Tableaux de bord de la sécurité réseau

3e édition

Cédric Llorens

Laurent Levier

Denis Valois

Benjamin Morin

Avec la contribution de Olivier Salvatori

Cette étude de cas, disponible en libre téléchargement sur le site des Éditions Eyrolles, vient en complément de l'ouvrage *Tableaux de bord de la sécurité réseau*, 3e édition, paru en septembre 2010 aux Éditions Eyrolles.

➡ www.editions-eyrolles.com/Livre/9782212128215/

EYROLLES

Table des matières

CHAPITRE 21

Partie VII

Étude de cas

L'étude de cas présentée dans cette partie a pour objectif d'illustrer les notions abordées tout au long de l'ouvrage dans une perspective pratique. Elle s'ouvre par la présentation d'une série d'outils maison que nous mettons à a disposition du lecteur et se poursuit par la mise en œuvre d'un exemple concret d'évolution d'une entreprise et de sa sécurité.

Au travers des outils maison, notre objectif est de faciliter le contrôle des configurations réseau et l'élaboration d'un tableau de bord de la sécurité réseau. Ces outils sont mis en pratique dans l'étude de cas dans un contexte de réseau d'entreprise.

Le chapitre 19 détaille ces outils, qui permettent de contrôler la consistance des ACL et la configuration d'équipements Cisco et Juniper, ainsi que de calculer une valeur de risque. La conception des programmes et des exemples sont également fournis.

L'étude de cas est découpée en deux chapitres, qui retracent les principales étapes de l'évolution du réseau d'une entreprise fictive, RadioVoie, depuis son premier réseau interne de type PME jusqu'au stade de la multinationale. Au travers de l'évolution de cette entreprise et de son réseau, sont illustrés à la fois les besoins de sécurité et les politiques correspondant à chaque étape du développement de l'entreprise.

Le chapitre 20 détaille la mise en place du premier réseau interne de RadioVoie puis son ouverture vers Internet et à des tierces parties et enfin son premier contrat de défense militaire, à très fortes contraintes de sécurité.

Le chapitre 21 présente la transformation de RadioVoie en une multinationale devant gérer un nombre important d'équipements.

Chaque étape de cette évolution idéale est structurée en une étude de risques, une politique de sécurité réseau tenant compte des besoins exprimés, une solution de sécurité adaptée, un bilan des risques couverts et non couverts et des tableaux de bord de la sécurité.

19

Outils maison de sécurité réseau

Nous avons vu au chapitre 18 les principes permettant d'évaluer la sécurité d'un réseau et de construire des tableaux de bord de la sécurité réseau. On constate souvent que les produits logiciels disponibles sur le marché présentent des limitations. Ces limitations sont généralement intrinsèques au modèle sous-jacent desdits produits, qui ne peuvent prendre en compte des besoins de sécurité spécifiques.

L'objectif de ce chapitre est de montrer, par des exemples concrets, la relative facilité avec laquelle il est possible de concevoir des automatismes de diagnostic, de mesures et de tableaux de bord pour répondre à ces besoins spécifiques.

Bien que ces outils maison n'offrent pas d'interfaces professionnelles, ils répondent de manière efficace à certains problèmes de sécurité (les sources de ces outils sont disponibles à l'adresse *http://tableaux.levier.org*).

Ils permettent en outre d'obtenir une puissance d'expression plus riche qu'une simple comparaison textuelle sur la base de patrons littéraux. Ce pouvoir d'expression a un sens uniquement sous l'hypothèse raisonnable qu'il traduit des fonctions effectivement calculables.

Ce chapitre est structuré selon une approche ascendante : les outils de base, permettant d'évaluer un critère atomique, sont présentés en premier, et les outils plus puissants à la fin. Bien que ces outils soient fortement orientés analyse de configuration de routeurs, le principe général de leur fonctionnement vaut aussi pour d'autres types d'équipement réseau, comme les pare-feu.

En règle générale, ces outils nécessitent d'avoir accès à un fichier de configuration afin de l'interpréter. Ils ont été développés selon une approche « offline » afin d'éviter toute interaction directe avec le réseau.

Tous les outils décrits dans ce chapitre sont accessibles à un deuxième niveau universitaire. Certains d'entre eux exigent des connaissances en cryptographie, en théorie des

langages et en compilation, en théorie des graphes, en structures des données et en algorithmique. Comme ces outils maison sont parfois comparés à des outils standards Unix, une connaissance de ceux-ci est utile.

Les exemples sont tous réalisés sur un système Linux appelé margot. Ainsi, l'entrée au niveau de l'interprète de commandes (shell) est identifiée par le prompt :

```
margot$
```

Analyse de la conformité des mots de passe

Un grand nombre de routeurs Cisco (plusieurs dizaines de milliers) sont gérés par une équipe, chaque membre de l'équipe devant pouvoir s'authentifier individuellement. Il est possible pour cela de déployer un serveur AAA et de configurer les routeurs pour relayer l'authentification à ce serveur. Un mécanisme, dit « mode catastrophe », doit être prévu pour pallier une perte éventuelle de connectivité entre les routeurs et le serveur AAA. Il s'agit de configurer également les mots de passe d'accès en urgence.

On distingue trois mots de passe catastrophe : l'accès session VTY, le mode privilégié ENABLE et l'accès console. Si le nombre d'équipements est important, un grand nombre de mots de passe doivent être partagés par l'équipe. Ces mots de passe ont par ailleurs une durée de vie très longue.

Ce modèle, que nous appelons « livret de mots de passe », est en fait une base de données relationnelle dans laquelle un mot de passe distinct est associé à un couple <routeur, type d'accès>.

La politique de sécurité demande de vérifier la conformité *a posteriori* de ces mots de passe dans chaque configuration. De plus, il faut vérifier la non-divulgation du mot de passe ENABLE dans chaque configuration.

Les outils GENPASS et Cisco_CRYPT sont écrits chacun en moins de 500 lignes C et permettent de réaliser ces contrôles.

Conception des outils

Plutôt que de générer aléatoirement des mots de passe pour ensuite les intégrer dans une véritable base de données (chiffrée), l'outil GENPASS est conçu pour offrir une solution de remplacement à la gestion d'une base de données distribuée.

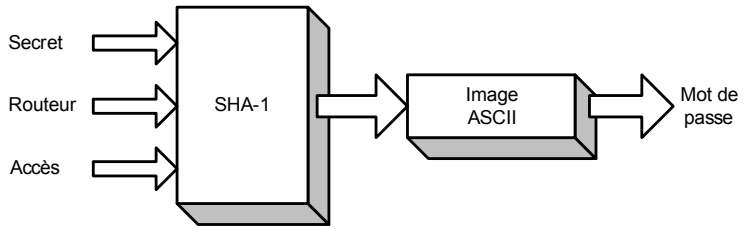
GENPASS s'appuie sur les caractéristiques communes à tous les générateurs de signature cryptographique (MD5, SHA-1, etc.) :

- Il est très difficile de retrouver le texte initial à partir de la signature.
- Les collisions de signatures sont très peu probables.

GENPASS accepte en paramètre un secret, le nom d'un routeur et le type d'accès. Il calcule alors la signature cryptographique des paramètres et imprime une image de cette signature, comme l'illustre la figure 19.1.

Figure 19.1

Génération de mots de passe



Les avantages de cette approche sont relativement évidents :

- Nul besoin de mise à jour de la base de données pour un nouveau routeur, ou un nouveau type d'accès ; le livret est une implémentation algorithmique sans base de données.
- L'implémentation est beaucoup plus efficace qu'une base de données ; le programme lui-même est public et le partage d'un unique secret est léger.

En revanche, il faut être conscient des risques suivants :

- C'est le secret qui définit le livret, donc l'ensemble des mots de passe ; il convient donc de le protéger adéquatement.
- Il est impossible de modifier un seul mot de passe sans tous les modifier également ; ce modèle de gestion de mots de passe ne convient donc qu'à des mots de passe ayant tous une même durée de vie.

Par souci de complétude, GENPASS intègre une fonctionnalité de génération aléatoire utilisée pour forger le secret. GENPASS utilise pour cela la bibliothèque cryptographique de OpenSSL, qui offre toutes les primitives nécessaires.

Dans le monde Cisco, les mots de passe ENABLE sont codés dans la configuration avec deux méthodes distinctes : PASSWORD-7 et SECRET-5. La méthode PASSWORD-7 est réversible, si bien que même un cryptanalyste amateur peut casser l'algorithme en peu de temps. Plusieurs décodeurs sont d'ailleurs disponibles sur Internet, et n'importe quel moteur de recherche peut les trouver rapidement.

La méthode SECRET-5 est fondée sur une signature MD5, qui est une implémentation directe de la méthode utilisée par les systèmes Unix. Cet algorithme est non réversible.

Le problème est que ces deux méthodes Cisco ne sont pas mutuellement exclusives et qu'il est possible de retrouver le même mot de passe ENABLE dans une configuration sous les deux types de codage. Quand les deux codages sont présents dans une configuration, la méthode SECRET-5 est utilisée prioritairement.

En conséquence, la politique de sécurité se décline ainsi : si les deux codages sont présents dans une configuration, le codage PASSWORD-7 ne doit pas révéler le véritable mot de passe encodé sous SECRET-5.

L'outil Cisco_CRYPT a été conçu pour intégrer les fonctionnalités d'encodage sous les deux modes et de décodage sous le mode PASSWORD-7.

Prise en main

Soit la ligne de commande suivante :

```
genpass -dr -n nombre -l longueur -a regex -w fichier-mots -f fichier-germe
➥ -s chaîne-germe clefs
```

- -d spécifie une génération déterministe des mots de passe, implémentant le mode livret.
- -r spécifie une génération aléatoire des mots de passe.
- -n nombre spécifie le nombre de mots de passe à générer.
- -l longueur spécifie le nombre de caractères pour chaque mot de passe généré.
- -a regex spécifie l'alphabet utilisé pour générer les mots de passe.
- -w fichier-mots spécifie un fichier contenant des mots à utiliser comme alphabet pour générer des phrases de passe.
- -f fichier-germe spécifie le fichier contenant un secret à utiliser pour la génération des mots de passe.
- -s chaîne-germe spécifie une chaîne de caractères à utiliser pour la génération des mots de passe.
- clefs spécifie les paramètres non secrets pour la génération des mots de passe.

Et la ligne de commande suivante :

```
cisco_crypt -e5 -e7 -s sel -d7 mot-de-passe
```

- -e5 spécifie le chiffrement du mot de passe sous le mode SECRET-5.
- -e7 spécifie le chiffrement du mot de passe sous le mode PASSWORD-7.
- -s sel spécifie le sel (*salt*) à utiliser dans les modes de chiffrement.
- -d7 spécifie le déchiffrement du mot de passe dans le mode PASSWORD-7.
- mot-de-passe : spécifie le mot de passe qui sera soit chiffré soit déchiffré.

Exemples

Dans un premier temps, nous utilisons GENPASS pour forger le secret d'un livret pour le réseau « client ». Le secret, de 256 caractères hexadécimaux, est conservé dans le fichier client.key. Le germe supplémentaire 'client' est injecté dans l'entropie du générateur aléatoire :

```
margot$ genpass -r -l 256 -a '[0-9A-F]' -s 'client' >client.key
```

Par exemple, le fichier client.key contient la chaîne suivante :

```
B61F695BDE0DBA1940707142B67281AF5B8DD610B57B26F7A3D71BC973437277EBBF626106D7110EEDCB
➥ 5AC1CCF85F2F0526502793024A204CBCB50F194C74A6FDF2A61756D95ECAA9F387B5F9B05411BF7CD
➥ F88F0BDF355D1FC0509D7BDCE93D716ACE94E0BAA90062272CBA335BE78545DF38B0225DF77C13AAC
➥ CF37E2EAC9
```

Nous déterminons ensuite le mot de passe à utiliser pour le routeur dont le nom est dans la variable ROUTEUR et pour le type d'accès dont le nom vty, enable ou console est dans la variable ACCES.

Les mots de passe comportent 12 symboles pris sur un alphabet de 64 caractères (alphanumériques et deux signes de ponctuation).

Dans l'exemple suivant, la variable ROUTEUR contient la chaîne client-01-04a et la variable ACCES la chaîne enable :

```
margot$ genpass -d -f client.key -l 12 -a '[a-zA-Z0-9./]' \
    $ROUTEUR $ACCES
Tdi4.TulMSr8
```

Notons que cette dernière commande doit être invoquée pour la configuration initiale du routeur.

La souplesse dans la spécification des paramètres de GENPASS permet de définir à la volée une structure de mots de passe par réseau, client, région, etc. Le nombre de clés et de paramètres n'est pas borné. Ainsi, GENPASS peut être utilisé pour la génération de secrets prépartagés IPsec ou WI-FI.

L'utilisation de l'outil cisco_crypt est triviale : les paramètres d7, e7 et e5 spécifient respectivement les modes de décodage-encodage PASSWORD-7 et l'encodage SECRET-5. Le paramètre s spécifie le sel si nécessaire.

La validation de la politique de conformité du mot de passe ENABLE peut s'implémenter de la façon suivante :

```
#!/bin/sh
# le paramètre $1 est le nom du routeur et du fichier de
# configuration
#
ENCODED_5=`awk '/^enable secret 5 / { print $4 }' $1`
EXPECTED=`genpass -d -f client.key -l 12 -a '[a-zA-Z0-9./]' \
     $1 "enable"`
EXPECTED_5=`cisco_crypt -e5 -s $ENCODED_5 $EXPECTED`

if [ $ENCODED_5 = $EXPECTED_5 ]
then
          echo "$1 CONFORME"
else
          echo "$1 NON CONFORME"
fi
```

De même, la validation de la politique de non-divulgation du mot de passe ENABLE peut s'implémenter comme suit :

```
# !/bin/sh
# le paramètre $1 est le nom du routeur et du fichier de
# configuration
#
```

```
ENCODED_7=`awk '/^enable password 7 / { print $4 }' $1`
ENCODED_5=`awk '/^enable secret 5 / { print $4 }' $1`
DECODED_7=`cisco_crypt -d7 $ENCODED_7`
RECODED_5=`cisco_crypt -e5 -s $ENCODED_5 $DECODED_7`

if [ $ENCODED_5 = $RECODED_5 ]
then
          echo "$1 DIVULGATION"
else
          echo "$1 NON DIVULGATION"
fi
```

Supposons maintenant que le fichier de configuration du routeur client-01-04a contienne les deux lignes suivantes :

```
enable secret 5 $1$ONFJ$fel1u1qDLLU1sW4fqZ3M60
enable password 7 14231602584A1E3E280500277A
```

Les invocations de cisco_crypt donnent :

```
margot$ cisco_crypt -e7 -s 14 Tdi4.TulMSr8
14231602584A1E3E280500277A
margot$ cisco_crypt -d7 14231602584A1E3E280500277A
Tdi4.TulMSr8
margot$ cisco_crypt -e5 -s '$1$ONFJ$' Tdi4.TulMSr8
$1$ONFJ$fel1u1qDLLU1sW4fqZ3M60
```

La configuration du routeur est conforme, puisque le mot de passe ENABLE est bien ce qu'il doit être, alors même que ce mot de passe est codé sous les deux modes, en violation de la politique de non-divulgation.

Analyse de la cohérence d'ACL

Valider une ACL est facile lorsqu'elle ne dépasse pas quelques dizaines de lignes. Malheureusement, il n'est pas rare d'être confronté à des ACL significativement plus longues, ce qui rend la validation automatique essentielle. Par exemple, un ingénieur pourrait déployer de longues ACL pour implémenter un « pare-feu du pauvre ». Cette section détaille un outil conçu pour automatiser la détection d'ACL incohérentes.

L'outil VACL analyse une ACL indépendamment de toutes les autres. Il n'est pas conçu pour gérer globalement l'ensemble des ACL d'un réseau, à la différence du produit Sol-Soft disponible sur le marché.

Comme les autres outils décrits dans ce chapitre, VACL fonctionne essentiellement offline, en lisant un fichier de configuration téléchargé par un autre moyen. VACL analyse une ACL et rapporte les diagnostics de redondance et d'inconsistance, même en cas d'incohérence partielle.

VACL est écrit en moins de 4 000 lignes de code C.

Conception de l'outil

Idéalement, toutes les règles d'une ACL devraient référer à des adresses IP, des ports et des protocoles distincts. Si deux règles d'une même ACL réfèrent aux mêmes adresses, ports et protocoles, on devrait y regarder de plus près pour investiguer la cohérence de ces deux règles.

Une ACL étendue (au sens de Cisco) est représentée par un 7-tuple dans un espace discret défini par les sept dimensions suivantes :

- permission permit ou deny ;
- protocole IP (ICMP, TCP, UDP, etc.) ;
- intervalle d'adresses IP sources ;
- ports sources, pour le protocole TCP ou UDP ;
- intervalle d'adresses IP destination ;
- ports destination, pour le protocole TCP ou UDP ;
- paramètres associés au protocole.

Ces sept dimensions sont des ensembles discrets et finis. L'idée sous-jacente est de considérer une règle d'ACL comme un hyper-rectangle dans cet espace multidimensionnel. La détection des incohérences est donc réduite au calcul des intersections dans un ensemble de solides.

VACL est écrit en C, avec un cadre LEX et YACC, et parcourt une ACL en suivant la syntaxe et la sémantique Cisco.

La syntaxe d'une ACL a été reconstruite par la grammaire hors contexte suivante :

```
acl → std_head std_line | ext_head ext_line |
   named_std_head named_std_body | named_ext_head named_ext_body
std_head → access-list std-acl-number
ext_head → access-list ext-acl-number
named_std_head → string access-list standard
named_ext_head → string access-list extended
named_std_body → std_line | named_std_body std_line
named_ext_body → ext_line | named_ext_body ext_line
std_line → permission addresses
ext_line → permission protocol addresses ports addresses ports
    protocol_flag precedence tos log
permission → permit | deny
protocol → string | number
addresses → host string | any | subnet-ip-addr netmask
ports → empty-string | eq port | neq port | lt port | gt port |
    range port port
port → string | number
protocol_flag → empty-string | string | number | number number
precedence → empty-string | precedence string | precedence number
tos → empty-string | tos string | tos number
log → empty-string | log
```

Une dépendance importante est que le système sur lequel s'exécute VACL doit pouvoir résoudre les noms DNS et les noms de protocoles et de ports de la même manière que le routeur analysé. En revanche, si les règles de l'ACL sont exprimées numériquement, cette dépendance ne s'applique plus.

Sur un PC Intel bas de gamme, sous Linux ou BSD, l'outil peut analyser rapidement des ACL de plus de 4 000 règles.

Prise en main

Soit la ligne de commande suivante :

```
vacl -a fichier-acl
```

- -a spécifie l'analyse d'une ACL.
- fichier-acl spécifie le nom du fichier contenant le texte d'une ACL.

Exemple

Considérons une ACL définie par les quatre règles suivantes :

```
access-list 101 permit IP 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit IP 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
access-list 101 permit IP 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
access-list 101 permit IP 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
```

Comme toutes les règles ne contiennent que des adresses sources et destination, nous pouvons visualiser l'ACL comme un ensemble de quatre rectangles, avec les coordonnées récapitulées au tableau 19.1.

Tableau 19.1 Exemple de règles ACL

Règle	Première IP SRC	Dernière IP SRC	Première IP DST	Dernière IP DST
1	10.0.0.0	10.255.255.255	10.0.0.0	10.255.255.255
2	47.4.0.0	47.7.255.255	47.0.0.0	47.255.255.255
3	47.7.6.0	47.7.6.255	47.7.6.0	47.7.6.255
4	47.0.0.0	47.255.255.255	47.4.0.0	47.5.255.255

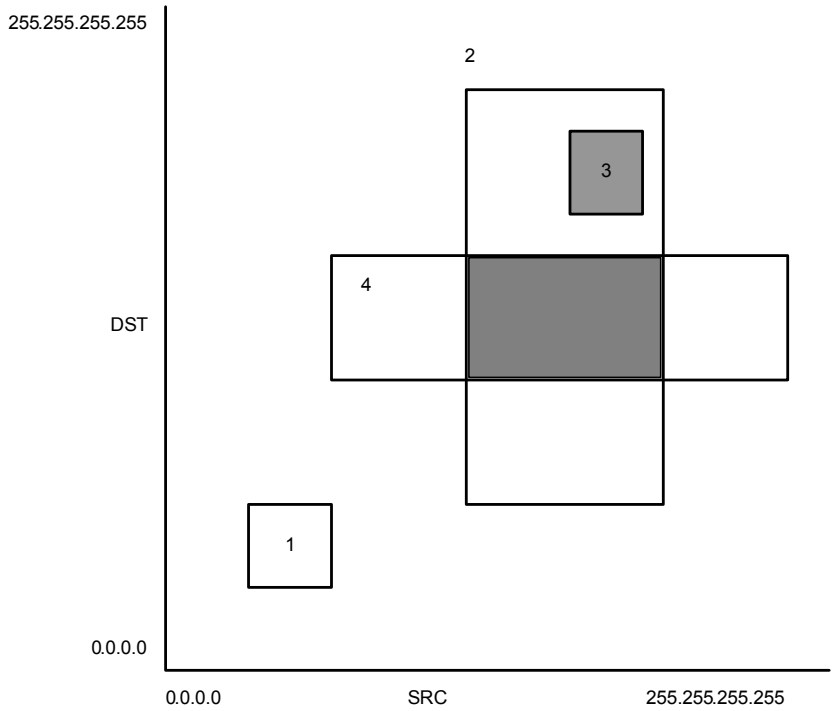
Pour une bonne compréhension visuelle, ces quatre rectangles sont illustrés à la figure 19.2.

Bien que les proportions ne soient pas respectées, on voit immédiatement que la règle 1 est totalement indépendante de toutes les autres. En revanche, la règle 3 est un sous-ensemble propre de la règle 2, qui intercepte d'ailleurs la règle 4.

Le calcul de l'intersection entre la règle 2 et la règle 4 donne le rectangle foncé, avec les adresses sources variant de 47.4.0.0 à 47.7.255.255 et les adresses destination variant de 47.4.0.0 à 47.5.255.255.

Figure 19.2

Intersections géométriques des lignes d'une ACL



Pour interpréter correctement l'incohérence, il faut se référer aux permissions associées aux règles 2 et 4. Comme il s'agit de permit dans les deux cas, nous en concluons que ces règles sont redondantes à l'intersection calculée. La règle 3 est totalement redondante avec la règle 2. Nous pouvons donc la supprimer.

