite protéger dans la perspective d'une évaluation.

Les critères communs reposent sur les concepts suivants :

- Les exigences fonctionnelles : Regroupées sous forme de classes, chaque classe couvre le domaine suivant comme l'illustre le tableau suivant :

Classe	**Description**
FAU	Cette classe traite des exigences de l'audit de sécurité
FCO	Cette classe traite des exigences associées à la non répudiation des émissions et réceptions
FCS	Cette classe traite des exigences associées à la gestion des crypto-systèmes
FDP	Cette classe traite des exigences des protections des données utilisateurs
FIA	Cette classe traite des exigences pour que des fonctions établissent et contrôlent l'identité
FMT	Cette classe traite des exigences de l'administration de la sécurité
FPR	Cette classe traite des exigences de la protection de la vie privée
FPT	Cette classe traite des exigences de la protection de l'ensemble des fonctions de sécurité
FRU	Cette classe traite des exigences de l'utilisation des ressources
FTA	Cette classe traite des exigences fonctionnelles pour contrôler l'établissement d'une session utilisateur
FTP	Cette classe traite des exigences de chemins et canaux de confiance

Table 4-1: Critères communs, les exigences fonctionnelles

NB: Chaque classe contient un ensemble de familles, et chaque famille contient un ensemble de composants. Chaque composant définit une exigence de sécurité.

- Les exigences d'assurance : Regroupées sous forme de classes, chaque classe couvre le domaine suivant comme l'illustre le tableau suivant :

Classe	Description
ACM	Cette classe traite des exigences de gestion de la configuration
ADO	Cette classe traite des exigences de livraison et d'exploitation
ADV	Cette classe traite des exigences de développement
AGD	Cette classe traite des exigences de documentation
ALC	Cette classe traite des exigences associées au cycle de vie
ATE	Cette classe traite des exigences des tests
AVA	Cette classe traite des exigences associées à l'identification des vulnérabilités
APE	Cette classe traite des exigences de l'évaluation du profil de protection
ASE	Cette classe traite des exigences de l'évaluation de la cible de sécurité

Table 4-2: Critères communs, les exigences d'assurance

NB: Chaque classe contient un ensemble de familles, et chaque famille contient un ensemble de composants. Chaque composant définit une exigence d'assurance.

- Les niveaux d'évaluation d'assurance EAL (Evaluation Assurance Level) : Ils certifient que le produit respecte un certain niveau d'assurance EAL. L'assurance est la confiance qui peut être accordée à la sécurité fournie par une cible d'évaluation. Chaque niveau d'évaluation EAL regroupe un ensemble d'exigences d'assurance. 7 niveaux d'assurance existent comme l'illustre le tableau suivant :

Niveau	Description
-	Niveau minimum de sécurité
EAL1	Tests fonctionnels
EAL2	Tests structurels
EAL3	Tests et vérifications méthodiques
EAL4	Conceptions, tests et vérifications méthodiques
EAL5	Conception semi-formelle et tests
EAL6	Vérification semi-formelle de la conception générale
EAL7	Vérification formelle de la conception générale

Table 4-3: Critères communs, les niveaux d'évaluation

- Les profils de protection : Un profil de protection permet de définir les exigences fonctionnelles d'un type de produit en fonction d'une cible d'évaluation. Un profil de protection est donc réutilisable par tous et présente l'avantage d'exposer des exigences reconnues comme étant nécessaires pour satisfaire les objectifs de sécurité. Par exemple, dans le domaine des cartes à puce, des sociétés ont défini des profils de protection pointant des domaines spécifiques tels que le domaine des circuits intégrés,

ou le domaine des applications financières. Les profils de protection permettent donc d'avoir un ensemble commun d'exigences de sécurité apportant le concept de réutilisabilité pour l'évaluation d'un type de produit.

- La cible de sécurité : La cible de sécurité contient les exigences de sécurité du produit à évaluer. La cible de sécurité est le dossier qui servira de base à l'évaluation.

Les Critères Communs permettent d'évaluer un produit de sécurité selon des exigences prédéfinies. Si l'évaluation s'avère positive, alors le produit de sécurité se voit décerner une certification, qui sera reconnue au niveau mondial.

4.3. L'analyse probabiliste des risques

4.3.1. L'historique

Les méthodes d'évaluation des risques ont été issues des programmes spatiaux, nucléaire et militaire américain au début des années 60. L'analyse des arbres de défaillance en est un exemple.

L'analyse probabiliste des risques a été constamment améliorée par les experts du domaine et a gagné de la crédibilité pendant les deux dernières décennies non seulement dans l'industrie nucléaire, mais également dans d'autres industries comme la pétrochimique, les plates-formes pétrolières et la défense [Bedford et al. 2001, Stamatelotos 2002].

En raison de son approche logique, systématique et compréhensive, l'analyse probabiliste des risques a prouvé à plusieurs reprises qu'elle était capable de découvrir des faiblesses de conception, qui avaient échappé aux experts. Cette méthodologie a aussi prouvé qu'il était très important d'examiner l'ensemble des scénarii ayant une faible probabilité d'occurrence, mais avec une forte conséquence sur le système concerné.

Après l'accident de la navette spatiale challengeur le 29 octobre 1986, la NASA a décidé que l'analyse probabiliste des risques devait être appliquée à tout le programme spatial, mais a aussi déclaré que les techniques d'analyse devaient être systématiquement améliorées par des données précises et un historique complet.

4.3.2. Les concepts généraux

Le concept du risque inclut deux types de conséquences indésirables. Par exemple, le nombre de personnes ayant une maladie donnée et la probabilité de l'occurrence de ce mal. Parfois, le risque est défini comme la valeur prévue de ces conséquences.

Une définition commune du risque est celle donnée par le triplet suivant. La détermination du risque consiste généralement à répondre aux questions suivantes :

1. Qu’est qui peut tourner mal ?
2. Est-ce que c’est probable ?
3. Quelles en sont les conséquences ?

La réponse à la première question consiste à définir un ensemble de scénarii d'accidents possibles. La deuxième question exige l'évaluation des probabilités associées à ces scénarii, et la troisième exige d'estimer leurs conséquences.

En plus des probabilités et des conséquences, la définition du triplet souligne le développement de scénarii d'accidents comme étant partie prenante de la définition du risque.

Le processus d'analyse des risques commence donc à déterminer un ensemble d’événements initiaux (IEs) qui perturbent le système. Pour chaque IE, l'analyse consiste à déterminer les échecs qui peuvent mener à des conséquences indésirables. Ensuite, les conséquences associées aux scénarii sont déterminées, ainsi que leurs fréquences. Notons que la multitude de tels scénarii permettent de créer un profil de risque de notre système.

L’analyse probabiliste des risques suit alors une méthodologie constituée des étapes suivantes :

- Définition des objectifs : Les objectifs de l'évaluation de risque doivent être bien définis et les conséquences indésirables identifiées.
- La connaissance du système : La connaissance du système concerné est primordiale. Elle couvre les aspects de conception jusqu'aux procédures de fonctionnement du système.
- Identification des événements initiaux : L'ensemble des événements initiaux déclenchant des scénarii d'accidents doit être identifié. Les événements initiaux indépendants qui mènent à des scénarii semblables doivent être groupés ainsi que leurs fréquences, afin d'évaluer les fréquences initiales.
- Modélisation des scénarii : Chaque scénario d'accident doit être modélisé avec des outils probabilistes appelés arbres d'événements (Event Tree). Un arbre d'événement commence avec un événement initial et s'étend avec la progression du scénario. Des séries de succès ou d'échecs des événements intermédiaires sont appelés les événements pivots, jusqu'à ce qu'un état final de l'arbre soit atteint (feuille).
- Modélisation des échecs : Chaque échec d'un événement pivot dans un scénario d'accidents doit être modélisé avec des outils probabilistes appelés arbre de défaillances (Fault Tree). La partie supérieure de l'arbre est un événement pivot défini dans un scénario d'accidents. La partie intermédiaire de l'arbre se compose des événements intermédiaires causant l'événement supérieur. Ces événements sont liés par des portes logiques aux événements de base, dont l'échec fait finalement produire l'événement supérieur. Les arbres de défaillance sont alors simplifiés, en utilisant des

règles de réduction booléennes, afin de renforcer la quantification des scénarii d'accidents.

- Collecte de données et analyse : Divers types de données doivent être rassemblés et traités. Les données rassemblées fournissent des informations sur les taux d'échec, les temps de réparation, les probabilités d'IE, les probabilités de défaillance de structure, les probabilités d'erreur humaines, les probabilités de processus d'échec.
- Quantification : La fréquence de l'occurrence de chaque état d'extrémité est le produit de la fréquence d'IE et des probabilités conditionnelles des événements pivots le long du chemin liant l'IE à l'état d'extrémité. Les scénarii sont groupés selon l'état d'extrémité du scénario définissant une conséquence donnée. Tous les états d'extrémité doivent être alors groupés, et leurs fréquences se résument alors à la fréquence d'un seul état représentatif d'extrémité.
- Analyse d'incertitude : Des analyses d'incertitude doivent être réalisées pour évaluer le degré de confiance que l'on peut porter sur les calculs numériques du risque. Des méthodes de simulation de type Monte-Carlo sont généralement employées pour réaliser une analyse d'incertitude.
- Analyse de sensibilité : Des analyses de sensibilité sont également fréquemment réalisées pour indiquer si des changements sur des valeurs d'entrée peuvent causer des changements importants des calculs numériques partiels ou finals du risque.

4.4. Les graphes de privilèges

Dacier a défini une méthode générale d'évaluation quantitative de la sécurité des systèmes informatiques [Dacier 1994, Dacier et al. 1996]. Elle est basée sur une représentation des vulnérabilités présentes dans un système informatique, appelé un graphe de privilèges. Un privilège est défini comme étant un ensemble de droits qu'un sujet peut posséder sur un objet.

Dans un graphe de privilèges, chaque nœud représente un ensemble de privilèges. L'existence d'un arc d'un premier ensemble de privilèges vers un second indique que la possession de ce premier ensemble permet d'acquérir ce second ensemble de privilèges, par application d'une ou de plusieurs méthodes. Les méthodes de transfert de privilèges à l'origine de l'existence des arcs dans un graphe correspondent à des vulnérabilités présentes dans le système. Ces vulnérabilités peuvent correspondre à des faiblesses du système, mais peuvent aussi également représenter des mécanismes de transfert de droits parfaitement licites et indispensables au fonctionnement du système.

Etant donné cette représentation, on peut envisager d'associer à chacune des vulnérabilités prises en compte lors de la construction du graphe des privilèges une valeur numérique correspondant à la probabilité de sa mise en œuvre. La mesure quantitative de la sécurité est alors définie comme étant la valeur du

temps ou d'effort correspondant à la difficulté pour l'attaquant d'obtenir les privilèges de la cible de sécurité [Ortalo 1998].

4.5. Les graphes d'attaques

Somest et al. ont proposé une nouvelle approche du graphe d'attaques en tenant compte à la fois du modèle de graphes de privilèges de Dacier [Dacier 1994, Dacier et al. 1996], mais aussi des graphes d'attaques de Swiler [Swiler et al. 2001]. Cependant, ils suggèrent une implémentation des graphes d'attaques sur un vérificateur de formule symbolique appelé Model Checker. Ce vérificateur permet de gérer un grand nombre d'états, de considérer simultanément plusieurs événements autres que des attaques, et de limiter la complexité en espace des graphes d'attaques par une analyse "backforwarding" que nous détaillerons ci-après [Somest et al. 2002].

Model Checker est une technique automatique pour vérifier des systèmes à états finis. Les spécifications du système sont exprimées en propositions de logique temporelle et le système est modélisé en un graphe d'états à transition. Une procédure de recherche permet alors de déterminer si les spécifications sont satisfaites par le graphe d'états à transition. Les dernières évolutions de Model Checker permettent de traiter un grand nombre d'états grâce à une représentation binaire efficace des transitions d'états [Cimatti et al. 2002].

Somest et al. ont donc transposé au graphe d'états à transition le graphe d'attaques, et aux spécifications du système les spécifications des privilèges. Ils ont aussi modifié le code source de Model Checker de manière à donner non pas un chemin qui viole les spécifications du système, mais l'ensemble des chemins du graphe d'attaques qui violent ces spécifications.

4.6. Conclusion et choix de la méthode d'évaluation

Des évaluations de la sécurité sont soit basées sur des critères d'évaluation comme les critères communs [CC 2002], soit basées sur une analyse probabiliste des risques [Bedford et al. 2001, Stamatelotos 2002]. Cependant, d'autres types d'évaluation de la sécurité ont été proposés [Philips et al. 1998, Swiler et al. 2001], on retiendra plus particulièrement le travail réalisé par le laboratoire LAAS [Ortalo 1998] qui définit une mesure de la sécurité comme l'effort pour un attaquant d'obtenir des privilèges sur des objectifs de sécurité. De plus, de nombreux travaux de recherche explorent le domaine des graphes d'attaques, afin de déterminer quel est le jeu minimum de règles de sécurité permettant d'assurer la sécurité d'un système [Somesh et al. 2002]. Bien que d'autres travaux couvrent aussi la gestion des risques pour un système d'information [Wulf et al. 1996, Williams et al. 1998, Bush et al. 2001], aucun papier n'évoque réellement la mesure de la sécurité d'un réseau de télécommunications.

Dans le cadre d'un réseau, la notion de privilèges sur un équipement réseau est très limitée et peut être réduite à "pas de privilège", "privilège de lecture",

"privilège d'écriture". De plus, les vulnérabilités de configuration offrant de tels droits et pouvant être exploitées par des attaques externes sont aisément détectées par nos outils de vérification comme nous le verrons par la suite.

Notre problème consiste plutôt à déterminer, pour la majorité des vulnérabilités qui ne peuvent pas être exploitées par une attaque externe, le risque pris si ces vulnérabilités ne sont pas corrigées. Notre besoin nous oriente donc vers un modèle permettant de décrire toutes les séquences d'événements pouvant impacter le réseau plutôt qu'un modèle basé sur des graphes d'attaques ou de privilèges.

Sachant de plus, que notre objectif est de quantifier le risque associé à la non application de la politique de sécurité, et que les vérifications réalisées sur les configurations des équipements réseaux sont de nature statique, nous avons donc choisi de baser notre méthode d'évaluation sur une évaluation probabiliste des risques.

Chapitre 5. Présentation du prototype de mesure de la sécurité

5.1. Introduction

Le prototype de mesure de la sécurité que nous avons mis en œuvre au sein du réseau de France Télécom/Equant se base sur une évaluation probabiliste des risques [Bedford et al. 2001, Stamatelotos 2002]. Un risque est défini comme une fonction à la fois de la probabilité qu'une menace exploite une vulnérabilité donnée, mais aussi de l'impact résultant de l'exploitation de cette vulnérabilité.

Dans le cadre d'un réseau, nous désignerons une menace comme :

- la probabilité de l'exploitation d'une vulnérabilité d'un équipement réseau par un événement (attaque, virus, etc.),
- une vulnérabilité comme une faiblesse de sécurité de nature logique (erreur de configuration, etc.) ou physique (faiblesse du système de protection électrique des équipements réseau, etc.) ou humaine (suite à un acte malveillant ou à une erreur),
- et une conséquence comme un impact (congestion du réseau, inondation de la salle contenant les équipements de télécommunications, perte financière, mauvaise publicité, etc.) sur le réseau de l'exploitation d'une vulnérabilité de sécurité.

Le prototype de mesure de la sécurité consiste à vérifier l'application de la politique de sécurité sur les configurations des équipements réseau, à définir des indicateurs de sécurité afin d'établir un tableau de bord de la sécurité réseau, et à quantifier le risque associé à la non application de la politique de sécurité.

Avant de détailler ces différentes parties, nous rappelons dans un premier temps les notions fondamentales concernant les tableaux de bord de la sécurité.

5.2. Les fondamentaux d'un tableau de bord de la sécurité

L'établissement d'un tableau de bord de la sécurité se réfère de manière fondamentale à la notion de mesure de la sécurité. De manière théorique, une "mesure" est définie comme le processus par lequel on affecte des nombres ou des symboles aux attributs d'entités appartenant au monde réel, de manière à les décrire par rapport à des règles clairement définies.

On distingue les mesures directes qui permettent d'attribuer une valeur à l'attribut d'une entité (par exemple, la taille d'un programme peut se mesurer

par le nombre de lignes de codes), mais aussi les mesures indirectes qui ne permettent pas d'attribuer une valeur à l'attribut d'une entité (la facilité de maintenance ne peut pas se mesurer directement).

La théorie de la mesure montre toute la difficulté de définir de manière cohérente et consistante un tableau de bord de la sécurité. Objectif utopique ou non, il n'en reste pas moins que l'on ne peut réduire un tableau de bord de sécurité à un indicateur entre 0 et 100 pour 100 pour un système complexe sans perdre d'informations essentielles.

5.2.1. Quels sont les objectifs ?

Malgré ces difficultés, il ne faut pas pour autant ne pas initier une telle démarche. L'établissement d'un tableau de bord de la sécurité doit absolument s'inscrire dans une démarche sécuritaire afin de répondre aux besoins de sécurité du réseau :

- Déterminer les éléments les plus critiques, les menaces et les conséquences qui pèsent sur le réseau.
- Définir une politique de sécurité permettant de se prémunir contre les menaces et les conséquences les plus critiques.
- Mettre en oeuvre des technologies répondant aux objectifs définis dans la politique de sécurité.
- Contrôler l'application de la politique de sécurité par des contrôles internes et externes récurrents.
- Consolider, corréler les informations des contrôles afin de bâtir un tableau de bord de la sécurité cohérent avec les objectifs de sécurité.

Les objectifs attendus d'un tableau de bord de la sécurité sont multiples, on peut cependant lister les points suivants :

- Un tableau de bord de la sécurité doit montrer régulièrement le niveau de sécurité d'un système. L'historique doit être gardé pour des analyses statistiques ultérieures.
- Un tableau de bord de la sécurité doit permettre de déclencher des actions ou alertes préventives. Ces actions ou alertes doivent prendre en considération l'historique des données collectées.
- Un tableau de bord de la sécurité doit permettre de prendre des décisions sur des critères de nature différente.
- Un tableau de bord de la sécurité ne doit pas être par nature un rapport *post-mortem* d'un incident de sécurité, mais doit plutôt être un rapport préventif afin d'éviter un incident de sécurité.

Quel que soit l'état d'avancement du tableau de bord de la sécurité, les personnes concernées doivent être impliquées, et des objectifs doivent être définis afin de corriger les failles de sécurité.

5.2.2. Quels sont les besoins opérationnels ?

De manière générale, l'opération d'un réseau complexe nécessite de la part des entités opérationnelles des qualités de réaction rapide et de définition des priorités. La sécurité n'échappe pas à cette règle et doit fournir à ces entités ces deux axes d'actions.

Le premier axe concerne la réaction rapide qui repose sur le fait qu'un tableau de bord de la sécurité doit permettre de déclencher des actions ou alertes préventives.

Le deuxième axe concerne la définition des priorités qui repose sur le fait qu'un tableau de bord de la sécurité doit permettre de prendre des décisions en tenant compte de critères de nature différente.

Enfin, les informations données aux entités opérationnelles doivent être non seulement précises, mais elles doivent aussi détailler les impacts réseau possibles associés à la correction des faiblesses détectées.

5.2.3. Quelles sont les problèmes d'échelle ?

L'une des caractéristiques qui font qu'une activité peut se voir attribuer le statut de science est la capacité d'obtenir et de manipuler des mesures relatives à l'objet de cette science.

On rencontre souvent dans la littérature les mots "mesure" et "métrique", il n'est pas simple de les distinguer de manière définitive. La langue française génère elle-même quelques confusions puisque les termes "mesure" et "métrique" ont tous les deux une connotation mathématique bien que dans des contextes différents.

Bien que le National Institute Standard Technologies (NIST) précise que le terme "métrique" devrait être utilisé pour la définition mathématique et algorithmique, et que le terme "mesure" devrait désigner la valeur numérique obtenue, on s'oriente cependant vers une utilisation systématique du terme "mesure".

Une "mesure" est définie comme le processus par lequel on affecte des nombres ou des symboles aux attributs d'entités appartenant au monde réel de manière à les décrire par rapport à des règles clairement définies [Fenton et al. 1996].

On distingue les mesures directes qui permettent d'attribuer une valeur directement à l'attribut d'une entité (par exemple, la taille d'un programme peut se mesurer par le nombre de lignes de codes), mais aussi les mesures indirectes qui ne permettent pas d'attribuer une valeur directement à l'attribut d'une entité (la facilité de maintenance ne peut pas se mesurer directement).

Le jugement de l'adéquation d'une mesure est basé sur le choix des attributs qui caractérisent une entité, mais aussi sur le fait que l'association de valeurs numériques aux attributs doit préserver certaines propriétés. De manière plus formelle, toutes les relations du système empirique doivent être préservées dans le système numérique.

Un énoncé est signifiant si sa vérité (ou sa fausseté) reste inchangé quand on passe d'une échelle à une autre échelle admissible. Plusieurs échelles existent comme l'illustre le tableau suivant :

Echelle	**Exemple**	**Opérations Statistiques possibles**
Nominale	Numérotation des joueurs de football	Fréquence
Ordinale	Classification en catégories (A, B, C, etc.)	Médiane, Percentile, etc.
Intervalle	Température	Moyenne, écart type, etc.
Ratio	Taille	Moyenne géométrique, coefficient de variation, etc.

Table 5-1: Exemples d'échelles de mesure

Dans le cadre de la mesure de la sécurité logique d'un réseau, nous définirons tout d'abord un ensemble le plus complet possible d'attributs caractérisant la sécurité des configurations d'un réseau. Nous baserons ensuite notre mesure sur le comptage du nombre de faiblesses détectées dans les configurations des équipements réseau. Nous adopterons alors une échelle de type ratio. Il doit être noté qu'une échelle de type ratio préserve l'ordre, la taille des intervalles, incluant l'élément 0.

5.2.4. Quelles sont les limitations ?

Les limitations ou les erreurs à ne pas commettre lors de la construction d'un tableau de bord de la sécurité sont nombreuses, on peut cependant lister les suivantes :

- Un tableau de bord de la sécurité doit être composé de plusieurs sous tableaux dépendant de la complexité du système et des objectifs de sécurité.
- On ne peut réduire un tableau de bord de sécurité à un indicateur entre 0 et 100 pour 100 pour un système complexe sans perdre d'information essentielle.
- Un tableau de bord de la sécurité doit être considéré comme un apport d'information sur la sécurité et non comme des valeurs réelles absolues de la sécurité. Le danger encouru avec les indicateurs de sécurité est qu'ils engendrent l'objectif de les rendre à tout prix positifs, sans prendre en compte que le tableau ne traduit pas réellement la sécurité.

5.3. Détection des vulnérabilités de sécurité

5.3.1. Approche générique

A partir des règles de sécurité génériques définies dans la politique de sécurité réseau, on décline alors un ensemble de tests de sécurité qui ont pour objectif de vérifier un ou plusieurs points de sécurité.

Pour chaque déviation de sécurité détectée, un test de sécurité signale alors une vulnérabilité. Comme illustré à la figure suivante, le moteur de vérification permet d'associer à un ensemble d'équipements réseau un ensemble de tests, afin d'établir un profil de contrôle et de signaler les vulnérabilités détectées :

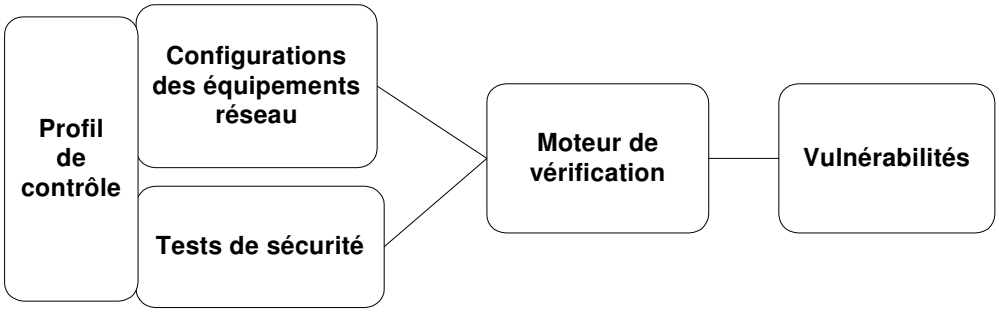


Figure 16: Détection des vulnérabilités

Plusieurs profils de contrôle peuvent être définis suivant la nature des équipements réseau et des tests de sécurité concernés. Nous avons détaillé un premier jeu de tests de sécurité, qui s'appliquent soit à une configuration d'un équipement réseau (vérifier la présence de lignes de configurations spécifiques, vérifier la consistance d'une liste de filtrage, etc.), soit à un ensemble de configurations d'équipements réseau (vérifier l'unicité du plan d'adressage, vérifier les topologies de routage, vérifier les périmètres de sécurité d'un réseau MPLS/VPN, etc.) [Valois et Llorens 2002, Llorens et al. 2003].

5.3.2. La détection des vulnérabilités de configuration d'un équipement

Les algorithmes employés dans les tests de sécurité peuvent être triviaux jusqu'à des algorithmes plus sophistiqués notamment pour la vérification des éléments constituant une liste de filtrage [Eppstein et al. 2001, Warkhede et al. 2001]. Les tests de sécurité sont développés principalement en langage AWK pour la facilité et la portabilité de ce langage, et en C pour des raisons de performance lorsque les algorithmes sont plus complexes. La table suivante donne quelques tests de sécurité disponibles pour vérifier les configurations d'équipement CISCO :

test script	Description
t.acl_01	Détecter les ACLs (Access Control List) définies mais pas référencées, mais aussi les ACLs référencées mais pas définies
t.acl_02	Vérifier la consistance d'une ACL
t.acl_03	Vérifier que les "lines" (permettant d'accéder à l'équipement en mode administration) sont protégées par une ACL pour le trafic entrant
t.acl_04	Vérifier que les "interfaces" réseau d'un équipement sont protégées par une ACL
t.acl_05	Vérifier que les communautés SNMP (Simple Network Management Protocol) sont définies avec une ACL
t.acl_06	Vérifier que l'ACL protégeant l'accès en administration d'un équipement vérifie les règles de sécurité définies
t.file_01	Vérifier que les configurations sauvegardées sur les systèmes d'administration ne datent pas de plus de 7 jours
t.file_02	Vérifier que le "hostname" est équivalent au nom du fichier de configuration
t.pw_01	Vérifier que le mot de passe "enable password 7" est différent du mot de passe "enable secret 5"
t.pw_02	Vérifier les mots de passe stockés
t.routing_01	Vérifier que l'équipement ne permet pas le routage à la source
t.version_01	Vérifier que la version d'IOS est supérieure à x.x

Table 5-2: Exemples de tests de configuration

On constate qu'il y a des tests de sécurité génériques "t.acl_01" et que d'autres effectuent des contrôles plus spécifiques "t.version_01". On constate aussi que des tests de sécurité sont faciles à implémenter comme "t.file_02", alors que d'autres nécessitent une approche algorithmique tel que "t.acl_01".