Si la règle 2 avait une permission deny, la règle 3 serait clairement incohérente, puisqu'elle permettrait un trafic ayant été auparavant refusé.

L'invocation de VACL sur l'exemple ci-dessus est donnée dans la transcription suivante :

```
margot$ vacl -a ./exemple1.txt
[2] access-list 102 permit ip 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
[3] access-list 102 permit ip 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
*** redundancy [3] < [2]
[2] access-list 102 permit ip 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
[4] access-list 102 permit ip 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
*** redundancy [4] * [2] = permit ip 47.4.0.0 0.3.255.255 47.4.0.0 0.1.255.255
```

En revanche, si nous avons l'ACL suivante :

```
access-list 101 permit icmp 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit tcp 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
access-list 101 permit udp 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
access-list 101 permit udp 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
```

l'invocation de VACL donne :

```
margot$ vacl -a ./exemple2.txt
margot$
```

Il n'y a plus de redondance ni d'inconsistance détectée. D'autres exemples sont disponibles sur le site de référence de l'ouvrage.

Analyse de configuration par patron

Il arrive fréquemment qu'une analyse sémantique de configuration ne soit pas appropriée. En revanche, une analyse syntaxique, comme la vérification de conformité sur un patron de configuration standard défini par des expressions régulières, est souvent pertinente. De plus, la structure syntaxique du corpus à analyser est quelques fois complexe ; un parcours syntaxique avec les outils standards (YACC, AWK, PERL, etc.) est fastidieuse et délicate. Le modèle AWK applique ainsi chaque ligne lue à toutes les règles de haut en bas, si bien que l'implémentation d'une syntaxe relativement simple requiert des variables de gestion d'état.

Rappelons qu'une expression régulière est un modèle de texte constitué de caractères ordinaires (par exemple les lettres de a à z) et de caractères spéciaux, appelés *métacaractères*. Le modèle décrit une ou plusieurs chaînes à mettre en correspondance lors d'une recherche effectuée sur un texte.

L'objectif de cet outil est d'exprimer un patron relativement complexe afin de répondre, par exemple, aux conformités suivantes :

- Le paramètre HOSTNAME est conforme au standard de nommage, exprimé par une expression régulière.
- L'interface LOOPBACK99 est définie, et ses sous-paramètres sont conformes à la politique de sécurité réseau. En particulier, l'adresse IP de l'interface est routée uniquement sur l'interface de gestion.
- Le contrôle d'accès BACKBONE existe et est conforme à la politique de sécurité réseau.
- Toutes les interfaces ETHERNET ont leur sous-paramètre IPREDIRECT désactivé conformément à la politique de sécurité réseau.

Bien qu'il soit possible d'écrire un script AWK ou un analyseur syntaxique (typiquement généré par YACC), ceux-ci sont peu souples, et leur adaptation à un nouveau patron peut se révéler complexe. Tous les ingénieurs ne sont pas nécessairement programmeurs, mais ils sont certainement capables d'écrire une expression régulière après une courte formation..

L'outil HAWK (Hervé's AWK) est un analyseur syntaxique qui permet d'exprimer des patrons modélisant des lignes de configuration. Ces lignes peuvent être des expressions régulières, et il doit être possible de les structurer avec les opérateurs classiques d'une expression régulière, à savoir la conjonction, la disjonction et la fermeture transitive.

Il est également possible de définir du code exécutable pour implémenter des vérifications sémantiques.

HAWK permet d'exprimer un patron comme une expression régulière, dans laquelle chaque élément est également une expression régulière. Un moteur tel que celui de HAWK permet de parcourir un fichier d'entrée et de rapporter toutes les lignes non conformes au patron.

Le source HAWK a environ 12 000 lignes de code C.

Le patron HAWK

HAWK permet d'exprimer un patron comme une suite de blocs (declare, begin, success, failure) et un pattern entre le bloc BEGIN et le bloc SUCCESS.

Un patron classique en HAWK s'écrit de la manière suivante :

```
DECL {
 déclaration1
 etc.
}

BEGIN {
 intruction1
 etc.
}

PATTERN


SUCCESS {
 intruction1
 etc.
}

FAILURE {
 intruction1
 etc.
}
```

Le bloc DECL contient la déclaration des variables (string ou entière), des tableaux (string ou entière) et des macros (instruction ou pattern). La notion de fonction n'existe pas en HAWK, et il n'y a donc pas de récursivité possible.

Le bloc BEGIN contient une suite d'instructions qui seront exécutées avant la validation du patron.

Le PATTERN entre les blocs DECLARE et BEGIN correspond au template de conformité que HAWK valide.

Le bloc SUCCESS contient des instructions qui seront exécutées si le pattern a complètement matché tout l'input.

Le bloc FAILURE contient des instructions qui seront exécutées si le pattern n'a pas pu être matché au complet sur l'input, c'est-à-dire si le moteur ne peut plus trouver un pattern pour une ligne d'input.

Les deux derniers blocs sont exécutés à la fin du processus et sont mutuellement exclusifs.

Les instructions HAWK

Comme le langage AWK, HAWK permet d'ajouter des instructions dans les blocs BEGIN, dans le PATTERN HAWK *via* des accolades { … } et dans les blocs SUCCESS et FAILURE. La liste des instructions possibles est riche et adopte le modèle du langage C.

HAWK permet de définir deux types de macros dans le bloc DECL : des macros de pattern ainsi que des macros action.

Des types de variables ou tableaux peuvent être définies, telles que variables de type chaîne de caractères, entier ou tableau. Comme AWK, HAWK implémente des tableaux associatifs dont l'index est une chaîne de caractères :

```
DECL {
  str this_line;
  int this_line_number;
  str line[];
}
```

Le langage HAWK implémente des instructions de condition, de boucle, etc.

L'instruction *if (expression) then statement else statement* permet de définir une condition de branchement :

```
DECL {
  str this_line;
}

if (this_line == "line") then
  {..}
else
  { ..}
```

L'instruction *forall (value-identifier = array-identifier[index-identifier]) statement* permet de définir une condition de bouclage sur l'ensemble des valeurs continues dans un tableau donné :

```
DECL {
  str this_line,line_index;
}

forall(this_line=line[line_index])  {
   ...
}
```

Plusieurs fonctions sont prédéfinies, notamment les suivantes :

- *str* field(*n*, *s*, *fs*) : renvoie le *n*-ième champ de la chaîne de caractères s séparé par fs.
- *str* FILENAME : renvoie le nom du fichier qui est lu.
- *int* length(*s*) : renvoie la longueur de la chaîne s.

Le pattern HAWK

Le pattern HAWK, correspondant au template de conformité que HAWK valide, peut être constitué d'une simple expression régulière de base, telle que la suivante, qui indique qu'il doit y avoir une ligne vty :

```
 [eis]:^line vty.*$
```

Une telle expression est constituée d'un flag entre crochets suivi du délimiteur deux-points et de l'expression régulière.

Le PATTERN peut aussi être constitué d'une suite d'expressions régulières telle que la suivante, qui indique qu'il doit y avoir une ligne vty et une access-class :

```
[fx] :line vty.*$
[fx] : access-class 99 in
```

HAWK permet de définir un bloc d'expressions régulières à l'aide de parenthèses, qui peuvent abriter des patterns ou des expressions mathématiques, comme la suivante, qui indique qu'il doit y avoir 0 ou plusieurs lignes vty et access-class associée).

```
*(
   [fx] :line vty.*$
   [fx] : access-class 10 in
)
```

HAWK permet d'utiliser les opérateurs de structure suivant à appliquer aux patterns :

- La disjonction logique pattern1 | pattern2 | pattern3, qui se traduit, dans le langage logique ou mathématique par l'opérateur OU logique. Dans l'exemple suivant, on réalise une disjonction entre deux expressions régulières (on a soit une access-class, soit un transport input none) :

```
   :^ access-class 99 in$
|
   :^ transport input none$
```

- La conjonction pattern1 & pattern2 & pattern3, qui se traduit, dans le langage logique ou mathématique, par l'opérateur ET logique. Dans l'exemple suivant, on réalise une conjonction entre deux expressions régulières (on a une access-class et un transport input telnet) :

```
    :^ transport input
&
    :^ transport input telnet
```

- La négation ! pattern, qui se traduit, dans le langage logique ou mathématique, par l'opérateur NON logique. Dans l'exemple suivant, on réalise une conjonction entre deux expressions régulières (on a un transport input mais pas de transport input telnet) :

```
    :^ transport input
&
    !:^ transport input telnet
```

Le moteur HAWK

Un moteur tel que celui de HAWK permet de parcourir un fichier d'entrée et de rapporter toutes les lignes non conformes au patron. HAWK est purement non déterministe, *a contrario* de ses prédécesseurs CDIFF et HDIFF, qui sont purement déterministes.

Il peut y avoir ainsi plusieurs patterns évalués en parallèle, mais aussi plusieurs actions. Bien que le résultat soit garanti, l'ordre d'évaluation, par exemple d'un pattern *versus* un autre dans une conjonction, n'est pas garanti.

HAWK implémente des algorithmes efficaces de la théorie des automates. En conséquence, il a le même comportement asymptotique que GREP, et le temps d'exécution de l'outil est proportionnel au nombre de lignes de son input. Le temps total est dominé par le traitement d'ouverture et de lecture des fichiers, et le temps consommé par le parcours est négligeable.

En situation réelle sur un PC récent (FreeBSD 6.2, release-p8), environ 70 000 fichiers de configuration (36 millions de lignes de configuration) sont validés en environ quatre minutes avec un patron HAWK contenant quinze règles.

Prise en main

Soit la ligne de commande suivante :

```
Hawk -f fichier-patron fichier-configuration
```

- -f fichier-patron spécifie le fichier contenant le patron.
- fichier-configuration spécifie le ou les fichiers à vérifier.

Exemple

Le langage IOS de Cisco possède une caractéristique intéressante : bien qu'une ligne AUX soit associée à un port matériel (de type VT100 ou modem), il est possible d'ouvrir une connexion distante sur le port AUX sous certaines conditions. En conséquence, il est important de configurer correctement les routeurs de manière à ce que ce phénomène ne soit pas possible.

Un fichier de configuration doit avoir au moins un port AUX configuré, et tous les ports doivent se conformer aux critères suivants :

- transport input *x*, où *x* n'est ni telnet ni all.

- access-group 99 in, où l'ACL 99 contient l'unique règle deny any.

Une première approche consiste à écrire un patron HAWK modélisant une configuration correcte :

```
DECL {
          macro line_aux  :^line aux [0-9]
                          ;
          macro indent    :^[ ]
                          ;
          macro transport :^ transport input
                          ;
          macro access    :^ access-group in
                          ;
}

# Patron
* ! line_aux
+ (
    line_aux
          (
              * (indent & ! transport)
              (
                  transport
               &
                  ! [ex] :^ transport input (telnet|all)
              )
              * (indent & ! transport)
          &
              * (indent & ! access)
              [fx] : access-group in 99
              * (indent & ! access)

          )
          * ! line_aux
)

SUCCESS {
          printf("PASS %s\n", FILENAME);
}

FAILURE {
    printf("FAIL %s\n", FILENAME);
}
```

Ce patron n'est pas très utile puisqu'il ne comporte aucune référence à un bloc AUX en cas d'erreur.

Une deuxième version adopte la stratégie inverse. Le patron HAWK recherche alors les erreurs de conformité selon les critères suivants :

- Une configuration n'a aucun port AUX configuré ou a au moins un port configuré avec une des erreurs suivantes :
  - n'a pas transport input ou a transport input telnet ou transport input all.
  - n'a pas access-group in ou a access-group *x* in avec $x \neq 99$.

```
DECL {
          macro line_aux  :^line aux [0-9]
                          ;
    macro t_23 [ex] : transport input (telnet|all)
                          ;
    macro a_99 [fx] : access-group in 99
                          ;
          int n_aux ;
          str aux[], this_aux, i ;
          int lineno[] ;
    str t_in[], a_in[], ok = "ok" ;
}

# Patron
* ! line_aux
+ (
    line_aux
    {
        aux[++n_aux] = LINE; lineno[n_aux] = LINENO;
        t_in[n_aux] = " no transport input";
              a_in[n_aux] = " no access-group in";
         }

         * (
        :^[ ]
      |
        (
            ( ! t_23 & :^ transport input
            ) { t_in[n_aux] = ok; }
        |
            t_23 { t_in[n_aux] = LINE; }
        )
      |
        (
            ( ! a_99 & :^ access-group in
            ) { a_in[n_aux] = LINE; }
        |
            a_99 { a_in[n_aux] = ok; }
        )
    )

    * ! line_aux
)

SUCCESS {
```

```
    if (n_aux == 0) printf("%s: no line aux\n", FILENAME);

    forall (this_aux = aux[i])
    {
        if (t_in[i] != ok)
            printf("%s %ld (%s): %s\n",
                    FILENAME, lineno[i], this_aux, t_in[i]);

         if (a_in[i] != ok)
             printf("%s %ld (%s): %s\n",
                    FILENAME, lineno[i], this_aux, a_in[i]);
    }
    printf("PASS %s\n", FILENAME);
}
```

Le patron ci-dessus détecte les fichiers non conformes. Pour toutes les erreurs détectées, le HAWK imprime le nom du fichier de configuration, le numéro de ligne où l'erreur a été trouvée, ainsi que le texte non conforme.

Analyse de configuration d'équipements réseau Juniper

Cette section traite des limitations inhérentes au développement d'un outil dédié à un problème complexe, sur un objet complexe. En un certain sens, l'outil Juniper est un échec en termes de simplicité et de souplesse, puisqu'il est complexe et lourd. De plus, toute modification, même mineure, requiert une programmation fastidieuse. Cependant, force est de constater qu'il est efficace.

Les équipements réseau Juniper ont un modèle de configuration hiérarchique, calqué sur la notion de bloc imbriqué, comme dans beaucoup de langages de programmation modernes. Heureusement, le constructeur publie la syntaxe complète dans la documentation disponible sur son site Internet.

Par opposition à l'outil HDIFF décrit à la section précédente, nous avons besoin d'un outil nous permettant de valider sémantiquement des configurations, et non seulement syntaxiquement. Les tests sémantiques peuvent être variés et ne sont pas définis à l'avance.

Dans ce cas précis, le parcours de configuration doit se faire avec une technique dite « dirigée par la syntaxe » (syntax-directed). En effet, un même mot-clé pouvant se retrouver dans plusieurs sections différentes, l'analyseur doit pouvoir discriminer totalement son contexte d'utilisation. De plus, la configuration Juniper étant en format libre, un paramètre peut occuper plusieurs lignes, entrecoupées de commentaires. Ici encore, l'outil HDIFF, avec son approche ligne par ligne, n'est pas approprié.

Conception de l'outil

L'outil générique Juniper est construit à l'aide du générateur de compilateur YACC, et son analyseur lexical est un automate généré par LEX. Les tests sémantiques sont écrits en C.

Plus précisément, les tests sémantiques sont codés directement dans le fichier YACC, associés à une règle syntaxique donnée. Si nous ajoutons un nouveau test sémantique, il nous faut mettre à jour les règles syntaxiques YACC et coder le test voulu.

La partie la plus fastidieuse est certainement le parcours syntaxique, bien qu'il soit possible de court-circuiter les règles non pertinentes.

Prise en main

Soit la ligne de commande suivante :

```
juniper fichier-configuration
```

fichier-configuration spécifie le fichier contenant la configuration de l'équipement Juniper.

Exemple

Nous désirons valider la liste des comptes d'accès configurés sur les équipements réseau Juniper. Ces comptes et leurs caractéristiques sont définis par le patron de configuration suivant :

```
system {
          login {

    /* Définit un profil d'utilisateur */
              class <identifiant> {
                  permissions [ <liste d'identifiants> ] ;
              }

    /* Définit un utilisateur identifié par son UID
      et de profile spécifié par sa classe */
              user <identifiant> {
                  uid <entier> ;
                  class <identifiant> ;
              }
          }
}
```

L'outil d'analyse doit ignorer les lexèmes apparaissant dans d'autres contextes, de façon que nous ne soyons pas obligés d'implémenter la totalité de la syntaxe Juniper.

Pour chaque utilisateur, l'analyseur doit vérifier l'unicité de son identifiant et de son UID. La classe d'un compte utilisateur peut être de type super-usager ou lecture seulement. Ces classes sont caractérisées par des identifiants de permission (la permission admin octroie les superpouvoirs et ne doit pas apparaître dans la classe non privilégiée). Au moins un compte de chaque classe doit être configuré.

Nous utilisons deux structures de fouilles (typiquement des ensembles) distinctes pour les classes et les utilisateurs. L'ensemble "usagers" pourrait avantageusement être doublement indexé par l'identifiant d'utilisateur et par l'UID.

Pour cet exemple, un pseudo-code YACC est préférable à une transcription littérale (les lexèmes littéraux sont en gras) :

```
/* dans le contexte system { login { ... } } */

class identifiant { permission [ liste_identifiants ] ; }
{
        entrer l'identifiant de classe dans l'ensemble "classes".
        si l'entrée existait déjà alors erreur « classe non unique »

        si le mot clef admin est dans la liste des permissions alors
         étiqueter cette classe "super-pouvoirs"
        sinon
         étiqueter cette classe « lecture seulement »
}

/* on suppose que les classes sont définies avant les usagers */
user identifiant { uid numero ; class identifiant ; }
{
  si l'ensemble "usagers" contient déjà une entrée
            de même "uid <entier>" alors
          erreur « uid non unique »
        si l'ensemble "usagers" contient déjà une entrée
         de même "user <identifiant>" alors
          erreur « usager non unique »

        si l'identifiant de classe est dans l'ensemble "classes" alors
         copier localement l'étiquette de la classe
        sinon
         erreur « classe non définie »

  entrer l'identifiant usager, son uid et son étiquette de classe
     dans l'ensemble "usagers"
}

/* en sortant du contexte login { ... } */
system { login { ... }
{
  rw ← 0 ; ro ← 0 ;

        pour tous les enregistrements dans l'arbre "usagers" faire
         si étiquette == "super-pouvoirs" alors
     incrémenter rw
         sinon
           incrémenter ro
        fin pour

        si rw == 0 alors erreur « pas de super-usager »
        si ro == 0 alors erreur « pas de compte lecture seulement »
}
```

Un fichier de configuration Juniper contient les déclarations de classes et d'utilisateurs suivants :

```
system {
  login {
   class c1 {
     permissions [admin];
   }

   class c2 {
     permissions [admin firewall];
   }

   class c3 {
     permissions [firewall];
   }


   user cedric {
     uid 1000;
     class c1;
   }

   user denis {
     uid 1000;
     class c2;
   }

   user margot {
     uid 1001;
     class c4;
   }
  }
}
```

Si nous exécutons le programme Juniper sur ce fichier de configuration, nous obtenons le résultat suivant :

```
margot$ ./juniper demo.conf
demo.conf
    utilisateur 'margot': classe 'c4' non déclarée.
    utilisateur 'cedric': uid '1000' dupliqué.
    Pas d'utilisateur read-only.
```

Le programme Juniper se restreint aujourd'hui à l'analyse sémantique des déclarations de classes et d'utilisateurs, mais il peut être étendu à d'autres vérifications syntaxiques et sémantiques.

Corrélation d'événements avec RTA

Le logiciel présenté dans cette section se distingue des précédents par deux caractéristiques. Premièrement, il n'est pas orienté sur l'analyse offline de configurations ou de topologies, mais plutôt sur une analyse d'événements en temps réel. Deuxièmement, ce n'est pas un logiciel complet, prêt à l'emploi, mais une sorte de « démonstration » de concept.

Trois exemples illustrent comment implémenter à peu de frais un moteur d'alertes de sécurité, et le lecteur est invité à implémenter la détection d'événements particuliers à son propre contexte. En ce sens, le logiciel correspond plus à un *toolkit*.

Historiquement, la détection en temps réel d'événements de sécurité a été pour la première fois mis en œuvre avec le Firewall Toolkit de Marcus Ranum (FWTK). Tous les proxies du FWTK envoyaient leurs événements sur le canal syslog, et la détection se faisait par une commande de type grep filtrant les événements sans danger. Un fichier contenait les expressions régulières décrivant ces syslogs, et le principe sous-jacent était que tous les syslogs résultants étaient par définition digne d'être rapportés. Notons que cette technique est dite hors contexte, le programme grep n'ayant pas de mémoire sur les événements passés et étant donc incapable de corréler deux événements.

Dans cette section, nous montrons comment plusieurs événements indépendants sans sévérité particulière peuvent être corrélés pour détecter un événement dangereux. L'architecture de base est simple et permet de faire facilement remonter les syslogs vers un serveur en charge de l'analyse de ces événements. Le trafic est basé sur le protocole classique SYSLOG, utilisant le protocole de transport UDP port 514.

Les aspects d'ingénierie réseau, comme la confidentialité, le chemin de transit ou la volumétrie, ne sont pas appréhendés. Gardons en mémoire que le protocole UDP n'a pas de livraison fiable et que le trafic est en clair (donc non confidentiel). Typiquement, les équipements réseau enverront le trafic syslog vers un serveur grâce à une ligne de configuration. Pour un système de type Unix, le fichier /etc/syslog.conf aura une ligne :

```
*.*         @ip
```

Elle spécifie que les événements de toutes les sévérités pour toutes les facilités seront envoyés vers l'adresse IP du serveur d'analyse et d'alerte. De son côté, ce système doit pouvoir accepter les trames syslog externes. Enfin, [il[qui ?] injecte le trafic syslog dans le programme RTA (Real-Time Analysis) par le biais de la ligne suivante dans le fichier /etc/syslog.conf :

```
*.*         |/var/log/rta
```

On notera que le fichier /var/log/rta est un pipeline FIFO sur disque. Finalement, le système injecte un marqueur dans le flux syslog toutes les minutes.

Conception de RTA

Le programme RTA est maintenant capable de lire toutes les trames syslog des équipements réseau. Le moteur principal réalise tout simplement une boucle de lecture et appelle des fonctions de détection et d'analyse.

Remarquons que le langage de programmation choisi (langage C) permet l'utilisation de mémoires secondaires, organisées avec une structure adéquate permettant la corrélation de l'événement courant avec un événement passé.

Première fonction de corrélation, RTA1

RTA1 a pour but de détecter les connexions extranet qualifiées de pirates. Pour notre exemple, nous considérons trois passerelles extranet gw1.tdbsr.com gw2.tdbsr.com et gw3.tdbsr.com suivantes. Ces trois passerelles génèrent par exemple les deux syslogs suivants respectivement au début et à la fin d'une connexion extranet :

```
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=X User=Y
Logout Vpn=extranet SrcIp=X User=Y
```

Les champs X et Y correspondent à une adresse IP et au compte utilisé pour l'authentification de la connexion. Ainsi, si un même compte est utilisé à partir de deux adresses IP différentes, ce compte est considéré comme piraté et une alarme doit être émise.

La corrélation d'une connexion est faite avec toutes les autres connexions extranet en cours. Pour ce faire, on conserve une liste doublement cousue, dans laquelle chaque élément contient le descriptif d'une connexion. La double couture permet de retirer efficacement une connexion de la liste lors de sa terminaison.

À la réception d'un syslog LoginSuceeded, la liste est parcourue pour trouver une autre connexion courante sur le même compte. Si cette connexion existe, une alarme est créée. Un descriptif de la connexion est ensuite inséré dans la liste. À la réception d'un syslog Logout, le descriptif de la connexion est retiré de la liste.

Deuxième fonction de corrélation, RTA2

La fonction RTA2 a pour but de détecter l'origine d'une attaque en force brute sur un mot de passe dans une session SSH. Comme le protocole SSH construit un tunnel chiffré, la technique classique par un équipement de détection d'intrusion (réseau) est inopérante. On doit donc baser la détection sur les syslogs générés par le processus SSHD.

Considérons que nous ayons les événements suivants :

```
sshd[29717]: Wrong password given for user 'denis'.
sshd[29821]: password authentication failed. Login to account toto not allowed
➥ or account non-existent.
```

Ces événements ne donnent aucune information sur l'origine de la connexion SSH. Par ailleurs, le processus SSHD fournit l'origine de chaque connexion par les événements générés lorsqu'une connexion est initialisée, donc avant toute tentative d'authentification :

```
sshd[29717]: INFO: DNS lookup for "ip1" gives <hostname>.
sshd[29821]: WARNING: DNS lookup failed for "ip1".
```

Il s'agit donc de conserver le contexte de chaque connexion SSH, avec comme information l'adresse IP d'origine. Le contexte est identifié uniquement par le PPID (le premier champ à gauche). Les contextes peuvent être gérés par toute structure de données permettant l'insertion, la destruction et une fouille rapide. Si le nombre de contexte n'est pas prohibitif, alors une liste doublement cousue est suffisante.

À la réception d'un nouveau contexte, un nouvel élément est inséré dans la liste avec l'adresse IP d'origine. À la réception d'un syslog de type authentification ratée, une fouille dans la liste permet de retrouver l'origine de la connexion, et une alerte est générée avec cette information. Un compteur peut également être prévu pour lever une alarme après un nombre déterminé de tentatives. Enfin, un contexte est détruit à la terminaison de la session SSH :

```
sshd[29761]: Remote host disconnected.
```

Troisième fonction de corrélation, RTA3

La fonction RTA3 supervise l'intégrité de certains systèmes critiques. Le critère d'intégrité choisi est simple : si aucun syslog n'est reçu d'un de ces systèmes depuis deux minutes, une alerte doit être générée. Pour que cette détection fonctionne, ces systèmes doivent générer eux-mêmes un marqueur syslog à chaque minute. Ce troisième exemple illustre une corrélation d'événements basée sur un critère temporel et non contextuel.

Une table contient donc les noms ou les adresses IP des systèmes sous surveillance. La table est triée, permettant une fouille dichotomique rapide. À chaque système correspond le tampon horodateur du dernier événement reçu.

À chaque trame syslog, la fonction effectue une fouille dans la table. Si le système n'est pas dans la table, la fonction se termine immédiatement, car ce système n'est pas sous surveillance. Si le système est dans la table, le tampon horodateur de l'événement courant est comparé à celui conservé dans la structure. Si la différence de temps est trop grande, une alerte est générée. Dans tous les cas, la structure de données est mise à jour avec la valeur du tampon.

Prise en main

Soit la ligne de commande suivante :

```
rta -f fichier-fifo
```

-f fichier-syslog spécifie le fichier input des syslogs. En utilisation normale, le fichier syslog est un pipeline FIFO.

Nous en verrons deux exemples de mise en œuvre au chapitre [20 ou 21 ?].

Gestion de graphes avec GRAPH

L'implémentation d'algorithmes de graphes est un exercice de programmation classique dans les cursus universitaires. De plus, des outils adéquats se trouvent facilement sur Internet.

Pourtant, nombre de ces outils ne sont que des jouets ou, au mieux, des exercices de style. Les autres produits sont souvent des programmes avec interface graphique ou des collections d'algorithmes sophistiqués implémentés sur une structure de données peu adaptée à notre contexte.

La bibliothèque GRAPH a été développée suivant des critères précis. Elle doit pouvoir manipuler des graphes dirigés ou non dirigés, de taille non bornée *a priori*. Elle doit être simple, avec une interface souple, et privilégier l'optimisation en temps par rapport à l'optimisation en espace. La bibliothèque doit être portable sur l'ensemble des plates-formes Unix, avec un minimum de dépendances.

La bibliothèque GRAPH et son interface shell nécessitent des connaissances de base en théorie des graphes, ainsi que de l'environnement Unix.

Cet outil est écrit en moins de 2 500 lignes de code C.

Conception de l'outil

Les choix fondamentaux, sous-jacents au développement de la bibliothèque GRAPH sont des conséquences directes de son contexte particulier d'utilisation : l'analyse de réseaux de routeurs interconnectés par des liens locaux (LAN) ou par des liens globaux (WAN).

Il est donc assumé que les graphes ne sont pas denses :

- Une structure de données non opaque donne une visibilité sur tous les champs internes, facilitant d'autant l'évolution de la bibliothèque avec des fonctions *ad hoc*. Cette approche évite de gérer une interface fonctionnelle fastidieuse.
- Une grande mémoire centrale est disponible sur beaucoup de plates-formes et permet de se libérer des contraintes mémoire. En fait, dans le compromis classique espace/temps, la bibliothèque GRAPH privilégie la rapidité d'exécution. La bibliothèque consomme $O(m^2 + n)$ octets pour un graphe de m nœuds et n arcs.
- Les algorithmes implémentés sont efficaces asymptotiquement. Par exemple, les plus courts chemins sont calculés avec des itérations sur l'algorithme de Dijkstra, en utilisant une liste prioritaire comme structure de données, plutôt que l'algorithme plus simple de fermeture transitive de Floyd-Warshall. En revanche, la clarté du codage est privilégiée par rapport aux optimisations de codage. Ainsi, les tableaux sont-ils référencés par indexation et non par arithmétique sur des pointeurs. Ce dernier point est relativement mineur, car la bibliothèque peut être compilée avec toutes les options d'optimisation.
- Plusieurs représentations internes d'un même graphe sont implémentées. Ce critère découle directement des précédents. Parce que différents algorithmes sont idéalement supportés par différentes structures de données, nous avons choisi de représenter un

graphe en parallèle par les trois structures classiques : listes d'adjacences, listes d'incidences et matrice d'incidences. Ces trois structures, bien que gourmandes en mémoire, permettent d'accéder à n'importe quel nœud ou arc en temps constant. La version 2 de la bibliothèque permet de choisir quelles représentations internes seront utilisées.

- Une programmation défensive, bien que plus lourde, assure la consistance des structures de données et permet de détecter les fautes classiques, comme une mauvaise récupération de mémoire *(memory leak)*. Le code est donc prévu pour inclure des vérifications internes sous forme d'assertions, et ce dans toutes les fonctions. L'effet de bord évident est de taxer l'efficacité temps. Bien sûr, l'utilisation fréquente de la bibliothèque permet de détecter les erreurs et de stabiliser le code. La bibliothèque peut être recompilée avec les assertions désactivées.
- La facilité d'utilisation est importante dans ce contexte. Il y a non seulement des primitives de bas niveau (ajouter un nœud, ajouter un arc, fouille en profondeur, etc.), mais également plusieurs fonctions de haut niveau (plus courts chemins, points d'articulation, composantes connexes, etc.). Ce principe est sous-jacent à une bibliothèque totalement dynamique, avec une structure de données autocroissante ; l'utilisateur ou le programmeur peut, sans y être obligé, précalculer le nombre maximal de nœuds ou d'arcs.
- Une interface utilisateur primitive mais complète est incluse, ce qui permet d'invoquer toutes les fonctions de la bibliothèque à partir d'une commande shell. À la section suivante, les exemples sont donnés avec cette interface.
- La généralité est importante. La bibliothèque peut supporter librement des arcs dirigés ou non dirigés dans un même graphe. En fait, la non-direction est une caractéristique d'arc, et non de graphe. De plus, il est possible de définir plusieurs arcs, avec leurs paramètres respectifs, entre une même paire de nœuds.
- Un traitement offline est préféré à une approche incrémentale online ; c'est pourquoi il n'y a pas de primitive de destruction ni de modification de nœuds et d'arcs. La bibliothèque assume que le graphe initial sera l'objet final du traitement, sans possibilité de mise à jour. Ainsi, l'interface shell lit un graphe puis applique directement les algorithmes choisis.

Prise en main

Soit la ligne de commande suivante :

```
graph -acfpsv fichier
```

- -c calcule et imprime toutes les composantes connexes.
- -f calcule et imprime tous les points d'articulation. Cette option est utile pour trouver les nœuds critiques *(single points of failure)*.
- -F calcule et imprime les points d'articulation comme avec l'option -f et imprime les partitions de nœuds autour de chaque point d'articulation. Essentiellement, cette option calcule comment un graphe serait déconnecté sans les nœuds critiques.

- -p calcule et imprime les chemins entre chaque paire de nœuds, détecte les chemins asymétriques et imprime le coût associé à chaque chemin.
- -s calcule et imprime toutes les composantes fortement connexes.
- -v imprime le graphe lui-même.
- fichier spécifie le fichier contenant la description du graphe.

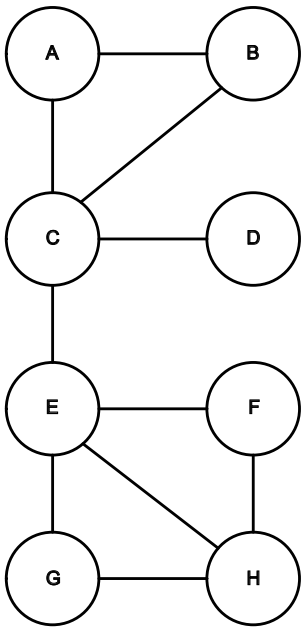
Exemples

Le premier exemple illustre le calcul des nœuds et arcs critiques *(single points of failure)*.

Dans le graphe non dirigé illustré à la figure 19.3, les huit nœuds représentent des équipements réseau, et les dix arcs les liens de communication entre ces équipements. Les arcs sont tous de coût zéro.

Figure 19.3

Exemple de graphe (graphe1)



Le fichier graphe1.dat représente ce graphe :

```
# graphe1.dat

U  A B
U  A C
U  B C
U  C D
U  C E
U  E F
U  E G
```

```
U  E H
U  F H
U  G H
```

Le fichier est constitué de 12 lignes, la première étant un commentaire, la deuxième ligne étant vide, et les 10 lignes suivantes encodant un seul arc. Un arc non dirigé est caractérisé par la lettre U, suivie des étiquettes des deux nœuds reliés par cet arc. Il est possible d'ajouter un quatrième champ spécifiant le coût associé à l'arc, lequel est 1 par défaut. Il n'est pas nécessaire de prédéclarer les nœuds, bien que ce soit possible. Ce dernier cas est utile pour associer un coût à un nœud.

L'invocation de l'outil GRAPH calcule les composantes connectées ainsi que les points d'articulation avec les partitions associées :

```
margot/19.graph$ graph -cF graphe1.dat
graphe1.dat: 8 nodes, 10 edges, 4512 bytes
connected component (8 nodes):
{ A B C D E F G H }
articulation point: C
node partition: { A B }
node partition: { D }
node partition: { E F G H }
articulation point: E
node partition: { A B C D }
node partition: { F G H }
```

Pour obtenir l'ensemble des éléments critiques, liens de communication compris, nous utilisons l'astuce suivante : remplacer chaque arc par un nœud artificiel, de poids identique à l'arc. Ce nouveau nœud est connecté par des arcs de poids nul.

Ainsi, l'exemple devient le graphe illustré à la figure 19.4.

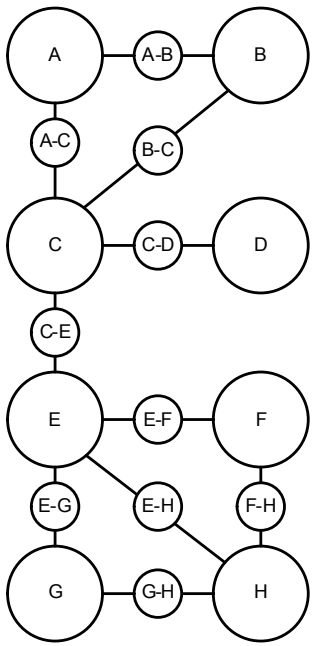
Le fichier graphe2.dat donne l'encodage de ce graphe :

```
# graphe2.dat

N  A-B 1
N  A-C 1
N  B-C 1
N  C-D 1
N  C-E 1
N  E-F 1
N  E-G 1
N  E-H 1
N  F-H 1
N  G-H 1

U  A A-B 0
U  B A-B 0
U  A A-C 0
U  C A-C 0
```

```
U  B B-C 0
U  C B-C 0
U  C C-D 0
U  D C-D 0
U  C C-E 0
U  E C-E 0
U  E E-F 0
U  F E-F 0
U  E E-G 0
U  G E-G 0
U  E E-H 0
U  H E-H 0
U  F F-H 0
U  H F-H 0
U  G G-H 0
U  H G-H 0
```

Figure 19.4

Exemple de graphe (graphe2)



Ici, les nœuds artificiels sont déclarés avec un coût de 1, correspondant au coût des arcs de l'exemple 1. Les arcs sont déclarés avec un coût nul.

L'invocation de l'outil GRAPH donne l'ensemble de tous les éléments critiques :

```
margot/19.graph$ graph -F graphe2.dat
graphe2.dat: 18 nodes, 20 edges, 23656 bytes
articulation point: C-D[1]
```

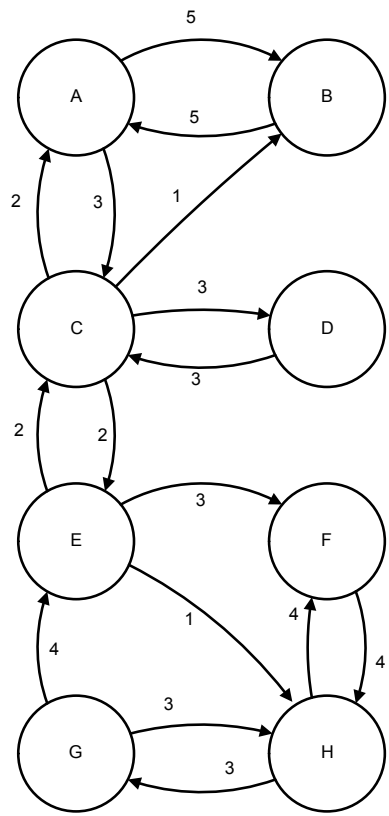
```
node partition: { A-B[1] A-C[1] B-C[1] C-E[1] E-F[1] E-G[1] E-H[1] F-H[1] G-H[1]
➥ A B C E F G H }
node partition: { D }
articulation point: C-E[1]
node partition: { A-B[1] A-C[1] B-C[1] C-D[1] A B C D }
node partition: { E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] E F G H }
articulation point: C
node partition: { A-B[1] A-C[1] B-C[1] A B }
node partition: { C-D[1] D }
node partition: { C-E[1] E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] E F G H }
articulation point: E
node partition: { A-B[1] A-C[1] B-C[1] C-D[1] C-E[1] A B C D }
node partition: { E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] F G H }
```

Nous avons donc les nœuds critiques C et E comme précédemment, puis les arcs critiques C-D et C-E.

Dans l'exemple illustré à la figure 19.5, le poids des arcs modélise le coût de chaque route. Le graphe est dirigé, afin de pouvoir représenter des routes de coûts asymétriques, et certains arcs sont absents à cause d'un contrôle d'accès sur le nœud bloquant tout trafic.

Figure 19.5

Exemple de graphe (graphe3)



Le fichier graphe3.dat encode cet exemple. Dans le premier champ, la lettre D spécifie un arc dirigé :

```
# graphe3.dat
#
D  A B  5
D  A C  3
D  B A  5
D  C A  2
D  C B  1
D  C D  3
D  C E  2
D  D C  3
D  E C  2
D  E F  3
D  E H  1
D  F H  4
D  G E  4
D  G H  3
D  H F  4
D  H G  3
```

Si nous souhaitons vérifier qu'il y a un chemin entre toutes les paires de nœuds (il n'y a qu'une seule composante fortement connexe), nous obtenons la liste des chemins asymétriques.

Cette liste est partiellement donnée dans la transcription suivante :

```
margot$ graph -asp graphe3.dat
graphe3.dat: 8 nodes, 16 edges, 4488 bytes
strongly connected component (8 nodes):
{ A B C D E F H G }
paths:
A <-> B
asymetric paths:
cost: 4 < (A -> C)[3] (C -> B) >
cost: 5 < (B -> A)[5] >
A <-> C
asymetric paths:
cost: 3 < (A -> C)[3] >
cost: 2 < (C -> A)[2] >
A <-> D
asymetric paths:
cost: 6 < (A -> C)[3] (C -> D)[3] >
cost: 5 < (D -> C)[3] (C -> A)[2] >
A <-> E
asymetric paths:
cost: 5 < (A -> C)[3] (C -> E)[2] >
cost: 4 < (E -> C)[2] (C -> A)[2] >
A <-> H
```

```
asymetric paths:
cost: 6 < (A -> C)[3] (C -> E)[2] (E -> H) >
cost: 11 < (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> A)[2] >
A <-> G
asymetric paths:
cost: 9 < (A -> C)[3] (C -> E)[2] (E -> H) (H -> G)[3] >
cost: 8 < (G -> E)[4] (E -> C)[2] (C -> A)[2] >
B <-> C
asymetric paths:
cost: 8 < (B -> A)[5] (A -> C)[3] >
cost: 1 < (C -> B) >
B <-> D
asymetric paths:
cost: 11 < (B -> A)[5] (A -> C)[3] (C -> D)[3] >
cost: 4 < (D -> C)[3] (C -> B) >
B <-> E
asymetric paths:
cost: 10 < (B -> A)[5] (A -> C)[3] (C -> E)[2] >
cost: 3 < (E -> C)[2] (C -> B) >
B <-> F
asymetric paths:
cost: 13 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> F)[3] >
cost: 14 < (F -> H)[4] (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> B) >
B <-> H
asymetric paths:
cost: 11 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> H) >
cost: 10 < (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> B) >
B <-> G
asymetric paths:
cost: 14 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> H) (H -> G)[3] >
cost: 7 < (G -> E)[4] (E -> C)[2] (C -> B) >
cost: 7 < (G -> H)[3] (H -> F)[4] >
```

En situation réelle, nous extrayons les informations de connectivité à partir des fichiers de configuration des routeurs composant un réseau. Ces informations sont typiquement extraites de la configuration des interfaces LAN et WAN. L'adresse IP du sous-réseau ainsi que son masque permettent de reconstruire la connectivité de tout un réseau, sous l'hypothèse que le plan d'adressage ne contienne pas de doublon ; les adresses doivent être uniques dans le réseau.

Dans certains cas particuliers, il est possible d'extraire de la configuration le coût associé à une route. Ce coût est alors reflété sur l'arc modélisant le lien de communication. Le coût des routes induit un graphe dirigé, mais symétrique sur la connectivité.

Un autre cas intéressant est l'analyse inter-VPN sur une structure MPLS. En effet, la connectivité étant asymétrique, un VPN peut être exporté et indépendamment importé. Dans ce cas particulier, le coût est constant. Ainsi, nous pouvons extraire de l'ensemble des configurations tous les paramètres d'import et d'export pour ensuite construire le graphe dirigé de connectivité inter-VPN.

Ce graphe n'est pas nécessairement symétrique. Une fouille en profondeur donne le périmètre d'un VPN, c'est-à-dire l'ensemble des routeurs accessibles à partir d'un seul point. Finalement, le calcul des routes asymétriques met en lumière les erreurs d'importation et d'exportation des tables de routage.

Sur un ordinateur personnel moyen de gamme, la bibliothèque GRAPH permet le traitement routinier de réseaux de plusieurs milliers de nœuds.

Calcul de risque avec BAYES

Nous avons vu au chapitre 18 comment évaluer les impacts et une valeur de risque par une modélisation probabiliste. Un arbre probabiliste associé à divers scénarios d'événements est engendré à partir d'une description des vulnérabilités et des règles de propagation. Cet arbre probabiliste peut être potentiellement très important et se recalcule chaque fois que le profil de vulnérabilité d'un équipement réseau change.

L'outil BAYES calcule les probabilités des impacts à partir d'un arbre probabiliste. Il calcule aussi la valeur de risque à partir de valeurs de conséquences associées aux impacts.

Cet outil est écrit en moins de 400 lignes de code C.

Conception de l'outil

BAYES calcule un arbre probabiliste à partir de données regroupées dans quatre fichiers distincts. Le premier fichier contient les vulnérabilités, le second les règles de propagation, le troisième les probabilités associées aux impacts (ou feuilles de l'arbre) et le dernier les conséquences associées aux impacts.

Comme nous le verrons dans les exemples ci-après, la formalisation des tests, des règles de propagation et des impacts est très ouverte et peut s'adapter à différents environnements.

La construction de l'arbre probabiliste par BAYES suit les règles décrites au chapitre 18, ainsi que les règles spécifiques suivantes :

- La racine de l'arbre probabiliste est le nœud "0".
- L'impact "0" doit être compris comme la feuille indiquant qu'il n'y a pas d'impact.
- Il y a une distribution équiprobable des probabilités, hormis celle de l'impact "0", associées aux nœuds issus du nœud racine.
- Un nœud final possède une feuille avec l'impact et la probabilité associée au nœud, mais aussi une feuille avec l'impact "0" prenant la valeur de probabilité restante.