La vérification des configurations consiste à lancer l'outil de vérification sur des configurations avec des tests de sécurité déterminés comme l'illustre le listing suivant :

```
$ conf_validation -e 't.acl_0[123]' conf[12]
BEGIN t.acl_01 conf1
ACLs referenced but not defined (références mais pas définies) :
      snmp-server community toto RW 80
       ip access-group 112 in
       access-class 23 in
```

```
END FAILED

BEGIN t.acl_04 conf1
      interface Ethernet0 has no access-group
END FAILED

BEGIN t.acl_01 conf2
ACLs defined but not referenced (définies mais pas référencées) :
      101
END FAILED
```

Afin d'illustrer plus largement les tests de sécurité ainsi que la détection des vulnérabilités, détaillons quelques exemples de tests. Le premier test de sécurité "t.acl_01" consiste à vérifier que les ACLs définies sont référencées et que les ACLs référencées sont définies.

Ce test de sécurité consiste à stocker dans deux tableaux associatifs distincts les ACLs référencées et les ACLs définies. Puis, le test vérifie que toutes les ACLs référencées sont définies, et que toutes les ACLs définies sont référencées. Ce test de sécurité signale donc autant de vulnérabilités qu'il peut détecter d'inconsistances dans la configuration d'un équipement. Rappelons qu'une telle vulnérabilité peut avoir un impact sur le réseau s'il s'agit d'une ACL destinée à protéger le réseau et qui n'est pas appliquée.

5.3.2.1. Le contrôle des mots de passe

Le deuxième exemple de test de sécurité "t.pw_01" consiste à vérifier le mot de passe administrateur d'un équipement de type CISCO. CISCO utilise deux algorithmes de hachage pour encoder les mots de passe dans une configuration, "enable password 7" et "enable secret 5". La première fonction "enable password 7" est facilement réversible avec un minimum d'efforts de cryptanalyse. La seconde fonction "enable secret 5" est basée sur la fonction de hachage MD5. L'outil "cisco_crypt", issu du domaine public, offre les options suivantes :

- **cisco_crypt –e7 :** Encode un mot de passe en mode faible, "enable password 7".
- **cisco_crypt –d7 :** décode un mot de passe faible, "enable password 7".
- **cisco_crypt –e5 :** Encode un mot de passe en mode fort, "enable secret 5".

Le test "t.pw_01" consiste donc à vérifier, si les mots de passe encodés par "enable password 7" et "enable secret 5" sont tous les deux présents dans la configuration de l'équipement, qu'ils sont différents. Rappelons aussi que dans ce cas, le mot de passe retenu par l'équipement est le mot de passe "enable secret 5". Pour vérifier ces mots de passe, voici en pseudo_code le test "t.pw_01" :

```
Extraire “enable password 7” de la configuration dans la variable  ENCODED_7
Extraire “enable secret 5” de la configuration dans la variable ENCODED_5
Invoquer "cisco_crypt –d7" pour décoder ENCODED_7 dans la variable DECODED_7
Invoquer "cisco_crypt –e5" pour encoder DECODED_7 dans la variable RECODED_5

if RECODED_5 != ENCODED_5
then
        imprimer "les mots de passe sont différents"
        return TEST_PASSED
else
        imprimer "les mots de passe sont égaux"
return TEST_FAILED
fi
```

Si le test réussi, alors les deux mots de passe sont différents et aucune vulnérabilité de sécurité est signalée. Dans le cas contraire, une vulnérabilité est signalée indiquant que le mot de passe administration n'est pas protégé. Rappelons qu'une telle vulnérabilité peut avoir un impact sur le réseau si le mot de passe est dérobé.

5.3.2.2. Le contrôle des listes de filtrages

Le troisième exemple de test de sécurité "t.acl_02" consiste à vérifier la consistance d'un filtre de trafic ou Access Control List [Valois et Llorens 2002]. Une ACL est une liste de règles ou chaque règle peut être représentée par un tuple de dimension 6 ou les dimensions sont les suivantes :

- Le protocole (ICMP, TCP, UDP,...),
- L’intervalle des adresses sources,
- Les ports sources (TCP ou UDP),
- L’intervalle des adresses destinations,
- Les ports destinations (TCP ou UDP),
- Les options.

Les 6 dimensions sont des ensembles finis. Deux types de permission peuvent être appliqués à une règle d'une ACL. La première permission "permit" laisse passer les paquets de données, la deuxième permission "deny" détruit les paquets de données si ceux-ci correspondent à la règle en cours. Détecter les redondances ou les inconsistances d'un ACL consiste à considérer les règles de l'ACL comme des formes géométriques multidimensionnelles (hyper-rectangles). Le problème de la détection des inconsistances consiste à calculer les intersections géométriques entre les rectangles déduits de ces règles. La

table suivante donne un exemple simplifié d'une ACL composée des règles suivantes :

Règle	Première adresse source	Dernière adresse Source	Première adresse destination	Dernière adresse destination
1	10.0.0.0	10.255.255.255	10.0.0.0	10.255.255.255
2	57.4.0.0	57.7.255.255	57.0.0.0	57.255.255.255
3	57.7.6.0	57.7.6.255	57.7.6.0	57.7.6.255
4	57.0.0.0	57.255.255.255	57.4.0.0	57.5.255.255

Table 5-3: Exemple d'une liste de filtrage

La figure ci-après illustre les intersections entre les rectangles définis ci-dessus :

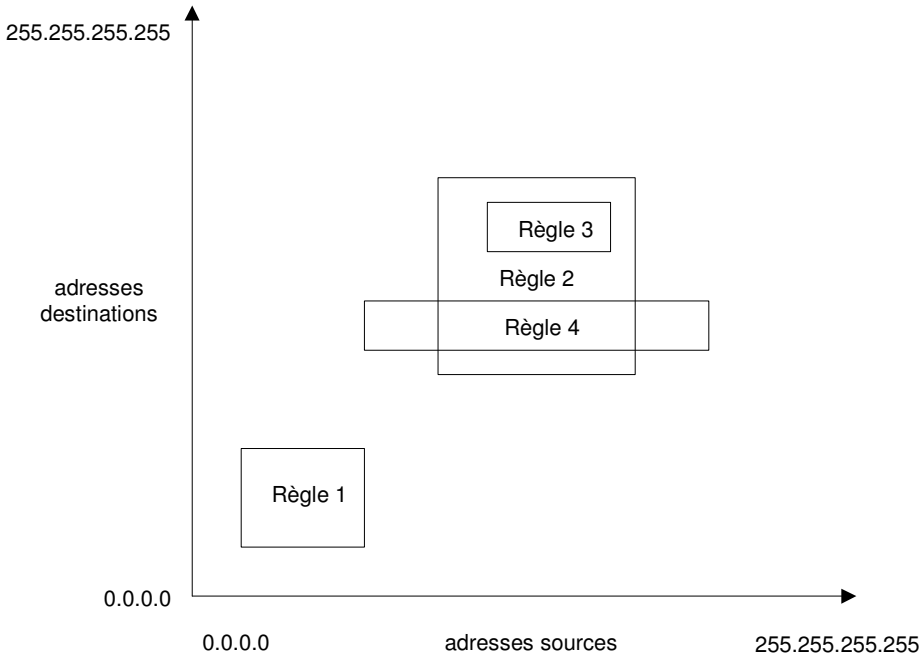


Figure 17: Analyse géométrique d'une liste de filtrage

On constate que la règle 1 est donc complètement isolée des autres règles. La règle 3 est incluse dans la règle 2 et il y a une intersection entre la règle 2 et 4. De manière plus précise, l'interprétation de l'intersection entre les règles 2 et 4 nécessite de connaître les permissions respectivement appliquées à chaque règle.

Dans notre exemple, les règles ont des permissions "permit" signifiant qu'il y a une redondance entre ces règles. Si les deux règles avaient eu des permissions différentes, alors il y aurait eu une inconsistance entre ces règles [Valois et Llorens 2002]. Si nous lançons le test de sécurité "t.acl_02" sur cette ACL contenue dans la configuration de l'équipement conf1, le test détecte alors deux vulnérabilités :

```
$ conf_validation -e't.acl_0[2]' conf[1]
[2] access-list 102 permit ip 57.4.0.0 0.3.255.255 57.0.0.0 0.255.255.255
[3] access-list 102 permit ip 57.7.6.0 0.0.0.255 57.7.6.0 0.0.0.255
*** redundancy   [3] < [2]
[2] access-list 102 permit ip 57.4.0.0 0.3.255.255 57.0.0.0 0.255.255.255
[4] access-list 102 permit ip 57.0.0.0 0.255.255.255 57.4.0.0 0.1.255.255
*** redundancy   [4] * [2] = permit ip 57.4.0.0 0.3.255.255 57.4.0.0 0.1.255.255
```

Rappelons qu'une telle vulnérabilité peut avoir un impact sur le réseau s'il s'agit d'une ACL destinée à protéger le réseau et qui comporte des inconsistances.

L'algorithme utilisé emploie une liste liant les lignes de l'ACL dans leur ordre de positionnement. Quand un paquet doit être comparé à une ACL, ce paquet est comparé à la 1ère ligne, puis à la 2ème, et dans l'ordre jusqu'à la dernière ligne. La comparaison s'arrête dès qu'une ligne correspond à un paquet. L'algorithme de validation peut alors s'écrire à l'aide du pseudo-code suivant [Valois et Llorens 2002] :

```
Pour chaque ligne de l'ACL faire

  Transformer la ligne en un hypercubre

  Pour toutes les lignes précédemment lues faire
      Calculer l'intersection entre la ligne courante et les autres lignes
      Si l'intersection n'est pas vide alors reporter l'intersection
  Fin pour

  Rajouter la ligne courante aux lignes lues

End for
```

Le 6-tuple se compose du protocole d'IP (ICMP, TCP, UDP...), de la plage des adresses IP source, des ports source (TCP et UDP), de la plage des adresses IP destination, des ports de destination (TCP et UDP), et des drapeaux d'états. La complexité en temps de cet algorithme est quadratique avec le nombre de lignes de ACL et linéaire avec le nombre d'équipement réseau.

5.3.2.3. Autres contrôles

Nous détaillons dans ce paragraphe un ensemble de test de sécurité associé à une configuration CISCO. Suivant les versions d'IOS de CISCO, la syntaxe des commandes de configuration peut changer et doit être prise en compte dans la définition des règles de configuration [CISCO 2001, NSA 2003].

Nous définissons aussi par l'expression "ip_admin_range", l'intervalle d'adresses IP autorisée à accéder aux équipements réseau à des fins d'administration.

A) Section: IOS: Configuration générale des routeurs

La configuration générale d'un routeur doit définir les paramètres et services suivants :

- no service tcp-small-servers, no service udp-small-servers : Cette commande désactive les services TCP et UDP echo, discard, daytime et chargen, qui ne sont pas utilisés de manière générale et peuvent permettre de récolter des informations utiles ou de lancer des attaques de type déni de service. Il est préférable que ces services à valeur ajoutée soient assurés par un serveur dédié.
- no ip bootp server : Cette commande désactive le service bootp, qui utilise le routeur pour récupérer des informations réseau et expose de ce fait ce dernier à des attaques de type déni de service. Cette fonctionnalité agit de manière identique au protocole RARP (Reverse Address Resolution Protocol) pour récupérer l'adresse IP ainsi que d'autres informations. Il est préférable que cette fonction soit réalisée par un serveur dédié.
- no service dhcp : Cette commande désactive le service DHCP. Il est préférable que le routeur ne distribue pas les adresses IP de manière dynamique, de façon à ne pas s'exposer à de possibles attaques par déni de service. Un serveur dédié peut jouer ce rôle.
- no finger service, no identd service : Cette commande désactive le service finger, qui permet d'obtenir des informations précieuses sur le système, comme la liste des utilisateurs connectés, les noms des utilisateurs, etc.
- no cdp run : Cette commande désactive le protocole CDP (CISCO Discovery Protocol), qui permet d'obtenir des informations très utiles sur le réseau et d'en déduire son architecture. Elle peut cependant être utilisé de manière ponctuelle afin d'aider à la résolution de problèmes réseau.
- no ip http server : Cette commande désactive le serveur HTTP d'administration. Un routeur est un équipement de routage et doit le rester, de façon à limiter son domaine de responsabilité. Il est préférable de s'orienter vers des accès d'administration sécurisés comme SSH.
- no ip source-route : Cette commande interdit les paquets routés depuis la source IP donnée par un paquet. Cette méthode permet de détourner le protocole de routage standard prévu pour mener des attaques potentielles.
- no boot network, no service config : Cette commande désactive le démarrage en téléchargeant la configuration *via* le réseau. Par principe, toute configuration doit être intégrée et ne doit pas s'exposer à des attaques de type man-in-the-middle, susceptibles de violer la chaîne d'intégrité des configurations des routeurs.
- no service pad : Cette commande désactive le service X.25 PAD (Packet Assembler Disassembler). Le service PAD permet de réaliser des accès Telnet sur l'équipement réseau. Il faut donc le désactiver par défaut, à

moins de l'utiliser dans le cadre d'accès distants définis. Dans ce cas, un chiffrement et une authentification forte de l'utilisateur sont requis.

- service timestamps {log | debug} datetime msec show-timezone localtime : Cette commande active l'horodatage détaillé des informations journalisées sur le routeur, informations fondamentales pour l'investigation de sécurité.
- service tcp-keepalives-in : Cette commande contrôle si les connexions (Telnet ou SSH, par exemple) sont encore actives afin d'éviter de bloquer tous les VTY (Virtual Teletype Terminal) disponibles avec des connexions dites orphelines, susceptibles d'être utilisées par des attaques.
- no ip domain-lookup : Cette commande désactive les requêtes DNS afin de limiter les informations fournies par l'équipement réseau.
- enable secret <mot de passe> : Cette commande active un mot de passe pour passer en mode enable ou mode administrateur. Le mot de passe est stocké après avoir appliqué la fonction de hachage MD5.
- service password-encryption : Cette commande active l'encodage des mots de passe par la fonction de hachage MD5 pour le mot de passe enable, et par l'algorithme de Vigenère pour les autres mots de passe.

B) Section: IOS: Configuration des interfaces des routeurs

Cette section détaille les commandes associées aux interfaces d'un routeur qui doivent être configurées. Il s'agit des interfaces du routeur utilisées pour les connexions vers d'autres systèmes ou routeurs.

- no ip directed-broadcast : Cette commande désactive le Directed Broadcast. De la sorte, le routeur ne réagit pas aux paquets reçus de type broadcast pointant sur une adresse IP. Dans les dernières versions, l'option est positionnée par défaut sur une interface. Cela permet de se prémunir des attaques par déni de service, smurf, etc., et évite que le réseau ne participe à ces attaques de manière indirecte.
- no ip proxy-arp : Cette commande désactive le relayage de messages Address Resolution Protocol sur de multiples segments de LAN. Cette option évite que le routeur, qui agit comme un intermédiaire pour les requêtes ARP, ne casse un périmètre de sécurité en acceptant de manière transparente des accès entre de multiples accès de LAN.
- no ip redirects : Cette commande désactive l'envoi de messages ICMP Redirect. Cette option permet de se prémunir contre les scannings fondés sur le protocole ICMP, lequel permet à un observateur de récolter des informations utiles sur le réseau (topologie, règles de filtrage, etc.).
- no ip unreachables : Cette commande désactive l'envoi de messages ICMP Destination Unreachable. Cette option permet de se prémunir contre les scannings fondés sur le protocole ICMP.

- ip accounting access-violations : Cette commande active la comptabilisation des paquets IP qui violent les ACL. Cette option permet de mesurer ou de quantifier ces violations.
- no ip mask-reply : Cette commande désactive les réponses aux messages ICMP Mask Reply. Cette option permet de se prémunir contre les scannings fondés sur le protocole ICMP.
- no cdp enable : Cette commande désactive CDP. Ce protocole donne des informations très utiles sur le réseau lui-même et permet de déduire son architecture. Elle peut être cependant utilisé de manière ponctuelle afin d'aider à la résolution de problèmes réseau.

C) Section: IOS: Filtrage du trafic sur les interfaces

Un filtre, ou ACL, doit être implémenté en périphérie du réseau, c'est-à-dire sur toutes les interfaces ayant des connexions vers l'extérieur. Cela permet de ne pas recevoir de trafic provenant de préfixes (classes d'adresses IP) non autorisés ainsi que d'éviter que le réseau n'envoie des préfixes non autorisés. Les lignes de configurations ci-après détaillent des filtres sur les flux réseau que l'on peut appliquer au trafic entrant ou sortant d'une interface réseau donnée :

```
# Interface sur laquelle sera appliquée le filtre pour le trafic entrant (ingress) et sortant (egress) du routeur
interface xy
 ip access-group 100 in
 ip access-group 100 out

# Élimination des préfixes non autorisés fondés sur les classes d'adresses IP [RFC1918]
access-list 100 deny ip host 0.0.0.0 any
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 deny ip 10.0.0.0 0.255.255.255 any
access-list 100 deny ip 172.16.0.0 0.15.255.255 any
access-list 100 deny ip 192.168.0.0 0.0.255.255 any
access-list 100 deny ip 192.0.2.0 0.0.0.255 any
access-list 100 deny ip 169.254.0.0 0.0.255.255 any
access-list 100 deny ip 240.0.0.0 15.255.255.255 any

# Filtrage du trafic ICMP autorisé
access-list 100 deny icmp any any fragments
access-list 100 permit icmp any any echo
access-list 100 permit icmp any any echo-reply
access-list 100 permit icmp any any packet-too-big
access-list 100 permit icmp any any source-quench
access-list 100 permit icmp any any time-exceeded
access-list 100 deny icmp any any
```

```
# Autorisation de vos préfixes
access-list 100 permit ip <les préfixes du réseau>
access-list 100 deny ip any
```

D) Section: IOS: Configuration TACACS

Le mot de passe d'un utilisateur peut être stocké localement généralement au format type 7 codé *via* l'algorithme Vigenère, qui est réversible. Les versions récentes d'IOS supportent les mots de passe au format MD5, qui n'est pas réversible.

Bien qu'il soit possible de définir des utilisateurs directement dans la configuration d'un routeur, il est cependant toujours plus sécurisé de définir les comptes utilisateurs, mots de passe et niveaux de privilèges sur un serveur TACACS+ ou RADIUS. L'exemple ci-dessous repose sur un serveur TACACS+.

```
# On définit l'authentification par le serveur TACACS ou par le compte enable si le TACACS n'est pas disponible
aaa new-model
aaa authentication login default group TACACS+ enable
aaa authentication enable default group TACACS+ enable

# Définition de l'autorisation des commandes par le serveur TACACS ou par une configuration locale si le TACACS n'est pas disponible
aaa authorization commands 15 default group TACACS+ local

# Définition de l'accounting associé au serveur TACACS et stockage des accès et des commandes passées sur le routeur
aaa accounting exec default start-stop group TACACS+
aaa accounting commands 15 default stop-only group TACACS+

# Définition des adresses IP des serveurs TACACS ainsi que des clés utilisées pour l'authentification des serveurs
TACACS-server host {IP}
TACACS-server key {clé}
```

E) Section: IOS: Configuration SNMP

Le protocole SNMP v1 doit être protégé par trois éléments de configuration afin d'assurer une sécurité minimale :

```
snmp-server community {communauté} view {nom} RO/RW {ACL}
```

Le premier champ correspond à la communauté SNMP. Il ne doit pas être trivial comme "private", "public" et doit être composé au minimum de 8 caractères et chiffres calculés de manière aléatoire.

Le deuxième champ correspond aux options des droits d'accès, soit lecture seulement, ou RO (Read Only), soit lecture-écriture, ou RW (Read Write). Il est fortement conseillé de ne garder que l'option RO afin d'éviter toute erreur volontaire ou involontaire d'écriture.

Le dernier champ correspond au filtrage des adresses IP autorisées à accéder à SNMP. Cette liste est généralement limitée aux adresses IP de la zone d'administration :

```
# Définition des communautés et des mots de passe dont l'accès est limité
# aux adresses d'administration
snmp-server community r3ad view cutdown RO 10
snmp-server community wr1te RW 10
snmp-server view cutdown ip.21 excluded
snmp-server enable traps <…>
snmp-server host x.x.x.x
snmp-server source loopback0

# ACL filtrant les adresses IP autorisées à accéder au service SNMP du routeur
access-list 10 permit ip_admin_range
```

F) Section: IOS: SSH (Secure SHell)

La configuration et l'activation de SSHv1 ou SSHv2 doit être faite par les commandes suivantes :

```
# Attribution d'un nom au système
hostname {nom}

# Renseignement du domaine DNS requis pour des sessions SSH
ip domain-name {domaine}

# Génération d'une clé publique/privée SSH avec des paramètres optionnels
crypto key generate rsa
ip ssh timeout 60
ip ssh authentication-retries 3
ip scp server enable

# Autorisation de l'accès SSH
line vty 0 4
  transport input SSH
```

G) Section: IOS: IPsec

Il est possible d'établir des tunnels IPsec au niveau d'un routeur pour monter des tunnels IPsec sur le réseau ou vers le routeur à des fins d'administration. Pour ce faire, toutes les options de configuration doivent être définies comme l'illustre la configuration suivante :

```
# Définition de la politique IKE de gestion des clés utilisant l'algorithme de hachage MD5, l'algorithme de chiffrement 3DES, une authentification par secret partagé utilisant l'algorithme de Diffie-Hellman et la clé d'authentification
crypto isakmp policy 1
 hash md5
 encryption 3des
 authentication pre-share
 group 2

# Définition du secret partagé
crypto isakmp key <key> address y.y.y.y

# Définition des transformations autorisées
crypto IPsec transform-set 3desmd5 esp-3des esp-md5-hmac

# Définition de la cryptomap regroupant toutes les définitions précédentes, les voisins IPsec, les transformations ainsi qu'une ACL filtrant les associations de sécurité
crypto map ma-cryptomap 10 ipsec-isakmp
set peer y.y.y.y
set transform-set 3desmd5
match address 110

# Application de la cryptomap aux interfaces nécessaires où l'on souhaite monter des tunnels IPsec ainsi qu'une ACL qui n'autorise que le trafic IPsec
interface xy
 ip address y.y.y.y 255.255.255.0
 crypto-map ma-cryptomap
 ip access-group 100 in

# Définition de l'ACL qui autorise le trafic IPsec entre les systèmes du réseau
access-list 100 permit udp host x.x.x.x host y.y.y.y eq 500
access-list 100 permit esp host x.x.x.x host y.y.y.y
access-list 100 permit ahp host x.x.x.x host y.y.y.y
access-list 100 permit ip <remoteLAN> <localLAN>

# Définition des associations de sécurité
access-list 110 permit ip x.x.x.x <wildcard> y.y.y.y <wildcard>
```

H) Section: IOS: Journalisation des événements

L'équipement ne conservant qu'une quantité limitée d'informations dans un tampon local volatile, il faut envoyer les messages vers un système déporté exécutant un daemon syslog afin de recevoir les journaux sur une plate-forme centrale. Ces journaux peuvent faire l'objet de traitements en cas de problème réseau ou d'investigation de sécurité. Le processus de journalisation doit être configuré avec les commandes suivantes :

```
no logging console
logging on
logging buffered 16384 debugging
logging trap debugging
logging console critical
logging facility local5
logging source-interface loopback0
logging {IP}
```

I) Section: IOS: NTP (Network Time Protocol)

Le temps, ou horloge, d'un équipement est fondamental pour la corrélation d'événements réseau comme une investigation de sécurité ou une analyse d'un problème réseau. L'architecture globale du protocole NTP s'appuie sur différents niveaux de bases de temps, ou strates.

La synchronisation *via* NTP doit être configurée avec les commandes suivantes :

```
# Définition du processus de base de temps fondé sur l'heure GMT avec une
authentification par clé et un filtrage sur les adresses IP autorisées
clock timezone GMT
ntp authentication-key {id} md5 {clé}
ntp authenticate
ntp trusted-key {id}
ntp update-calendar
ntp server {IP}
ntp access-group {query-only | serve-only | serve | peer} protect-ntp
ntp source loopback0

# Définition des classes d'adresses IP autorisées à échanger des messages NTP
ip access-list standard protect-range
 permit ip_admin_range
```

J) Section: IOS: Configuration d'un message d'avertissement

Voici un message type qui doit être configuré :

```
L'accès a ce système n'est autorise qu'aux seuls personnels habilités. Toutes tentatives
d'accès ou tout accès non autorisé sera poursuivi conformément à la loi.

Only authorized users are can access this system. All attempts of intrusion or
intrusions will be prosecuted according laws.
```

K) Section: IOS:Configuration des accès au routeur

Cette section détaille les accès d'administration à un routeur pour se prémunir de toute faiblesse de configuration sur une des parties les plus sensibles de la configuration des routeurs.

Il existe plusieurs méthodes pour se connecter à un équipement, soit directement par le port console, soit par le port AUX (généralement réservé aux accès modem), soit encore par le biais du réseau, au travers d'une interface VTY (Virtual Teletype Terminal).

Dans les exemples ci-dessous, les connexions sont protégées par mot de passe. Les authentifications de type TACACS permettent de gérer des comptes individuels d'accès associés à des types de profils définis, dans lesquels les commandes autorisées sont clairement spécifiées et limitées. On filtre ainsi les classes d'adresses IP autorisées à accéder au routeur, tout en limitant les temps de connexion au routeur sans activité.

La "line console" permet d'accéder directement à l'équipement à l'aide d'un terminal. Par défaut, on se protège au moyen des lignes de configurations suivantes :

```
line con 0
 # Définition d'un mot de passe local
 password 7 …….
 # Définition d'un temps maximal de connexion sans activité
 exec-timeout 15 0
 # Interdiction de réaliser des connexions sortantes
 transport output none
```

La "line aux" permet d'accéder à l'équipement en général à l'aide d'un modem pour des accès de type backup. Par défaut, on se protège au moyen des lignes de configurations suivantes :

```
line aux 0
 # Définition d'un mot de passe local
 password 7 …….
 # Filtrage des adresses IP autorisées à se connecter
 access-class xx in
 # Interdiction de réaliser des connexions entrantes
 transport input none
 # Définition d'un temps maximal de connexion sans activité
 exec-timeout 15 0
  # Interdiction de réaliser des connexions sortantes
 transport output none
```

La "line vty" permet d'accéder à l'équipement, par exemple pour des besoins d'administration. Par défaut, on se protège au moyen des lignes de configurations suivantes :

```
line vty 0 3
 # Définition d'un mot de passe local
 password 7 …….
```

```
# Filtrage des adresses IP autorisées à se connecter:
access-class xx in
# Définition d'un temps maximal de connexion sans activité
exec-timeout 15 0
# Définition des protocoles autorisés à se connecter
transport input ssh
# Interdiction de réaliser des connexions sortantes
transport output none
```

La configuration de chaque équipement réseau est donc primordiale afin de s'assurer de l'intégrité du réseau et de ses services.

5.3.3. Le contrôle des topologies de routage interne IS_IS

Comme nous l'avons vu au chapitre 1, le routage IS_IS est organisé hiérarchiquement de sorte qu'un large domaine puisse être administrativement divisé en aires. On utilise alors des aires IS_IS de niveau 1 et de niveau 2. Le cheminement entre les domaines administratifs est réalisé par les systèmes intermédiaires de bordure. Les systèmes IS_IS du niveau 1 se chargent des systèmes dans leur propre aire, et reroutent sinon le trafic vers les systèmes intermédiaires IS_IS du niveau 2.