Nous allons détailler les formats des quatre fichiers de données nécessaires à la construction de notre arbre probabiliste.

Fichier de vulnérabilités

Le fichier de vulnérabilités contient trois éléments par ligne. La première ligne correspond à la première vulnérabilité, et ainsi de suite. Les vulnérabilités apparaissent dans le fichier par le numéro de test auquel elles sont rattachées.

Les éléments sont les suivants :

- Le test (un test peut détecter une ou plusieurs vulnérabilités). Il s'agit d'un entier positif.
- L'objet sur lequel a été détectée la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- L'impact associé au test de sécurité. Il s'agit d'un entier positif.

Le fichier suivant contient deux vulnérabilités associées à deux tests différents ayant des impacts différents :

```
margot$ cat exemple1.txt
1 A 1
2 A 2
```

Fichier de propagations

Le fichier de propagations contient un ensemble de règles de propagation.

Chaque règle de propagation contient les éléments suivants :

- Un test (un test peut détecter une ou plusieurs vulnérabilités). Il s'agit d'un entier positif.
- L'objet sur lequel a été détectée la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- L'objet sur lequel peut se propager la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- La liste des tests avec lesquels nous déterminons la propagation de la vulnérabilité (un test peut détecter une ou plusieurs vulnérabilités).

Le fichier suivant contient trois règles de propagation. La première indique que le nœud racine peut propager les vulnérabilités associées aux tests 1 et 2 sur l'objet A. La deuxième indique que les vulnérabilités associées au test 1 peuvent se propager de l'objet A vers l'objet A sur les vulnérabilités détectées par les tests 1 et 2. La dernière indique que les vulnérabilités associées au test 2 peuvent se propager de l'objet A vers l'objet A sur les vulnérabilités détectées par les tests 1 et 2.

```
margot$ cat exemple1.rule
0 A A 1 2
1 A A 1 2
2 A A 1 2
```

Fichier de probabilités

Le fichier de probabilités contient les probabilités associées aux impacts (un élément par ligne). La première ligne correspond à la probabilité du premier impact, et ainsi de suite.

Les contraintes associées au fichier de probabilités sont les suivantes :

- La probabilité est un réel compris entre 0 et 1.
- La somme des probabilités de toutes les lignes n'est pas nécessairement égale à 1.

- La contrainte est que, pour toute ligne i, la somme des probabilités de l'impact 0 et de l'impact i est inférieure ou égale à 1.

Le fichier suivant indique trois valeurs de probabilités (associées aux impacts 0, 1, 2) :

```
margot$ cat exemple1.proba
0.1
0.3
0.3
```

Fichier de conséquences

Le fichier de conséquences contient les conséquences associées aux impacts (un élément par ligne). La première ligne correspond à la valeur de conséquence du premier impact, et ainsi de suite.

La conséquence est un réel positif. Cette valeur est arbitraire et sans contrainte.

Le fichier suivant indique trois valeurs de conséquences (associés aux impacts 0, 1, 2) :

```
margot$ cat exemple1.cons
0
10
50
```

Prise en main

Soit la ligne de commande suivante :

```
bayes fichier-vulnérabilités fichier-propagation fichier-conséquences profondeur
```

- fichier-vulnérabilités spécifie le fichier contenant les vulnérabilités.
- fichier-propagations spécifie le fichier contenant les règles de propagation.
- fichier-probabilités spécifie le fichier contenant les valeurs des probabilités associées aux impacts.
- fichier-conséquences spécifie le fichier contenant les valeurs des conséquences associées aux impacts.
- profondeur est un nombre entier spécifiant la profondeur maximale d'exploration de l'arbre probabiliste.

Exemple élémentaire

Définissons notre premier arbre probabiliste avec les fichiers de données suivants :

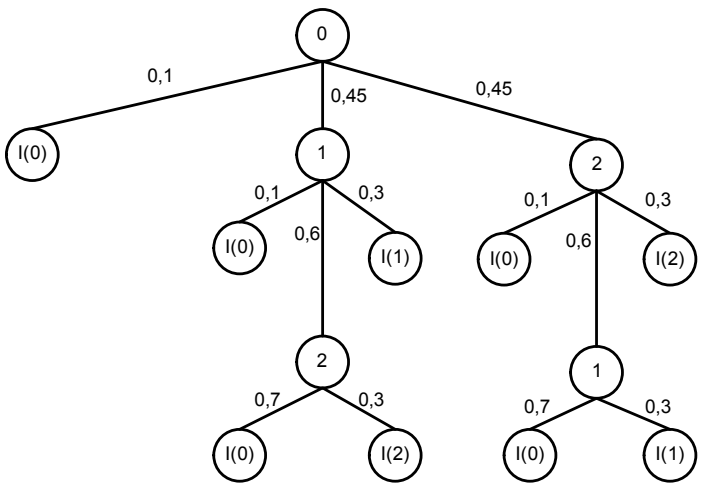
```
margot$ cat exemple1.rule
0 A A 1 2
1 A A 1 2
2 A A 1 2
```

```
margot$ cat exemple1.proba
0.1
0.3
0.3

margot$ more exemple1.cons
0
10
50

margot$ cat exemple1.txt
1 A 1
2 A 2
```

La figure 19.6 illustre l'arbre probabiliste associé.

Figure 19.6

Arbre probabiliste élémentaire



Le calcul des impacts 0, 1 et 2 est donné par les formules suivantes :

$I(0) = 0,1 + 0,7 \times 0,6 \times 0,45 + 0,1 \times 0,45 + 0,7 \times 0,6 \times 0,45 + 0,1 \times 0,45 = 0,568$

$I(1) = 0.3 \times 0,45 + 0,3 \times 0,6 \times 0,45 = 0,216$

$I(2) = 0,3 \times 0,6 \times 0,45 + 0,3 \times 0,45 = 0,216$

Si nous exécutons le programme BAYES sur ces fichiers de données, nous retrouvons ces mêmes valeurs :

```
margot$ gmake exemple1
normalise exemple1.rule exemple1.proba exemple1.txt exemple1.cons
bayes exemple1.txt.ref.dat[1234] 10

------------------------------------------------------------------------
nb_vulnerabilités par test :
```

```
    test:0 | nb de vulnérabilités:1 | impact:0
    test:1 | nb de vulnérabilités:1 | impact:1
    test:2 | nb de vulnérabilités:1 | impact:2
nb_impacts (3) =  0  1  2
nb_probabilités (3 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (3 impacts) =  0.000000e+00  1.000000e+01  5.000000e+01
------------------------------------------------------------------------
distribution des probabilités (impacts): 5.680000e-01  2.160000e-01
➥ 2.160000e-01 / somme=1.000000e+00
risque : 1.296000e+01
nombre de noeuds de l'arbre : 5.000000e+00
nombres de feuilles par impact: 5.000000e+00  2.000000e+00
➥ 2.000000e+00 / somme=9.000000e+00
profondeur de l'arbre : 2
------------------------------------------------------------------------
```

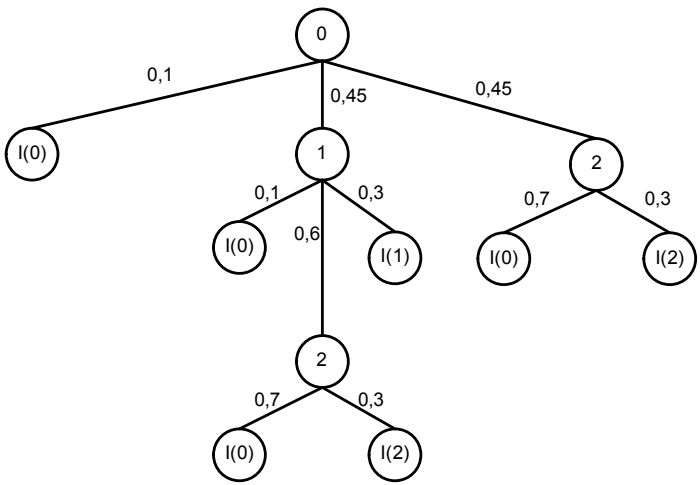
Modifions l'arbre probabiliste par le fichier de propagation :

```
margot$ cat exemple2.rule
0 A A 1 2
1 A A 1 2
```

La figure 19.7 illustre l'arbre probabiliste associé.

Figure 19.7

Arbre probabiliste élémentaire modifié



Le calcul des impacts 0, 1 et 2 est donné par les formules suivantes :

$$I(0) = 0{,}1 + 0{,}7 \times 0{,}6 \times 0{,}45 + 0{,}1 \times 0{,}45 + 0{,}7 \times 0{,}6 \times 0{,}45 + 0{,}7 \times 0{,}45 = 0{,}649$$

$$I(1) = 0{,}3 \times 0{,}45 = 0{,}135$$

$$I(2) = 0{,}3 \times 0{,}6 \times 0{,}45 + 0{,}3 \times 0{,}45 = 0{,}216$$

Si nous exécutons le programme BAYES sur ces fichiers de données, nous retrouvons ces mêmes valeurs :

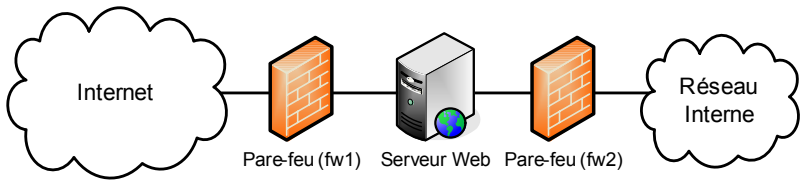
```
margot/19.bayes$ make exemple2
normalise exemple2.rule exemple2.proba exemple2.txt exemple2.cons
bayes exemple2.txt.ref.dat[1234] 10

-------------------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:1 | nb de vulnérabilités:1 | impact:1
    test:2 | nb de vulnérabilités:1 | impact:2
nb_impacts (3) =  0  1  2
nb_probabilités (3 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (3 impacts) =  0.000000e+00  1.000000e+01  5.000000e+01
-------------------------------------------------------------------------
distribution des probabilités (impacts): 6.490000e-01  1.350000e-01
➥ 2.160000e-01 / somme=1.000000e+00
risque : 1.215000e+01
nombre de noeuds de l'arbre : 4.000000e+00
nombres de feuilles par impact: 4.000000e+00  1.000000e+00
➥ 2.000000e+00 / somme=7.000000e+00
profondeur de l'arbre : 2
-------------------------------------------------------------------------
```

Exemple réseau

Prenons maintenant le cas de figure d'un réseau constitué de deux pare-feu (fw1, fw2) et d'un serveur Web (web), comme illustré à la figure 19.8.

Figure 19.8

Architecture sécurisée d'accès à un réseau



La modélisation pour notre calcul de risque est la suivante : pour chaque objet (fw1, fw2, web), il y a trois tests possibles pouvant référencer une ou plusieurs vulnérabilités ; de plus, il y a trois impacts possibles (faible, moyen, fort), comme le résume le tableau 19.2.

Dans ce modèle, si nous tenons compte de la topologie réseau et du fait que les attaques viennent uniquement de l'extérieur, les règles de propagation sont les suivantes :

```
margot$ cat exemple3.rule
0 fw1 fw1 1 2 3
0 web web 4 5 6
1 fw1 fw1 1
2 fw1 fw1 2
3 fw1 fw1 3
```

```
4 web web 4
5 web web 5
6 web web 6
7 fw2 fw2 7
8 fw2 fw2 8
9 fw2 fw2 9
3 fw1 web 4 5 6
6 web fw1 1 2 3
6 web fw2 7 8 9
9 fw2 web 4 5 6
```

Tableau 19.2 Répartition des tests et des impacts de l'exemple 3

Objet	Test	Impact
fw1	1	1 (faible)
	2	2 (moyen)
	3	3 (fort)
web	4	1 (faible)
	5	2 (moyen)
	6	3 (fort)
fw2	7	1 (faible)
	8	2 (moyen)
	9	3 (fort)

Si nous prenons différents fichiers de vulnérabilités *(voir tableau 19.3)* et que nous exécutions le programme BAYES pour chacun de ces fichiers, nous obtenons la distribution des probabilités des impacts ainsi que l'évolution dans le temps du risque illustrée par les courbes des figures 19.9 et 19.10.

Tableau 19.3 Fichiers de vulnérabilités de l'exemple réseau

Fichier 1 exemple 3	Fichier 2 exemple 31	Fichier 3 exemple 32	Fichier 4 exemple 33	Fichier 5 exemple 34	Fichier 6 exemple 35
1 fw1 1 1 fw1 1 1 fw1 1 1 fw1 1 1 fw1 1	1 fw1 1 1 fw1 1 3 fw1 3 3 fw1 3 5 web 2 6 web 3	1 fw1 1 1 fw1 1 3 fw1 3 3 fw1 3 6 web 3 6 web 3 9 fw2 3	1 fw1 1 1 fw1 1 3 fw1 3 3 fw1 3 9 fw2 3 9 fw2 3 9 fw2 3	1 fw1 1 1 fw1 1 3 fw1 3 3 fw1 3 5 web 2 5 web 2 9 fw2 3	1 fw1 1 1 fw1 1 4 web 1 4 web 1 4 web 1 4 web 1 7 fw2 1 7 fw2 1

Cette simulation prend en compte les fichiers de probabilités et de conséquences suivantes :

```
margot$ cat exemple3.proba
0.1
0.3
0.3
0.8

margot$ cat exemple3.cons
0
10
50
100
```

L'exécution de BAYES donne alors les résultats suivants :

```
margot$ gmake exemple3 | grep "distribution"
distribution des probabilités (impacts): 3.774880e-01  6.225120e-01  0.000000e+00
➥ 0.000000e+00 / somme=1.000000e+00
distribution des probabilités (impacts): 4.208056e-01  1.479440e-01  4.828375e-02
➥ 3.829667e-01 / somme=1.000000e+00
distribution des probabilités (impacts): 3.272332e-01  1.493427e-01  0.000000e+00
➥ 5.234241e-01 / somme=1.000000e+00
distribution des probabilités (impacts): 3.880000e-01  2.160000e-01  0.000000e+00
➥ 3.960000e-01 / somme=1.000000e+00
distribution des probabilités (impacts): 4.539200e-01  1.440000e-01  1.540800e-01
➥ 2.480000e-01 / somme=1.000000e+00
distribution des probabilités (impacts): 4.643200e-01  5.356800e-01  0.000000e+00
➥ 0.000000e+00 / somme=1.000000e+00

margot$ gmake exemple3 | grep "risque"
risque : 6.225120e+00
risque : 4.219029e+01
risque : 5.383584e+01
risque : 4.176000e+01
risque : 3.394400e+01
risque : 5.356800e+00
```

À partir de ces données, la figure 19.9 illustre la distribution dans le temps des probabilités associées aux impacts. Remarquons notamment que l'impact est le plus fort pour le fichier numéro 3.

La figure 19.10 illustre l'évolution dans le temps du calcul du risque. Ces valeurs de risques traduisent à nouveau un risque important pour le fichier 3 associé aux vulnérabilités.

Différentes valeurs de probabilités, de règles de propagation ou de conséquences peuvent être choisies, l'important étant de valider le comportement et la pertinence des mesures de sécurité réalisées.

Figure 19.9

Distribution des probabilités associées aux impacts

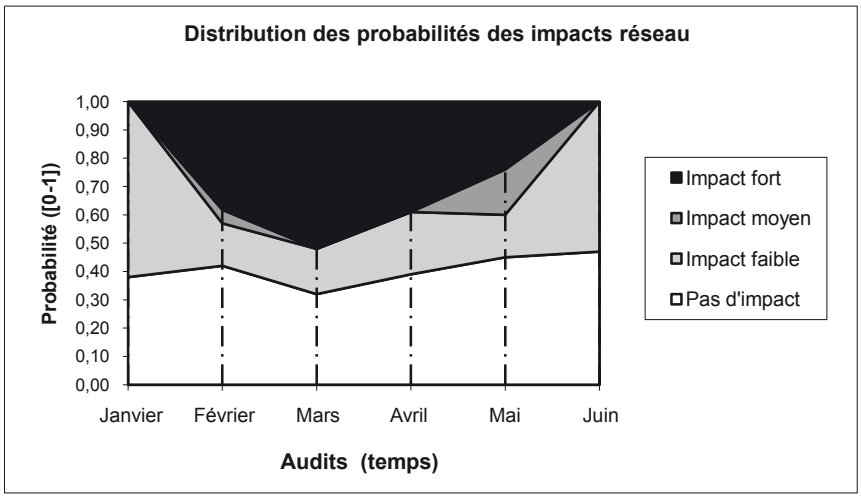
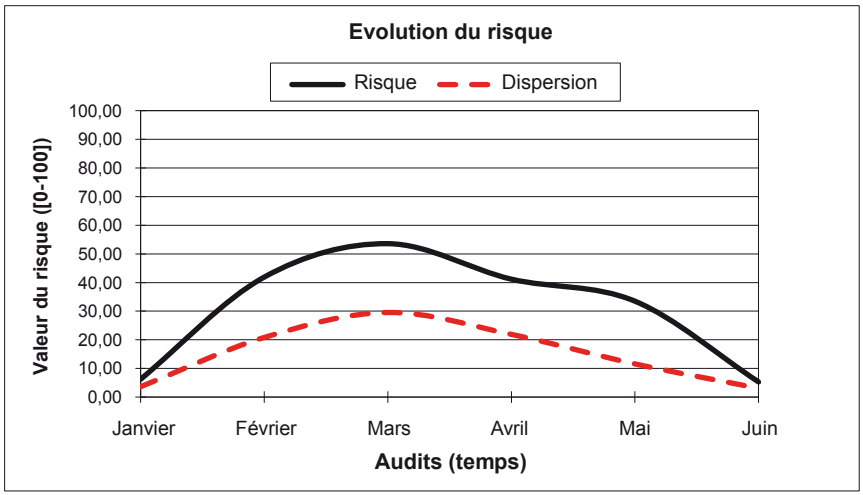


Figure 19.10

Évolution du risque dans le temps



Exemple de réduction combinatoire

Sachant que l'objectif est de calculer les probabilités associées aux impacts réseau, il est 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 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 possible grâce à une simplification combinatoire fondée sur la répétition de k objets parmi n objets lors de la construction des branches de l'arbre probabiliste.

Cette réduction permet de déterminer des sous-branches identiques dans notre arbre probabiliste et ainsi de ne pas les construire. Bien que cette approche permette de prendre en compte un nombre important de vulnérabilités détectées, nous perdons cependant de la granularité dans les règles de propagation en considérant des groupes de vulnérabilités plutôt que des vulnérabilités.

Si nous considérons, dans l'exemple suivant, une vulnérabilité par test (exemple 5) ou groupons plusieurs vulnérabilités par test (exemple 4), nous obtenons les mêmes résultats pour le calcul des probabilités des impacts et du risque résultant :

```
margot$ cat exemple4.rule
0 A A 1 2
1 A A 1 2
2 A A 2

margot$ cat exemple4.txt
1 A 1
1 A 1
2 A 2
2 A 2

margot$ cat exemple5.rule
0 A A 1 2 3 4
1 A A 1 2 3 4
2 A A 1 2 3 4
3 A A 3 4
4 A A 3 4

margot$ cat exemple5.txt
1 A 1
2 A 1
3 A 2
4 A 2
```

Exécutons maintenant l'exemple 4 :

```
margot$ gmake exemple4
normalise exemple4.rule exemple4.proba exemple4.txt exemple4.cons
bayes exemple4.txt.ref.dat[1234] 10

-------------------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:1 | nb de vulnérabilités:2 | impact:1
    test:2 | nb de vulnérabilités:2 | impact:2
nb_impacts (3) =  0  1  2
nb_probabilités (3 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (3 impacts) =  0.000000e+00  1.000000e+01  5.000000e+01
-------------------------------------------------------------------------
distribution des probabilités (impacts): 5.096800e-01  1.620000e-01
```

```
➥ 3.283200e-01 / somme=1.000000e+00
risque : 1.803600e+01
nombre de noeuds de l'arbre : 9.000000e+00
nombres de feuilles par impact: 9.000000e+00  2.000000e+00
➥ 6.000000e+00 / somme=1.700000e+01
profondeur de l'arbre : 4
------------------------------------------------------------------------
```

Exécutons maintenant l'exemple 5 :

```
margot$ gmake exemple5
normalise exemple5.rule exemple5.proba exemple5.txt exemple5.cons
bayes exemple5.txt.ref.dat[1234] 10

------------------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:1 | nb de vulnérabilités:1 | impact:1
    test:2 | nb de vulnérabilités:1 | impact:1
    test:3 | nb de vulnérabilités:1 | impact:2
    test:4 | nb de vulnérabilités:1 | impact:2
nb_impacts (3) =  0  1  1
nb_probabilités (3 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (3 impacts) =  0.000000e+00  1.000000e+01  5.000000e+01
------------------------------------------------------------------------
distribution des probabilités (impacts): 5.096800e-01  1.620000e-01
➥ 3.283200e-01 / somme=1.000000e+00
risque : 1.803600e+01
nombre de noeuds de l'arbre : 2.500000e+01
nombres de feuilles par impact: 2.500000e+01  4.000000e+00
➥ 2.000000e+01 / somme=4.900000e+01
profondeur de l'arbre : 4
------------------------------------------------------------------------
```

Nous constatons que le nombre d'éléments parcourus est nettement inférieur dans l'exemple 4 (9 éléments parcourus) que dans l'exemple 5 (25 éléments parcourus) pour un même résultat.

Précisons que le groupement de vulnérabilités pour un même test doit faire l'objet d'un accord entre les experts de sécurité.

Limites

Le problème de la construction de notre arbre probabiliste est lié, dans le pire des cas, au problème d'énumération des permutations d'un ensemble de *N* éléments. N'oublions pas qu'il s'agit d'un problème à la combinatoire explosive. D'autres limitations viennent du fait que nous ne considérons pas des analyses d'incertitude, de sensibilité, etc. Cette restriction est liée ici à la complexité intrinsèque d'un réseau.

Le facteur principal dans le contrôle de l'explosion combinatoire est la définition des règles de propagation. En effet, si toutes les vulnérabilités peuvent engendrer toutes les

vulnérabilités, le calcul devient impossible à réaliser en pratique, car il demanderait un temps d'exécution prohibitif.