Le rôle du protocole de routage IS_IS est de calculer le meilleur chemin interne jusqu'à la frontière d'une aire. Il offre également une détection d'échec rapide et une forte convergence des tables de routage.

Le contrôle des topologies de routage IS_IS nécessite d'avoir l'ensemble des configurations des équipements réseau. Nous décrirons dans un premier temps la politique de sécurité du routage IS_IS, puis nous détaillerons comment vérifier cette politique dans les configurations des équipements réseau.

5.3.3.1. La politique de sécurité du routage IS_IS

Le routage IS_IS peut être sécurisé par les règles de sécurité suivantes.

A) Règle de sécurité des mots de passe

L'authentification des sessions de routage doit être réalisée pour renforcer la sécurité du processus de routage IGP et se protéger contre les attaques de type spoofing et hijacking [RFC3567].

- Règle de sécurité : un mot de passe doit être défini pour chaque session de routage IS_IS afin d'authentifier la session de routage, ainsi que les mises à jour de routage. La granularité des mots de passe par session facilite la gestion opérationnelle des mots de passe du réseau.

B) Règle de sécurité des distances de routage

Les valeurs administratives de distance de routage peuvent être configurées pour marquer les routes internes du cœur de réseau avec une valeur de poids spécifique. Ainsi, toutes les autres routes seront marquées par défaut avec une distance de poids maximum et ne seront pas propagées dans le cœur de réseau.

- Règle de sécurité : des distances administratives de routage doivent être définies afin de contrôler le routage interne IGP du cœur de réseau.

C) Règles de sécurité topologique de routage

La conception de la topologie de routage constitue un des éléments-clés de sécurité d'un cœur de réseau. Sachant que toutes les configurations IS_IS sont d'une nature asymétrique par configuration, tous les graphes IS_IS déduits sont dirigés. Afin d'assurer une résilience des topologies de routage, nous définissons les règles de sécurité suivantes :

- Règle de sécurité : le graphe "IS_IS aire" doit être connexe, où l'aire IS_IS de niveau 2 doit être bi-connecté aux aires de niveau 1 et être un point d'articulation pour le graphe comme l'illustre la figure suivante :

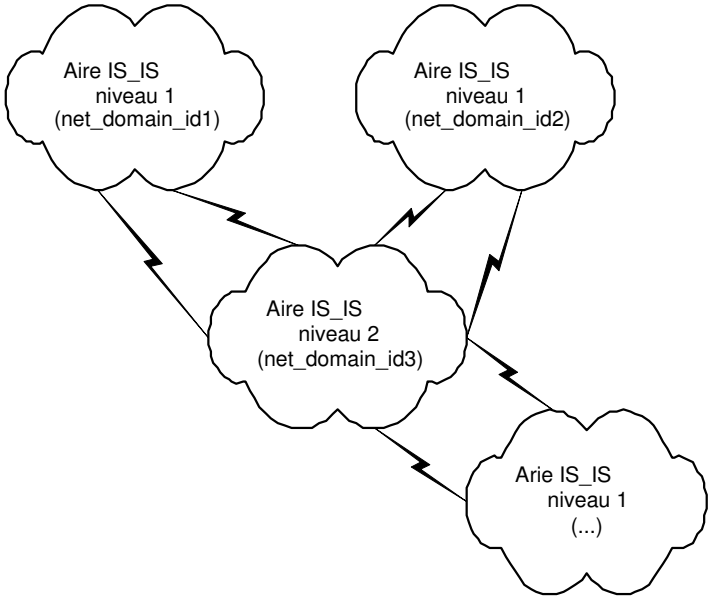


Figure 18 : Vérification de la topologie de routage des aires IS_IS

- Règle de sécurité : pour chaque aire IS_IS de niveau 1 et 2, les graphes "IS_IS aire routeurs" doivent être connexes et sans point d'articulation comme l'illustre la figure suivante :

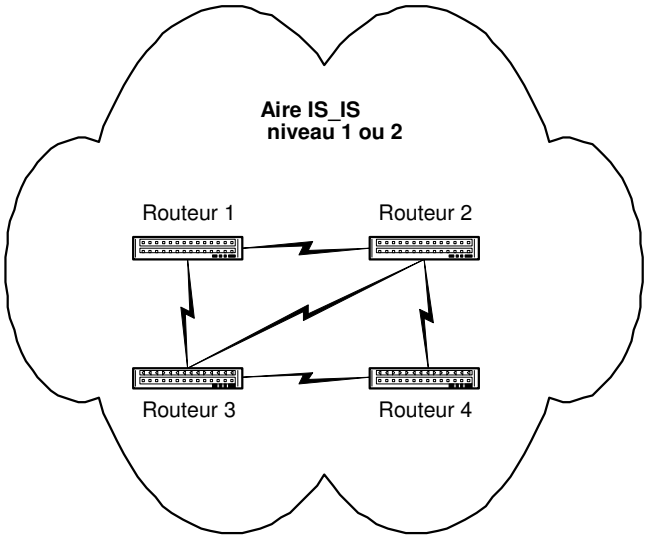


Figure 19: Vérification de la topologie de routage des équipements réseau au sein d'une aire IS_IS

5.3.3.2. Le contrôle de la politique de sécurité du routage IS_IS

A) La vérification des mots de passe

Les mots de passe IS_IS sont présents dans la configuration d'un équipement réseau. Nous pouvons donc extraire cette information en analysant chaque configuration participant au routage IS_IS. Pour une configuration CISCO, la commande de configuration est la suivante :

> isis password password {level-isis}: définit un mot de passe pour une session de routage.

Le processus de contrôle consiste à analyser toutes les configurations des équipements réseau et à vérifier si chaque mot de passe est conforme à la politique de sécurité.

Si la politique déclare que les mots de passe sont identiques pour toutes les sessions de routage dans toutes les configurations des équipements réseau, alors la complexité en temps est linéaire avec le nombre de mots de passe. Autrement, si la politique déclare que les mots de passe sont différents, alors la complexité en temps est de l'ordre de O(|M|*log(|M|)) si |M| est le nombre de mots de passe [Brassard et al. 1996].

B) La vérification des distances de routage

L'information administrative de distance IS_IS est présente dans la configuration d'un équipement réseau. Nous pouvons donc extraire cette information en analysant chaque configuration appartenant au réseau IS_IS. Pour une configuration CISCO, les commandes sont les suivantes :

> distance weight {ip-address {ip-address mask}} [ip access-list] : définit un poids pour les routes.

> access-list access-list-number [dynamic list-name [timeout value]] {deny | permit} protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log| log-input] : définit une liste d'adresses IP.

C) La vérification des topologies de routage

L'information de topologie de routage IS_IS est présente dans la configuration d'un équipement réseau. Nous pouvons donc extraire cette information en analysant chaque configuration d'un équipement réseau appartenant au routage IS_IS. Pour une configuration CISCO, les commandes sont les suivantes :

> hostname name: nom du routeur.
>
> router isis: définit un processus de routage IS_IS.
>
> *i*s-type {level-1 | level-2 | level-1-2}: définit l'aire IS_IS.
>
> net network-entity-title: définit l'adresse de l'aire IS_IS ainsi que son identifiant.
>
> ip router isis: Configure une session de routage IS_IS sur une interface réseau.
>
> ip address ip-address [subnet_mask] : définit l'adresse IP d'une interface réseau.
>
> isis metric default-metric {level-1 | level-2}: définit la métrique IS_IS sur l'interface réseau.

Il doit être noté que d'autres éléments de configuration pourraient être pris en considération comme les notions de priorité IS_IS, les intervalles de retransmission d'état de lien IS_IS, etc.

Le processus de contrôle se compose d'étapes suivantes.

C.1) Extraire les informations de routage

Il faut extraire les informations de routage dans les configurations des équipements réseau afin de créer le fichier "topologie" structuré par les champs suivants :

> <router_name> extrait de la commande **"**hostname name"
>
> <is_type> extrait de la commande **"**is-type {level-1 | level-2}"
>
> <net> extrait de la commande **"**net network-entity-title*"*
>
> <ip_address> extrait de la commande **"**ip address ip-address [subnet_mask] "
>
> <netmask> extrait de la commande **"**ip address ip-address [subnet_mask] "
>
> <IS_IS_metric> extrait de la commande **"**isis metric default-metric {level-1 | level-2}"
>
> <IS_IS_level_number> extrait de la commande **"**isis metric default-metric {level-1 | level-2}"

Ces informations sont alors utilisées pour construire les graphes IS_IS, mais aussi pour valider la conformité et la consistance des configurations IS_IS.

C.2) Construction du graphe "IS_IS aire"

Nous pouvons déduire par la requête algébrique suivante les sommets et les arcs du graphe. Il doit être noté que nous pouvons déduire du champ <net> une marque unique d'aire IS_IS que nous appelons "net_aire" ou <net.net_aire> :

```
/* Liste les aires IS_IS */
Pour chaque valeur dans topologie[net.net_aire] faire

    /* Liste des interconnexions des aires IS_IS */
    topologie[net.net_aire] as a join topologie[net.net_aire] as b
        on bitwise(a[ip_address], a[netmask]) = bitwise(b[ip_address], b[netmask])
    where
        a[net.net_aire] = valeur and
        a[net.net_aire] != b[net.net_aire]

FinFaire
```

Note: 2 routers sont connectés s'ils pointent sur le même domaine réseau comme l'illustre la condition suivante : bitwise(a[ip_address], a[netmask]) = bitwise(b[ip_address], b[netmask])

C.3) Construction du graphe "'IS_IS aire routeur"

Nous pouvons déduire par la requête algébrique suivante les sommets et les arcs du graphe:

```
/* Liste les aires IS_IS */
Pour chaque valeur dans topologie[net.net_aire] faire

    /* Liste les interconnexions des routeurs */
    topologie[router_name] as a join topologie[router_name] as b
        on bitwise(a[ip_address], a[netmask]) = bitwise(b[ip_address], b[netmask])
    where
        a[net.net_aire] = valeur and
        a[net.net_aire] = b[net.net_aire]

FinFaire
```

Note: 2 routers sont connectés s'ils pointent sur le même domaine réseau comme l'illustre la condition suivante : bitwise(a[ip_address], a[netmask]) = bitwise(b[ip_address], b[netmask])

C.4) Vérification des topologies des graphes de routage

Comme ces graphes de routage ne sont pas denses (|E| n'est pas comparable à $|X|^2$), on prend une structure de données par liste d'adjacences afin d'améliorer les complexités en temps des algorithmes utilisés [Brassard et al. 1996].

Si |X| est le nombre de sommets et |E| est le nombre d'arcs dans le graphe, alors la complexité en temps pour vérifier que le graphe "IS_IS aire" est connexe est de l'ordre de O(|X|+|E|). De même pour le graphe "IS_IS aire routeur". L'algorithme est basé sur une recherche en profondeur pour vérifier la connexité et les points d'articulation d'un graphe [Tarjan 1972].

5.3.4. Le contrôle des topologies de routage BGP

Comme nous l'avons vu au chapitre 1, le protocole BGP est un protocole de routage pour interconnecter des systèmes Autonomes. Ces informations de routage de réseau incluent notamment l'information sur la liste des systèmes autonomes (ASs) traversées.

Nous décrirons dans un premier temps la politique de sécurité du routage BGP, puis nous détaillerons comment vérifier cette politique dans les configurations des équipements réseau [Valois et Llorens 2002, Feamster et al. 2003, Feamster 2004].

5.3.4.1. La politique de sécurité du routage BGP

Le routage BGP peut être sécurisé par les règles de sécurité suivantes.

A) La règle de sécurité des mots de passe

L'authentification des sessions de routage doit être réalisée pour renforcer la sécurité du processus de routage BGP et se protéger contre les attaques de type spoofing et hijacking [RFC3567].

- Règle de sécurité : Un mot de passe BGP doit être défini pour chaque session de routage. La granularité des mots de passe par session de routage facilite la gestion opérationnelle des mots de passe du réseau.

B) Les règles de sécurité topologique du routage

La conception de la topologie de routage constitue un des éléments principal de sécurité des sessions de routage d'un réseau avec l'extérieur. Sachant que toutes les configurations BGP sont d'une nature asymétrique par configuration, tous les graphes BGP déduits sont dirigés. Afin d'assurer une résilience des topologies de routage, nous définissons les règles de sécurité suivantes [Llorens et al. 2003] :

- Règle de sécurité : Le graphe “BGP_AS” doit être connexe, et doit avoir au minimum deux sessions de routage pour chacun des AS appartenant à l'opérateur de télécommunications. Chaque session de routage correspond à une session eBGP. Pour les autres sessions, deux sessions de routage sont requises pour les sessions entre les AS de l'opérateur de télécommunications et les AS des opérateurs de transit et les fournisseurs d'accès, et au minimum une session pour les AS des clients. Nous dirons que le graphe est “AS-Connexe” comme l'illustre la figure suivante :

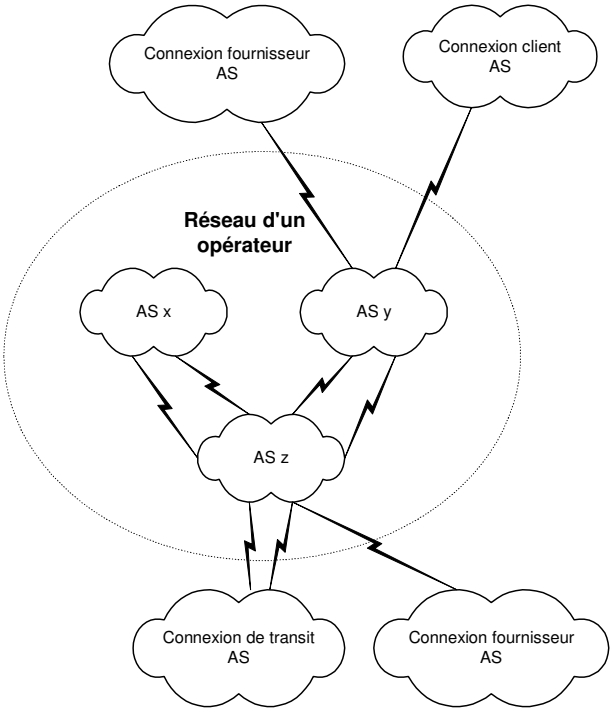


Figure 20: Vérification de la topologie de routage des systèmes autonomes BGP

Cette règle de sécurité tient compte du fait que la topologie de routage d'un réseau doit tenir compte des aspects géographiques et physiques de déploiement du réseau.

- Règle de sécurité : Le graphe “BGP_SUBAS” définit au sein d'un AS doit être connexe et sans point d'articulation. Il s'agit du modèle de confédération BGP. Chaque session de routage correspond à une session eBGP comme l'illustre la figure suivante :

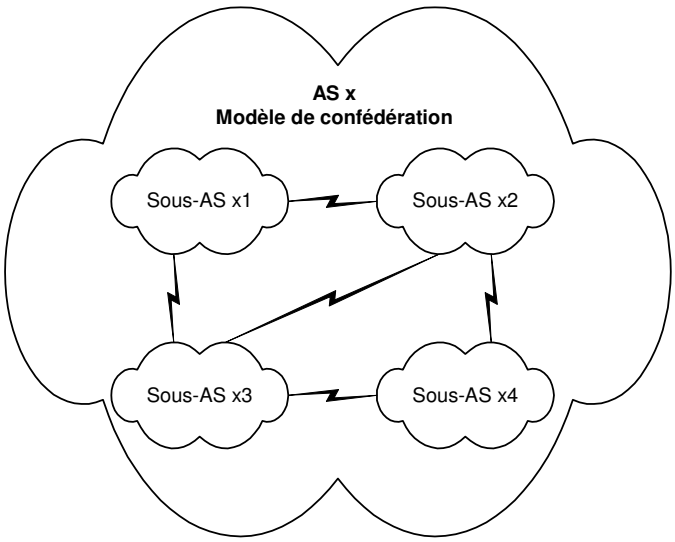


Figure 21 : Vérification de la topologie de routage des sous-systèmes autonomes BGP

Cette règle de sécurité tient compte du fait que la topologie de routage d'un réseau doit tenir compte des aspects géographiques et physiques de déploiement du réseau. Il doit être noté que le modèle "BGP confederation" est de moins en moins utilisé principalement du à la complexité des configurations engendrées.

Règle de sécurité : Le graphe “BGP_SUBAS router” défini au sein d'un SUB_AS ou d'un AS doit être complet, on parle alors du modèle **“**full meshing". Il doit être connexe et sans point d'articulation si l'on parle du modèle “Route Reflector”. Chaque session de routage correspond à une session iBGP comme le montre la figure suivante [RFC2796] :

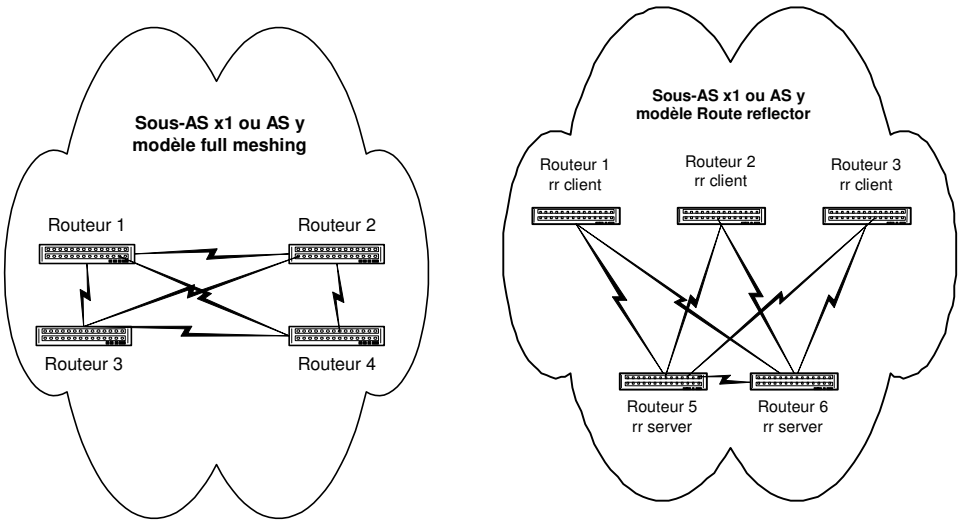


Figure 22 : Vérification de la topologie de routage des équipements réseau BGP

Il doit être noté qu'une combinaison des deux modèles est possible afin d'éviter d'avoir uniquement un modèle **“**full meshing", très consommateur en termes de mémoire et de temps processeur, mais aussi d'avoir un modèle “Route Reflector", apportant des problématiques de routage sous-optimales.

C) Les règles de sécurité des interconnexions de routage

Les interconnexions avec les clients, fournisseurs ou opérateurs de transit doivent être sécurisées afin de protéger le réseau contre les attaques de routage et les problèmes de mauvaises configurations [Valois et Llorens 2002, Feamster et al. 2003, Feamster 2004].

C.1) Les règles de sécurité du trafic de routage entrant

- Règle de sécurité : Un contrôle des adresses IP associées aux mises à jour de routage doit être mis en place afin de protéger le réseau contre les attaques de routes non autorisées ou non allouées [RFC1918, RFC3882]. Pour y parvenir, une liste de préfixes IP non souhaités doit être configuré dans les équipements réseau.

- Règle de sécurité : Une limite maximum des préfixes IP qui peuvent être annoncés doit être mise en place afin de se protéger contre les attaques de type de-aggregation et flooding. Pour y parvenir, chaque session de routage configurée dans les équipements réseau doit mettre en œuvre une limite maximale.
- Règle de sécurité : Un contrôle des AS annoncés doit être mis en place afin de protéger le réseau contre les attaques de type route et AS spoofing. Pour y parvenir, une liste composée d'expressions régulières caractérisant des AS doit être configurée dans les équipements réseau.
- Règle de sécurité : Un contrôle des communautés étendues des mises à jour de routage doit être défini afin de se protéger contre les attaques de type erreur de configuration, des options "additive" et "transitive" des communautés étendues BGP. Pour y parvenir, une liste composée d'expressions doit être configurée dans les équipements réseau pour implémenter une politique de routage.
- Règle de sécurité : Un contrôle des instabilités de routes annoncées doit être mis en place afin de protéger le réseau contre les attaques pouvant impacter le processus de routage. Pour y parvenir, des paramètres de contrôle de l'instabilité des routes annoncées doivent être configurés dans les équipements réseau. Les paramètres de contrôle doivent suivrent certaines recommandations afin de limiter les impacts des instabilités sur les serveurs DNS maîtres d'Internet [RFC2439, RFC3345].

C.2) La règle de sécurité du trafic de routage sortant

- Règle de sécurité : Un contrôle des adresses IP annoncées dans les mises à jour de routage doit être mis en place afin de ne pas propager des adresses non autorisées ou non allouées [RFC1918, RFC3882]. Pour y parvenir, une liste de préfixes IP non souhaités doit être configurée dans les équipements réseau.
- Règle de sécurité : Un contrôle des communautés étendues des mises à jour de routage doit être défini afin de se protéger contre les attaques de type erreur de configuration, des options "additive" et "transitive" des communautés. Pour y parvenir, une liste de filtrage des communautés doit être configurée dans les équipements réseau pour implémenter une politique et des décisions de routage.

5.3.4.2. Le contrôle de la politique de sécurité du routage BGP

A) La vérification des mots de passe

Les mots de passe BGP sont présents dans la configuration d'un équipement réseau. Nous pouvons donc extraire cette information en analysant chaque configuration participant au routage BGP. Pour une configuration CISCO, la commande de configuration est la suivante :

> neighbor {ip-address | peer-group-name} password string: définit un mot de passe pour une session de routage.

Le processus de contrôle consiste à analyser toutes les configurations des équipements réseau et à vérifier si chaque mot de passe est conforme à la politique de sécurité.

Si la politique déclare que les mots de passe sont identiques pour toutes les sessions dans toutes les configurations des équipements réseau, alors la complexité en temps est linéaire avec le nombre de mots de passe. Autrement, si la politique déclare que les mots de passe sont différents, alors la complexité en temps est de l'ordre de O(|M|*log(|M|)) si |M| le nombre de mots de passe [Brassard et al. 1996].

B) La vérification des topologies de routage

La topologie de routage BGP est directement présente dans les configurations des équipements réseau. Nous pouvons donc extraire cette information en analysant chaque configuration participant au routage BGP. Pour une configuration CISCO, les commandes de configuration sont les suivantes :

> hostname name: nom du routeur.
>
> ip address ip-address [subnet_mask] : définit une adresse IP qui sera utilisé pour définir les sessions de routage.
>
> router bgp autonomous-system: définit le système autonome du processus BGP.
>
> bgp confederation identifier autonomous-system: définit l'identifant de la confédération.
>
> bgp confederation peers autonomous-system [autonomous-system] : définit les sessions de routage au sein d'un confédération.
>
> neighbor ip-address …: définit les sessions de routage

Il doit être noté que nous allons extraire toutes les adresses IP des interfaces réseau afin de retrouver toutes les sessions de routage BGP. Cette approche est primordiale, car nous n'avons au premier abord pas de connaissance des sessions de routage.

Le contrôle des topologies de routage suit les étapes suivantes.

B.1) Extraire les informations de routage

Il faut extraire les informations de routage BGP à partir des configurations des équipements réseau afin de créer le fichier “topologie” structuré par les champs suivants :

> <router_name> extrait de la commande "hostname name"
>
> <bgp_as_id> extrait de la commande "router bgp autonomous-system"
>
> <bgp_subas_id> extrait de la commande "bgp confederation identifier autonomous-system"
>
> <bgp_ip_address> extrait de la commande "neighbor ip-address"

et un fichier “donnée” structuré par les champs suivants :

> <router_name> extrait de la commande "hostname name"
> <ip_address> extrait de la commande "ip address ip-address [subnet_mask] "

Ces informations sont utilisées afin de déduire les graphes de routage BGP par une jointure algébrique entre les fichiers “topologie” et “donnée”. Il doit être noté que la symétrie des sessions de routage est possible lorsqu’il s’agit de sessions de routage internes. Dans le cas de sessions de routage externes, les lignes non résolues par l'opération de jointure signifieront qu’il s’agit soit de sessions de routage de routage vers des clients, des fournisseurs ou des opérateurs de transit. Cette opération de jointure permet aussi de valider et de trouver les inconsistances de configuration (contrôle de la définition des AS, contrôle de la définition des peers, etc.)

B.2) Construire le graphe “BGP_AS”

Si on considère les données contenues dans les fichiers "topologie" et "donnée", on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe :

```
/* Liste les aires BGP_AS */
Pour chaque valeur dans topologie[bgp_as_id] faire

  /* Liste les sessions de routage entre les BGP_AS  */
  topologie[bgp_as_id] as a join donnée as b join topologie[bgp_as_id] as c
        on a[bgp_ip_address] = b[ip_address] and
        b[router_name] =c[router_name]
  where
        a[bgp_as_id] = valeur and
        a[bgp_as_id] != c[bgp_as_id];

FinFaire
```

Note: 2 routeurs sont BGP connectés, si un routeur a une session de routage vers l'adresse IP d'une interface du second routeur.

B.3) Construire le graphe “BGP_SUBAS”

Si on considère les données contenues dans les fichiers "topologie" et "donnée, on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe :

```
/* Liste les aires BGP_AS */
Pour chaque valeur in topologie[bgp_as_id] faire

    /* Liste les aires BGP_SUBAS */
```

```
    Pour chaque valeur1 dans topologie[bgp_subas_id] where bgp_as_id = valeur
faire

      /* Liste les sessions de routage entre les BGP_SUBAS */
      topologie[bgp_subas_id] as a join donnée as b join topologie[bgp_subas_id] as c
            on a[bgp_ip_address] = b[ip_address] and
            b[router_name] =c[router_name]
      where
            a[bgp_subas_id] = valeur1 and
            a[bgp_subas_id] != c[bgp_subas_id]

      FinFaire
FinFaire
```

Note: 2 routeurs sont BGP connectés, si un routeur a une session de routage vers l'adresse IP d'une interface du second routeur.

B.4) Construire le graphe "BGP_SUBAS router"

Si on considère les données contenues dans les fichiers "topologie" et "donnée, on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe :

```
/* Liste les aires BGP_AS */
Pour chaque valeur dans topologie[bgp_as_id] faire

   /* Liste les aires BGP_SUBAS */
   Pour chaque valeur1 in topologie[bgp_subas_id] where bgp_as_id = valeur  do

      /* Liste sessions de routage entre les routeurs */
      topologie[router_name] as a join donnée as b join topologie[router_name] as c
            on a[bgp_ip_address] = b[ip_address] and
            b[router_name] =c[router_name]
      where
            a[bgp_subas_id] = valeur1 and
            a[bgp_subas_id] = c[bgp_subas_id]

   FinFaire
FinFaire
```

Note: 2 routeurs sont BGP connectés, si un routeur a une session de routage vers l'adresse IP d'une interface du second routeur.