Rappelons tout de même que l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées.

En résumé

L'automatisation du contrôle des configurations d'équipements réseau est nécessaire lorsque les configurations deviennent complexes ou qu'elles concernent un grand nombre d'équipements.

Nous avons montré dans ce chapitre qu'un effort raisonnable de développement permettait d'obtenir un retour sur investissement significatif. Le développement de petits outils ad hoc est rapide et peu coûteux en comparaison de l'investissement dans de gros outils commerciaux souvent dispendieux. Nous avons de surcroît la liberté complète de nos choix fondamentaux et pouvons obtenir des résultats immédiats.

Nous ne prétendons pas pour autant que tous les outils commerciaux peuvent être remplacés par des outils maison, mais plutôt que les deux se complètent harmonieusement.

Les deux chapitres suivants décrivent l'évolution d'une entreprise, de ses besoins en télécommunications, des différentes solutions mises en œuvre et des besoins en sécurité associés. Ces chapitres mettent en pratique de manière concrète nos outils maison.

20

RadioVoie, du réseau initial au premier gros contrat

Au travers de l'évolution d'une entreprise fictive, RadioVoie, et de son réseau, sont illustrés dans ce chapitre et le suivant à la fois les besoins de sécurité et les politiques correspondant à chaque étape du développement de l'entreprise.

RadioVoie a développé une technologie révolutionnaire permettant la transmission de la voix par ondes radio *via* un appareil de très petite taille mais à très longue portée, fonctionnant selon le principe du talkie-walkie.

Grâce au démarchage de son fondateur, cette société a décroché un premier contrat visant à équiper de ses appareils le personnel urbain de la ville de Paris afin que celui-ci puisse être en contact permanent avec le centre de contrôle. RadioVoie a breveté sa technologie révolutionnaire pour tous les pays. La fabrication de la solution est sous-traitée à une société de montage liée par une clause de confidentialité.

Pour des raisons financières, RadioVoie sous-traite la production de ses équipements à une tierce partie.

Cette étude de cas reprend de façon pratique les différentes étapes d'une bonne stratégie de sécurité. Pour chaque évolution du réseau de RadioVoie, nous détaillons l'analyse des besoins, la définition de la politique de sécurité, les solutions techniques et les contrôles de sécurité, les risques couverts et non couverts par la solution technique proposée, ainsi que l'établissement d'un tableau de bord de sécurité.

Le premier réseau RadioVoie

Composée initialement d'une seule personne, la société RadioVoie embauche du personnel et s'installe dans des locaux afin de faire face aux projets en cours.

Les enjeux sont importants pour l'entreprise, qui doit s'assurer de disposer d'un réseau disponible et dont les accès sont protégés. Son premier système d'information doit supporter ses services internes (comptabilité, commercial, technique, secrétariat), qui comptent une demi-douzaine de personnes.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- Le réseau de données interne doit avoir un bon débit pour permettre le partage de ressources (serveurs de fichiers, imprimantes, etc.) de stations de travail hétérogènes (Windows, Macintosh, Unix, etc.).
- Cantonné au siège de l'entreprise situé à Paris, le réseau doit séparer logiquement le département recherche et développement.
- Les spécifications techniques de la technologie révolutionnaire doivent être protégées.

Étude de risques

Comme il s'agit d'un réseau interne, sans ouverture vers d'autres réseaux externes, les risques ou attaques de sécurité sont limités à des menaces internes. Ces dernières sont rapidement détectables puisque le personnel est en nombre limité. La menace interne ne doit toutefois pas être sous-estimée, car c'est un risque récurrent et potentiellement considérable pour toute entreprise.

La disponibilité du réseau n'est pas non plus vitale pour l'entreprise, qui peut intervenir directement sur les systèmes en cas de problème.

Par contre, la confidentialité des données relatives aux brevets et projets est essentielle, de même que les accès aux ressources avec des droits d'accès limités. La sauvegarde de ses données est non moins vitale pour l'entreprise.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, la politique de sécurité réseau minimale de RadioVoie est constituée des règles suivantes :

- *« Les accès aux équipements réseau de l'entreprise sont limités aux administrateurs réseau. »*
- *« Le réseau est subdivisé en deux réseaux logiquement distincts. »*
- *« Les données confidentielles de l'entreprise sont chiffrées avant d'être émises sur le réseau interne de l'entreprise. Aucune donnée confidentielle n'est émise en dehors de ce réseau interne. Par données confidentielles, il convient d'entendre non les données*

bureautiques de l'entreprise, mais les spécifications techniques concernant les brevets ainsi que celles des appareils en cours de production (plans, schémas, etc.). »

- *« Aucun réseau sans fil (Wi-Fi, etc.) n'est autorisé au sein de l'entreprise. »*

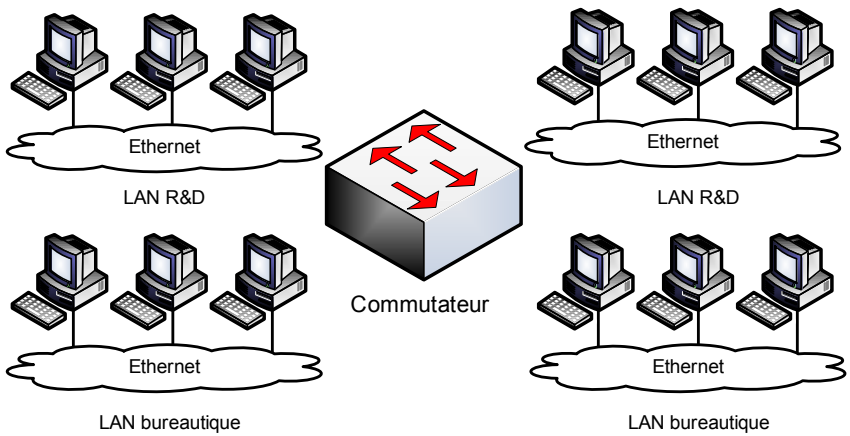
Solution de sécurité

Pour satisfaire ces besoins, RadioVoie construit un réseau Ethernet point à point 100BaseT à 100 Mbit/s. Le réseau est centralisé dans une armoire de brassage et est contrôlé par un commutateur disposant de la capacité de créer des réseaux virtuels (Virtual LAN).

Les différents réseaux locaux (bureautique, recherche et développement) sont raccordés au commutateur, comme illustré à la figure 20.1.

Figure 20.1

Le premier réseau de RadioVoie



La possibilité de créer des réseaux virtuels (configuration de VLAN) au niveau du commutateur permet au réseau local bureautique d'être séparé logiquement du réseau local recherche et développement, comme l'illustre la figure 20.2.

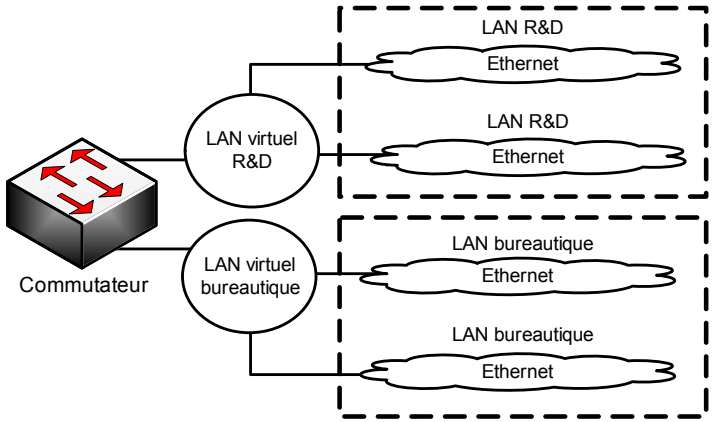
Afin de limiter les coûts de chiffrement des données confidentielles circulant sur le réseau, RadioVoie adopte une solution applicative (par programme), qui chiffre les données avant ou pendant leur transmission.

Deux scénarios de chiffrement peuvent être envisagés :

- La solution chiffre les données pendant leur transit. Le serveur et le client partagent un secret visant à garantir la confidentialité de l'information. Sans ce secret, il n'est pas possible d'accéder à l'information, ni à l'endroit où elle est stockée.
- La solution chiffre les données avant qu'elles transitent sur le réseau. L'endroit où sont stockées les données peut dans ce cas être accessible à des personnes non autorisées. La sécurité repose donc entièrement sur la qualité du secret et de l'algorithme de chiffrement.

Figure 20.2

Séparation logique des VLAN



Pour la seconde solution, une procédure de transit est nécessaire :

1. La donnée est en clair sur la machine de l'utilisateur.
2. La donnée est chiffrée efficacement avec un secret.
3. La donnée est envoyée sur le réseau.
4. La donnée en clair est détruite localement (selon la sécurité physique de la machine).
5. Le secret est communiqué de manière confidentielle au récepteur de la donnée.
6. Le récepteur de la donnée peut utiliser le processus inverse pour y accéder.

Risques réseau couverts

Si le commutateur n'est pas contrôlable à distance par l'administrateur réseau et qu'il est configuré de manière sécurisée, les risques d'attaque permettant d'accéder au commutateur tendent vers zéro. Cependant, la mise en place de contrôles au niveau du commutateur nécessite généralement la capacité de prise de contrôle à distance et peut engendrer un risque de sécurité pour le commutateur.

Pour pallier ce risque, un nouveau réseau virtuel (VLAN) doit être installé à l'endroit où sont hébergées les plates-formes de contrôle. Dans ce cas, seule une attaque visant à casser le compartimentage des réseaux virtuels peut permettre d'accéder au VLAN de supervision. Ce risque peut être réduit par la mise en place d'un contrôle d'accès au niveau MAC.

Le commutateur renforce la disponibilité du réseau, même s'il ne peut la garantir. Il reste en effet toujours un risque d'inonder le réseau de broadcast ou qu'un programme fortement consommateur de bande passante cherche à se dupliquer rapidement.

Par la commutation des paquets au niveau 2 du modèle OSI, le réseau voit sa disponibilité, sa performance et sa confidentialité renforcées. Le commutateur n'envoie vers une machine que les paquets qui lui sont destinés (normaux ou broadcast), limitant de ce fait le risque de saturation et d'écoute du réseau. Il élimine également le risque de refus de

service par rupture du moyen de communication, à la différence des technologies en bus, par exemple.

Il est fréquent que de tels commutateurs offrent la fonctionnalité de n'accepter que les paquets réseau provenant d'une adresse MAC (Media Access Control) spécifique. Une telle solution offre évidemment un contrôle plus fin des connexions.

Enfin, une fonctionnalité NAC (Network Access Control) peut être déployée sur le commutateur afin de contrôler en profondeur les systèmes qui s'y connectent en empêchant certaines machines de s'échanger des données, malgré qu'elles soient sur le même VLAN.

Risques réseau non couverts

Une attaque directe sur le commutateur par un système interne afin de pénétrer un réseau virtuel (VLAN) non autorisé est possible. Ce type d'attaque s'appuie cependant sur le principe que l'attaquant envoie des paquets avec l'adresse MAC qu'il désire écouter ou avec toutes les adresses MAC du VLAN auquel il désire accéder. La mise en place, au niveau du commutateur, de contrôles d'accès de niveau MAC fait tendre ce risque vers zéro.

Il est aussi possible d'attaquer le commutateur par le protocole IEEE 802.1q si sa configuration n'est pas verrouillée. Pour réduire ce risque, les ports autorisés à émettre/recevoir du trafic 802.1q doivent être identifiés et n'être reliés à aucune machine ou prise réseau accessible des bureaux. Les fonctions de type dynamic trunking doivent bien sûr être également désactivées.

Un refus de service est également possible sur le commutateur par l'exploitation de faiblesses. À ce niveau, seul le constructeur peut résoudre le problème, et aucune solution automatisée (au niveau des postes de travail) ne peut être mise en place.

Reste un risque de saturation d'un périphérique particulier du réseau (serveur de fichiers, etc.) ou de tout le réseau pour peu que l'attaquant dispose d'une bande passante totale (100 % du réseau) ou qu'il inonde le réseau de broadcast. Ce type d'attaque est toutefois facilement détectable, et le système responsable est rapidement retrouvé, du fait du nombre limité d'équipements.

Tableau de bord de sécurité

Après avoir défini la politique de sécurité ainsi que les solutions possibles associées, cette section détaille les principaux contrôles à mettre en place, fournit des éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le commutateur est l'élément critique du réseau. L'objectif du contrôle consiste donc à vérifier par une supervision que le commutateur est toujours actif mais aussi que les LAN virtuels sont toujours implémentés.

Pour vérifier que le commutateur est actif, une supervision à l'aide du protocole SNMP (Simple Network Management Protocol) permet de contrôler les informations offertes par la MIB (Management Information Base).

Pour vérifier que les LAN virtuels sont actifs, il suffit de récupérer régulièrement la configuration du commutateur et de l'analyser afin de s'assurer qu'elle implémente bien les VLAN désirés par le biais du VLAN d'administration. La fréquence de ce contrôle dépend des moyens mis en œuvre pour le satisfaire. S'il s'agit d'une opération manuelle, il ne faut pas espérer plus d'un contrôle par jour. Si le contrôle peut être automatisé, il devient possible d'effectuer plusieurs contrôles par jour.

La figure 20.3 illustre la création d'un VLAN dédié à la supervision du commutateur. Ce VLAN étant isolé logiquement des autres VLAN, l'administration à distance du commutateur devient acceptable.

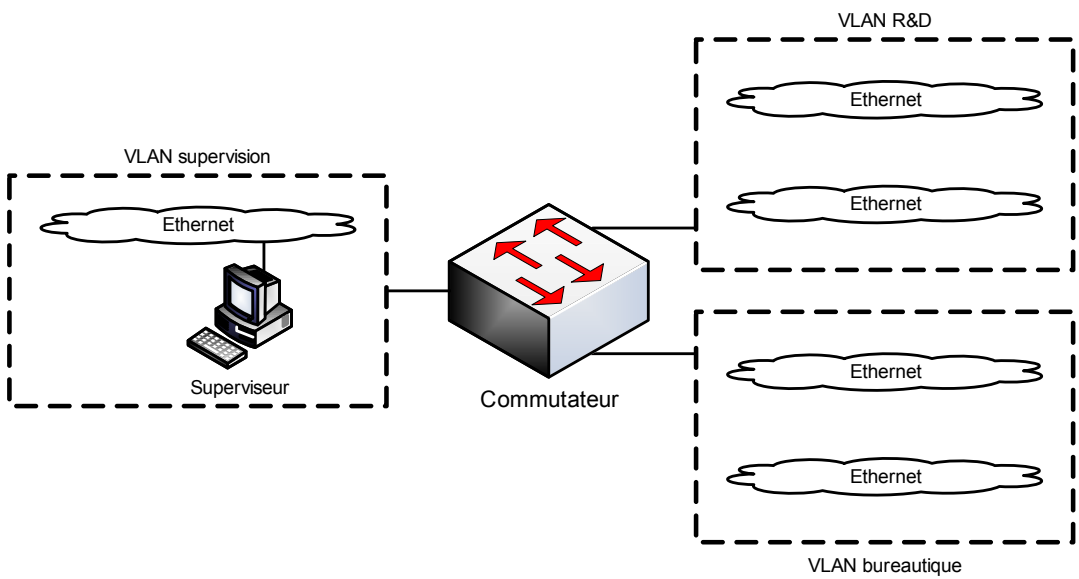


Figure 20.3

Le VLAN d'administration du réseau de RadioVoie

Mise en œuvre des outils maison

Comme indiqué à propos des contrôles de sécurité, les configurations du ou des commutateurs doivent être analysées afin de détecter toute mauvaise configuration avec le patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration d'un commutateur Catalyst de Cisco suivant :

```
margot/20.1/hawk$ cat caxt.txt
interface GigabitEthernet1/0/1
 no ip address
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport trunk allowed vlan 3,4,5
 switchport mode access
 switchport nonegotiate
!
interface GigabitEthernet1/0/2
 no ip address
 switchport mode trunk
 switchport access vlan 3
 switchport port-security
 switchport port-security maximum 1
 switchport port-security violation restrict
 switchport port-security aging time 10
 switchport port-security aging type inactivity
 switchport nonegotiate
!
interface GigabitEthernet1/0/3
 no ip address
 switchport access vlan 3
 switchport port-security
 switchport port-security maximum 1
 switchport port-security violation restrict
 switchport port-security aging time 10
 switchport port-security aging type inactivity
 switchport nonegotiate!
```

La justification des éléments de configuration est fournie à la partie III de l'ouvrage relative à la configuration des équipements réseau.

Pour analyser ces configurations, nous utilisons notre outil HAWK, avec le patron de sécurité suivant :

```
margot/20.1/hawk$ cat cat.tp
DECL {
   str this_interface, interface[], i, strict_state;
   int nb_int, check_trunk[], check_access[];
}

BEGIN {
   nb_int = 0;
   strict_state = "no interface";
}

# verification de la presence d'une interface en mode strict
*!:^interface
+ (
     :^interface
```

```
     {
       interface[++nb_int] = LINE;
       strict_state = "no ip address";
     }

     # verification de la premiere ligne en mode strict
     :^ no ip address

     # verification des lignes suivantes en mode laxiste
     *(
         [e]:^ switchport( mode)? trunk
         { check_trunk[nb_int] = 1; }
         |
         [e]:^ switchport( mode)? access
         { check_access[nb_int] = 1; }
         |
         :^[ ]
     )

     # poursuite de l'analyse
     *!:^interface
)

SUCCESS {
forall(this_interface = interface[i])
   {
      if (check_trunk[i] != 0 && check_access[i] != 0)
          printf("%s;%s;(erreur) configuration en mode trunk et access\n",
          ➥ FILENAME, this_interface);
   }
}

FAILURE {
printf("%s;%s (line %d);(erreur) configuration non conforme au mode strict;%s\n",
➥ FILENAME, LINE, LINENO, strict_state);
}
```

Si nous exécutons le programme HAWK sur une configuration Catalyst (cat.txt) qui ne respecte pas le patron de sécurité sur les contrôles en mode laxiste, nous obtenons les résultats suivants :

```
margot/20.1/hawk$ hawk -f ./cat.tp ./cat.txt
./cat.txt;interface GigabitEthernet1/0/1;(erreur) configuration en mode trunk
➥ et access
./cat.txt;interface GigabitEthernet1/0/2;(erreur) configuration en mode trunk
➥ et access
```

Cet exemple illustre en première erreur qu'une interface GigabitEthernet1/0/1 est définie en mode trunk et access et, en deuxième erreur, qu'une seconde interface GigabitEthernet1/0/2 est définie en mode trunk et access.

Si nous exécutons le programme HAWK sur une autre configuration Catalyst (cat1.txt) qui ne respecte pas le patron de sécurité sur les contrôles en mode strict, nous obtenons les résultats suivants :

```
margot/20.1/hawk$ hawk -f ./cat.tp ./cat1.txt
./cat1.txt; switchport trunk encapsulation dot1q (line 3);(erreur) configuration
➥ non conforme au mode strict;no ip address
```

Cet exemple illustre qu'il manque la ligne de configuration no ip address sur une des interfaces du Catalyst.

L'outil HAWK permet ainsi de contrôler en profondeur les configurations des commutateurs et de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Analyse de périmètres

S'il est important de contrôler les configurations des équipements réseau, il est primordial de valider les VLAN implémentés dans les commutateurs. Pour y parvenir, nous utilisons notre outil GRAPH, ainsi qu'un script d'extraction, utilisé pour déterminer les nœuds et arcs de notre graphe VLAN.

Notre graphe VLAN est un graphe non dirigé, composé des éléments suivants :

- Nœuds. Nous définissons un nœud comme la composée du nom du routeur, du nom de l'interface et du nom du VLAN appliqué à l'interface (déterminé par la commande switchport access vlan x). Par exemple, cat1-GigabitEthernet1/0/3-vlan3 désigne le nœud associé à la configuration cat1, sur l'interface GigabitEthernet1/0/3 où est appliqué le vlan3.
- Arcs. Si deux nœuds sont dans un même VLAN, il existe un arc bidirectionnel entre ces deux nœuds.

Une fois les nœuds et les arcs extraits de la ou des configurations des commutateurs, nous fournissons ces données à l'outil GRAPH, qui calcule les composants connexes du graphe VLAN. Les nœuds contenus dans un composant connexe impliquent donc qu'ils communiquent entre eux.

Si nous appliquons cette méthode sur l'exemple suivant, constitué de deux configurations de Catalyst (cat1 et cat2), nous obtenons les résultats suivants :

```
margot/20.1/graph_vlan$ ./vlans_graph.sh
<stdin>: 8 nodes, 18 edges, 5552 bytes
connected component (4 nodes):
{ cat1-GigabitEthernet1/0/3-vlan3 cat1-GigabitEthernet1/0/6-vlan3
➥ cat2-GigabitEthernet1/0/3-vlan3 cat2-GigabitEthernet1/0/6-vlan3 }
connected component (3 nodes):
{ cat1-GigabitEthernet1/0/4-vlan4 cat2-GigabitEthernet1/0/4-vlan4
➥ cat2-GigabitEthernet1/0/5-vlan4 }
connected component (1 nodes):
{ cat1-GigabitEthernet1/0/5-vlan5 }
```

Les résultats de l'outil GRAPH indiquent que les trois composants connexes suivants ont été trouvés, comme l'illustre la figure 20.4 :

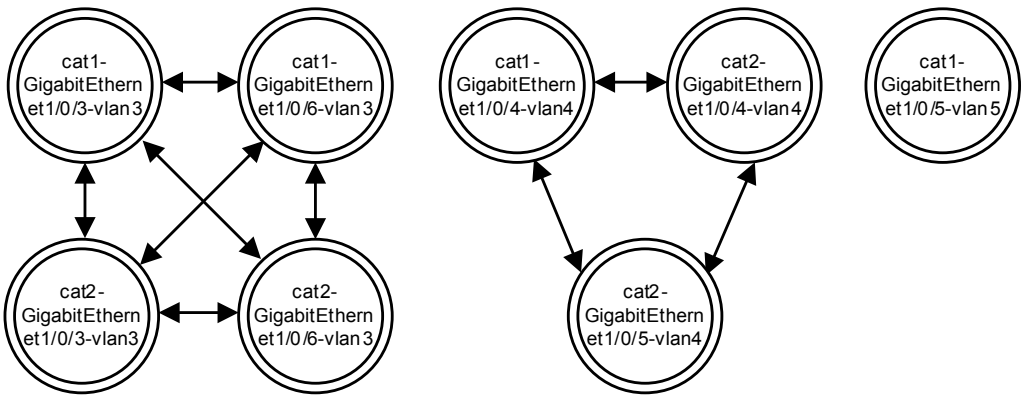


Figure 20.4

Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau

- Composant 1 : les interfaces GigabitEthernet1/0/3 et GigabitEthernet1/0/6 du Catalyst cat1 et les interfaces GigabitEthernet1/0/3 et GigabitEthernet1/0/6 du Catalyst cat2 peuvent communiquer par le biais du vlan3.
- Composant 2 : l'interface GigabitEthernet1/0/4 du Catalyst cat1 et les interfaces GigabitEthernet1/0/4 et GigabitEthernet1/0/5 du Catalyst cat2 peuvent communiquer par le biais du vlan4.
- Composant 3 : l'interface GigabitEthernet1/0/5 du Catalyst cat1 est isolée.