B.5) Vérifier les topologies des graphes

Comme ces graphes de routage ne sont pas denses (|E| n'est pas comparable à $|X|^2$), on prend une structure de données par liste d'adjacences afin d'améliorer les complexités en temps des algorithmes utilisés [Brassard et al. 1996].

La vérification du graphe “BGP_AS” graphe consiste à valider que le graphe est "as-connected". Si |X| est le nombre de sommets et |E|est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|).

La vérification du graphe “ BGP_SUBAS ” consiste à valider que le graphe est connexe et sans point d'articulation. Si |X| est le nombre de sommets et |E| est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|) [Tarjan 1972].

La vérification du graphe “BGP_SUBAS router” consiste à valider que le graphe est complet pour le modèle "full meshing. Pour le modèle "Route Reflector”, il s'agit de vérifier que le graphe est connexe et sans point d'articulation. Si |X| est le nombre de sommets et |E|est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|).

L'algorithme utilisé pour vérifier les graphes est basé sur une recherche en profondeur pour vérifier la connexité et les points d'articulation d'un graphe [Tarjan 1972].

C) La vérification des interconnexions de routage

Les éléments d'interconnexion sont présents dans les configurations des équipements réseau. Nous pouvons donc extraire ces informations en analysant chaque configuration. Pour une configuration CISCO, les commandes de configuration sont les suivantes :

- neighbor {ip-address | peer-group-name} maximum-prefix maximum [threshold] [warning-only] : limite le nombre de préfixes qui doivent être envoyé à une session de routage.
- neighbor {ip-address | peer-group-name} prefix-list prefix-listname {in | out}: filtre les mises à jour de routage de et vers les sessions de routage externes.
- neighbor {ip-address | peer-group-name} route-map route-map-name {in | out}: applique un contrôle des routes lors des mises à jour de routage entrantes et sortantes.
- set community {community-number [additive]} | none: permet de définir un attribut étendu de routage BGP, appelé communauté.
- set comm-list community-list-number | community-list-name delete: permet de retirer une communauté dans les mises à jour de routage.
- match as-path path-list-number: permet de filtrer les valeurs des systèmes autonomes.
- match community standard-list-number|expanded-list-number|community-list-name [exact-match] : permet de filtrer les mises à jour de routage par une liste de communauté.
- bgp dampening [half-life reuse suppress max-suppress-time] [route-map map] : permet de définir une politique de contrôle des instabilités des mises à jour de routage.

La vérification consiste à extraire et à valider les éléments de filtrage du routage. On doit aussi vérifier la consistance des listes de filtrage. La complexité en temps est linéaire avec le nombre de routeurs et quadratique si on considère le nombre d'éléments contenus dans les listes de filtrages.

5.3.5. Le contrôle des périmètres des réseaux privés virtuels MPLS/VPN

Comme nous l'avons vu au chapitre 1, l'isolation de routage d'un MPLS/VPN est réalisée par le protocole MP-BGP, mais aussi par les importations et les exportations des routes-targets afin de définir la topologie d'un MPLS/VPN.

Nous décrirons dans un premier temps la politique de sécurité des topologies de routage MP-BGP et sa vérification. Nous détaillerons ensuite la politique de sécurité des périmètres MPLS/VPNs et sa vérification. Enfin, nous décrirons deux possibles méthodes permettant de classifier les périmètres des MPLS/VPN.

5.3.5.1. Le politique de sécurité du routage MP-BGP

Les topologies de routage MP-BGP sont équivalentes aux topologies de routage BGP et peuvent être considérées à 2 niveaux. Le premier niveau de topologie considère les systèmes autonomes. Le deuxième niveau de topologie considère les équipements réseau au sein d'un système autonome.

Afin d'assurer une résilience des topologies de routage, nous définissons les règles de sécurité suivantes :

- Règle de sécurité : Le graphe "MP-BGP_AS" doit être connexe, et doit avoir au minimum deux sessions de routage pour chacun des AS appartenant à l'opérateur de télécommunications. Chaque session de routage correspond à une session MP-eBGP. Pour les autres sessions de routage, deux sessions sont requises pour les sessions de routage entre les AS de l'opérateur de télécommunications et les AS des opérateurs de transit et les fournisseurs d'accès, et au minimum une session pour les AS des clients. Nous dirons que le graphe est "AS-Connexe".
- Règle de sécurité : Le graphe "MP-BGP_SUBAS router" défini au sein d'un AS doit être complet, on parle alors du modèle **"**full meshing". Il doit être connexe et sans point d'articulation si l'on parle du modèle "Route Reflector". Chaque connexion correspond à une session MP-iBGP.

5.3.5.2. Le contrôle de la politique de routage MP-BGP

Les éléments de topologie de routage sont présents dans les configurations des équipements réseau. Nous pouvons donc extraire ces informations en analysant chaque configuration participant au routage MP-BGP. Pour une configuration CISCO, les commandes de configuration sont les suivantes :

- hostname name: nom du routeur.
- ip address ip-address [subnet_mask] : définit une adresse IP qui sera utilisé pour les sessions de routage.
- router BGP autonomous-system: définit le système autonome BGP/MP-BGP.
- neighbor ip-address …: définit les sessions de routage unicast Ipv4.
- address-family ipv4 vrf: définit les tables de routage locales VRF.
- address-family vpnv4 : définit les sessions de routage VNPv4. Chacune d'entre elles sont uniques grâce à l'ajout champ RD à l'adresse IP.

Il doit être noté que nous allons extraire toutes les adresses IP des interfaces réseau afin de retrouver toutes les sessions de routage MP-BGP. Cette approche est primordiale, car nous n'avons au premier abord pas de connaissance des sessions de routage MP-BGP.

Les étapes de contrôle des topologies de routage sont les suivantes:

A) Extraire les informations de routage

Il faut extraire les informations de routage MP-BGP à partir des configurations des équipements réseau (PE) afin de créer le fichier “topologie” structuré par les champs suivants :

<router_name>: name: Extrait de la commande "hostname name"

<BGP_as_id>: autonomous-system: Extrait de la commande "router BGP autonomous-system"

<BGP_ip_address>: ip-address: Extrait de la commande "router BGP autonomous-system” and **“**neighbor ip-address"

<MP-BGP-ipv4>: ip-address: Extrait de la commande "address-family ipv4 vrf“ and **“**neighbor ip-address"

<MP-BGP-vpnv4>: ip-address: Extrait de la commande "address-family vpnv4“ et **“**neighbor ip-address"

<type_router>: Extrait de la commande "hostname**,** précise si le routeur est un P, PE ou CE.

et le fichier “donnée” (P, PE and CE routers) structuré par les champs suivants :

<router_name>: name: extrait de la commande "hostname name"

<ip_address>: ip-address: Extrait de la commande "ip address ip-address [subnet_mask] "

<type_router> extrait de la commande "hostname**,** précise si le routeur est un P, PE ou CE.

Ces informations sont utilisées afin de déduire les graphes par une jointure algébrique entre les fichiers “topologie” et “donnée”. Il doit être noté que la symétrie des sessions de routage est possible lorsqu’il s’agit de sessions de routage internes. Dans le cas de sessions de routage externes, les lignes non résolues de l’opération signifieront qu’il s’agit de sessions de routage externes. Cette opération de jointure permet aussi de valider et de trouver les

inconsistances de configuration (contrôle de la définition des AS, contrôle de la définition des peers, etc.).

B) Construire le graphe “MP-BGP AS”

Si on considère les données contenues dans les fichiers "topologie" et "donnée, on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe:

```
/* Liste les domaines MP-BGP AS */
Distinct topologie[BGP_as_id]

/* Détermine les sommets du graphe */
Pour chaque valeur dans topologie[BGP_as_id] faire

    /* Liste les sessions de routage entre les MP-BGP AS */
    topologie[BGP_as_id] as a join donnée as b join topologie[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] = valeur and
          a[BGP_as_id] != c[BGP_as_id];

FinFaire
```

Note: 2 routeurs sont MP-BGP connectés, si un routeur a une session de routage vers l'adresse IP d'une interface du second routeur.

C) Construire le graphe “MP-BGP AS router”

Si on considère les données contenues dans les fichiers "topologie" et "donnée", on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe:

```
/* Liste les domaines MP-BGP AS */
Pour chaque valeur in topologie[BGP_as_id] do

    /* Détermine les sommets du graphe */
    Distinct topologie[router_name] where BGP_as_id = valeur

        /* Liste les sessions de routage entre les routeurs MP-BGP */
        topologie[router_name] as a join donnée as b join topologie[router_name] as c
        on
              a[MP-BGP-vpnv4] = b[ip_address] and
              b[router_name] =c[router_name]
        where
```

```
            a[BGP_as_id] = valeur and
            a[BGP_as_id] = c[BGP_as_id] and
            a[type_router] != “CE” and
            b[type_router] != “CE”

    /* Liste les sessions de routage entre les routeurs BGP */
    topologie[router_name] as a join donnée as b join topologie[router_name] as c
    on
        a[MP-BGP-ipv4] = b[ip_address] and
        b[router_name] =c[router_name]
    where
        a[BGP_as_id] = valeur and
        a[BGP_as_id] = c[BGP_as_id] and
        a[type_router] = “PE” and
        b[type_router] = “CE”
FinFaire
```

Note: 2 routeurs sont MP-BGP connectés, si un routeur a une session de routage vers l'adresse IP d'une interface du second routeur.

Comme ces graphes de routage ne sont pas denses (|E| n'est pas comparable à $|X|^2$), on prend une structure de données par liste d'adjacences afin d'améliorer les complexités en temps des algorithmes utilisés [Brassard et al. 1996].

Les inconsistances seront détectées durant la construction du graphe.

La vérification du graphe “MP-BGP_AS” consiste à valider que le graphe est "AS-connected". Si |X| est le nombre de sommets et |E| est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|).

La vérification du graphe “MP-BGP_SUBAS router” consiste à valider que le graphe est complet pour le modèle "full meshing". Pour le modèle "Route Reflector”, il s'agit de vérifier que le graphe est connexe et sans point d'articulation. Si |X| est le nombre de sommets et |E| est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|).

L'algorithme utilisé pour vérifier les graphes est basé sur une recherche en profondeur pour vérifier la connexité et les points d'articulation d'un graphe [Tarjan 1972].

5.3.5.3. La politique de sécurité du périmètre d'un MPLS/VPN

Le mécanisme par lequel un MPLS VPN contrôle la topologie de routage d’un VPN est l'utilisation de route-target ou communauté étendue MP-BGP. Une route-target suit un format prédéfini afin de créer un VPN en important ou en exportant cette route-target. L'importation d'une route-target signifie que vous apprenez les routes de cette route-target, l’exportation d’une route-target

indique que vous envoyez vos routes associées à cette route-target. Un VPN est donc créer en important et en exportant un même route-target. Notons qu'un VPN est généralement configuré sur plusieurs PEs.

On peut déduire des configurations des équipements réseau le graphe MPLS VPN où chaque sommet représente un VPN, et où un arc dirigé entre deux VPNs est déduit des exportations et importations des routes-targets. L'asymétrie des configurations indique que le graphe est dirigé.

Sachant que les erreurs de configuration peuvent affecter l'intégrité d'un VPN, nous définissons les règles de sécurité génériques suivantes :

- Règle de sécurité : La configuration d'un VPN doit correspondre à ce qui a été demandée par le client. Toute violation peut mettre en péril le périmètre d'un VPN et son intégrité.
- Règle de sécurité : Toute exportation d'une route-target VPN doit référer, au sein du réseau MPLS VPN, à une importation de cette route-target VPN.
- Règle de sécurité : Toute importation d'une route-target VPN doit référée, au sein du réseau MPLS VPN, à une exportation de cette route-target VPN.
- Règle de sécurité : L'importation ou l'exportation de routes-targets non autorisées (violation du format prédéfini) ou interdites (utilisation à des fins administratives) ne doivent pas faire partie d'une configuration d'un VPN.

5.3.5.4. Le contrôle de la politique de sécurité du périmètre d'un MPLS/VPN

Sachant que les éléments topologiques des VPNs sont présents dans les configurations des équipements réseau. Nous pouvons donc extraire ces informations en analysant chaque configuration contenant des VPNs. Pour une configuration CISCO, les commandes de configuration sont les suivantes :

> ip vrf vrf_name: permet de créer une table de routage VRF avec le suffixe vrf_name.
>
> route-target *{*import|export|both*}* *route-target-ext-community* : permet de créer une topologie de connexité pour une VRF donnée en important ou exportant des routes-targets.

Les étapes permettant de contrôler les périmètres des VPNs sont les suivantes :

A) Extraire les informations topologiques des VPNs

Cette étape consiste à extraire les informations topologiques des VPNs à partir des configurations des équipements réseau (PE) afin de créer le fichier "topologie" structuré par les champs suivants :

> <router_name>: name: Extrait de la commande **"**hostname name"
>
> <vrf_name>: name: Extrait de la commande "ip vrf vrf_name"
>
> <*rt*>: *route-target-ext-community: E*xtrait de la commande "route-target {import | export} *route-target-ext-community*"

> <im_ex>: “export” |”import”: Extrait de la commande "route-target {import | export} *route-target-ext-community*"

Ces informations sont alors utilisées afin de construire le graphe MPLS VPN.

B) Construire le graphe MPLS VPN:

Cette étape consiste à déterminer les sommets du graphe à partir du fichier "topologie". Les sommets peuvent être directement déduit à partir du champ “vrf_name”.

Pour les arcs du graphe, il faut déterminer les interconnexions entre les VPNs à partir des exportations et des importations des routes-targets. Nous considérerons aussi que le nom d’une VRF est unique et désigne un unique VPN parmi l’ensemble des configurations des équipements réseau.

Par l’asymétrie des configurations des routes-targets, si un VPN(A) réalise une exportation du route-target RT(x), et que le VPN(B) réalise une importation du RT(x), il existe alors un arc entre le VPN(A) et le VPN(B). Si l’on souhaite avoir une connexion réseau entre ces deux VPNs, le VPN(A) doit alors réaliser une importation du route-target RT(x), et que le VPN(B) doit réaliser une exportation du RT(x), afin de créer un arc entre le VPN(B) et le VPN(A).

Si pour chaque route-target RT(x), on construit la liste des VRFs qui réalisent une exportation et la liste des VRFSs qui réalisent une importation, on peut alors déduire tous les arcs du graphe MPLS VPN comme illustré à la figure suivante :

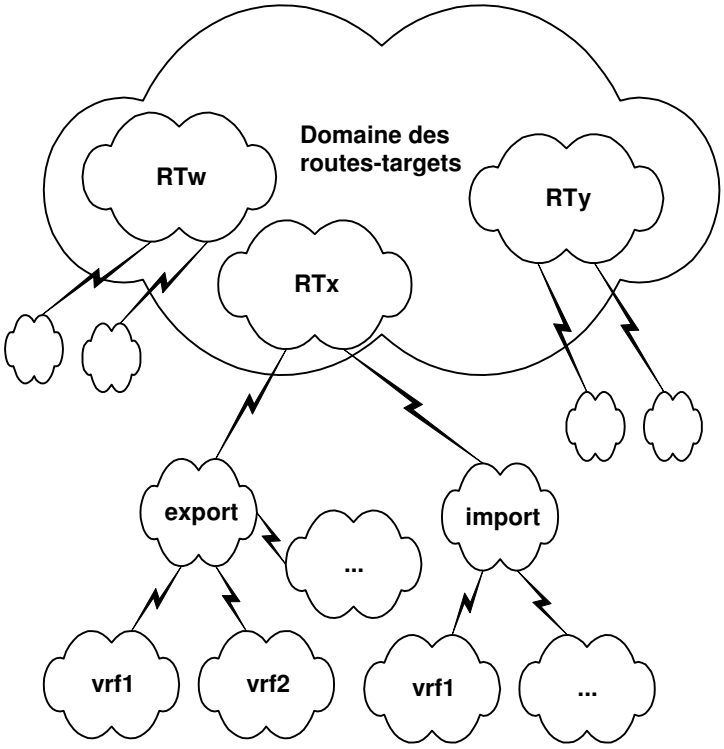


Figure 23: Hiérarchie des routes-targets associées aux MPLS/VPNs

Si on considère les données contenues dans le fichier "topologie", on peut alors déduire par la requête algébrique suivante les sommets et les arcs du graphe MPLS/VPN :

```
/* Détermine les sommets du graphe */
Distinct topologie[vrf_name]

/* Détermine les arcs du graphe */
Pour chaque valeur dans topologie[rt] faire

    /* Liste les interconnexions entre les VPNs */
   topologie[vrf_name] as a join
   topologie[vrf_name] as b
   on
            a[rt] = b [rt] = value
   where
           a[im_ex] = “export” and b[im_ex] = “import”
           and  a[vrf_name]! = b[vrf_name]

FinFaire
```

Note: la VRF utilisée pour l'administration du réseau doit être connectée par définition à toutes les autres VRFs.

Comme ces graphes de routage ne sont pas denses (|E| n'est pas comparable à $|X|^2$), on prend une structure de données par liste d'adjacences afin d'améliorer les complexités en temps des algorithmes utilisés [Brassard et al. 1996].

Les inconsistances seront détectées durant la construction du graphe.

Les composantes fortement connexes du graphe VPN donnent les périmètres de sécurité réels des VPNs qui ont pu être définis dans les configurations. Ces périmètres de sécurité montrent alors soit l'isolation d'un VPN donné, soit des interconnexions avec d'autres VPNs. Le calcul des points d'articulation du graphe VPN permet aussi de connaître les points de passage obligé dans une composante fortement connexe, et donne ainsi des informations topologiques de sécurité intéressantes. Si |X| est le nombre de sommets et |E| est le nombre d'arcs, alors la vérification se réalise avec une complexité en temps de l'ordre de O(|X|+|E|) [Tarjan 1972].

Enfin, si les composantes connexes (si il existe un chemin entre toute paire de sommets (x,y) de la composante) du graphe VPN ne sont pas égales aux composantes fortement connexes (si pour toute paire de sommets (x,y) de la composante, il existe un chemin de x à y et de y à x) du graphe VPN, il y a alors des inconsistances de configuration des VPNs. De même, toute

configuration non bidirectionnelle entre deux sommets montre aussi des inconsistances de configuration des VPNs.

5.3.5.5. Classification du périmètre d'un VPN par une analyse de graphe

Aucune preuve mathématique sur les éléments théoriques utilisés ainsi que la validité de cette mesure ne sont démontrées. Ce paragraphe permet cependant de donner une possible voie de recherche sur des métriques de sécurité réseau.

Si le périmètre d'un VPN(x) se définit comme l'ensemble des VPNs qui sont reliés directement ou indirectement à ce VPN(x), le périmètre d'un VPN(x) peut alors être caractérisé par les attributs minimaux suivants :

- Nombre de connexions directes de ce VPN(x) à d'autres VPNs (NDC : Network Direct Connection) : Cet attribut représente les connexions directes au VPN(x). Cet attribut peut être interprété comme un niveau de menaces directes ou de chemins directs de pénétration. Cet attribut est calculé en comptant le nombre d'arcs directs à ce VPN dans le composant fortement connexe du graphe MPLS VPN le contenant.
- Nombre de connexions de ce VPN(x) à d'autres VPNs (NPC : Network Path Connection) : Cet attribut représente toutes les connexions à un VPN. Cet attribut peut être interprété comme un niveau potentiel de menaces ou de chemins de pénétration. Il doit être noté que l'effort exigé pour établir un chemin indirect de pénétration est probablement beaucoup plus difficile comparé à une connexion directe. Cet attribut est calculé en comptant le nombre de chemins possibles à ce VPN dans le composant fortement connexe du graphe MPLS VPN le contenant.
- Est-ce un point de passage obligé entre VPNs ? (A : Articulation point) : Il indique si le VPN(x) agit comme un point de passage obligé entre différents VPNs. Il peut être interprété comme un point de faiblesse, car la plupart des chemins de pénétration entre les différents VPNs doivent donc passer par ce VPN. Cet attribut est calculé en déterminant les points d'articulation du graphe MPLS VPN.
- Nombre d'éléments contenus dans le périmètre d'un VPN(x) (NE : Network Element) : Cet attribut représente le nombre de VPNs reliés directement ou indirectement à ce VPN. Cet attribut peut être interprété comme la cardinalité du périmètre de sécurité du VPN. Plus cette cardinalité est important, plus le VPN pourrait faire face à des menaces importantes. Cet attribut est calculé en comptant le nombre de sommets contenus dans le composant fortement connexe du graphe MPLS VPN le contenant.

Ces attributs sont limitatifs et peuvent être sujets à de nombreux commentaires suivant différent types de topologies réseau. C'est un point très limitatif de la méthode qui doit être revu dans de futures recherches.

Différentes topologies de connexité de VPNs peuvent être envisagées comme l'illustre les figures suivantes (chaque connexion est bidirectionnelle) :

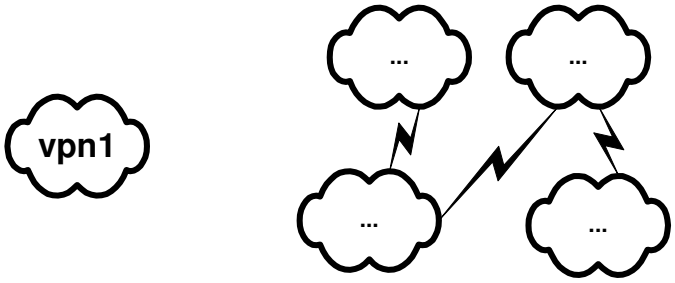


Figure 24: Mesure du périmètre de sécurité, le VPN est isolé

Si on considère le VPN_1, les caractéristiques de son périmètre sont les suivantes : 0 connexion directe, 0 chemin de connectivité, le VPN_1 n'est pas un passage obligé et il y a un sommet $\{VPN_1\}$ dans le composant fortement connexe contenant le VPN_1.

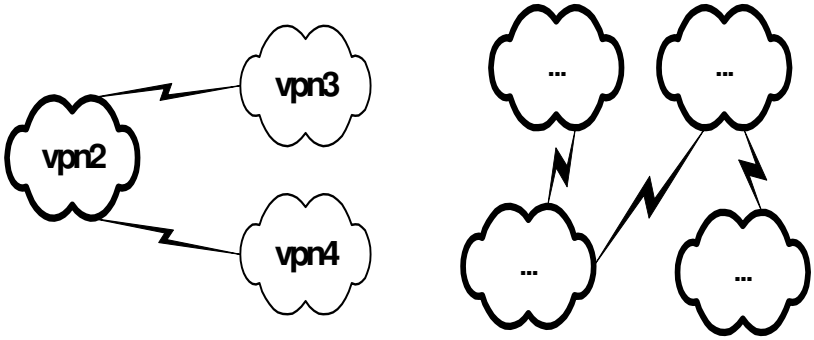


Figure 25: Mesure du périmètre de sécurité, le VPN a des interconnexions

Si on considère le VPN_2, les caractéristiques de son périmètre sont les suivantes : 2 connexions directes, 2 chemins de connectivité, le VPN_2 est un passage obligé et il y a 3 sommets $\{VPN_2, VPN_3, VPN_4\}$ dans le composant fortement connexe contenant le VPN_2.

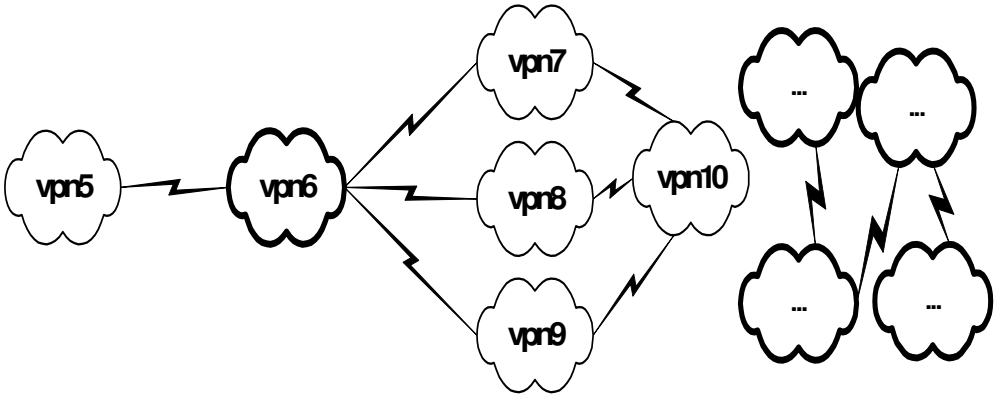


Figure 26: Mesure du périmètre de sécurité, le VPN a des interconnexions indirectes

Si on considère le VPN_6, les caractéristiques de son périmètre sont les suivantes : 4 connexions directes, 7 chemins de connectivité, le VPN_6 est un

passage obligé et il y a 6 sommets {VPN_5, VPN_6, VPN_7, VPN_8, VPN_9, VPN_{10}} dans le composant fortement connexe contenant le VPN_6.

Basé sur ces attributs, on peut alors définir une classification (basé sur l'exposition des menaces) du périmètre d'un VPN. Pour y parvenir, nous utiliserons une méthode de décision multicritères que nous décrirons ci-après afin de comparer des attributs de nature différente. En guise d'exemple, nous calculerons cette classification pour les périmètres des VPNs suivants dans les topologies réseau A, B, C :

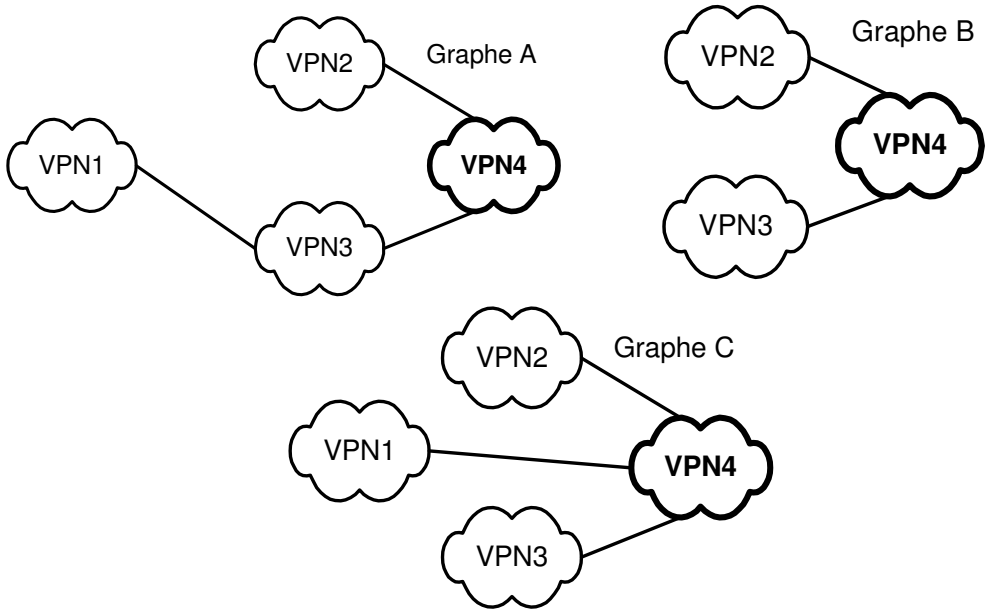


Figure 27: Exemples de graphes de VPNs

Notre mesure se base sur la méthode AHP (Analytic Hierarchy Process). La méthode AHP se base elle-même sur les attributs de sécurité définis précédemment.