Le contrôle de sécurité consiste à vérifier si ces interfaces doivent faire partie du VLAN considéré. En cas d'erreur, l'isolation du VLAN n'est plus assurée.

Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Exemple de tableau de bord de sécurité réseau

Le tableau 20.1 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité.

Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés. Par exemple, l'évolution dans le temps du nombre total de faiblesses de sécurité (détectées par les contrôles interne et externe sur les commutateurs constituant le réseau) par impact réseau donne une indication de la non-application de la politique de sécurité réseau, comme l'illustre la figure 20.5.

La pertinence de ces courbes nécessite une revue permanente, à la fois des évolutions des configurations des équipements et de la politique de sécurité réseau. Ces courbes ne retranscrivent pas forcément un risque de sécurité mais donnent un indicateur de l'application de la politique de sécurité réseau.

Tableau 20.1 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés
Administration	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés

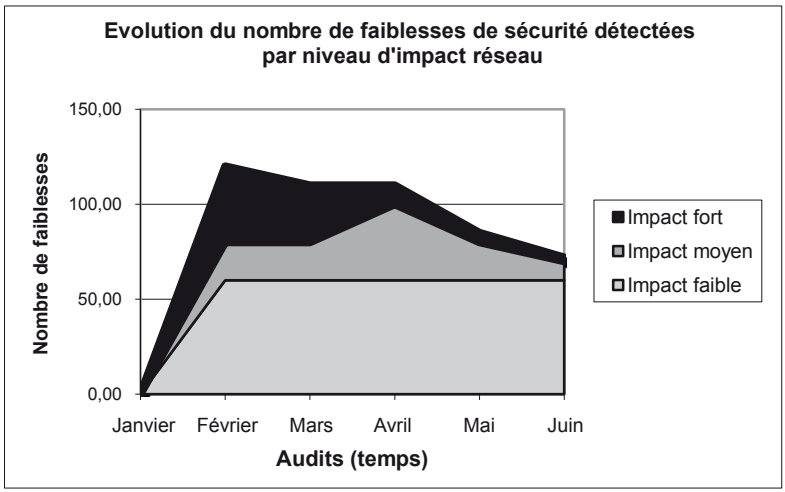


Figure 20.5

Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau

Extension du réseau RadioVoie

Suite à un fort accroissement de la demande lié au succès de son produit, désormais utilisé par la Mairie de Paris mais aussi à Lille, Lyon, Bordeaux et Marseille, RadioVoie s'agrandit.

Afin de ne pas trop subir le poids fiscal et les diverses contraintes imposées par la région parisienne (circulation, grèves, etc.), l'entreprise décide de créer un nouveau site à Mouans-Sartoux, dans les Alpes-Maritimes. Ce nouveau site n'accueille dans un premier temps que du personnel administratif.

Pour des raisons financières, RadioVoie sous-traite la production de ses équipements à une tierce partie.

Besoins à satisfaire

L'entreprise souhaite interconnecter les deux sites pour échanger des informations. De plus, les commerciaux de RadioVoie doivent pouvoir se connecter à distance au réseau interne pour connaître les dernières informations afin de les transmettre à leurs clients.

Les informations échangées, tant au niveau de l'interconnexion des sites que de l'accès à distance par les commerciaux, ne nécessitent pas une bande passante importante (inférieure à 512 Kbit/s) mais doivent être considérées comme confidentielles. Il n'est pas prévu que les commerciaux en accès à distance aient besoin d'utiliser l'interconnexion entre les sites.

Les coûts de connexion réseau, de maintenance et de sécurité sont des éléments décisifs de choix des solutions techniques.

Étude de risques

L'élément critique de RadioVoie est le réseau dédié à la production des équipements radio, qui est le cœur de l'activité de l'entreprise. Connaissant les contraintes financières, RadioVoie demande à la tierce partie à qui elle sous-traite la production de répondre à de nouvelles exigences de sécurité physique et de confidentialité. De plus, RadioVoie planifie des audits afin de contrôler le niveau de sécurité du site de production de la tierce partie.

Concernant les interconnexions des sites et des accès distants, une solution clés en main de sécurité doit être mise en place. Comme les changements de configuration ne seront pas fréquents, une solution incluant plusieurs fonctions de sécurité doit être choisie.

Les temps de réponse des interconnexions n'est pas une contrainte. La solution peut donc s'appuyer sur un réseau IP de bout en bout.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, RadioVoie adopte une politique de sécurité réseau minimale, constituée des règles suivantes :

- *« L'interconnexion réseau entre les sites de l'entreprise est authentifiée et chiffrée. »*
- *« L'interconnexion réseau entre les sites de l'entreprise ne reste pas indisponible pendant plus de vingt-quatre heures ouvrées. »*
- *« Les accès réseau à distance aux sites de l'entreprise sont authentifiés, chiffrés et limités aux commerciaux de l'entreprise. L'authentification des accès distants est individuelle. »*
- *« Les ordinateurs utilisés pour les accès distants respectent les standards de l'entreprise pour les machines nomades. Cela signifie au minimum que l'ordinateur de l'utilisateur distant est protégé des virus et du réseau (Internet, etc.) et qu'il ne peut invalider ou modifier ces protections. »*
- *« Tout vol ou problème de sécurité déclenche une procédure de modification de la protection des accès distants. »*
- *« Un utilisateur distant connecté à l'entreprise ne peut permettre, consciemment ou non, à un tiers d'atteindre le réseau de l'entreprise. »*
- *« Les flux réseau entre les sites de l'entreprise sont filtrés. »*
- *« Le réseau de production est physiquement et logiquement séparé des autres réseaux de l'entreprise. »*

De plus et de manière plus spécifique, la politique de sécurité réseau pour la relation avec le réseau Internet édicte les règles suivantes :

- *« La relation avec Internet respecte les contraintes légales françaises (droit du travail, loi de sécurité électronique, etc.). »*
- *« Les flux réseau en provenance d'Internet sont filtrés et contrôlés. »*
- *« Les flux entre l'entreprise et Internet sont contrôlés par une solution antivirus. »*
- *« Les flux entre l'entreprise et Internet sont contrôlés par une solution de lutte contre les attaques (applets hostiles, etc.). »*
- *« Aucun flux en provenance d'Internet n'est autorisé à atteindre directement le réseau interne de l'entreprise. »*
- *« Les machines qui sont en relation directe avec Internet sont sécurisées en permanence. L'établissement d'un standard associé à des procédures et des contrôles assure le respect de cette règle. »*
- *« Toutes les machines sont équipées d'une solution antivirus. »*
- *« Les machines en relation avec Internet sont administrées depuis le réseau bureautique par l'intermédiaire des flux chiffrés. »*
- *« L'entreprise n'est autorisée à utiliser sur Internet que les services de messagerie, de noms, de transfert de fichiers et de consultation de données en clair ou chiffrées. »*
- *« Le courrier entrant est filtré pour éliminer les courriers non sollicités (Spam, Scam, etc.). »*
- *« Tous les flux en relation avec Internet sont notés, stockés et archivés pendant une année. »*

- *« L'utilisation d'Internet pour un usage non professionnel est tolérée. »*
- *« Il est possible de limiter l'étendue de la consultation des données afin de prévenir les déviances non professionnelles ou illégales (sites pornographiques, pédophiles, d'échange de logiciels piratés, etc.). »*
- *« Une procédure est créée afin de garantir la restauration du service au plus vite en cas de refus de service. »*
- *« Une procédure et un processus sont établis pour la gestion des incidents de sécurité (infection par virus, attaque, etc.). »*

Solution de sécurité

Il s'agit de connecter les sites de Paris et de Mouans-Sartoux ainsi que les commerciaux au site de Mouans-Sartoux, comme l'illustre la figure 20.6.

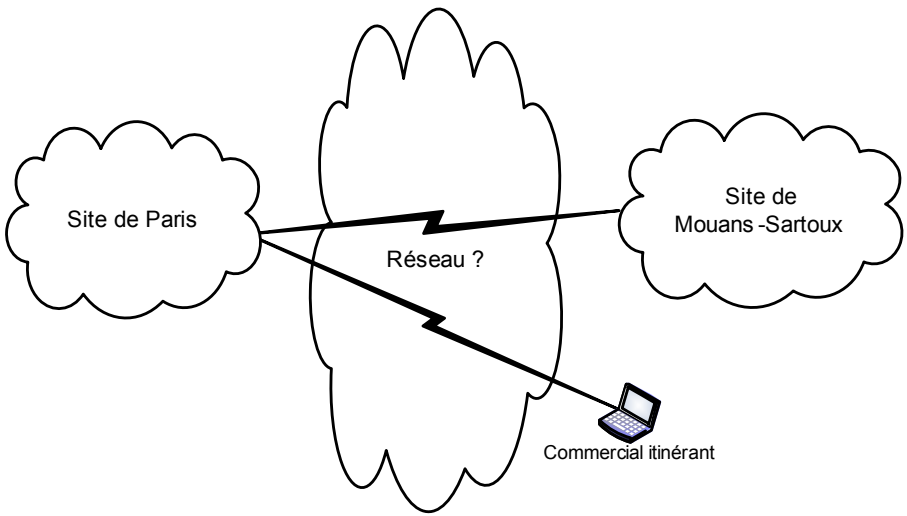


Figure 20.6

Interconnexion des sites de RadioVoie

Les aspects financiers sont décisifs dans le choix technique final. En revanche, les besoins en bande passante entre les sites ne sont guère importants et ne nécessitent pas la mise en œuvre de solutions réseau complexes.

Les flux réseau étant chiffrés, il s'agit de définir un réseau privé virtuel, ou VPN (Virtual Private Network), entre les sites de l'entreprise.

Les diverses offres de connexions réseau des opérateurs de télécommunications reposent généralement sur des liaisons louées ou publiques. Bien que les liaisons louées (Numéris, etc.) offrent une fiabilité en terme de qualité de service, ou QoS (Quality of Service), et une sécurité élémentaire, la rigidité et les coûts financiers de ce type de connexion

risquent d'être rédhibitoires. La tarification est en effet généralement établie au prorata de la distance entre les sites et de la bande passante consommée.

Un VPN passant par un réseau public comme Internet réclame un investissement sécurité plus important, car il nécessite de protéger l'entreprise des risques associés au réseau public. Cette solution n'offre de surcroît aucune garantie de qualité de service du fait de l'utilisation du protocole IP sur un réseau sous le contrôle d'un nombre inconnu d'entreprises. Malgré ses inconvénients, cette solution reste cependant plus évolutive et financièrement plus attractive. RadioVoie opte donc pour un VPN au travers du réseau public Internet.

Le délai maximal d'indisponibilité des interconnexions réseau est assez important (vingt-quatre heures ouvrées). Il est donc essentiel de ne pas mettre en œuvre de liens réseau redondants entre les sites afin de limiter les coûts. Un accord contractuel permet de protéger l'entreprise de ce risque d'indisponibilité du lien d'interconnexion et de ses conséquences financières.

Pour garantir le lien de bout en bout, RadioVoie doit utiliser de part et d'autre le même fournisseur de service. Il peut être également envisageable de disposer d'un accès Internet de type « particulier », avec une forte bande passante montante tel que peut l'offrir l'ADSL.

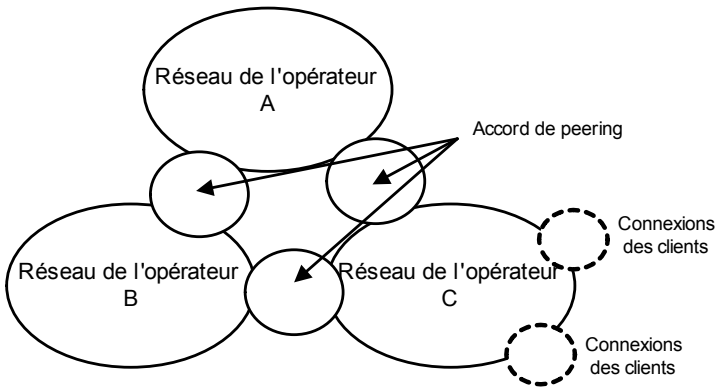
Solution sécurisée d'interconnexion entre les sites

Le réseau Internet est constitué de nombreux réseaux, gérés individuellement par les opérateurs et fournisseurs de télécommunications. Chacun de ces réseaux est interconnecté par le protocole IP à d'autres réseaux par des accords de peering, permettant de constituer la toile internationale, ou Web.

Les accords de peering définissent les paramètres d'interconnexion entre deux réseaux IP (type d'interconnexion, paramètres de débit, échange des tables de routage, etc.), comme illustré à la figure 20.7.

Figure 20.7

Interconnexions entre opérateurs Internet



Les routes d'Internet représentent aujourd'hui de l'ordre de 120 000 entrées de préfixes dans les tables de routage des nœuds réseau.

Dans sa version 4, le protocole IP (IPv4) n'implémente pas de fonction de sécurité. Les travaux entrepris sur la version 6 (IPv6) permettent d'ajouter une telle fonction par le biais d'IPsec.

Le protocole IPsec permet d'établir, par le biais d'une connexion Internet, un tunnel IP sécurisé (mode tunnel) chiffré (mode ESP) et authentifié (mode AH) entre deux acteurs.

Afin de prendre en compte la politique de sécurité, l'interconnexion des sites de Radio-Voie s'appuie sur cette suite de sécurité IPsec pour chiffrer et authentifier les boîtiers de chiffrement correspondant aux sites.

L'authentification IPsec utilise des clés générées préalablement de manière aléatoire. Il serait dangereux de télécharger sur Internet des outils de génération de ce type de clés. En effet, pour la plupart, ces outils n'implémentent pas un réel caractère aléatoire pour la génération des clés, mettant en péril la confidentialité des clés générées.

La petite taille de l'entreprise permet d'instaurer la procédure suivante de transmission sécurisée des clés :

- Le fondateur de l'entreprise génère la clé.
- Le fondateur de l'entreprise installe la clé sur le boîtier IPsec parisien.
- Le fondateur de l'entreprise installe la clé sur un média amovible.
- Le fondateur de l'entreprise amène le média amovible sur le site de Mouans-Sartoux en s'assurant que ce média n'est accessible à personne d'autre.
- Le fondateur de l'entreprise installe la clé sur l'autre boîtier.
- Le fondateur de l'entreprise dépose le média dans le coffre-fort d'une banque.

Le fondateur de l'entreprise étant en même temps son propriétaire, il n'est guère imaginable d'avoir un risque plus bas dans le respect de la confidentialité de la clé.

Un filtrage des protocoles entrants et sortants est effectué par chaque site avant le passage des données dans le tunnel IPsec ou leur réception du tunnel IPsec, comme illustré à la figure 20.8.

Le coût financier étant un critère important pour une PME, RadioVoie opte pour un équipement permettant à la fois de créer des tunnels IPsec et d'intégrer des fonctions de filtrage de protocoles.

Les paramètres du tunnel IPsec ainsi que le filtrage des protocoles sont soigneusement définis préalablement à toute implémentation dans les équipements.

Le large choix d'équipements IPsec certifiés par l'ICSA (International Computer Security Association) et offrant des services de réseau privé virtuel est récapitulé au tableau 20.2.

Après étude approfondie, RadioVoie opte pour le boîtier VPN/IPsec VPN Router Family de Nortel, qui inclut toutes les fonctions de sécurité nécessaires, limitant de ce fait les coûts financiers.

Bien que le cumul de fonctions de sécurité ne soit jamais recommandé, la configuration des règles de sécurité du boîtier est *a priori* stable dans le temps. La configuration initiale demande toutefois l'intervention d'un expert.

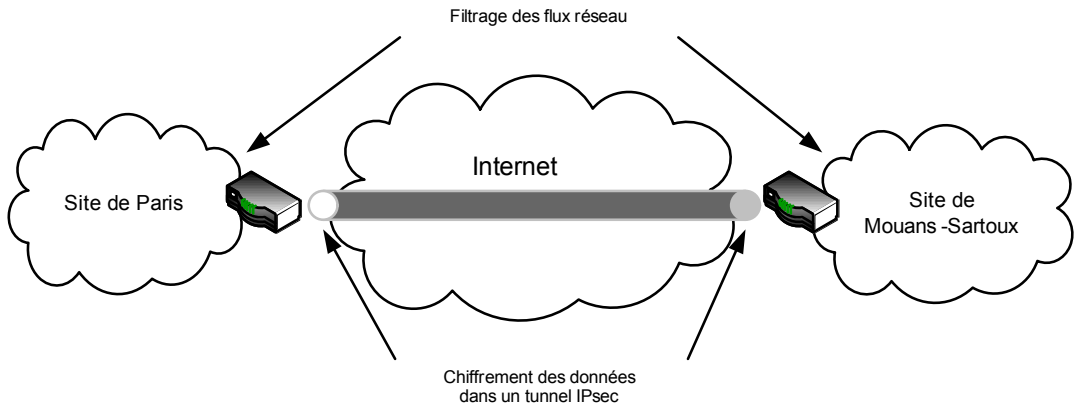


Figure 20.8

Mécanismes de sécurité à déployer entre les sites

Tableau 20.2 Équipements IPsec certifiés par l'ICSA

Société	Produit	Version	OS
3Com	3Com x-family United Security Platform	X3.0.0.2090	Propriétaire
Fortinet, inc.	FortiGate Multi-Layared Security System Family	FortiOS 3.0	Propriétaire
Juniper Networks	Ssg, isg, ns series family	ScreenOS 6.1	Propriétaire
Nokia	Security IPsec product group	IPSO4.2	Propriétaire
Secure Computing Corporation	Sidewinder 7	70002	Propriétaire
WatchGuard Technologies, inc.	Firebox X Edge e-Series	8.6.3	Propriétaire
ZyXEL	ZvWALL USG Series	ZLD 2.01	Propriétaire

La famille de boîtiers VPN permet de monter des tunnels IPsec avec des algorithmes de chiffrement symétrique 3DES, AES et RC4 et des fonctions de hachage SHA-x.

Les méthodes d'authentification des utilisateurs s'appuient sur RADIUS, LDAP, SecureID, des certificats X.509, etc.

Les clients VPN/IPsec sont disponibles pour la plupart des plates-formes (Microsoft 95, 98, 2000, etc., IBM-AIX, SUN-Solaris, HP-UX, Linux).

Deux boîtiers VPN/IPsec sont nécessaires. Ils doivent être connectés avant l'équipement d'interconnexion au réseau Internet fourni par l'opérateur de télécommunications, comme l'illustre la figure 20.9.

Les boîtiers assurent la fonction de chiffrement mais peuvent également filtrer le flux réseau au sein du tunnel IPsec.

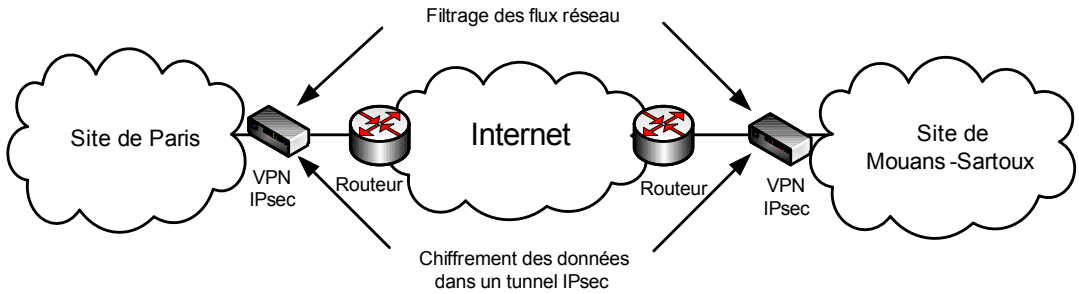


Figure 20.9

Interconnexions des sites de RadioVoie par des boîtiers IPsec

Le filtrage des protocoles peut être assuré par le boîtier, qui intègre au sein du tunnel un pare-feu de type filtrage dynamique implémentant de nombreuses options de filtrage du trafic. L'avantage de ce type de filtrage est qu'il permet de pallier les limites du filtrage statique, qui ne couvre pas l'allocation dynamique des ports sources.

Il va de soi que l'exposition d'un boîtier à Internet présente un risque de sécurité, qu'il est nécessaire de couvrir par l'ajout de filtrages complémentaires.

Puisque RadioVoie dispose d'un routeur, il est possible d'utiliser ses capacités filtrantes pour assurer un premier « nettoyage » des flux en provenance d'Internet (principe du routeur *choke*). Le routeur filtre donc les flux polluants, comme les connexions 137/UDP en provenance des stations de travail Windows mal configurées. En fait, il n'accepte que les flux IPsec.

Il est aussi possible de monter une zone démilitarisée (DMZ) sur le pare-feu et de mettre la patte externe du tunnel IPsec sur la DMZ et la patte interne sur une autre DMZ. Ainsi le pare-feu contrôle les deux côtés du boîtier.

Cette solution de pare-feu est donc mise en place pour assurer un véritable service de filtrage, comme illustré à la figure 20.10.