La méthode AHP est un outil permettant de prendre des décisions dans un système complexe multicritères. Pour y parvenir, la méthode précise à la fois une échelle de comparaison entre les différents critères à comparer, mais fournit aussi une méthode pour déterminer la décision la plus judicieuse à prendre.

La méthode consiste à construire une matrice AHP qui est symétrique, où chaque élément est représenté dans l'échelle fournie. Une fois la matrice construite, T.Saaty a démontré que le vecteur propre associé à la plus grande valeur propre réelle λ_{max} permettait de déterminer le meilleur choix possible [Saaty 1988].

De cette manière, si on compare les attributs entre eux, et chaque VPN à chacun des autres VPNs pour chaque attribut, on peut alors obtenir une classification du périmètre d'un VPN pour une topologie réseau donnée. De même, on peut obtenir une mesure du périmètre d'un VPN pour différentes topologies réseau.

A) Calcul du périmètre d'un VPN pour une topologie réseau

La matrice AHP de comparaison des attributs est la suivante (la matrice a été déterminée par des experts) :

	NDC	NPC	A	NE
NDC	1	3	7	5
NPC	1/3	1	5	3
A	1/7	1/5	1	1/5
NE	1/5	1/3	5	1

Le vecteur propre pour la valeur propre réelle maximale "4,24" est le suivant (0.56, 0.26, 0.05, 0,14).

La matrice AHP de comparaison des VPNs pour le graphe B par rapport à l'attribut NDC est la suivante (la matrice a été déterminée par des experts) :

	VPN2	VPN3	VPN4
VPN2	1	1	1/3
VPN3	1	1	1/3
VPN4	3	3	1

(comparé sur le critère "NDC")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.2, 0.2, 0.6).

La matrice AHP de comparaison des VPNs pour le graphe B par rapport à l'attribut NPC est la suivante (la matrice a été déterminée par des experts) :

	VPN2	VPN3	VPN4
VPN2	1	1	1
VPN3	1	1	1
VPN4	1	1	1

(comparé sur le critère "NPC")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.33, 0.33, 0.33)

La matrice AHP de comparaison des VPNs pour le graphe B par rapport à l'attribut A est la suivante (la matrice a été déterminée par des experts) :

	VPN2	VPN3	VPN4
VPN2	1	1	1/3
VPN3	1	1	1/3
VPN4	3	3	1

(comparé sur le critère "A")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.2, 0.2, 0.6).

La matrice AHP de comparaison des VPNs pour le graphe B par rapport à l'attribut NE est la suivante (la matrice a été déterminée par des experts) :

	VPN2	VPN3	VPN4
VPN2	1	1	1
VPN3	1	1	1
VPN4	1	1	1

(comparé sur le critère "NE criteria)

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.33, 0.33, 0.33).

Comme il est défini dans la méthode AHP, nous calculons ainsi la classification finale basée sur les priorités des vecteurs déterminés précédemment comme l'illustre le calcul suivant :

	NDC	NPC	G	NE				
						0.56		0,25
VPN2	0,2	0,33	0,2	0,33	*	0.26	=	0,25
VPN3	0,2	0,33	0,2	0,33		0.05		0,50
VPN4	0,6	0,33	0,6	0,33		0.14		

On peut donc quantifier les périmètres des VPNs dans une topologie réseau donnée. Dans notre exemple, le VPN_2 et le VPN_3 sont moins exposés aux menaces que le VPN_4 (0,5) dans la topologie réseau B.

B) Le périmètre d'un VPN pour différentes topologies réseau

On peut aussi de quantifier le périmètre d'un VPN pour différentes topologies réseau. Calculons le périmètre du VPN_4 dans les topologies réseau A, B et C.

La matrice AHP de comparaison des attributs est la suivante (la matrice a été déterminée par des experts) :

	NDC	NPC	A	NE
NDC	1	3	7	5
NPC	1/3	1	5	3
A	1/7	1/5	1	1/5
NE	1/5	1/3	5	1

Le vecteur propre pour la valeur propre réelle maximale est le suivant "4,24" (0.56, 0.26, 0.05, 0,14).

La matrice AHP de comparaison du VPN_4 pour les graphes A, B, C par rapport à l'attribut NPC est la suivante (la matrice a été déterminée par des experts) :

	A	B	C
A	1	1	1/3
B	1	1	1/3
C	3	3	1

(comparé sur le critère "NDC") ---

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.2, 0.2, 0.6).

La matrice AHP de comparaison du VPN_4 pour les graphes A, B, C par rapport à l'attribut NPC est la suivante (la matrice a été déterminée par des experts) :

	A	B	C
A	1	3	3
B	1/3	1	1
C	1/3	1	1

(comparé sur le critère "NPC")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.6, 0.2, 0.2).

La matrice AHP de comparaison du VPN_4 pour les graphes A, B, C par rapport à l'attribut A est la suivante (la matrice a été déterminée par des experts) :

	A	B	C
A	1	1	1
B	1	1	1
C	1	1	1

(comparé sur le critère "A")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.33, 0.33, 0.33).

La matrice AHP de comparaison du VPN_4 pour les graphes A, B, C par rapport à l'attribut NE est la suivante (la matrice a été déterminée par des experts) :

	A	B	C
A	1	3	1
B	1/3	1	1/3
C	1	3	1

(comparé sur le critère "NE")

Le vecteur propre pour la valeur propre réelle maximale "3" est le suivant (0.43, 0.14, 0.43).

Comme il est défini dans la méthode AHP, nous calculons ainsi la classification finale basée sur les priorités des vecteurs déterminés précédemment comme l'illustre le calcul suivant :

	NDC	NPC	G	NE				
A	0,2	0,6	0,33	0,43		0.56		0,34
B	0,2	0,2	0,33	0,14	*	0.26	=	0,20
C	0,6	0,2	0,33	0,43		0.05		0,46
						0.14		

On peut donc quantifier le périmètre d'un VPN dans différentes topologies réseau. Dans notre exemple, le graphe B (0,20) offre un périmètre moins exposé aux menaces pour le VPN_4 que le graphe A (0,34), et le graphe A (0,34) offre un périmètre moins exposé aux menaces pour le VPN_4 que le graphe C (0,46).

5.3.5.6. Classification du périmètre d'un VPN par une analyse bayésienne

Aucune preuve mathématique sur les éléments théoriques utilisés ainsi que la validité de cette mesure ne sont démontrées. Ce paragraphe permet cependant de donner une possible voie de recherche sur des métriques de sécurité réseau.

Si le périmètre d'un VPN se définit comme l'ensemble des VPNs qui sont reliés directement ou indirectement à ce VPN, le périmètre d'un VPN peut alors être caractérisé par l'effort pour un attaquant de pénétrer ce VPN.

Un réseau bayésien est la convergence de la théorie des graphes et de la théorie des probabilités. Dans le cadre d'un réseau, le graphe d'un VPN peut être vu comme un réseau bayésien s'il subit une certaine transformation. Le graphe doit être en effet un graphe acyclique dirigé pour être un réseau bayésien (Directed Acyclic Graph). Cette approche permet de mesurer la probabilité qu'un VPN soit pénétré par d'autres VPNs si nous prenons en compte la topologie du graphe MPLS VPN et les distributions de probabilités.

Le graphe associé à un VPN n'est pas par défaut par graphe acyclique dirigé. Cependant, si nous souhaitons calculer le périmètre d'un VPNi au sein d'un réseau bayésien, nous devons alors transformer le graphe en un graphe $DAG(VPN_i)$. Nous limitons le calcul de notre périmètre du VPN à un unique DAG déduit du graphe initial. La raison est que le nombre de DAGs qui peuvent être déduits d'un graphe est d'ordre exponentiel. Le graphe DAG est alors calculé à partir du nœud VPN_i en utilisant un algorithme d'arbre

recouvrant de coût minimum [Cogis et al. 2003]. C'est un point très limitatif de la méthode qui doit être revu dans de futures recherches.

L'objectif de cette approche consiste à définir une mesure basée sur la probabilité de pénétrer un VPN. L'idée fondamentale de cette mesure est d'additionner toutes les probabilités qui pourraient permettre de pénétrer un VPN dans une topologie réseau donné. Ainsi, plus la valeur numérique de cette mesure est importante, plus le périmètre de sécurité du VPN est à risque. De plus, cette mesure permet aussi de quantifier un niveau de sécurité du périmètre d'un VPN pour différentes topologies réseau.

Par définition, un réseau bayésien est un graphe acyclique dirigé où chaque sommet (c.-à-d. un VPN) est une variable aléatoire. Ainsi, si un réseau bayésien se compose des variables aléatoires suivantes (X_1, X_2 X_n), où X_i est "vrai" si le VPN_i a été pénétré et "faux" autrement, pour $i \in [\ 1..n\ ]$, alors la probabilité que le VPN_i soit pénétré est [Jensen 2001] :

Pour $i \in [1..n]$ et si k est le nombre d'événements pour lequel X_i est dans l'état "Vrai", nous avons alors les formules suivantes :

$P(X_i = Vrai) = P(X_i = Vrai, X_1(1), X_2(1) \dots X_n(1), .., X_1(k), X_2(k) \dots X_n(k))$

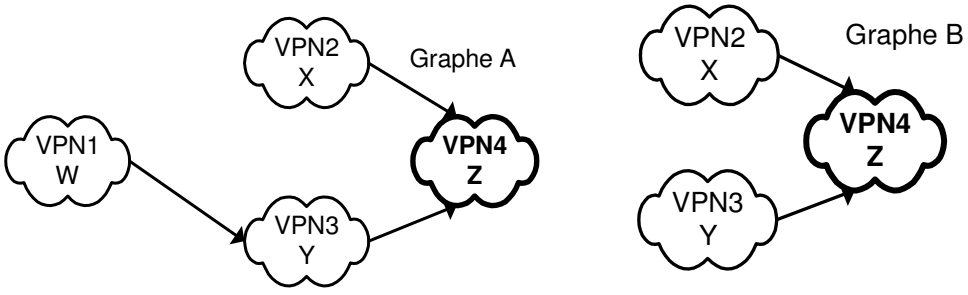
Si $(X_i = Vrai, X_1(1), X_2(1) \dots X_n(1))$, …, $(X_i = Vrai, X_1(k), X_2(k) \dots X_n(k))$ sont mutuellement exclusifs, nous avons alors la formule suivante :

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

Si on considère qu'il s'agit d'une distribution de probabilités, où $Pa(X_j)$ représente les parents de X_j (où causes) dans le réseau bayésien, alors nous avons la formule suivante :

$$P(X_i = Vrai) = \sum_{m=1}^{k} P(X_i = Vrai \mid X_1(m), X_2(m) \dots X_n(m)) * \prod_{j=1}^{n} P(X_j(m) \mid Pa(X_j(m)))$$

Basé sur ces formules, on peut alors définir une classification (basé sur l'exposition des menaces) du périmètre d'un VPN. En guise d'exemple, calculons le périmètre de sécurité des VPNs suivants dans les topologies réseau A, B, C :



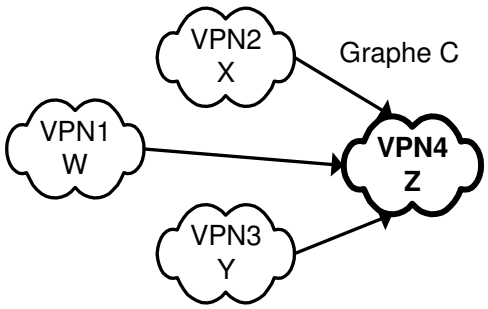


Figure 28: Exemples de graphes bayésiens contenant différents VPNs

Le graphe A contient 4 VPNs qui peuvent être associés aux variables aléatoires discrètes suivantes (W,X,Y,Z). Chaque variable aléatoire a deux états, l'état "Vrai" signifie que le VPN a été pénétré, et l'état "Faux" signifie que le VPN n'a pas été pénétré.

Si nous souhaitons mesurer le périmètre de sécurité du VPN_4 associé à la variable aléatoire Z, nous avons alors besoin de fixer les probabilités suivantes :

P(W=Vrai) = 0,5	P(W=Faux) = 0,5
P(X=Vrai) = 0,5	P(X=Faux) = 0,5
P(Y=Vrai) = 0,5	P(Y=Faux) = 0,5
P(Z=Vrai) = 0,5	P(Z=Faux) = 0,5

Table 5-4: Probabilités de pénétration d'un VPN

Nous considérons par défaut que la probabilité qu'un VPN soit pénétré ou non est la même dans l'absolu. Il doit être noté que d'autres valeurs de probabilités peuvent être considérées.

Nous devons aussi considérer les probabilités conditionnelles suivantes :

Sachant que	W = T	W = F
P(Y = T)	0,8	0,5
P(Y = F)	0,2	0,5

Table 5-5: Probabilités conditionnelles (1) de pénétration d'un VPN

Sachant que	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
P(Z = T)	0,8	0,8	0,8	0,5	0,5	0,8	0,8	0,8
P(Z = F)	0,2	0,2	0,2	0,5	0,5	0,2	0,2	0,2

Table 5-6: Probabilités conditionnelles (2) de pénétration d'un VPN

Nous considérons par défaut que la probabilité qu'un VPN soit pénétré ou non est la même dans l'absolu si les parents n'ont été pénétrés. Nous considérerons

cependant des probabilités différentes si un des parents a été pénétré. Il doit être noté que d'autres valeurs de probabilités peuvent être considérées.

Ainsi, on peut alors obtenir une mesure du périmètre d'un VPN pour une topologie réseau donnée. De même, on peut obtenir une mesure du périmètre d'un VPN pour différentes topologies réseau.

A) Le périmètre d'un VPN pour une topologie réseau

Si on calcule les probabilités que les $VPN_{2,}$ $VPN_{3,}$ VPN_4 soient pénétrés pour la topologie réseau B, nous obtenons alors une classification des périmètres comme l'illustre les formules suivantes :

Pour le VPN_2 (Pour le graphe B) :

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

Pour le VPN_3 (Pour le graphe B) :

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

Pour le VPN_4 (Pour le graphe B) :

$$P(Z=V) = P(Z=V|X=V,Y=V) * P(X=V,Y=V)$$
$$+$$
$$\dots$$
$$+ P(Z=V|X=F,Y=F) * P(X=F,Y=F)$$

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

$$P(Z=V) = 3*0.8*0.5*0.5+0.5*0.5*0.5$$
$$P(Z=V) = 0.725$$

On peut donc quantifier les périmètres des VPNs pour une topologie réseau donnée. Dans notre exemple, le VPN_2 et le VPN_3 (0.5) sont moins exposés que le VPN_4 (0.725) dans la topologie réseau B.

B) Le périmètre d'un VPN pour différentes topologies réseau

Maintenant, si on calcule la probabilité que le VPN_4 soit pénétré pour les topologies réseau A, B, C, on obtient alors une classification du périmètre d'un VPN pour différentes topologies réseau comme l'illustre les formules suivantes :

Pour le graphe A (VPN_4) :

P(Z=V) = P(Z=V|W=V,X=V,Y=V) * P(W=V,X=V,Y=V)

+

…

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

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

P(Z=V) = 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=V) = 0,7475

Pour le graphe B (VPN_4) :

P(Z=V) = P(Z=V|X=V,Y=V) * P(X=V,Y=V)

+

…

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

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

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

P(Z=V) = 0.725

Pour le graphe C (VPN_4) :

P(Z=V) = P(Z=V|W=V,X=V,Y=V) * P(W=V,X=V,Y=V)

+

…

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

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

$$P(Z=V) = 7*0.8*0.5*0.5*0.5+0.5*0.5*0.5*0.5$$
$$P(Z=V) = 0.7625$$

On peut donc quantifier le périmètre d'un VPN pour différentes topologies réseau. Dans notre exemple, le graphe B (0.725) offre un périmètre moins exposé aux menaces pour le VPN_4 que le graphe A (0,7475), et le graphe A (0,7475) offre un périmètre moins exposé aux menaces pour le VPN_4 que le graphe C (0.7625).

5.3.6. Les limitations

Beaucoup d'autres services réseau peuvent être déployés sur le réseau et doivent être sujets à des contrôles de sécurité. Par exemple, si on considère un service réseau mettant en œuvre un service de VPN basé sur le protocole IPSEC, de nombreux tests de sécurité doivent alors être écrits pour contrôler la politique de sécurité de ce service. De même, si des services d'accès de type DSL sont déployés, des tests spécifiques devront être conçus.

La limitation est donc que l'ensemble des tests n'est jamais exhaustif de part la complexité intrinsèque d'un réseau de télécommunications. Cet ensemble des tests devra donc être modifié en permanence afin de tenir compte des évolutions d'architecture et des services réseau.

Après avoir détaillé des exemples de détection des vulnérabilités, nous présentons comment définir une mesure de la sécurité pour le réseau.

5.4. Calcul des scénarii d'impact réseau

5.4.1. Approche générique

Cette étape consiste à calculer les scénarii d'événements possibles par le biais d'un arbre probabiliste basé sur les vulnérabilités préalablement détectées. En plus des vulnérabilités, deux autres entrées sont nécessaires au moteur de calcul des scénarii pour construire un tel arbre.

Le premier correspond aux règles de propagation des événements exploitant les vulnérabilités détectées. Le second correspond à la topologie du réseau afin de valider l'existence d'un chemin réseau dans le déclenchement d'un événement conditionné par un autre événement comme illustré à la figure suivante :

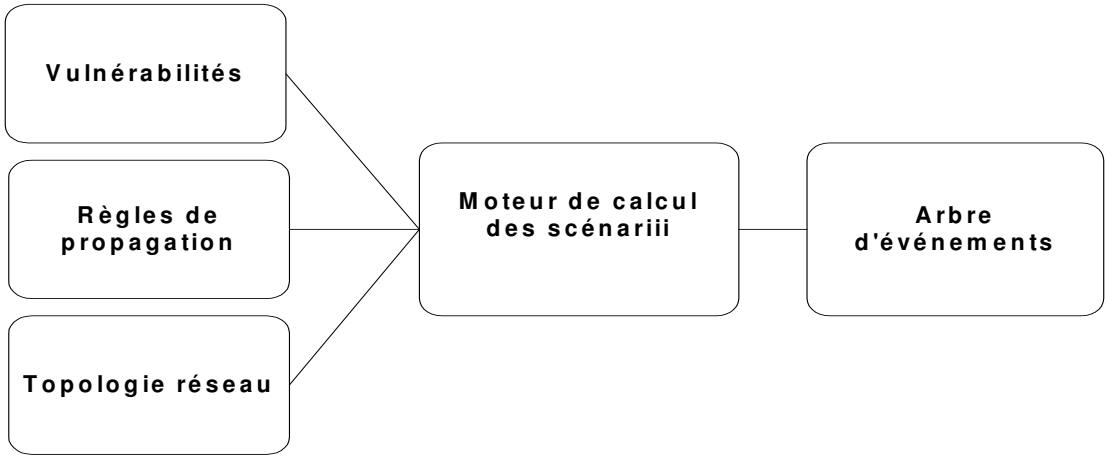


Figure 29: Calcul d'un arbre probabiliste à partir des vulnérabilités

L'algorithme associé au moteur de calcul des scénarii calcule un arbre probabiliste en respectant les fondements de la théorie des probabilités [Capinski et al. 2001].

5.4.2. Les restrictions d'un arbre probabiliste

Un arbre probabiliste suit des règles de construction qu'on peut résumer par les principes suivants :

- Un arbre a un seul point de départ. On dit que ce point est au niveau 0 de l'arbre.
- Tout point d'arrivée d'un arbre élémentaire est soit un point d'arrivée de l'arbre, soit un point de départ pour un autre arbre. Ce point est aussi appelé un nœud de l'arbre.
- Entre deux points d'un arbre, il y a un trajet orienté et un seul.
- Un chemin (ou trajet, ou séquence) maximal est un chemin allant de la racine à une extrémité de l'arbre, et un événement un ensemble de chemins maximaux.

- Chaque branche reliant deux nœuds successifs est affectée de la probabilité de passer du premier au second.

De plus, un arbre probabiliste suit aussi des règles relatives aux probabilités affectées qu'on peut résumer par les règles suivantes :

- La somme des probabilités affectées aux branches issues d'un même nœud est égale à 1.
- La probabilité affectée à chaque chemin (maximal ou non) est le produit des probabilités affectées à chacune des branches qui le composent.
- La probabilité d'un événement correspondant à plusieurs chemins maximaux est la somme des probabilités correspondant à chacun de ces chemins.

Aucune donnée ou statistique ne permet de déterminer la probabilité qu'un événement exploite une vulnérabilité donnée ou de déterminer une loi de probabilité quelconque. Nous considérerons alors que les probabilités sont de manière générale équiprobable pour chaque branche d'un nœud donné de l'arbre probabiliste. Toutefois, ceci ne constitue pas un inconvénient majeur, puisque l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées. De plus, il est à noter que d'autres distributions de probabilités peuvent être facilement mises en oeuvre dans ce modèle.

5.4.3. La modélisation simplifiée d'un nœud de l'arbre

Les vulnérabilités signalées par le moteur de vérification sont décrites par les champs suivants [Valois et Llorens 2002, Llorens et al. 2003] :

- L'équipement réseau dont la configuration contient cette vulnérabilité.
- Une description de la vulnérabilité.
- Le test de sécurité qui a détecté la vulnérabilité.
- L'impact réseau associé à la vulnérabilité. Il est à noter que l'impact réseau dépend directement dans notre modèle du test de sécurité.

Sachant que l'objectif est de quantifier les impacts réseau associés aux vulnérabilités détectées, l'étape suivante consiste à construire un arbre probabiliste basé sur ces vulnérabilités, à quantifier les probabilités de chaque branche et à calculer les probabilités de l'arbre associées aux impacts réseau [Bedford et al. 2001, Stamatelotos 2002].

Pour construire un tel arbre, nous considérerons une configuration où chaque nœud est composé :

- d'une branche indiquant qu'il n'y a pas d'impact réseau après l'exploitation de la vulnérabilité,
- d'une série de branches indiquant tous les événements exploitant d'autres vulnérabilités à partir du nœud en cours,
- et d'une branche indiquant un impact réseau.

On notera qu'on considère dans la figure suivante le nœud de l'arbre associé à l'exploitation de la vulnérabilité Vx :

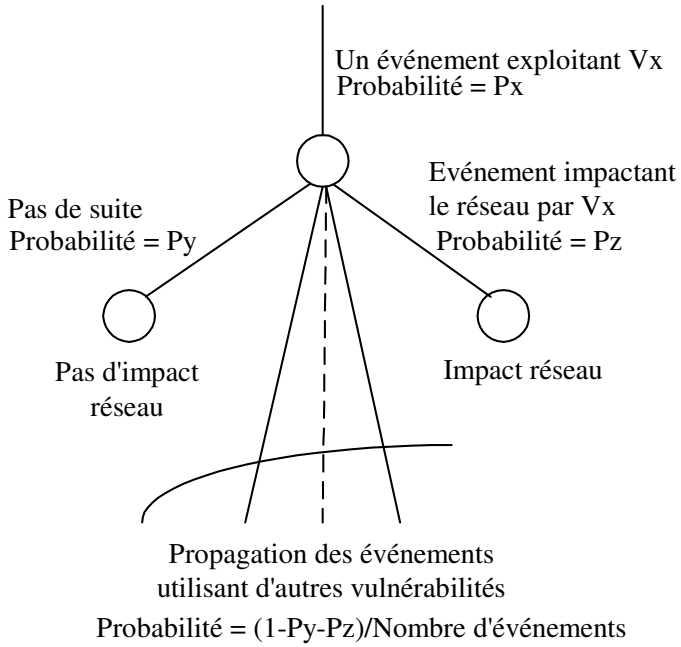


Figure 30: Modélisation d'un nœud d'un arbre d'événements

Nous considérons qu'il n'y a pas de cycles dans l'arbre, et qu'il n'y a pas de branche ayant un impact réseau et qui pointerait vers une série de branches indiquant tous les événements exploitant d'autres vulnérabilités. Cette restriction n'est pas un réel problème car cette série de branches est déjà présente dans les branches que nous avons définies précédemment. Ainsi, cette nouvelle série de branches peut être déduite et facilement implémentée.

Nous nous baserons cependant sur une configuration de nœud simplifiée que nous détaillons ci-après. Il doit être cependant noter que le modèle de configuration le plus complexe peut être déduit de ce modèle simplifié.

Le modèle de configuration simplifié consiste à dire que tout événement exploitant une vulnérabilité déclenche avec succès tous les autres événements exploitant les autres vulnérabilités. Il s'agit du pire des cas. Dans une telle configuration, si on considère l'arbre probabiliste suivant basé sur ces 3 vulnérabilités (V_1,V_2,V_3), ayant respectivement les impacts réseau suivants (NI/V1=fort, NI/V2=moyen, NI/V3=moyen), et où E/Vx est l'événement exploitant une vulnérabilité égale à Vx, nous obtenons alors l'arbre probabiliste suivant :

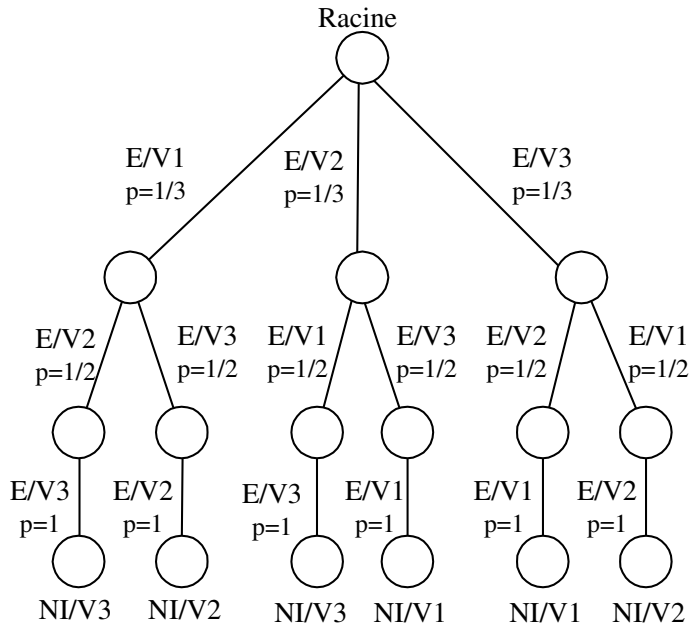


Figure 31: Exemple d'un arbre probabiliste

Si on considère une distribution de probabilités équiprobable pour chaque nœud de l'arbre, la probabilité d'avoir un impact réseau “fort” est alors égale à 2*(1/3*1/2*1) = 2/6, et la probabilité d'avoir un impact réseau “moyen” est égale à 4*(1/3*1/2*1) = 4/6. Ces calculs sont basés sur le principe des réseaux bayésiens [Jensen 2001].

5.4.4. Les limitations

Les limitations viennent du fait que nous considérons que les arbres probabilistes sans tenir compte de la modélisation des échecs, analyse d'incertitude, analyse de sensibilité, etc. Cette restriction est liée à la complexité intrinsèque d'un réseau de télécommunications. Cette généralisation pourra cependant faire l'objet de travaux de recherche ultérieurs.