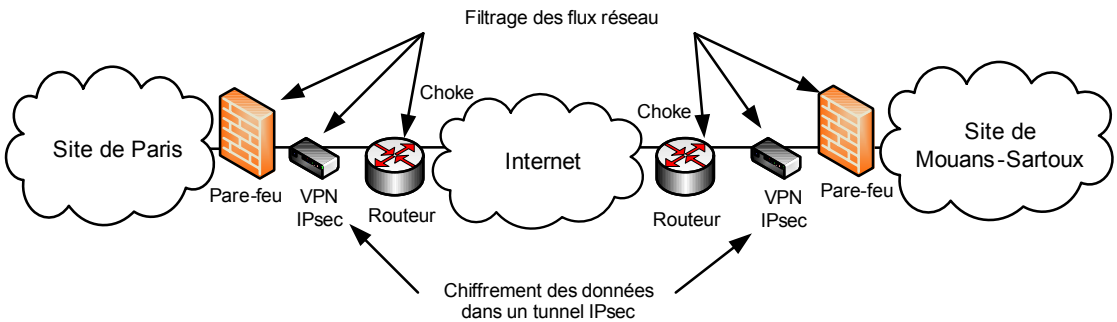


Figure 20.10

Filtres mis en place entre les sites de RadioVoie

Pour le tunnel IPsec créé entre les deux boîtiers VPN/IPsec, RadioVoie retient les options ESP (Encapsulating Security Payload) en mode tunnel, avec une authentification fondée sur des clés préalablement générées et installées manuellement dans les boîtiers.

Les règles de filtrage du trafic non chiffré prennent comme paramètres le type de protocole, la direction du flux de trafic, les adresses source et destination IP, les ports source et destination et l'établissement de la connexion TCP. Les filtrages sont bien entendus appliqués avant le chiffrement ou après le déchiffrement des paquets à émettre ou à recevoir.

En dehors des règles de filtrage, le pare-feu protège des attaques réseau classiques, telles que l'inondation SYN (SYN flooding), le bombardement UDP (UDP bombing), les attaques de type smurf, etc.

Le tableau 20.3 recense les produits pare-feu certifiés par l'ICSA.

Tableau 20.3 Produits pare-feu certifiés par l'ICSA

Société	Produit	OS
Alcatel-Lucent	Lucent VPN Firewall Family	Propriétaire
Balabit IT Security	Zorp Professional	Linux
Check Point Software	Check Point SecurePlatform NG	Linux
Chenghu Huawei Storage & Network	Quidway Eudemon Family	Propriétaire
D-Link Corporation	NetDefend Firewall Family	Propriétaire
Global Technology Associates Inc.	GTA Firewall Family	Propriétaire
IBM Internet Security Systems	Proventia M Series Integrated Security Appliance	Propriétaire
Ingate	Ingate Firewall 1500	Propriétaire
SonicWALL	Pro Series Firewall Family	Propriétaire
VarioSecure Networks Inc.	VarioSecure	Propriétaire

Solution sécurisée des accès à distance des commerciaux au site de Paris

En accord avec la politique de sécurité réseau, les accès à distance des commerciaux s'effectuent au travers d'Internet pour atteindre le point d'accès réseau du site de Paris, c'est-à-dire le boîtier VPN/IPsec.

Internet peut être atteint par des points d'accès téléphoniques mais également depuis une entreprise reliée à Internet, comme illustré à la figure 20.11.

Les tunnels entre le poste du commercial et le pare-feu s'établissent de la façon suivante :

1. L'ordinateur portable du commercial établit une connexion PPP avec le point d'accès du réseau ou le concentrateur d'accès LAC (L2TP Access Concentrator) géré par un fournisseur ou opérateur réseau.

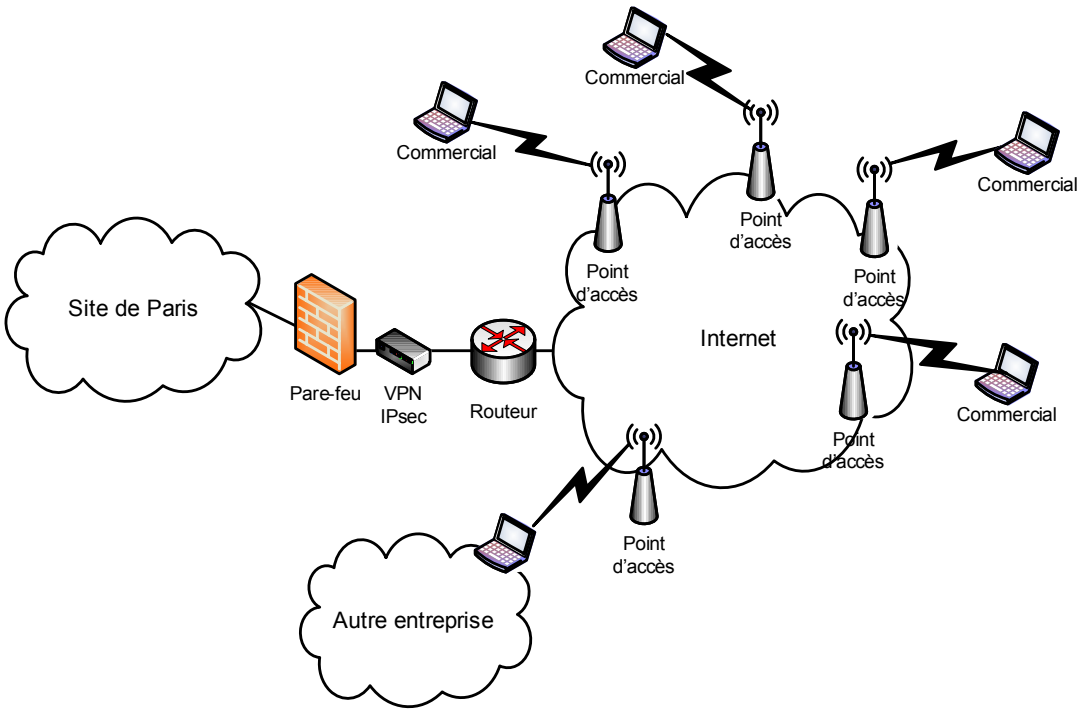


Figure 20.11

Accès distants au site de Paris

2. Suite aux informations fournies par le protocole PPP (Point-to-Point Protocol), le LAC initie une connexion L2TP avec le serveur réseau, ou LNS (L2TP Network Server), du site de destination, c'est-à-dire l'équipement d'interconnexion géré par un fournisseur ou opérateur réseau.
3. Comme le stipule la politique de sécurité, l'utilisateur fournit une authentification appropriée pour que le tunnel chiffré puisse être établi.
4. Une fois le tunnel L2TP établi, une session IPsec entre l'ordinateur portable du commercial et le pare-feu peut être établie afin de sécuriser les données transmises.

Pour l'authentification des utilisateurs, plusieurs solutions sont possibles au niveau du pare-feu, telles que des serveurs d'authentification gérant comptes et mots de passe, des tokens, des certificats, etc.

Une fois l'authentification acceptée et le tunnel établi, la connexion entre l'entreprise et l'utilisateur se déroule comme illustré à la figure 20.12.

Le tunnel part directement de la machine du commercial, qui chiffre donc les flux *via* un logiciel client, puis transite au travers d'Internet pour arriver au boîtier IPsec de l'entreprise, où les flux sont déchiffrés et où les contrôles d'accès filtrent les flux réseau.

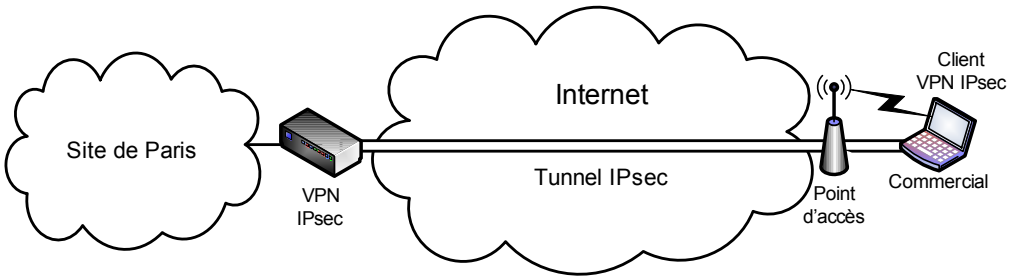


Figure 20.12

Connexion via *le tunnel IPsec établi pour les accès distants*

Malgré le sentiment de sécurité procuré par IPsec, certains risques demeurent. IPsec n'étant qu'une interface réseau supplémentaire pour la station de travail, la station peut être utilisée sciemment ou non comme relais pour accéder au réseau d'entreprise, comme l'illustre la figure 20.13.

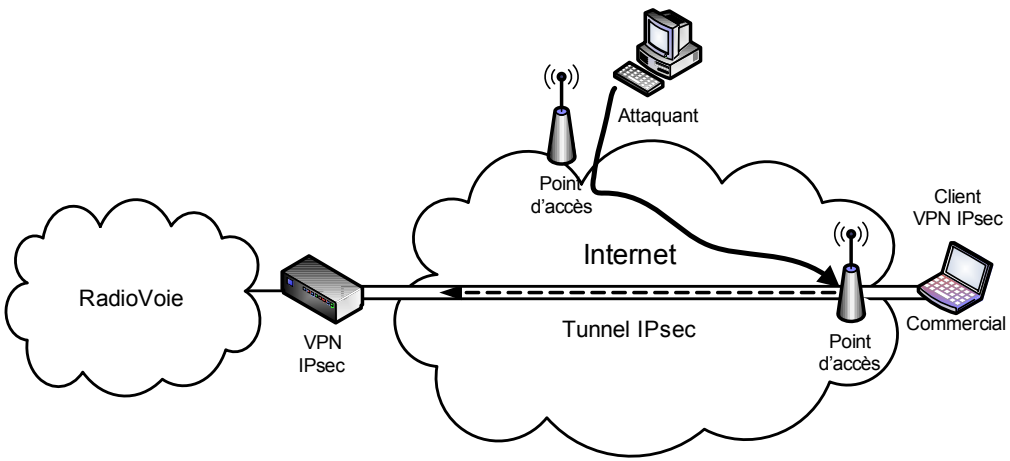


Figure 20.13

Attaque par rebond sur un tunnel IPsec

Sur cette figure, l'intrus prend le contrôle de la machine du commercial par l'intermédiaire d'un cheval de Troie ou en rebondissant sur un relais applicatif installé sur la machine, qui masque son adresse IP. L'intrus dispose de ce fait des mêmes droits réseau que l'utilisateur distant dont le système a été pénétré.

Pour réduire ce risque, RadioVoie configure les boîtiers VPN/IPsec de Nortel en désactivant la fonction Split Tunneling. Cela provoque une modification du comportement réseau de la machine distante. Le client modifie toutes les routes, y compris la route locale *(localhost),* afin que tous les paquets soient routés *via* le tunnel IPsec. Il n'est plus possible pour un intrus d'utiliser la machine comme relais.

Reste un dernier risque : un paquet infecté ne nécessitant pas nécessairement de réponse réseau (avec le protocole UDP, par exemple) peut affecter une machine. Un vers de type SQL Hammer, par exemple, si la station héberge un serveur MS-SQL, est une menace pour l'entreprise malgré toutes les précautions prises.

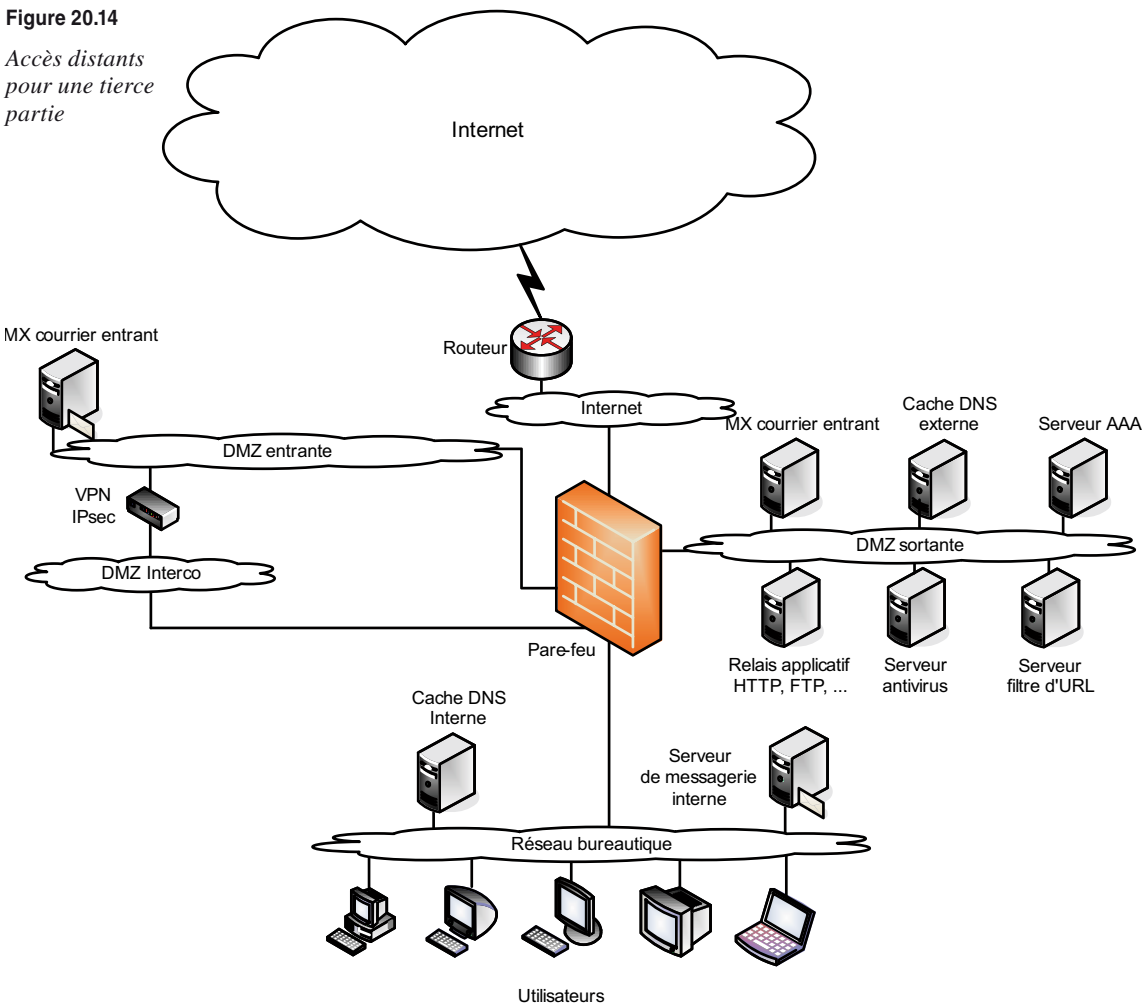
Rappelons que la politique de sécurité stipule, pour faire face à un tel risque, que le client soit protégé d'Internet, autrement dit que son trafic réseau soit filtré par un pare-feu personnel et que la configuration de celui-ci ne soit pas modifiable par l'utilisateur.

Solution sécurisée pour la relation Internet

Afin de respecter la politique de sécurité Internet, RadioVoie met en place l'architecture illustrée à la figure 20.14.

Figure 20.14

Accès distants pour une tierce partie



L'ensemble de la solution repose sur un commutateur entièrement dédié. Ce commutateur utilise la fonction de filtrage d'adresses MAC pour s'assurer que les paquets viennent bien des machines connues. Seule la connexion entre le pare-feu et le réseau bureautique est rattachée au commutateur de l'entreprise.

Chaque réseau est spécialisé dans un type de service :

- Le lien Internet sert à héberger les machines sans protection (ici le routeur).
- La DMZ entrante sert à l'accueil des trafics à l'initiative d'Internet.
- La DMZ sortante sert à l'accueil des trafics qui sortent vers Internet, ainsi que des trafics retour.
- La DMZ interco sert à isoler et filtrer le trafic en provenance du boîtier IPsec.
- Le dernier lien est rattaché au réseau bureautique.

Pour la messagerie, les flux réseau autorisés sont les suivants :

- Les messages sortants passent obligatoirement par le serveur de messagerie interne, équipé d'un antivirus, qui les réachemine au MX (Mail eXchanger), ou relais de messagerie, de courrier sortant, également équipé d'un antivirus, lequel les envoie vers Internet. Le MX de courrier sortant n'accepte que les courriers en provenance du serveur de messagerie interne et signés avec le nom de domaine de l'entreprise.
- Les messages entrants passent obligatoirement par le MX de courrier entrant, équipé d'un antivirus, qui les réachemine uniquement vers le serveur de messagerie interne. Le MX de courrier entrant est configuré pour ne relayer que les messages envoyés vers le domaine de l'entreprise. Il est également équipé d'une solution de filtrage du courrier par analyse du contenu pour lutter contre le Spam et les virus.
- Le serveur de messagerie interne héberge les boîtes aux lettres et est équipé d'une solution antivirus afin de prévenir la propagation d'un virus exclusivement en interne.

Les flux réseau autorisés pour obtenir une résolution de nom DNS sur le réseau Internet sont les suivants :

- La station de travail émet sa demande vers le serveur de noms cache DNS interne de l'entreprise.
- Le serveur cache DNS interne relaie la demande vers la DMZ sortante au serveur DNS cache installé sur le cache DNS externe afin d'optimiser les flux de messagerie.
- Le cache DNS externe va chercher la réponse et la renvoie au serveur cache DNS interne.
- Le cache DNS interne renvoie la réponse au demandeur.

Les flux réseau autorisés pour l'accès à Internet (HTTP, FTP, etc.) se déroulent de la façon suivante :

1. Les stations de travail se connectent à Internet *via* leur navigateur.
2. De manière transparente, le pare-feu valide l'URL demandée par rapport aux autorisations installées dans la solution de filtrage d'URL.

3. Si l'URL peut passer, le pare-feu envoie l'URL ou le flux FTP demandé vers le relais applicatif (proxy).
4. Le relais applicatif envoie l'URL ou le flux FTP demandé vers le serveur antivirus.
5. Le serveur antivirus va chercher l'information demandée, valide son contenu et la renvoie au relais applicatif.
6. Le relais applicatif renvoie la réponse au pare-feu.
7. Le pare-feu renvoie la réponse au navigateur du client.

Les flux réseau autorisés pour la relation avec le serveur AAA (Authentication, Authorization and Accounting) en charge de la gestion des comptes des commerciaux utilisant IPsec se déroulent de la façon suivante :

1. Le boîtier IPsec reçoit des données d'authentification.
2. Il demande au serveur AAA de les valider.
3. Le serveur répond.
4. Le tunnel est autorisé ou non.

Les flux réseau autorisés pour la relation entre le boîtier IPsec et le réseau bureautique sont les flux de type HTTP, FTP, etc. Ils sont réacheminés vers la solution antivirus de la même manière qu'est réacheminé le trafic sortant de l'entreprise.

Quelle que soit la provenance du flux, tous les flux non autorisés sont bloqués et font l'objet d'une trace (log). Tous les flux transitant par les relais applicatifs divers (relais applicatifs, serveur antivirus, relais de courrier, serveur de filtrage d'URL, etc.) font l'objet d'une trace. Ces traces sont archivées selon les *desiderata* de la politique de sécurité.

Certaines de ces fonctions peuvent se trouver regroupées sur une même machine. Le relais applicatif, par exemple, peut également être un cache DNS externe, tout comme le cache DNS interne peut être installé sur la même machine que le MX de courrier sortant. On peut imaginer que le serveur antivirus et le filtre d'URL partagent également une même machine.

Risques réseau couverts

De par l'architecture adoptée, RadioVoie a la garantie que les entités capables d'établir un tunnel IPsec sont connues et autorisées.

Tous les flux susceptibles d'atteindre le réseau interne de l'entreprise depuis l'extérieur sont contrôlés et filtrés. Tous les flux connus pour être vecteurs de risque (HTTP, FTP, SMTP, etc.) font l'objet d'un contrôle particulier au niveau applicatif afin d'être nettoyés de tout virus ou attaque.

L'entreprise ne peut être utilisée pour la propagation de Spam et peut lutter contre le Spam qui l'atteindrait. De plus, une attaque éventuelle depuis Internet sur le commutateur externe ne permet pas d'atteindre le réseau bureautique.

Le serveur AAA qui héberge les détails sur les comptes et les méthodes d'authentification autorisées peut difficilement être atteint par une personne non autorisée. L'entreprise

peut disposer d'une trace de tous les trafics réseau entre son réseau bureautique et un quelconque autre réseau.

Risques réseau non couverts

Si le pare-feu est franchi ou ignoré, tous les réseaux de l'entreprise peuvent être touchés. Si le pare-feu ou le commutateur externe est compromis, c'est toute la solution Internet qui peut être mise en déni de service. Si le commutateur externe est compromis, la confidentialité, l'authenticité et l'intégrité des flux de tous les réseaux peuvent également être compromis.

Si un intrus réussit à prendre le contrôle d'une machine acceptant les flux entrants, comme le MX de courrier entrant, grâce à une faille de sécurité, par exemple, il peut retenter cette attaque sur le serveur de messagerie interne. S'il réussit à nouveau, le réseau bureautique entier peut être compromis.

Si une attaque permet de passer outre le boîtier IPsec, l'attaquant peut bénéficier des mêmes autorisations que le plus privilégié des utilisateurs du boîtier.

Tableau de bord de sécurité

Après avoir défini la politique de sécurité ainsi que les solutions associées, cette section détaille les principaux contrôles à mettre en place, fournit des éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

RadioVoie a prévu des contrôles de sécurité à de multiples niveaux dans son réseau.

Au niveau du commutateur, un réseau virtuel logiquement isolé de tous les autres réseaux de l'entreprise est installé. Ce réseau utilise un plan d'adressage incompatible avec le réseau et n'a aucune relation avec les autres réseaux au-delà du niveau 2 du modèle OSI. Sa faiblesse réside donc dans le commutateur.

Ce réseau, destiné exclusivement à la supervision et au contrôle de sécurité du commutateur, héberge une machine chargée des tâches suivantes :

- Surveiller l'état du commutateur *via* des requêtes SNMP.
- Collecter *via* SSH v2 (protocole chiffré) la configuration du commutateur et l'analyse. La fréquence de l'analyse est fonction de qui l'effectue, l'homme ou la machine. En cas d'erreur, la solution relève une erreur sur sa console afin d'alerter l'équipe sécurité ou réseau. Les configurations collectées sont historisées.

L'adresse sur le commutateur permettant d'accéder à ces services à risque est uniquement celle située sur le VLAN de supervision. Cela signifie que le commutateur ne peut être administré que depuis ce même VLAN.