5.5. Calcul des probabilités associées aux impacts réseau

5.5.1. La réduction combinatoire du nombre des sous-branches de l'arbre probabiliste

Pour de grands réseaux, tel que celui d'un opérateur de télécommunications, il est possible d'avoir à traiter plusieurs milliers de vulnérabilités à un instant donné. De plus, si tous les événements exploitant ces vulnérabilités peuvent déclencher d'autres événements exploitant d'autres vulnérabilités, nous faisons alors face à une explosion combinatoire des séquences possibles. Ce problème se réduit alors à l'énumération des permutations possibles sur les vulnérabilités [Sedgewick 1977].

Sachant que l'objectif est de calculer les probabilités associées aux impacts réseau, il est alors possible de réduire la combinatoire de la construction de l'arbre en raisonnant non plus sur les vulnérabilités, mais directement sur les tests de sécurité. De même, il est encore possible de réduire la combinatoire de la construction de l'arbre en raisonnant non plus sur les tests de sécurité, mais directement sur les impacts réseau. Cela signifie que tous les tests de sécurité ayant un même impact réseau peuvent être vus comme un seul test. Cette réduction est effectivement possible grâce à une simplification combinatoire basée sur la répétition de k objets parmi n objets lors de la construction des branches de l'arbre probabiliste. Cette réduction permet donc de déterminer des sous-branches identiques dans notre arbre probabiliste et ainsi de ne pas les construire. Bien que cette approche permette aussi de prendre en compte un nombre important de vulnérabilités détectées, on perd cependant de la granularité dans les règles de propagation en considérant des groupes de vulnérabilités plutôt que des vulnérabilités.

En considérant les impacts réseaux plutôt que les vulnérabilités pour construire notre arbre probabiliste, on peut alors poser les hypothèses et formules suivantes :

Hypothèses

- "N" est le nombre de vulnérabilités, $N \in \mathbb{N}^*$
- "nb.impact" est le nombre d'impacts réseau, $nb.impact \in \mathbb{N}^*$
- "$nb.vul(impact_i)$" est le nombre de vulnérabilités pour un impact réseau donné, $i \in [1, nb.impact]$
- La distribution de probabilités est équiprobable pour tous les nœuds de l'arbre
- Chaque événement exploitant une vulnérabilité déclenche tous les événements exploitant les autres vulnérabilités (ou vulnérabilités restantes).
- "nb.impact < N"

Formules

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul(impact_i)!}$$ est le nombre de feuilles de l'arbre (1)

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul\,(impact_i)!*N}$$ est le nombre de feuilles de l'arbre pour un impact réseau (2)

$$\prod_{i=1}^{nb.impact} nb.vul(impact_i)!$$ est le nombre de feuilles identiques pour une branche de l'arbre (3)

$$\prod_{i=0}^{N-1} \frac{1}{\sum_{j=1}^{nb.impact} nb.vul(impact_j) - i}$$ est la probabilité d'une branche de l'arbre (4)

Si on considère les vulnérabilités détectées, le nombre de feuilles de l'arbre est égal au nombre de permutations possibles sur les vulnérabilités. Si N est le nombre de vulnérabilités, alors le nombre de feuilles est égal à *N!*

En théorie combinatoire, si on a k_1 répétitions de l'objet$_1$, k_2 répétitions de l'objet$_2$ … k_m répétitions de l'objet$_m$ parmi n objets, alors le nombre de permutations est :

$$\frac{n!}{(k_1)!*(k_2)!*...*(k_m)!}$$

Si on considère les impacts réseau, notre arbre probabiliste est composé de répétitions d'impact réseau dans les branches de l'arbre. Si nous avons nb.vul(impact$_1$) répétitions de impact$_1$, nb.vul(impact$_2$) répétitions du impact$_2$… nb.vul(impact$_m$) répétitions de impact$_m$ parmi N objets, alors le nombre de permutations devient :

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul\,(impact_i)!} \textit{(formule 1)}$$

Si on considère les impacts réseau, nous savons que le nombre de feuilles identiques pour une branche de l'arbre est égal au nombre de branches identiques, ou permutations possibles, pour les impacts réseau définis comme l'illustre la formule suivante :

$$nb.vul(impact_1)*...*nb.vul(impact_{nb.impact})*(nb.vul(impact_1)-1)*...*(nb.vul(impact_{nb.impact})-1)...*1$$

$$= \prod_{j=1}^{nb.impact} \prod_{i=0}^{nb.vul\,(impact_j)-1} (nb.vul\,(impact_j) - i) = \prod_{j=1}^{nb.impact} nb.vul\,(impact_j)!\,\textit{(formule 3)}$$

Si on multiplie les formules 1 & 3, on retrouve le nombre total de permutations si on avait à considérer les vulnérabilités :

$$\prod_{j=1}^{nb.impact} nb.vul\,(impact_j)!*\frac{N!}{\prod_{j=1}^{nb.impact} nb.vul\,(impact_j)!} = N!$$

Enfin, si on considère les vulnérabilités, la probabilité associée à une branche de l'arbre est égale à :

$$\prod_{i=0}^{N-1} \frac{1}{N-i} = \frac{1}{N} * \frac{1}{N-1} * \ldots * 1$$

$$\prod_{i=0}^{N-1} \frac{1}{N-i} = \frac{1}{\sum_{j=1}^{nb.impact} nb.vul(impact_j)} * \frac{1}{\sum_{j=1}^{nb.impact} nb.vul(impact_j) - 1} * \ldots * 1$$

$$\prod_{i=0}^{N-1} \frac{1}{N-i} = \prod_{i=0}^{N-1} \frac{1}{\sum_{j=1}^{nb.impact} nb.vul(impact_j) - i} \quad \textit{(formule 4)}$$

Prenons l'exemple de vérification des configurations donné ci-après. Calculons alors les impacts réseau en considérant l'arbre probabiliste associé aux vulnérabilités et l'arbre probabiliste associé aux impacts réseau :

Numéro de Vulnérabilité	**Numéro de test**	**Impact réseau**
1	1	1
2	1	1
3	3	1
4	4	2

Tableau IV : Résultat d'une vérification des configurations

Nous pouvons alors illustrer l'arbre probabiliste basé sur ces vulnérabilités par la figure suivante, où E/Vx est un événement exploitant une vulnérabilité égale à x :

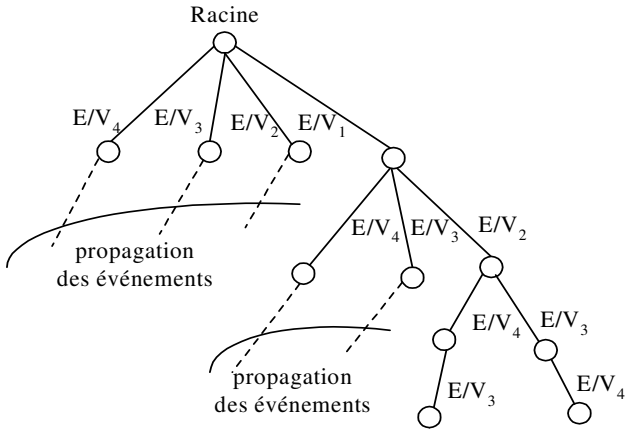


Figure 10 : Un arbre probabiliste basé sur les vulnérabilités

L'arbre a alors les propriétés suivantes si on considère les vulnérabilités :

- Le nombre de feuilles est égal à 4! = 4*3*2 = 24. Le nombre de feuilles possibles pour un test est égal à 4!/4 = 6.
- La probabilité d'une branche est égale à 1/4*1/3*1/2 = 1/24.
- La probabilité qu'un impact réseau soit égal à “1” est 3*6*1/24=3/4, La probabilité qu'un impact réseau soit égal à “2” est 1*6*1/24=1/4.

Si on considère les impacts réseau, nous pouvons représenter l'arbre probabiliste basé sur ces impacts réseau par la figure suivante, où E/Nx est un événement exploitant une vulnérabilité ayant un impact réseau égal à x. On peut ainsi le comparer à l'arbre probabiliste basé sur les vulnérabilités :

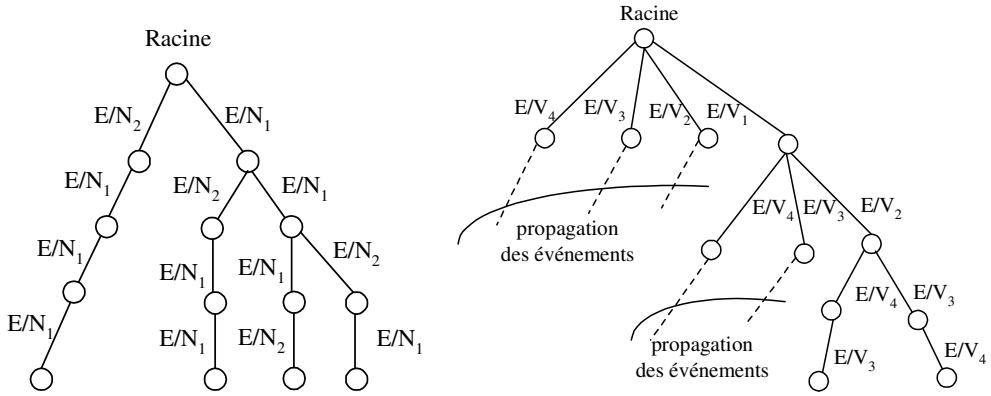


Figure 11 : Comparaison des arbres probabilistes

L'arbre a alors les propriétés suivantes si on considère les impacts réseau :

- Sachant que nb.impact = 2, nb.vul($Impact_1$) = 3, nb.vul($Impact_2$) = 1, alors le nombre de feuilles est égal à 4!/(3!*1!) = 4. Le nombre de feuilles possibles pour un impact réseau est égal à 4/4 = 1.
- Le nombre de feuilles identiques pour une branche est égal à 3*2*1=6.
- La probabilité d'une branche d'un arbre est égale à 1/4*1/3*1/2 = 1/24.
- La probabilité qu'un impact réseau soit égal à “1” est 3*1*6*1/24=3/4, La probabilité qu'un impact réseau soit égal à “2” est 1*1*6*1/24=1/4.

En considérant les impacts réseau plutôt que les vulnérabilités pour construire notre arbre probabiliste, nous avons réduit le nombre de branches explorées en réduisant le nombre de permutations à explorer.

Dans notre exemple, on passe de 24 à 4 permutations à explorer. Cette réduction améliore donc la complexité en temps du calcul des probabilités associées aux impacts réseau.

L'algorithme "bayes" que nous présentons maintenant est récursif et exploite cette réduction combinatoire afin de diminuer la complexité en temps théorique initiale.

L'algorithme "bayes" utilise les variables globales suivantes :

- nb_impacts : il s'agit du nombre d'impacts réseau. Cette variable a été renseignée lors de la vérification des configurations des équipements réseau.
- nb_vuln[NB_IMPACTS] : il s'agit de la table contenant le nombre de vulnérabilités pour chaque impact réseau. Cette table a été renseignée lors de la vérification des configurations des équipements réseau.
- next_impact[NB_IMPACTS][NB_IMPACTS] : il s'agit de la matrice définissant les règles de propagation entre les impacts réseau. Cette

matrice a été renseignée lors de la vérification des configurations des équipements réseau.

- probability_network_impact[NB_IMPACTS] : il s'agit de la table contenant les valeurs initiales des probabilités pour chaque impact réseau. Cette table a été renseignée lors de la vérification des configurations des équipements réseau.

L'algorithme "bayes" utilise les variables locales suivantes :

- no_impact : il s'agit de la variable décrivant l'impact réseau courant appelé par la fonction "bayes".
- probability : il s'agit de la variable décrivant la probabilité du nœud courant dans l'arbre probabiliste associé à l'appel de la fonction "bayes". Cette variable est utilisée pour mettre à jour la table "probability_network_impact [NB_IMPACTS]".
- mult : il s'agit de la variable décrivant le nombre de sous-branches identiques en utilisant la table "nb_vuln[NB_IMPACTS]". Cette variable est utilisée pour mettre à jour la table "probability_network_impact[NB_IMPACTS]".
- card : il s'agit du nombre de branches de l'arbre.

L'algorithme "bayes" peut alors s'écrire de la manière suivante en pseudo-code (tous les paramètres sont passés par valeur) :

```
 procedure bayes(no_impact, probability, mult) {

/* Mise à jour du nombre de sous-branches identiques */
mult :=  mult *  nb_vuln[no_impact];

/* Décrémente le nombre de vulnérabilités associées à no_impact */
nb_vuln[no_impact] := nb_vuln[no_impact] - 1;

/* Détermine si il existe d'autres branches pour le nœud en cours */
card := 0;
for i := 1 to  nb_impact
{
    if next_impact[no_impact][i] is defined then card := card + nb_vuln[i];
}

if (card = 0)
{
    /*  Nous sommes au niveau d'une feuille, mise à jour des probabilités associées */
    probability_network_impact[no_impact] :=
                probability_network_impact[no_impact]+ probability * mult;
}
```

```
else
{
        /*  Nous avons d'autres branches pour le nœud en cours */
        /* Calcule une distribution de probabilités équiprobable des branches */
        probability := probability * 1.0 / card;

       /* Appel de tous les impacts réseau */
       for i: = 1 to  nb_impacts
       {
              /* Appel récursif après la vérification des règles de propagation */
              /* et du nombre de vulnérabilités restantes */
              if (next_impact[no_impact][i] is defined  and nb_vuln[i] ≠ 0)
              {
                    bayes(i, probability, mult);
              }
      }
}

/* Incrémente le nombre de vulnérabilités pour l'impact réseau en cours */
/* afin qu'il puisse participer aux autres appels récursifs des autres branches*/
nb_vuln[no_impact] := nb_vuln[no_impact] +1;

return;
}
```

5.5.2. La complexité en temps de l'algorithme de parcours de l'arbre

Le meilleur cas est lorsqu'un événement exploitant une vulnérabilité avec un impact réseau donné ne peut générer d'autres événements exploitant d'autres vulnérabilités avec des impacts réseau différents. Dans un tel contexte, l'algorithme a une complexité en temps linéaire avec le nombre de vulnérabilités.

Le plus mauvais cas est lorsqu'il y a une parfaite répartition des vulnérabilités sur les impacts réseau, et lorsqu'un événement exploitant une vulnérabilité avec un impact réseau donné génère d'autres événements exploitant d'autres vulnérabilités quel que soit l'impact réseau. Dans un tel contexte, une limite supérieure de la complexité en temps est fournie par le nombre de feuilles de l'arbre probabiliste multiplié par la profondeur maximale de l'arbre :

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul(impact_i)!} * N \quad < \quad \frac{N!}{((\frac{N}{nb.impact})!\ )^{nb.impact}} * N$$

Si on utilise la formule de Stirling, nous obtenons :

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul(impact_i)!} * N < \frac{\sqrt{2\pi N} * (\frac{N}{e})^N * N}{(\sqrt{\frac{2\pi N}{nb.impact}} * (\frac{N}{nb.impact * e})^{\frac{N}{nb.impact}})^{nb.impact}}$$

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul(impact_i)!} * N < \frac{\sqrt{2\pi N} * (nb.impact)^N * N}{\sqrt{\frac{2\pi N}{nb.impact}}^{nb.impact}}$$

$$\frac{N!}{\prod_{i=1}^{nb.impact} nb.vul(impact_i)!} * N < \sqrt{2\pi N} * (nb.impact)^N * N$$

L'algorithme a une complexité en temps de l'ordre de $N^{3/2}*3^N$ au pire des cas. Bien que cette réduction combinatoire permette de limiter l'explosion de la complexité en temps initiale, on peut réduire cette complexité en temps en précisant par exemple qu'un impact réseau "faible" ne peut pas générer des événements avec des impacts réseau "moyen" et "fort". On pourra aussi modéliser une granularité plus fine d'impacts réseau avec pour objectif d'ajuster le plus précisément possible les propagations entre les impacts réseau. Enfin, la limite du modèle correspond au pire des cas à la limite d'énumérer les permutations d'un ensemble d'éléments [Sedgewick 1977].

5.5.3. Les limitations

Le problème de la construction de notre arbre probabiliste est lié au pire des cas au problème d'énumération des permutations d'un ensemble de N éléments. Le tableau ci-dessous nous donne les limitations de complexité en temps lié à l'énumération des permutations de N éléments [Sedgewick 1977] :

N	Nombre de permutations	Million/Sec	Billion/Sec	Trillion/Sec
10	3628800	Insignifiant	Insignifiant	Insignifiant
11	39916800	Secondes	Insignifiant	Insignifiant
12	479001600	Minutes	Insignifiant	Insignifiant
13	6227020800	Heures	Secondes	Insignifiant
14	871178291200	Jour	Minute	Insignifiant
15	1307674368000	Semaines	Minutes	Secondes
16	20922789888000	Mois	Heures	Minutes
17	355687428096000	Années	Jours	Heures
18	6402373705728000	Impossible	Mois	Jours
19	121645100408832000	Impossible	Impossible	Mois
20	2432902008176640000	Impossible	Impossible	Impossible

Table 5-7 : Temps de calcul de l'énumération des permutations de N éléments

Cette limitation de complexité en temps s'applique à notre contexte si on désire définir un nombre d'états finals supérieur à 20.

De plus, la complexité en espace associée à la construction de notre arbre d'événements dépend au pire des cas du nombre de permutations possibles multiplié par la taille de stockage d'un élément. Cette contrainte est beaucoup moins limitative que la complexité en temps et peut être optimisée [Bryant 1986].

5.6. Mise en place d'un tableau de bord de la sécurité réseau

5.6.1. Approche générique

Une fois que l'on a calculé les probabilités des impacts réseau, il suffit alors de quantifier les conséquences associées aux impacts réseau afin de calculer le risque associé à la non application de la politique de sécurité.

Ce risque est calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau comme nous le détaillerons par la suite. La figure suivante schématise le processus de calcul :

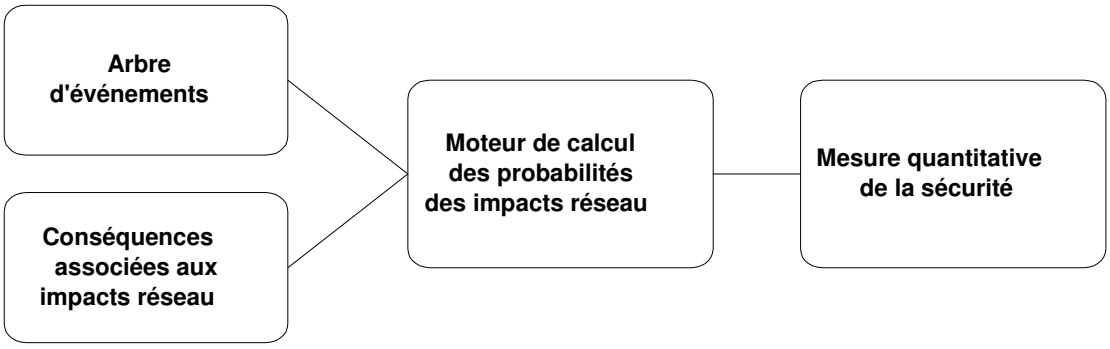


Figure 32: Calcul d'un risque

Avant de quantifier le risque, il est cependant possible de définir un premier tableau de bord de la sécurité avec les vulnérabilités détectées comme le détaille le paragraphe suivant. Nous détaillons ensuite comment améliorer ce tableau de bord en ajoutant une mesure quantitative de la sécurité.

5.6.2. Les indicateurs de base

L'ensemble des tests de sécurité défini permet de signaler des vulnérabilités, mais aussi de construire un premier tableau de bord de la sécurité réseau constitué des courbes ou indicateurs de sécurité suivants :

- L'évolution dans le temps du nombre de vulnérabilités de sécurité détectées par les contrôles successifs permet de donner une mesure de l'application de la politique de sécurité réseau.
- En s'appuyant sur les vulnérabilités de sécurité détectées lors des contrôles, on en déduit la liste des équipements ayant des vulnérabilités de sécurité ou étant impactés. Si l'on divise le nombre d'équipements ayant des vulnérabilités de sécurité par le nombre total d'équipements, on obtient le pourcentage d'équipements impactés. Cette courbe permet donc de savoir si les faiblesses de sécurité impactent le réseau dans son ensemble ou une partie du réseau seulement, avec un indicateur entre 0 et 100 p. 100.

- En divisant le nombre de vulnérabilités de sécurité détectées par le nombre total d'équipements, on obtient le nombre moyen de vulnérabilités de sécurité par équipement. En revanche, si l'on divise le nombre de vulnérabilités de sécurité par le nombre d'équipements impactés, on obtient le nombre moyen effectif de vulnérabilités de sécurité par équipement. L'écart entre les deux courbes permet de mesurer l'impact des faiblesses de sécurité sur l'ensemble des équipements. Un grand écart entre les deux courbes signifie à un instant donnée que les faiblesses de sécurité s'appliquent à un nombre limité d'équipements. À l'inverse, une égalité entre les courbes signifie que les faiblesses de sécurité s'appliquent au nombre total d'équipements.
- L'évolution dans le temps du nombre total de vulnérabilités de sécurité par niveau d'impact donne une indication des conséquences possibles sur le réseau. Pour y parvenir, nous devons définir pour chaque vulnérabilité de sécurité détectée un niveau d'impact réseau (faible, moyen, fort), et les cumuler comme l'illustre la figure suivante :

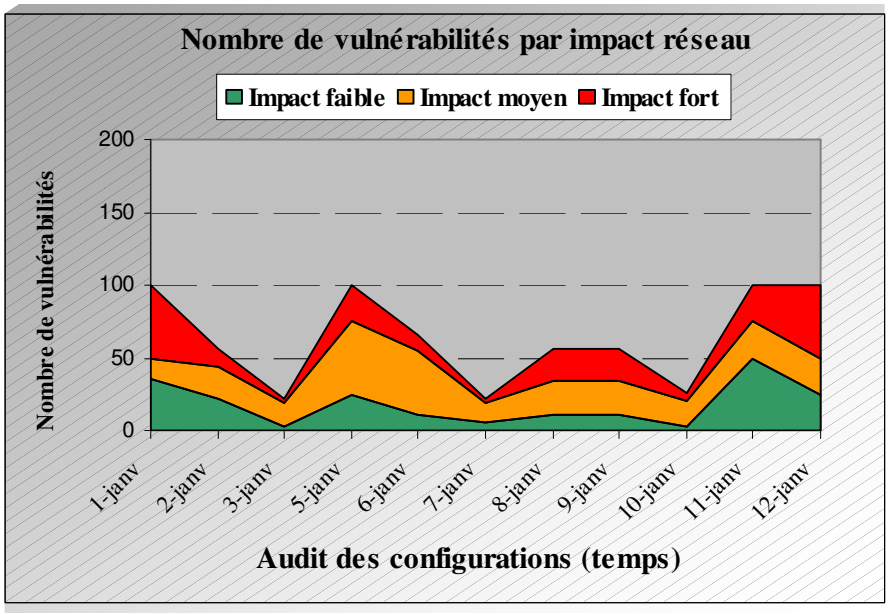


Figure 33 : Tableau de bord, nombre moyen de vulnérabilités par impact réseau

5.6.3. La mesure du risque

Une fois que l'on a calculé les probabilités des impacts réseau, il suffit alors de quantifier les conséquences associées aux impacts réseau afin de calculer le risque associé à la non application de la politique de sécurité. Ce risque est alors calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau [Bedford et al. 2001, Stamatelotos 2002].

5.6.3.1. Les conséquences associées aux impacts réseau

Nous prendrons des valeurs de conséquences prédéterminées comme l'illustre le tableau ci-après. Toutefois, ceci ne constitue pas un inconvénient majeur

dans notre expérience, puisque l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées. Il est à noter que d'autres distributions de conséquences peuvent être facilement mises en oeuvre dans ce modèle.

Enfin, nous considérerons qu'une valeur de risque comprise entre]50,100] définit un risque fort nécessitant une action immédiate. Une valeur de risque comprise entre]10,50] définit un risque moyen nécessitant la mise en place d'actions correctives. Une valeur de risque comprise entre]1,10] définit un risque faible nécessitant soit la mise en place d'actions correctives soit l'acceptation du risque.

Le tableau suivant illustre les différentes valeurs de conséquence et de probabilité associées :

Conséquence **Probabilité**	**Valeur = 10** **impact réseau faible**	**Valeur = 50** **Impact réseau moyen**	**Valeur = 100** **impact réseau fort**
Forte **= 1.0**	*Risque faible* 10*1,0=10	*Risque moyen* 50*1,0=50	*Risque fort* 100*1,0=100
Moyenne **= 0,5**	*Risque faible* 10*0,5=5	*Risque moyen* 50*0,5=25	*Risque moyen* 100*0,5=50
Faible **= 0,1**	*Risque faible* 10*0,1=1	*Risque faible* 50*0,1=5	*Risque faible* 100*0,1=10

Tableau V : Les conséquences des impacts réseau

5.6.3.2. Le calcul du risque

Il est alors possible de construire un tableau de bord amélioré de la sécurité réseau constitué des deux courbes suivantes.

La première courbe représente l'évolution dans le temps de la distribution des probabilités des impacts réseau pour chaque vérification des configurations :

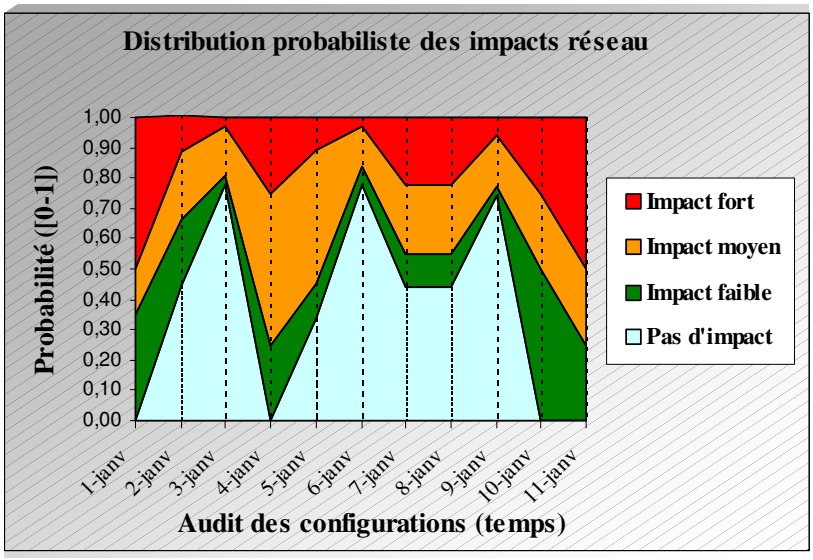


Figure 13 : Distribution des probabilités des impacts réseau

La seconde courbe représente l'évolution dans le temps du risque pour chaque vérification des configurations :

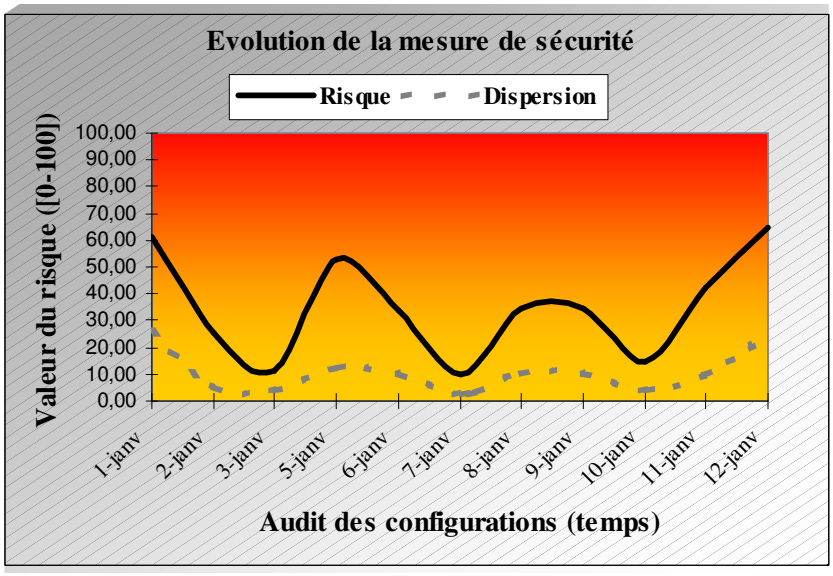


Figure 14 : Evolution de la mesure de sécurité

5.6.4. Les limitations

Les limitations viennent du fait que nous considérons pas des analyses d'incertitude, analyses de sensibilité, etc. Cette restriction est liée à la complexité intrinsèque d'un réseau de télécommunications. Toutefois, l'objectif était de valider le comportement et la pertinence des mesures de sécurité réalisées. Cette généralisation pourra cependant faire l'objet de travaux de recherche ultérieurs.

5.7. Prédiction des mesures de sécurité dans le temps

5.7.1. Approche générique

Les différents contrôles menés sur les configurations des équipements réseau représentent un ensemble d'information qui peut permettre de répondre aux questions suivantes :

- Comment la sécurité du réseau évolue-t-elle dans le temps ?
- Peut-on définir un meilleur ordre de priorité des corrections associées aux vulnérabilités de sécurité ?

Il est en effet important de savoir si l'évolution dans le temps des vulnérabilités de sécurité peut impacter des parties sensibles du réseau. En effet, certains domaines du réseau sont plus critiques que d'autres, notamment le domaine du routage réseau.

Cette projection dans le temps peut aussi permettre de définir un meilleur ordre de priorité des corrections associées aux vulnérabilités de sécurité, plutôt qu'un ordre séquentiel sur les vulnérabilités les plus critiques.

Les chaînes de Markov offrent un cadre mathématique adapté afin d'apporter une première réponse à ces questions. Nous détaillons dans un premier temps la modélisation d'une chaîne de Markov basée sur les vulnérabilités détectées, nous décrivons ensuite les indicateurs de stabilité et de prédiction retenus pour contrôler la qualité informative. Enfin, nous suggérons deux méthodes pour déterminer les priorités afin de corriger les vulnérabilités les plus critiques.

5.7.2. Une modélisation Markovienne d'ordre 1 de la sécurité du réseau

Une chaîne de Markov met en relation des observations successives d'une même variable aléatoire discrète. Nous considérerons ici une variable catégorielle Xt dont nous connaissons la valeur pour les différentes périodes de temps t = 0,1,2,3,….

Ces périodes correspondent alors aux différents contrôles de sécurité, et notre variable comporte un nombre fini d'états m. Une chaîne de Markov d'ordre 1 dit que l'ensemble du passé de l'époque t est résumé par l'époque t-1. La probabilité de la variable X_t pour un état est donné par le formule suivante [Berchtold 1998] :

$P(X_t=j0 \mid X_{t-1} = i1, X_{t-2} = i2, \ldots) = P(X_t=j0 \mid X_{t-1} = i1) = q_{i1,j0}(t)$

Où j0, i1, i2, … appartiennent à l'ensemble des valeurs de X_t

La matrice $Q_1(t-1,t)$ de dimension (m x m), appelée matrice de transition de l'époque t-1 à l'époque t, contient donc l'ensemble des probabilités possibles comme l'illustre la matrice suivante :

$$Q_1 = \begin{pmatrix} Q_{1,1} & . & . & . & Q_{1,m} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ Q_{m,1} & . & . & . & Q_{m,m} \end{pmatrix} = Q_1(t-1,t), \forall t$$

où chaque ligne de la matrice est une loi de probabilité.

Figure 34: Matrice de transition Markovienne

Dans le cadre de nos contrôles de sécurité, nous pouvons considérer 3 états par équipement réseau. Ces états sont ($R_{i,faible}$, $R_{i,moyen}$, $R_{i,fort}$) où i est le numéro de l'équipement, $R_{i,faible}$ signifie le nombre d'impact réseau faible détecté, $R_{i,moyen}$ signifie le nombre d'impact réseau moyen détecté, $R_{i,fort}$ signifie le nombre d'impact réseau fort détecté.

L'ensemble des équipements réseau (N) détermine donc l'ensemble des états possibles de la chaîne de Markov = 3* N. La dimension de la matrice de transition est égale à (3* N x 3* N) comme illustré par la matrice suivante :

$$Q_1 = \begin{array}{c} \\ R_{1,faible} \\ \\ \\ R_{N,fort} \end{array} \begin{pmatrix} R_{1,faible} & . & . & . & R_{N,fort} \\ . & & & & . \\ . & & & & . \\ . & & & & . \end{pmatrix} = Q_1(t-1,t), \forall t$$

*où N est nombre d'équipements réseau, et (3*N x 3*N) la dimension de la matrice de transition.*

Figure 35: Matrice de transition Markovienne associée au réseau

Sachant que notre cœur de réseau multi-services est composé de l'ordre de 500 équipements réseau, nous obtenons alors une matrice de dimension (1500 x 1500). Ne connaissant pas les probabilités associées à la matrice de transition, nous allons construire cette matrice à partir de la table de contingence associée à un contrôle de sécurité.

Par ailleurs, la propagation des événements est prépondérante dans notre approche, il convient donc de considérer la propagation des impacts réseau dans la construction de la table de contingence. La construction de la table de contingence d'un contrôle de sécurité consiste à renseigner, pour chaque état d'un équipement (i.e. pour chaque ligne), la colonne correspondante si ce même état d'impact a été détecté lors de ce contrôle.

Le passage de la table de contingence à la matrice de transition consiste à diviser chaque valeur d'une ligne par la somme des valeurs de chaque ligne afin d'obtenir une loi de probabilité par ligne.

Comme il est défini dans les chaînes de Markov, la probabilité des états futurs (t+2, t+3, etc.) est déterminer par une élévation à la puissance de la matrice de transition comme l'illustre la formule suivante [Berchtold 1998] :

$P(X_{t+2}=j \mid X_t=i) = P(X_{t+2}=j, X_{t+1}=k \mid X_t=i)$

$P(X_{t+2}=j \mid X_t=i) = P(X_{t+2}=j, X_{t+1}=k, X_t=i)/P(X_t=i)$

$P(X_{t+2}=j \mid X_t=i)=P(X_{t+2}=j|X_{t+1}=k)*P(X_{t+1}=k|X_t=i)* P(X_t=i)/P(X_t=i)$

$P(X_{t+2}=j \mid X_t=i) = P(X_{t+1}=k \mid X_t=i) * P(X_{t+2}=j \mid X_{t+1}=k)$

$P(X_{t+2}=j \mid X_t=i) =$ *(ligne i de Q) * (ligne j de Q)*

$P(X_{t+2}=j \mid X_t=i) = q^{(2)}_{i,j}$ *(équation de Chapman-Kolmogorov)*

Dans le contexte d'un réseau, l'élévation à la puissance de la matrice de transition va nous permettre de déterminer si des parties critiques du réseau peuvent être fortement impactés dans le temps comme illustré à la figure suivante :

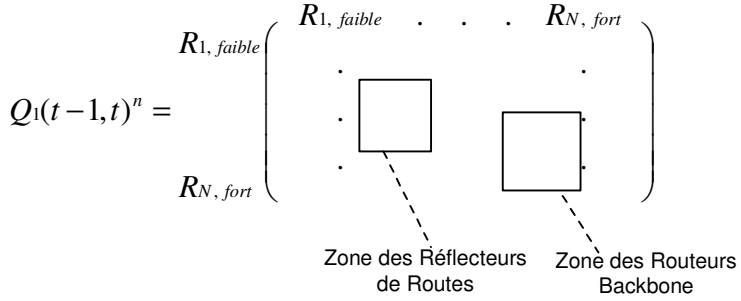


Figure 36: Prédiction des mesures de sécurité

Avant toute simulation, il est nécessaire de calculer quelques indicateurs sur la matrice de transition définie afin de s'assurer de la quantité de l'information contenue dans la matrice ainsi que de son pouvoir prédictif. Nous avons retenu les deux indicateurs suivants :

La mesure de stabilité maximale :

$D_m(Q1(t-1,t)) = c * \max_i ((1-\min_j(s_{i,j}))/(n_i+1))$, pour une matrice de transition de dimension (r x c) en ne tenant pas compte des zéros structurels.

Le pouvoir prédictif basé sur l'entropie de Shannon :

$$pp_h(Q1(t-1,t)) = 1 + \frac{\sum_i w_{i*} \sum_j S^2_{i,j} * \log_2 S_{i,j}}{\log_2 * c}$$, pour une matrice de transition de dimension (r x c) en ne tenant pas compte des zéros structurels.

5.7.3. Le calcul des prédictions

Basé sur les indicateurs choisis précédemment, voici l'évolution dans le temps des indicateurs de stabilité et de prédiction pour chaque vérification des configurations :

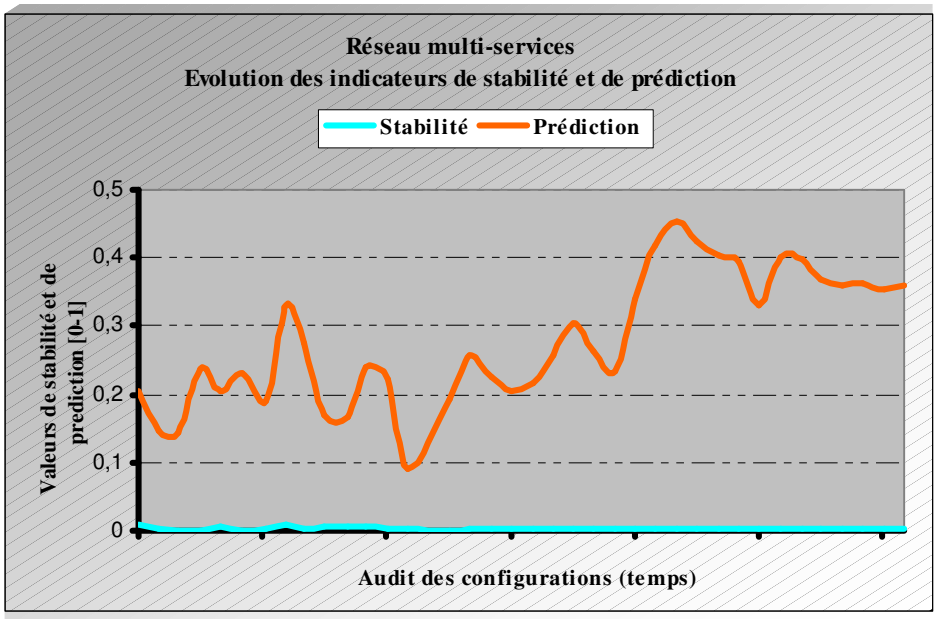


Figure 37: Evolution dans le temps des indicateurs de stabilité et de prédiction

Les courbes illustrent que la mesure de stabilité est petite, ce qui suggère que la distribution de probabilités est stable. Elles illustrent aussi que la mesure de prédiction est importante, ce qui suggère que l'incertitude est faible.

5.7.4. Le calcul des puissances successives de la matrice de transition

Sachant que notre réseau multi-services est composé de l'ordre de 500 équipements pour le cœur de réseau, nous obtenons une matrice de transition de dimension (1500 x 1500).

Pour le calcul des puissances successives de la matrice de transition, nous avons utilisé la librairie BLAS (Basic Linear Algebra Subprograms) codée pour les processeurs INTEL afin d'avoir un facteur accélérateur de l'ordre de 250 face à un codage classique [Cormen et al. 2002].

5.7.5. Le calcul des priorités par un parcours de la matrice

Le calcul des priorités de la matrice élevée à la puissance consiste à déterminer les routeurs ou les groupes de routeurs qui sont le plus impactés.

Pour y parvenir, une approche consiste à parcourir les lignes de la matrice et à calculer pour chaque routeur colonne les critères suivants :

- Critère : Le nombre de routeurs qui ont déclenché un impact (probabilité supérieure à 0) sur ce routeur.

- Critère : La somme des probabilités qui ont déclenché un impact sur ce routeur.

Le pseudo-code suivant implémente cet algorithme et calcule ces deux indicateurs en parcourant la matrice Q^n :

```
/* Parcours des colonnes de la matrice */
for(j=0;j<dimension(matrice);j++) {
      Somme_Colonne=0;
      Nombre_Colonne=0;

       /* Parcours des lignes  de la matrice */
      for(i=0;i<dimension(matrice);i++) {
            Somme_Colonne+=matrice[i][j];
            if (matrice[i][j]>0) Nombre_Colonne++;
      }

      /* Affichage des valeurs trouvées pour chaque colonne */
      printf("%f %f %s\n,Nombre_Colonne,Somme_Colonne, router[i]);

}
```

Un autre critère consiste aussi à considérer les routeurs qui sont plus critiques que d'autres. Par exemple, les routeurs réflecteurs de routes, qui centralisent les tables de routage du réseau multi-services, sont plus critiques pour la stabilité du réseau que les routeurs de périphérie PE.

Le choix des priorités consiste alors à déterminer les plus grandes valeurs relatives aux critères énoncés [Saaty 1988].

5.7.6. Le calcul des priorités par un parcours du graphe associé à la matrice

Une autre manière de déterminer les groupes de routeurs mutuellement impactés consiste à calculer les composants fortement connexes du graphe associé à la matrice de transition.

Si chaque routeur correspond à un sommet et un arc est défini entre le sommet i et j si A[i,j]>0, alors le calcul des composantes fortement connexes du graphe permet de déterminer les groupes de routeurs propageant des impacts comme l'illustre la figure suivante :

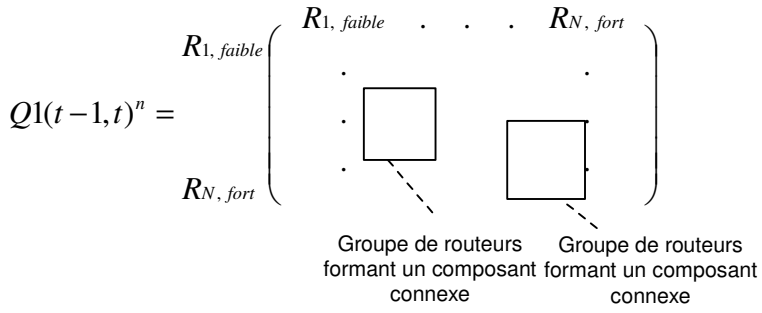


Figure 38: Détermination des groupes de routeurs impactés

5.7.7. Les limitations

Le choix d'une chaîne d'ordre 1 s'impose compte tenu de la dimension requise par une chaîne d'ordre 2.

Rappelons que dans le cas d'une chaîne d'ordre 1, chaque état de la chaîne est identifié comme étant une des valeurs de la variable aléatoire discrète X_t. On considère qu'il y a m valeurs possibles.

Dans le cas d'une chaîne d'ordre V, chaque état est composé d'une suite d'observations de la variable aléatoire discrète aux époques t-v à t-1, ce qui donne un total de m^v états différents.

Dans notre modélisation, sachant que le nombre de valeurs possibles est égal à 500*3=1500, nous aurions une matrice de dimension 1500^2 = 2.250.000 pour une chaîne d'ordre 2.

Le calcul des puissances successives nécessiterait alors des puissances de calcul importantes pour une détermination de priorités qui peut être faite avec une chaîne d'ordre 1.

Enfin, il doit être précisé que cette modélisation Markovienne est une première approche qui a donné des résultats intéressants. Il est cependant nécessaire de la généraliser dans des travaux de recherche ultérieurs pour confirmer sa validité.

Conclusion générale

L'objectif de cette thèse était de tenter de répondre à ces trois questions :

- Quel est le jeu minimal de règles de sécurité à mettre en place pour protéger un réseau ?
- Quel est le risque pris par le réseau si les vulnérabilités de sécurité détectées ne sont pas corrigées ?
- Peut-on prédire dans le temps si des parties critiques du réseau seront impactées par des vulnérabilités non corrigées ?

Nous avons fourni une première réponse à ces trois questions par la mise en œuvre du prototype de mesure de la sécurité du réseau multi-services de France Télécom/Equant.

Le prototype de mesure de la sécurité permet de contrôler près de 40.000 équipements réseau représentant près de 20 millions de lignes de configurations à partir de règles de sécurité génériques.

Près de ½ million de vérifications sont réalisées régulièrement sur les parties les plus critiques du cœur de réseau, et un tableau de bord de la sécurité réseau est mis à jour. La réduction combinatoire des sous-branches de l'arbre probabiliste, couplé à des règles de propagation réduites, permettent de considérer plusieurs milliers de vulnérabilités. Nous avons effectivement constaté avec des réseaux représentant plusieurs millions de lignes de configurations, que le nombre de vulnérabilités détectées n'était jamais négligeable. Le calcul d'un risque ainsi que l'établissement de priorités des vulnérabilités à corriger ont pu être calculés en tenant compte des limitations imposées à notre modèle.

Les mesures de la sécurité que nous avons pu mener ont montré que notre modèle donnait une mesure intéressante de la sécurité si on considère les incidents de sécurité survenus. Cependant, aucune courbe réelle ou information ne peuvent être données pour des raisons bien compréhensibles de confidentialité. De plus, de nombreux champs de recherche demeurent (se référer aux paragraphes limitations) et pourront faire l'objet de travaux de recherche ultérieurs.

L'étude d'un tableau de bord de la sécurité d'un système d'information contenant à la fois le réseau, les systèmes ainsi que tous les autres éléments constituant ce système reste un vaste champ de recherche couvrant à la fois les domaines de graphes d'attaques, de la détection d'intrusion, etc. Concernant la partie réseau, d'autres métriques doivent être développées afin de prendre en compte les évolutions du réseau mais aussi les services déployés.

Rappelons que l'un des objectifs majeurs d'un tableau de bord de sécurité est de refléter l'application de la politique de sécurité réseau. En aucun cas, il ne

faut assimiler ces indicateurs à un niveau de sécurité du réseau, mais plutôt à un état d'application de la politique de sécurité. De plus, ce tableau de bord permet de déterminer un niveau de risque si cette politique de sécurité n'est pas appliquée. Enfin, quel que soit l'état d'avancement du tableau de bord de la sécurité, les personnes concernées doivent être impliquées dans la construction du tableau de bord, et des objectifs doivent être définis afin de corriger les faiblesses de sécurité détectées.

Annexe A : Rappels sur la théorie de la complexité

La théorie de la complexité algorithmique s'attache à savoir entre différents algorithmes réalisant une même tâche, quel est le plus rapide et dans quelles conditions.

Dans les années 1960 et au début des années 1970, alors qu'on en était à découvrir des algorithmes fondamentaux (quicksort, Boyer-Moore...), on ne mesurait pas leur efficacité. On se contentait de dire: " cet algorithme (de tri) se déroule en 6 secondes avec un tableau de 50 000 entiers choisis au hasard en entrée, sur un ordinateur IBM 360/91. Une telle démarche rendait difficile la comparaison des algorithmes entre eux. Une approche indépendante des facteurs matériels était nécessaire pour évaluer l'efficacité des algorithmes. Donald Knuth fut un des premiers à l'appliquer systématiquement dès les premiers volumes de sa série "The art of computer programming". Si l'on élimine de l'analyse de la complexité la question de la vitesse d'exécution de la machine et la qualité du code produit par le compilateur, il ne reste comme paramètre significatif que " la taille des données sur lesquelles il s'exécute. On exprime donc le temps d'exécution en nombre d'opérations élémentaires. On évalue le nombre d'opérations élémentaires en fonction de la taille de la donnée. Si 'n' est la taille, on calcule une fonction f(n). Sachant que le nombre d'opérations élémentaires peut varier substantiellement pour deux données de même taille, on retiendra deux critères:

- Analyse au sens du plus mauvais cas : f(n) est le temps d'exécution du plus mauvais cas et le maximum sur toutes les données de taille n.
- Analyse au sens de la moyenne : fm(n) est l'espérance sur l'ensemble des temps d'exécution munis d'une distribution de probabilités des temps d'exécution.

Pour borner une complexité, on dit que la fonction *f(n)* est en grand O de *c(n)*. Cela signifie qu'il existe une constante *A* telle que, pour toutes les valeurs de N supérieures à une valeur suffisamment grande, la double inéquation $0 \leq f(n) \leq A \times c(n)$ est toujours vérifiée.

Enfin, les classes de complexité les plus courantes sont les suivantes :

Linéaire : an + b

Polynomiale : $a_i n^i + a_{i-1} n^{i-1} \ldots + a_1 n + a_0$

Exponentielle : a^n

Factorielle : n!

Annexe B : Rappels sur la théorie des graphes

Voici quelques définitions relatives à la théorie des graphes [Lacomme et al. 2003] :

- Un graphe G est un couple G = (X, E) constitué d'un ensemble X non vide et fini, et d'un ensemble E de paires d'éléments de X. Les éléments de X sont les sommets du graphe G, ceux de E sont les arcs du graphe G. Si e = {x, y} est un arc de G, on dit que les sommets de x et y, qui sont les extrémités de l'arc e, sont adjacents ou voisins dans le graphe G.

- Un parcours dans un graphe est une liste ordonnée de sommets tel que deux sommets consécutifs soient adjacents.

- Un graphe G est connexe lorsque pour toute paire {x,y} de ses sommets, il existe dans G une chaîne reliant x et y. Les composantes connexes d'un graphe sont ses sous-graphes connexes maximaux, maximaux dans le sens qu'ils ne sont sous-graphes stricts d'aucun autre sous-graphe connexe du graphe.

- Un graphe G est fortement connexe lorsque pour toute paire {x,y} de ses sommets, il existe dans G une chaîne reliant x et y, et une chaîne reliant y et x. Si G n'est pas fortement connexe, il est formé de p>1 composantes fortement connexes. Le nombre p est appelé nombre de forte connexité.

- S'il existe deux chaînes distinctes reliant deux sommets x et y d'un graphe G, alors ce graphe admet un cycle.

- Un point d'articulation d'un graphe connexe est un sommet dont la suppression déconnecte le graphe. Un point d'articulation d'un graphe est un point d'articulation d'une de ses composantes connexes.

- Un isthme d'un graphe connexe est un arc dont la suppression déconnecte le graphe. Un isthme d'un graphe est un isthme d'une de ses composantes connexes.

- Un graphe complet est un graphe dont tous les sommets sont reliés deux à deux.

Annexe C : BLAS (Basic Linear Algebra Subprograms)

Les bibliothèques scientifiques sont des ensembles de sous-programmes testés, validés, optimisés. Utiliser des librairies scientifiques permet de se consacrer uniquement au problème à traiter. Elles ont des qualités intrinsèques qui les rendent très intéressantes :

- Elles ont une interface générique quelle que soit l'architecture système.
- Elles supportent différents types de données : réel ou complexe, simple ou double précision.
- Elles prennent en compte différents types de stockage : matrice bande, symétrique, hermitienne.

BLAS est une librairie de routines qui effectuent des opérations de base impliquant des matrices et des vecteurs. Cette bibliothèque sert de support à d'autres librairies scientifiques. Chaque constructeur les écrit en langage machine pour qu'elles soient optimisées pour une architecture donnée.

Il existe trois niveaux de bibliothèques dans BLAS :

- Le niveau 1 effectue des opérations de bas niveau comme le produit scalaire et la combinaison linéaire de vecteurs. Soit de l'ordre de O(N) opérations flottantes, où N est la dimension des vecteurs impliqués.
- Le niveau 2 comprend les opérations courantes sur les matrices/vecteurs qui apparaissent dans les algorithmes d'algèbre linéaire, soit de l'ordre de O(N^2) opérations flottantes.
- Le niveau 3 correspond aux opérations matrice/matrice, soit de l'ordre de O(N^3) opérations flottantes. Il doit être noté que les matrices peuvent être partitionnées en blocs et les opérations sur des blocs distincts peuvent être effectuées en parallèle.

Annexe D : Exemple d'analyse des périmètres de sécurité IPSEC

De même que pour les ACL et afin d'assurer les services réseau attendus, les configurations des services réseau doivent être analysées afin de s'assurer de l'application de la politique de sécurité. L'exemple ci-après détaille les vérifications à mettre en place afin de contrôler les configurations IPSEC.

A) Consistance de configuration des CryptoMap

Nous proposons ici de détailler une configuration CISCO "conf_test" suivante contenant des commandes IPSEC :

```
hostname test
!
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto ipsec transform-set chiff_auth esp-3des esp-md5-hmac
!
crypto map VPN_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set transform-set chiff_auth
 match address 110
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map VPN_1_1
!
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
!
end
```

La consistance de l'implémentation d'IPSEC consiste tout d'abord à vérifier que les éléments définis sont appliqués, et que les éléments appliqués sont définis. Voici le script écrit en langage AWK qui s’exécute sur une

configuration CISCO. Ce script est un exemple non exhaustif et devra donc être complété.

```
# !/usr/bin/awk -f

#---------------------------------------------------------------
# Stockage des éléments IPSEC définis dans ipsec_def
#---------------------------------------------------------------
$1 == "crypto" && $2 == "ipsec" && $3 == "transform-set" {
          if (!($4 in ipsec_def) && $4!="") {
                  ipsec_def[ $4 ] = $0 "(line "FNR")";
          }
  next;
}

/^crypto / && $2 == "map" {
        if (!($3 in ipsec_def) && $3!="") {
                  ipsec_def[ $3 ] = $0 "(line "FNR")";
        }
  next;
}

/^access-list /  {
        if (!($2 in ipsec_def) && $2!="") {
                  ipsec_def[ $2 ] = $0 "(line "FNR")";
        }
           next;
}

#---------------------------------------------------------------
# Stockage des éléments IPSEC référencés dans ipsec_ref
#---------------------------------------------------------------
/^interface/ {
        interface = $2;
        next;
}

/^ crypto / && $2 == "map" {
        if (!($3 in ipsec_ref) && $3!="") {
                  ipsec_ref[$3]=interface";"$0"(line "FNR")";
        }
        next;
}
```

```
/^ set / && $2 == "transform-set" {
        if (!($3 in ipsec_ref) && $3!="") {
                ipsec_ref[ $3 ] = $0 "(line "FNR")";
        }
        next;
}

/^ match / && $2 == "address" {
        if (!($3 in ipsec_ref) && $3!="") {
                ipsec_ref[ $3 ] = $0 "(line "FNR")";
        }
        next;
}

END {

#-------------------------------------------------------------
# Vérification que les éléments définis sont référencés
#-------------------------------------------------------------
for (id in ipsec_def) {
        if (!(id in ipsec_ref) && id!="") {
                print FILENAME "; déf/non réf;" id ";" ipsec_def[id];
        }
}

#-------------------------------------------------------------
# Vérification que les éléments référencés sont définis
#-------------------------------------------------------------
for (id in ipsec_ref) {
        if (!(id in ipsec_def) && id!="") {
                print FILENAME "; réf/not déf;" id " ;" ipsec_ref[id];
        }
}
}
```

Si on exécute ce script sur la configuration IPSEC, on obtient alors le résultat suivant (aucune inconsistance n'a été détectée) :

```
bash$ awk -f./ipsec.sh./conf_test
bash$
```

Modifions la configuration IPSEC afin d'introduire des inconsistances comme l'illustre la configuration CISCO "conf_test" suivante :

```
hostname conf_test
!
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto ipsec transform-set chiff_auth1 esp-3des esp-md5-hmac
!
crypto map VPN_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set transform-set chiff_auth
 match address 120
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map VPN_1_11
!
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
!
end
```

Si on exécute ce script sur la nouvelle configuration IPSEC, on obtient alors le résultat suivant pointant sur les inconsistances de configuration:

```
bash$ awk -f./ipsec.sh./conf_test
./conf_test; déf/non réf;110;access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0
0.0.0.255(line 22)
./conf_test; déf/non réf;chiff_auth1;crypto ipsec transform-set chiff_auth1 esp-3des esp-
md5-hmac(line 11)
./conf_test; déf/non réf;VPN_1_1;crypto map VPN_1_1 10 ipsec-isakmp(line 13)
./conf_test; réf/not déf;chiff_auth ; set transform-set chiff_auth(line 15)
./conf_test; réf/not déf;120 ; match address 120(line 16)
./conf_test; réf/not déf;VPN_1_11 ;FastEthernet0; crypto map VPN_1_11(line 20)
bash$
```

Il doit être cependant noter que CISCO ne permet pas de supprimer une "crypto map" si celle-ci est appliquée sur des interfaces, à moins de la supprimer sur toutes les interfaces préalablement. En revanche, l'inconsistance de configuration d'une ACL, non définie mais appliquée dans une "crypto map", peut effectivement bloquer le routeur.

B) Consistance de configuration des politiques ISAKMP

Si on désire vérifier la consistance de configuration de la politique IPSEC isakmp de deux routeurs, une technique un peu simpliste consiste à parcourir les configurations, à extraire les éléments clés de la politique isakmp, et de contrôler les éléments qui ne sont pas des doublons. Voici le script écrit en langage AWK qui s'exécute sur deux configurations. Ce script est un exemple non exhaustif et devra donc être complété.

```
# !/bin/sh

awk '
/crypto isakmp policy/, /!/ {
if ($0 ~ /crypto isakmp policy/) {
        policy=$0;
} else {
        # imprime une ligne de la politique de sécurité
        if ($0 !~ /!/) print policy, $0;
}

# On trie et on imprime les doublons
}' $1 $2 | sort | uniq -u
```

Si on exécute ce script sur deux fichiers ("conf_test" et "conf_test1") contenant la même configuration isakmp IPSEC, on obtient alors le résultat suivant (les politiques isakmp sont identiques) :

```
bash$./ipsec1.sh./ conf_test./conf_test1
bash$
```

En revanche, si on modifie dans "conf_test" la politique isakmp (3des en des et group 2 en group 1), on obtient le résultat suivant pointant sur les déviations de la politique isakmp:

```
bash$./ipsec1.sh./ conf_test./conf_test1
crypto isakmp policy 10  encr 3des
crypto isakmp policy 10  encr des
crypto isakmp policy 10  group 1
crypto isakmp policy 10  group 2
```

C) Consistance de configuration des périmètres IPSEC

Si on désire vérifier les périmètres de configuration des réseaux privés virtuels IPSEC, l'approche consiste à analyser le graphe IPSEC engendré par les configurations des VPN IPSEC. Pour cela, nous considérerons tout d'abord que le nom d'une cryptomap suit la règle de configuration suivante :

VPN_X_Y
  X: identifiant unique d'un VPN IPSEC
  Y: instance d'une nouvelle politique pour un VPN IPSEC

Par exemple, la cryptomap VPN_1_1 correspond au VPN 1 et à la politique de sécurité 1. De même, VPN_1_2 correspond au VPN 1 et à la politique de sécurité 2.

Ensuite, si pour chaque configuration, on arrive à renseigner les champs de la table IPSEC suivante (il peut avoir plusieurs enregistrements par configuration de routeur), il est alors possible de construire le graphe IPSEC que nous détaillerons par la suite:

table IPSEC
	champ: NomRouteur: nom du routeur
	champ: CryptoMapId: identifiant unique d'un VPN IPSEC
	champ: IpAdresse: adresse ip de l'interface ou est appliquée une cryptomap
	champ: IpAdresseDestination: adresse ip destinatrice du tunnel IPSEC

Voici le script écrit en langage AWK qui s'exécute sur une configuration CISCO permettant d'extraire ces informations. Ce script est un exemple non exhaustif et devra donc être complété.

```
# !/usr/bin/awk -f

# Stocke le nom de la cryptomap dans le tableau cryptomap
/^crypto / && $2 == "map" {
		if ( !($3 in cryptomap)) {
			i = 0;
			this_cryptomap = $3;
			cryptomap[ $3 ] = $0;
		 }
		 next;
}

# Stocke les adresses des peers relatif à la cryptomap en cours
/^ set / && $2 == "peer" {
		 peer[this_cryptomap, i++] = $3;
		 next;
}

# Stocke l'adresse IP d'une interface
/^ ip address/ {
		address = $3;
		next;
}
```

```
# Imprime tous les couples: adresse IP de l 'interface et les adresses IP des peers IPSEC
/^ crypto / && $2 == "map" {
    if ($3 in cryptomap) {
        N = split($3, tmp, _");
        if (n == 3)  {
                for (i = 0; (peer[$3,i]) != ""; i++) {
                        print FILENAME, tmp[2], address, peer[$3,i];
                }
        }
    }
}
```

Si on exécute ce script sur la configuration CISCO suivante :

```
hostname conf_test
!
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
crypto isakmp key pvntl2o9xsra5 address 192.168.1.155
crypto isakmp key 6rtzlmkw6awvp address 192.168.1.156
crypto isakmp key p0vzuxb74uvjx address 192.165.1.154
crypto isakmp key pfjgkw1ml3hl8 address 192.165.1.155
crypto isakmp key qgp5h3fblo92p address 192.165.1.156
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto ipsec transform-set chiff_auth1 esp-3des esp-md5-hmac
!
crypto map VPN_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set peer 192.168.1.155
 set peer 192.168.1.156
 set transform-set chiff_auth
 match address 110
!
crypto map VPN_2_1 10 ipsec-isakmp
 set peer 192.165.1.154
 set peer 192.165.1.155
 set peer 192.165.1.156
 set transform-set chiff_auth
```

```
 match address 120
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map VPN_1_1
!
interface FastEthernet1
 ip address 192.165.1.1 255.255.255.0
 crypto map VPN_2_1
!
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
access-list 120 permit ip 10.0.3.0 0.0.0.255 10.0.4.0 0.0.0.255
!
end
```

On obtient alors le résultat suivant :

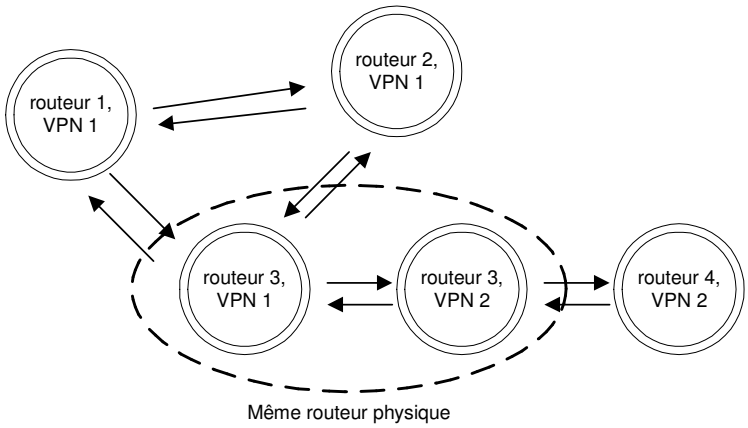
```
bash$ awk -f./ipsec2.sh./conf_test
./conf_test 1 192.168.1.1 192.168.1.154
./conf_test 1 192.168.1.1 192.168.1.155
./conf_test 1 192.168.1.1 192.168.1.156
./conf_test 2 192.165.1.1 192.165.1.154
./conf_test 2 192.165.1.1 192.165.1.155
./conf_test 2 192.165.1.1 192.165.1.156
```

Une fois la table IPSEC construite à partir de l'extraction des informations contenues dans les configurations, le produit cartésien de la table IPSEC par elle-même, conditionné par le fait que l'adresse ip de l'interface (où est appliquée une "crypto map") soit égale à l'adresse ip destinatrice du tunnel IPSEC, et que les CryptoMapId soient identiques, donne alors tous les arcs de notre graphe IPSEC comme l'illustre la requête SQL suivante :

```
SELECT
        Ipsec.NomRouteur, Ipsec.CryptoMapId, Ipsec.IpAdresse, Ipsec.IpAdresseDestination,
        Ipsec_1.NomRouteur, Ipsec_1.CryptoMapId, Ipsec_1.IpAdresse,
Ipsec_1.IpAdresseDestination
FROM
        Ipsec, Ipsec AS Ipsec_1
WHERE
        Ipsec.IpAdresseDestination=Ipsec_1.IpAdresse and
        Ipsec.CryptoMapId = Ipsec_1.CryptoMapId;
```

Un sommet du graphe IPSEC est donc représenté par le couple (NomRouteur,CryptoMapId), et un arc par un enregistrement trouvé par le

produit cartésien précédemment décrit. De plus, l'asymétrie de configuration d'un tunnel IPSEC indique que le graphe IPSEC construit est orienté comme l'illustre le schéma suivant :



Dans notre première configuration IPSEC, la table IPSEC serait alors renseignée par les données suivantes si nous avions dans "conf_test1" la configuration IPSEC associée :

Nom routeur	CryptoMapId	Adresse ip de l'interface	Adresse ip du tunnel IPSEC
Conf_test	1	192.168.1.1	192.168.1.154
Conf_test1	1	192.168.1.154	192.168.1.1

Maintenant, si l'on réalise le produit cartésien précédemment décrit, on obtient alors les informations suivantes (en fait les adresses IP nous permettent d'établir les relations de connectivité entre les sommets) :

Nom routeur	CryptoMapId	Adresse ip de l'interface	Adresse ip du tunnel IPSEC	Nom routeur	CryptoMapId	Adresse ip de l'interface	Adresse ip du tunnel IPSEC
Conf_test	1	192.168.1.1	192.168.1.154	Conf_test1	1	192.168.1.154	192.168.1.1
Conf_test1	1	192.168.1.154	192.168.1.1	Conf_test	1	192.168.1.1	192.168.1.154

Les arcs du graphe IPSEC sont les suivants :

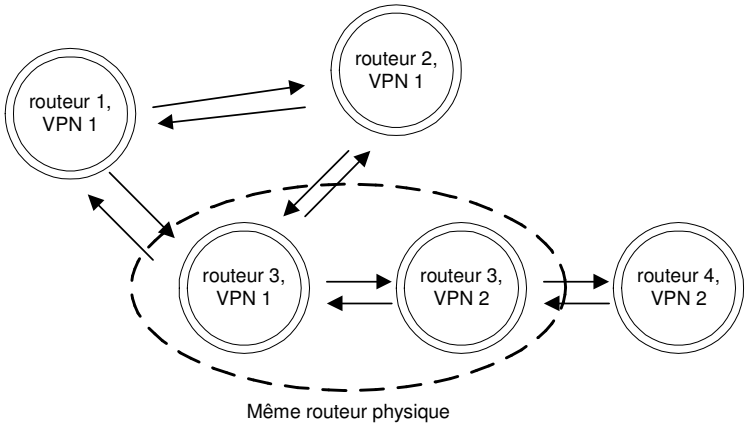
> (Conf_test,1) est ipsec-connecté à (Conf_test1,1)
> (Conf_test1,1) est ipsec-connecté à (Conf_test,1)

On a donc bien un tunnel IPSEC entre (Conf_test,1) et (Conf_test1,1) par l'asymétrie des connexions entre ces deux sommets. De manière plus générique, si on considère le graphe IPSEC ayant pour sommets (Conf_test,1) et (Conf_test1,1), alors un VPN IPSEC correspond à une composante fortement connexe du graphe IPSEC (si pour toute paire de sommets (x,y) de la composante, il existe un chemin de x à y et de y à x). Dans notre cas, il y a

une seule composante fortement connexe qui est {(Conf_test,1), (Conf_test1,1)} et qui représente le périmètre de sécurité du VPN 1 [Llorens et al. 2003].

De manière théorique, les composantes fortement connexes du graphe IPSEC donnent les périmètres de sécurité des VPN IPSEC qui ont pu être définies dans les configurations. Ces périmètres de sécurité montrent alors soit l'isolation d'un VPN IPSEC donné, soit des interconnexions avec d'autres VPN IPSEC. De plus, l'extraction de toutes les composantes fortement connexes d'un graphe est un problème facile.

Prenons un réseau plus complexe composé des routeurs et des configurations IPSEC suivantes :



Si on extrait des configurations les éléments permettant de construire la table IPSEC, on obtient alors après le produit cartésien les informations suivantes :

> (routeur1,1) est ipsec-connecté à (routeur2,1) && (routeur2,1) est ipsec-connecté à (routeur1,1)
> (routeur1,1) est ipsec-connecté à (routeur3,1) && (routeur3,1) est ipsec-connecté à (routeur1,1)
> (routeur2,1) est ipsec-connecté à (routeur3,1) && (routeur3,1) est ipsec-connecté à (routeur2,1)
> (routeur2,2) est ipsec-connecté à (routeur4,2) && (routeur4,2) est ipsec-connecté à (routeur2,2)
> (routeur3,2) est ipsec-connecté à (routeur4,2) && (routeur4,2) est ipsec-connecté à (routeur3,2)
> (routeur3,2) est ipsec-connecté à (routeur2,2) && (routeur2,2) est ipsec-connecté à (routeur3,2)

Les composantes fortement connexes de notre graphe IPSEC sont donc {(routeur1,1), (routeur2,1), (routeur3,1)} et {(routeur2,2), (routeur3,2), (routeur3,2)} qui permettent de vérifier les périmètres configurés. Dans notre exemple, les périmètres de sécurité sont bien restreints aux VPN IPSEC définies dans les configurations (VPN 1 et VPN 2).

Il est donc possible à partir d'un grand nombre de configurations réseau de retrouver les périmètres des réseaux privés virtuels IPSEC par une analyse des composantes fortement connexes du graphe IPSEC.

Enfin, si les composantes connexes (s'il existe un chemin entre toute paire de sommets (x,y) de la composante) du graphe IPSEC ne sont pas égales aux composantes fortement connexes (si pour toute paire de sommets (x,y) de la

composante, il existe un chemin de x à y et de y à x) du graphe IPSEC, alors il y a des inconsistances de configuration des VPNs IPSEC. De même, toute configuration non bidirectionnelle entre deux sommets montrent aussi des inconsistances de configuration des VPNs IPSEC.

Références bibliographiques

[Bedford et al. 2001] Bedford (T.), Cooke (R.M.), Probabilistic risk analysis: foundations and methods, *Cambridge University Press*, ISBN 0-521-77320-2, pp. 97-255, 2001.

[Berchtold 1998] Berchtold (A.), Chaînes de markov et modèles de transition, *Hermes*, ISBN 2-86601-661-0, pp. 13-102, 1998.

[Brassard et al. 1996] Brassard (G.), Bratley (P.), Fundamentals Of algorithmics, *Prentice-Hall*, ISBN 0-13-335068-1, pp. 219-258, 1996.

[Bryant 1986] Bryant (R.E.), Graph-based algorithms for boolean function manipulation, *IEEE Transactions on Computers*, C-35(8), 1986.

[Bush et al. 2001] Bush (S.F.), Evans (S.C.), Complexity-Based information assurance, *General Electrics Corporate Research and Development report*, number 2001CRD084, 2001.

[Capinski et al. 2001] Capinski (M.), Zastawniak (T.), Probalility through problems, *Springler*, ISBN 0-387-95063-X, pp. 117-167, 2001.

[CC 2002] Common Criteria, Common Criteria for information technology security evaluation, v2.2, 2004.

[Cimatti et al. 2002] Cimatti (A.), Clarke (E.M.), Giunchiglia (E.), Giunchiglia (F.), Pistore (M.), Roveri (M.), Sebastiani (R.), Tacchella (A.), NuSMV2 : An openSource tool for symbolic modelchecking, *Proc. 14th Intl Conf. Computer AidedVerification (CAV 2002)*, Springer-Verlag, Lect. Notes Comp. Sci. 2404, pp. 359-364, 2002.

[CISCO 2001] CISCO Systems, Essential IOS features every ISP should consider, *CISCO report*, v 2.9, 2001.

[Cogis et al. 2003] Cogis (O.), Robert (C.), Au-delà des ponts de Königsberg, *Vuibert*, ISBN 2-7117-5321-2, pp. 108-118, 2003.

[Cormen et al. 2002] Cormen (T.H.), Leverson (C.E.), Rivest (R.L.), Stein (C.), Introduction à l'algorithmique, *Dunod*, ISBN 2-10-003922-9, pp. 701-741, 2002.

[CTCPEC 1993] CTCPEC, The Canadian Trusted Computer Product Evaluation Criteria, Canadian System Security Center, Communications Security Establishment, Governement of Canada, version 3.0e, 1993.

[Dacier 1994] Dacier (M.), Vers une évaluation quantitative de la sécurité informatique, *Thèse de doctorat, Institut Polythechnique de Toulouse*, no 971, 1994.

[Dacier et al. 1994] Dacier (M.), Deswarte (Y.), Privilege Graph: an Extension to the Typed Access Matrix Model, *in Third European Symposium on Research in Computer Security (ESORICS 94)*, (D.Gollman, Ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, 875, Springler-Verlag, ISBN 3-540-58618-0, pp. 317-334, 1994

[Dacier et al. 1996] Dacier (M.), Deswarte (Y.), Kaâniche (M.), Models and Tools for Quantitative Assessment of Operational Security, *in 12th IFIP Information Systems Security Conference (IFIP/SEC'96)*, (S. K. Katsikas, D. Gritzalis, Eds.), Samos, Greece, May 21-23, pp.177-186, ISBN 0-412-78120-4, Chapman & Hall, 1996.

[Eppstein et al. 2001] Eppstein (D.), Muthukrishnan (S.), Internet packet filter management and rectangle geometry, *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pp. 827-835, 2001.

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

[Feamster et al. 2003] Feamster (N.), Balakrishnan (H.), Towards a logic for wide-area Internet routing, *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pp. 289-300, 2003.

[Fenton et al. 1996] Fenton (N.E.), Pfleeger (S.L.), Software Metrics: A Rigorous Approach & Practical Approach Revised, *PWS Publishing Company*, Second Edition, ISBN 0534954251, pp. 23-76, 1996.

[IS_IS 1992] ISO/IEC 10589, Information technology – Télécommunications and information exchange between systems – Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connexionless-mode Network Service (ISO 8473), 1992.

[ITSEC 1991] ITSEC, Critères d'évaluation de la sécurité des systèmes informatiques, *Office des publications officielles des Communautés Européennes*, ISBN 95-826-3005-6, Luxembourg, v1.2, 1991.

[Lacomme et al. 2003] Lacomme (P.), Prons (C.), Sevaux (M.), Algorithmes et graphes, *Eyrolles*, ISBN 2-212-11385-4, pp. 135-175, 2003.

[Jensen 2001] Jensen (F.V.), Bayesian networks and decision graphs, *Springer-Verlag*, ISBN 0-387-95259-4, pp. 1-157, 2001.

[JCSEC 1992] JCSEC, The Japanese Computer Security Evaluation Criteria-Functionality Requirements, *Ministry of International Trade and industry*, Draft V1.0, August, 1992.

[Laprie 1995] Laprie (J.C.), Guide de la Sûreté de Fonctionnement, *Cépaduès Editions*, 2ème édition, ISBN 2-85428-341-4, pp. 324-325, 1995.

[Llorens et al. 2003] Llorens (C.), Valois (D.), Le Teigner (Y.), Gibouin (A.), Computational complexity of the network routing logical security, *Proceedings*

of the IEEE international Information Assurance Workshop, Darmstadt, Germany, pp. 37-49, 2003.

[Mahajan et al. 2002] Mahajan (R.), Wetherall (D.), Anderson (T.), Understanding BGP misconfiguration, *ACM Proceedings of the 2002 conference on applications, technologies, architectures, and protocols for computer communications*, pp. 3-16, 2002.

[NSA 2003] National Security Agency, Security recommendations guides for router configuration, *NSA report*, v1.1, 2002.

[Ortalo 1998] Ortalo (R.), Évaluation quantitative de la sécurité des systèmes d¹information, *Thèse de Doctorat de l¹Institut National Polytechnique de Toulouse*, 1998.

[Philips et al. 1998] C. Phillips and L. Swiler, A graph-based system for network-vulnerability analysis, *in Proceedings of the 1998 Workshop on New Security Paradigms*, pp. 71–79, 1998.

[RFC0791] Postel (J.), Internet Protocol, *www.ietf.org*, Standard, 1981.

[RFC0792] Postel (J.), Internet Control Message Protocol, *Internet Engineering Task Force,* www.ietf.org, Standard, 1981.

[RFC0793] J. Postel (J.), Transmission Control Protocol, *Internet Engineering Task Force,* www.ietf.org, Standard, 1981.

[RFC1142] Oran (D.), OSI IS_IS Intra-domain Routing Protocol, *Internet Engineering Task Force,* www.ietf.org, Informational, 1990.

[RFC1195] Callon (R.W.), Use of OSI IS_IS for routing in TCP/IP and dual environments, *Internet Engineering Task Force,* www.ietf.org, Standard, 1990.

[RFC1774] Traina (P.), BGP-4 Protocol Analysis, *Internet Engineering Task Force,* www.ietf.org, Informational, 1995.

[RFC1918] Rekhter (Y.), Moskowitz (B.), Karrenberg (D.), de Groot (G.J.), Lear (E.), Address Allocation for Private Internets, *Internet Engineering Task Force, www.ietf.org*, Best current practice, 1996.

[RFC2385] Heffernan (A.), Protection of BGP Sessions via the TCP MD5 Signature Option, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 1998.

[RFC2439] Villamizar (C.), Chandra (R.), Govindan (R.), BGP Route Flap Damping, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 1998.

[RFC2547] Rosen (E.), Rekhter (Y.), BGP/MPLS VPNs, *Internet Engineering Task Force,* www.ietf.org, Informational, 1999.

[RFC2796] Bates (T.), Chandra (R.), Chen (E.), BGP Route Reflection-An Alternative to Full Mesh IBGP, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, www.ietf.org, 2000.

[RFC2917] Muthukrishnan (K.), Malis (A.), A Core MPLS IP VPN Architecture, *Internet Engineering Task Force,* www.ietf.org, Informational, 2000.

[RFC3031] Rosen (E.), Viswanathan (A.), Callon (R.), Multiprotocol Label Switching Architecture, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 2001.

[RFC3032] Rosen (E.), Tappan (D.), Fedorkow (G.), Rekhter (Y.), Farinacci (D.), Li (T.), Conta (A.), MPLS Label Stack Encoding, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 2001

[RFC3034] Conta (A.), Doolan (P.), Malis (A.), Use of Label Switching on Frame Relay Networks Specification, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 2001.

[RFC3036] Andersson (L.), Doolan (P.), Feldman (N.), Fredette (A.), Thomas (B.), LDP Specification, *Internet Engineering Task Force,* www.ietf.org, Standard, 2001.

[RFC3065] Traina (P.), McPherson (D.), Scudder (J.), Autonomous System Confederations for BGP, *Internet Engineering Task Force,* www.ietf.org, Proposed standard, 2001

[RFC3345] McPherson (D.), Gill (V.), Walton (D.), Retana (A.), Border Gateway Protocol (BGP) Persistent Route Oscillation Condition, *Internet Engineering Task Force,* www.ietf.org, Informational, 2002.

[RFC3567] Li (T.), Atkinson (R.), Intermediate System to Intermediate System (IS_IS) Cryptographic Authentication, *Internet Engineering Task Force,* www.ietf.org, Informational, 2003.

[RFC3882] Turk (D.), Configuring BGP to Block Denial-of-Service Attacks, *Internet Engineering Task Force,* www.ietf.org, Informational , 2004.

[Saaty 1988] Saaty (T.L.), Mathematical models for decision support archive, *Source-Nato Asi Series archive, Springer-Verlag*, ISBN 0-387-50084-7, pp. 89-107, 1988.

[Sedgewick 1977] Sedgewick (R.), Permutation generation methods, *ACM Computing Surveys*, Volume 9, Issue 2, pp. 137-164, 1977.

[Somesh et al. 2002] Somesh (J.), Sheyner (O.), Wing (J.M.), Minimization and reliability analyses of attack graphs, *Proceedings of the Computer Security Foundations Workshop*, Nova Scotia, pp. 49-63, 2002.

[Stamatelotos 2002] Stamatelotos (M.), Probabilistic risk assessment procedures guide for NASA managers and practitionners, *NASA report*, v1.1, 2002.

[Stoneburner et al. 2001] Stoneburner (G.), Goguen (A.), Ferringa (A.), Risk management guide for information technology systems, *National Institute of Standards and Technology*, SP 800-30, 2001.

[Swiler et al. 2001] Swiler (L.P.), Philips (C.), Ellis (D.), Chakerian (S.), Computer-attack graph generation tool, *DISCEX' 01 : DARPA Information Survivability Conference and Exposition II.*, pp. 307-321, 2001.

[Tarjan 1972] R.Tarjan (R.), Depth-First Search and Linear Graph Algorithms, *SICOMP*, Vol. 1, Number 2, pp. 146-160, 1972.

[TCSEC 1985] TCSEC, Trusted Computer System Evaluation Criteria, Department of Defense (DoD), DoD Standard, DoD 5200.28-STD, 1985.

[Valois et Llorens 2002] Valois (D.), Llorens (C.), Network Device Configuration Validation, *Proceedings of 14th annual FIRST conference*, Hawaii, 2002.

[Warkhede et al. 2001] Warkhede (P.), Suri (S.), Varghese (G.), Fast packet classification for two-dimensional conflict-free filters, *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.3, pp. 1434-1443, 2001.

[Williams et al. 1998] Williams (J.R.), Jelen (G.F.), A framework for reasoning about assurance, *Project report supported by National Securiy Agency*, contract number MDA904-97-C-0223, 1998.

[Wulf et al. 1996] Wulf (W.A.), Kienzle (D.M.), A practical approach to security assessment, *MOAT project report supported by DARPA*, contract number N66001-96-C-8527, 1996.