Grâce à l'architecture Internet, qui place le pare-feu comme goulet d'étranglement pour tous les trafics réseau à l'exception de ceux visant le routeur depuis Internet, il est possible, pour les relations avec l'extérieur, de tracer, voire d'écouter tous les flux réseau.

Le pare-feu peut tracer tous les trafics, autorisés ou non, et, selon la solution choisie, écouter le réseau.

Pour s'assurer de la nature de tous les flux réseau échangés par l'entreprise avec l'extérieur, il faut récupérer les traces du routeur, du pare-feu et du boîtier IPsec. Il faut également collecter les traces du serveur d'authentification AAA afin de s'assurer de la légitimité des connexions effectuées au travers de cette solution ainsi que des tentatives possibles de pénétration d'un compte.

L'ensemble de ces informations peut être concentré sur un système disposant d'un logiciel corrélateur d'événements. Cela permet de détecter rapidement les tentatives d'infraction de sécurité tout en évitant trop de faux positifs.

Des audits de sécurité sont prévus à échéance régulière sur les solutions afin de s'assurer de leur efficacité. Ces audits peuvent inclure des tests de pénétration et de vérification de la configuration par rapport à un standard, etc.

Le cas échéant, il reste possible d'ajouter une sonde d'intrusion. Rappelons que, contrairement à son nom, une sonde d'intrusion détecte rarement les intrus. Sa fonction est de détecter des comportements réseau anormaux, définis comme tels par l'administrateur de la sonde, et de les relever comme une alerte de sécurité potentielle (il peut exister des degrés d'alerte). Ces sondes sont parfaitement adaptées pour détecter les comportements réseau associés à des infections en provenance de vers *(worms)* ou à des attaques sur des services réseau comme FTP ou HTTP.

S'il s'agit d'une sonde préventive d'intrusion, celle-ci peut de surcroît détecter des évolutions anormales de trafic réseau sur tel ou tel flux spécifique (le port Netbios, par exemple), et ainsi relever une montée en puissance suspecte de ce flux, signe caractéristique d'un ver.

Mise en œuvre des outils maison

Cette section détaille la génération et la vérification des secrets pour des accès distants, ainsi que la vérification de la consistance des ACL réseau.

Gestion de secrets

Comme indiqué précédemment, des secrets partagés doivent être créés pour authentifier les connexions fondées sur le protocole IPsec. La gestion de secrets étant souvent délicate, nous utilisons notre outil GENPASS pour les gérer.

Nous générons une clé de manière pseudo-aléatoire. Elle sera la clé maîtresse pour la génération déterministe des autres clés :

```
margot/20.2/genpass$ cat generate.sh
# Generate the domain key
rm ./key.domain.com
umask 077
genpass -r -l 256 -a '[0-9a-zA-Z]' -s 'domain.com' > ./key.domain.com
chmod 400 ./key.domain.com
```

Le fichier key.domain.com contient la clé maîtresse. Il est composé de 256 caractères choisis par '[0-9a-zA-Z]'.

Nous générons de manière déterministe la clé pour l'utilisateur cedric.llorens :

```
# Generate the key for cedric llorens
echo "clé pour cedric llorens";
DOMAIN=key.domain.com; ACCOUNT=cedric.llorens;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
OJTEgstcsKhaW76jgRWNiSL1TBOkV24NLUlACkXbodhckc98XD78E01P
```

de même que celle pour l'utilisateur denis.valois :

```
# Generate the key for denis valois
echo "clé pour denis valois";
DOMAIN=key.domain.com; ACCOUNT=denis.valois;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
OJTEgstcsKhaW76jgRWNiSL1TBOkV24NLUlACkXbodhckc98XD78E01P
```

et celle pour l'utilisateur laurent.levier :

```
# Generate the key for laurent levier
echo "clé pour laurent levier";
DOMAIN=key.domain.com; ACCOUNT=laurent.levier;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
gcVonrPsHOJONEnt6KSqsFNkry2pMS4Vga67Z54ZESTSccIFCk7Xvkom
```

Ces clés sont uniques et ne sont pas stockées dans une base de données. Il suffit de connaître la clé maîtresse et des paramètres de génération pour retrouver ou contrôler les clés.

Nous pouvons aussi générer d'autres clés pour d'autres usages, ainsi que contrôler ces clés si elles sont présentes dans des configurations d'équipements réseau et fournir ainsi des données utiles pour l'établissement d'un tableau de bord de sécurité.

Analyse des ACL

Dans cette évolution de réseau, les ACL ont un rôle important et doivent être vérifiées afin de s'assurer quelles ne comportent pas d'inconsistances ou de redondances.

Prenons l'exemple de l'ACL suivante :

```
Margot/20.2/vacl$ cat acl.txt
access-list 100 permit ip any any
access-list 100 permit ip any 10.10.10.0 0.0.0.255
access-list 100 deny ip any 10.10.10.0 0.0.0.255
```

Pour analyser cette configuration d'ACL, nous utilisons notre outil VACL de la manière suivante :

```
Margot/20.2/vacl$ vacl -a acl.txt
[1] access-list 100 permit ip any any
[2] access-list 100 permit ip any 10.10.10.0 0.0.0.255
```

```
*** redundancy   [2] < [1]
[1] access-list 100 permit ip any any
[3] access-list 100 deny ip any 10.10.10.0 0.0.0.255
*** inconsistency   [3] < [1]
[2] access-list 100 permit ip any 10.10.10.0 0.0.0.255
[3] access-list 100 deny ip any 10.10.10.0 0.0.0.255
*** inconsistency   [3] = [2]
```

Les lignes 1 et 2 de l'ACL indiquent une redondance ; en revanche les lignes (1 et 3) et les lignes (2 et 3) indiquent une inconsistance.

Il est donc possible avec l'outil VACL de contrôler en profondeur les ACL et de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Exemple de tableau de bord de sécurité réseau

Le tableau 20.4 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité.

Tableau 20.4 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, les routeurs, les LAN et les systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, analyse des configurations des VLAN, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, les LAN et les systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)

Sous-réseau	Catégorie	Élément
Internet (suite)	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateurs, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

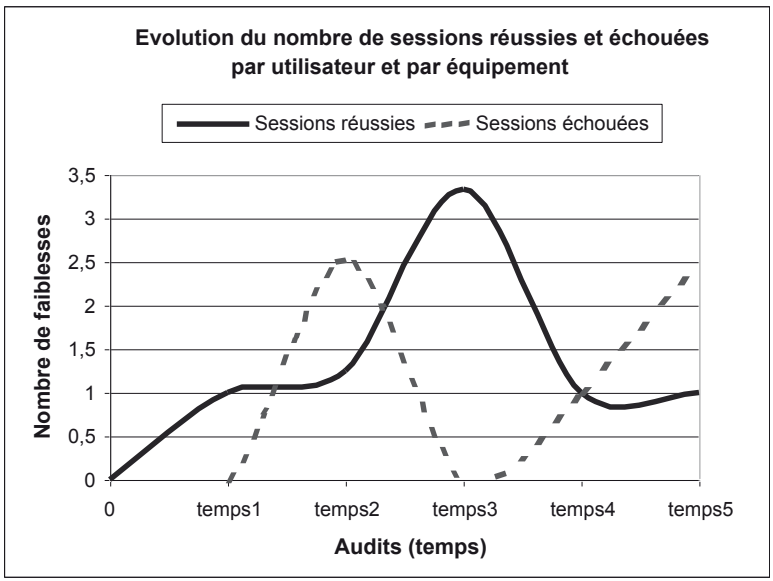
Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous considérons que chaque équipement de sécurité remonte des événements sur les accès réussis et échoués, l'évolution dans le temps du nombre de sessions réussies et échouées par utilisateur et par équipement permet de donner un point de vue sur la sécurité des accès au réseau de l'entreprise.

La figure 20.15 illustre une forte probabilité de tentative de pénétration entre les temps 1 et 2 (sessions réussies > 1) ainsi qu'une activité anormale de sessions échouées entre les temps 3 et 4. Une investigation de sécurité doit être menée pour clarifier ces variations.

Figure 20.15

Évolution du nombre de sessions réussies et échouées par utilisateur et par équipement



RadioVoie sous-traite son service de support

RadioVoie a constaté qu'elle ne pouvait assurer un service de support vingt-quatre heures sur vingt-quatre et sept jours sur sept pour ses clients, car elle ne dispose pas des infrastructures, outils et moyens financiers nécessaires à un support de qualité.

L'entreprise décide donc de sous-traiter ce service à une entreprise tierce, qui le lui facture au temps homme consommé.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- L'entreprise tierce partie a besoin d'accéder en permanence aux bases de connaissance, mais également aux schémas techniques des produits, hormis la technologie révolutionnaire, pour assurer un support de qualité.
- La base de connaissance est située sur le réseau bureautique de l'entreprise. Le serveur de fichiers bureautique contient les schémas techniques.

Étude de risques

Le risque principal réside dans l'accès par une entreprise tierce à des données importantes localisées au sein du réseau bureautique de l'entreprise. La méthode d'accès aux données représente un risque supplémentaire.

Une telle entreprise dispose de sa propre infrastructure, de sa propre politique de sécurité, si tant est qu'il existe une volonté de sécurité dans cette entreprise, de ses propres contraintes et de celles du pays où elle est située. Celles-ci ne sont pas nécessairement acceptables pour RadioVoie, qui doit donc négocier certaines d'entre elles et mettre en place des contre-mesures techniques destinées à pallier des failles de sécurité.

Ces mesures peuvent être appliquées au niveau réseau mais aussi au niveau applicatif.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, RadioVoie édicte une politique de sécurité réseau minimale constituée des règles suivantes :

- *« L'entreprise tierce partie accède uniquement aux informations dont elle a besoin pour assurer son service. »*
- *« Les informations auxquelles accède la tierce partie ne sont pas hébergées au sein du réseau bureautique. »*
- *« Chaque personne physique appartenant à la tierce partie qui accède au réseau de l'entreprise est identifiée et est dotée d'un accès qui lui est propre. »*
- *« Lorsqu'une machine de la tierce partie est connectée au réseau, elle ne peut plus être utilisée comme relais pour une autre machine. »*
- *« La machine de la tierce partie est protégée par une solution antivirus à jour agréée par l'entreprise. »*

- *« La tierce partie dispose d'un accès de secours en cas de défaillance de l'accès principal. »*
- *« L'authenticité de l'accès d'une tierce partie au réseau est garantie. »*
- *« La tierce partie ne peut accéder à Internet* via *la sortie de l'entreprise. »*

Solution de sécurité

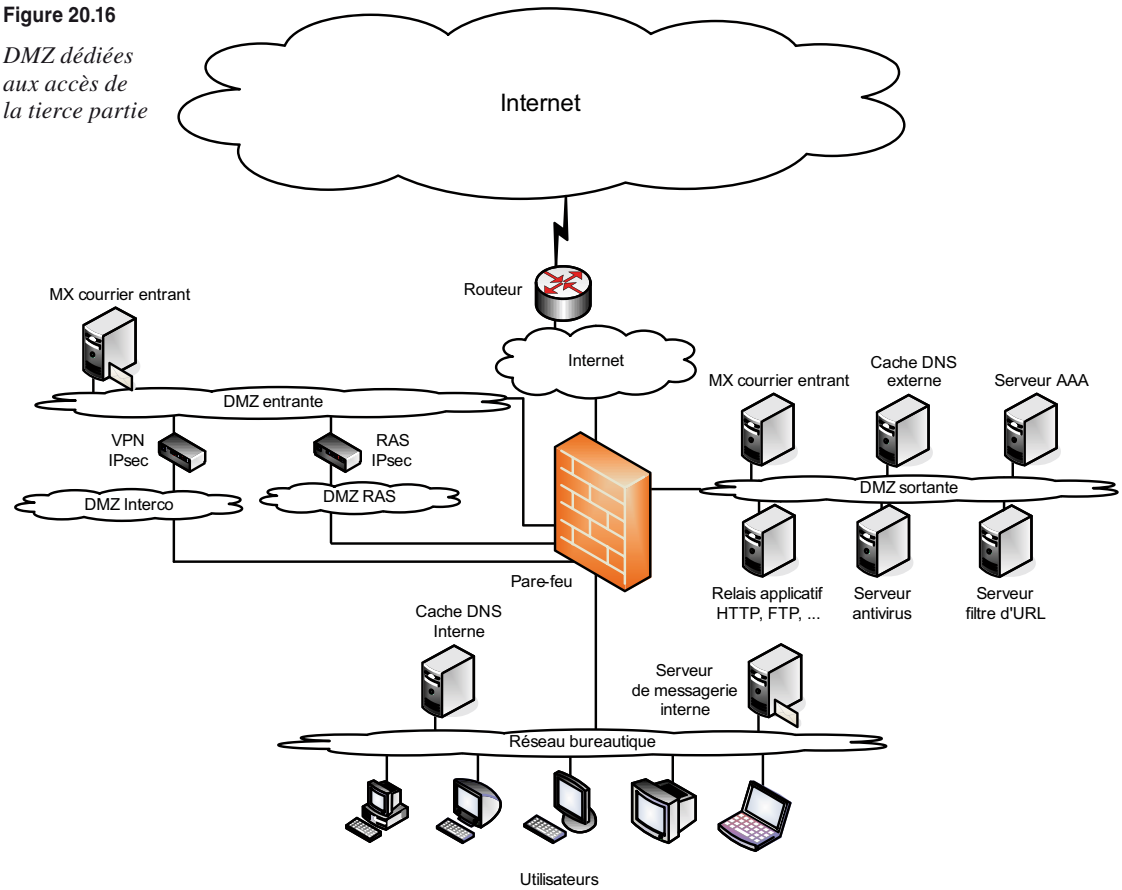
La satisfaction de ce nouveau besoin peut être résolue techniquement de la même manière que pour les accès distants des commerciaux.

Si une tierce partie peut accéder au réseau DMZ interco, elle peut être à même d'écouter les flux de données (alors en clair) et risque ainsi de compromettre la confidentialité des communications.

RadioVoie doit donc modifier son architecture Internet pour séparer les accès distants tierce partie de ceux de son réseau, comme illustré à la figure 20.20.

Figure 20.16

DMZ dédiées aux accès de la tierce partie



Un nouveau réseau est créé. Il s'agit d'une DMZ spécialisée dans l'accès distant des tierces parties (DMZ RAS). La figure 20.17 détaille les modifications de la nouvelle proposition.

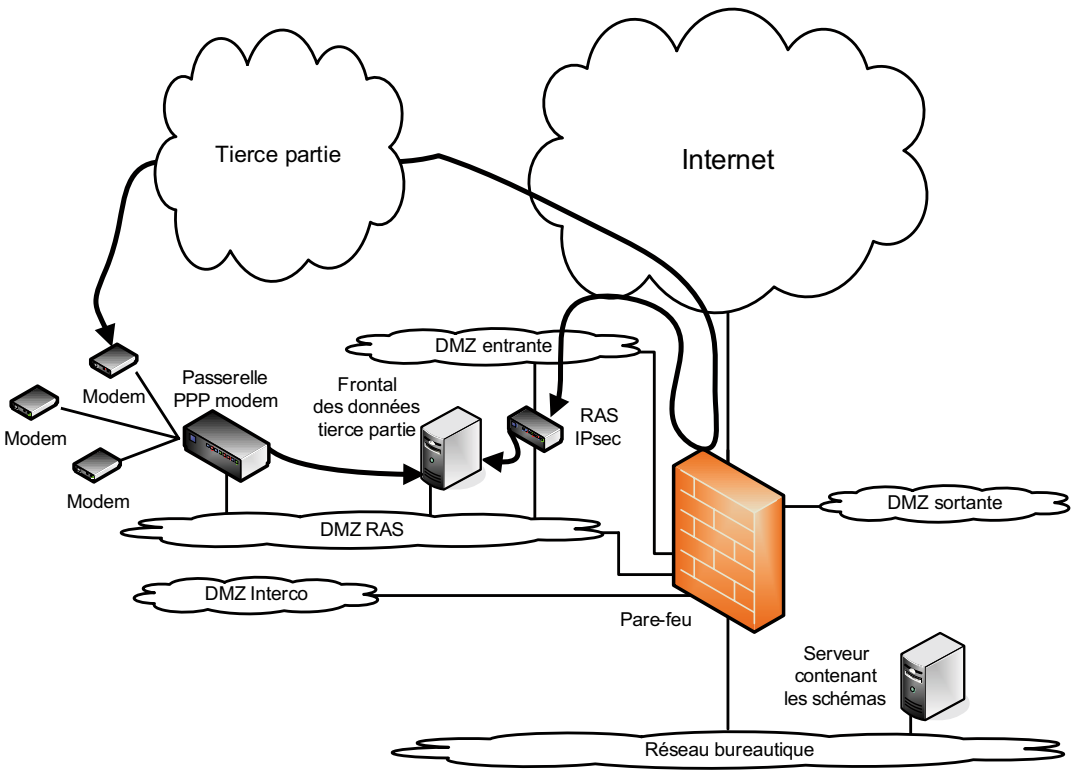


Figure 20.17

Serveur de données dédié aux accès distants de la tierce partie

Un boîtier IPsec est dédié à la tierce partie (RAS IPsec). Il est situé en parallèle du boîtier VPN IPsec déjà utilisé par RadioVoie, à cette différence près que l'interface interne est placée sur un réseau uniquement utilisé par les tierces parties.

Une solution d'accès par modem est également installée, comme l'exige la politique de sécurité. Cette solution est configurée avec une authentification de même qualité que celle utilisé sur les boîtiers IPsec (hormis le chiffrement) et avec une contrainte de rappel *(callback)* d'un numéro associé à chaque profil utilisateur.

Afin d'éviter que la tierce partie n'accède au réseau bureautique, un serveur frontal est placé dans la DMZ RAS. Ce serveur contient les données auxquelles la tierce partie a besoin d'accéder. Il ne s'agit en fait que de déplacer le serveur précédemment situé dans le réseau bureautique.

Les schémas techniques toujours situés sur le réseau bureautique sont poussés vers ce serveur frontal par le serveur bureautique, évitant d'avoir un seul flux sortant du réseau DMZ RAS, sans discrimination de destination.

Les contraintes placées par la solution VPN/IPsec (client garanti et désactivation du Split Tunneling) sont également appliquées à la tierce partie.

Enfin, un accord contractuel régit les éléments restants découlant de l'accord entre les parties (contraintes antivirus, etc.).

Risques réseau couverts

La solution adoptée permet à la tierce partie d'accéder aux informations dont elle a besoin, sans qu'aucun autre trafic n'entre dans un réseau de l'entreprise. La tierce partie se connecte de manière sécurisée et chiffrée et aboutit dans un réseau en cul de sac.

En cas de panne, la tierce partie peut s'appuyer sur une solution de connexion par modem rappelant un numéro fixe et prévenant ainsi le risque de piratage d'un compte.

Le tunnel IPsec tient à jour des listes d'accès pour autoriser la tierce partie à n'accéder qu'au serveur de données qui lui est réservé.

La tierce partie n'a aucune possibilité d'écouter des informations intéressantes sur ce réseau.

À l'autre extrémité du tunnel IPsec, la machine de la tierce partie ne peut être utilisée comme relais que pour atteindre le réseau DMZ RAS.

Risques réseau non couverts

La solution d'accès distant par modem et le boîtier IPsec sont placés sur le même réseau que la tierce partie. Si ces solutions disposent d'une administration à distance, elles peuvent être attaquées depuis ce réseau.

Si la tierce partie réussit à compromettre le boîtier IPsec ou à passer outre les filtres réseau au sein du tunnel IPsec, elle peut attaquer le pare-feu, la solution d'accès distant par modem ou le boîtier IPsec afin de les compromettre ou de les outrepasser.

Un attaquant qui aurait réussi à obtenir des données d'authentification valides pour la solution d'accès distant par modem et qui serait placé sur le réseau téléphonique entre les modems et le commutateur téléphonique public pourrait atteindre le réseau DMZ RAS en trompant la solution d'accès distant par modem, qui croirait être en contact avec le modem du numéro de rappel téléphonique.

Tableau de bord de sécurité

Cette section détaille les éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Outre les contrôles déjà assurés par la solution Internet (traces et audits réguliers des équipements), les traces du boîtier IPsec et de la solution d'accès distant par modem sont récupérées et analysées.

Les traces du serveur frontal peuvent être également collectées et analysées pour déceler des tentatives de manipulation des données hébergées.

Mise en œuvre des outils maison

Cette section détaille la corrélation d'événements et l'analyse de risques lorsque Radio-Voie sous-traite son service de support.

Corrélation d'événements

Nous détaillons ici le codage de deux règles de corrélation couvrant, pour la première, l'analyse des traces des accès distants à un réseau d'entreprise et, pour la seconde, l'analyse des traces des accès à un système réseau donné.

Règle de corrélation sur les traces des accès distants à un réseau

Les connexions d'accès distantes sont critiques et peuvent avoir des conséquences sur la sécurité du réseau interne de l'entreprise. Il est donc nécessaire d'analyser en temps réel les informations de traces disponibles.

Dans notre exemple, le format syslog pour l'analyse des connexions et déconnexions distantes de type VPN est donné par l'expression régulière suivante :

```
^(LoginSucceeded|Logout).* SrcIp=" RE_IPADDR ".* User=" RE_USERNAME
```

Nous allons écrire un programme (rta1) qui a pour objectif de détecter les connexions sur des sessions utilisateur existantes et de lever une alerte si c'est le cas. Sachant que le programme est exécuté pour chaque nouvelle entrée syslog, il doit obligatoirement gérer les contextes de session des utilisateurs.

Le pseudo-code des trois fonctions rta1, nécessaires pour écrire une corrélation d'événements par le biais de l'outil maison RTA, illustrent un tel codage :

```
/* Exécuter au démarrage du programme générique rta */
void rta1_preprocess(...)
{
  Initialisation
}

/* Exécuter pour chaque entrée syslog traitée par le programme générique rta */
void rta1_process(...)
{

  Extraction/décodage des champs du syslog

  Si il s'agit d'une connexion
  {
```

```
          Parcours de la liste pour trouver les éventuelles
          autres connexions du même usager
          {
                 Si une autre connexion est trouvée
                        Lever une alarme
          }

          Insérer la nouvelle connexion dans la liste
  }

  Si il s'agit d'une déconnexion
  {
        Retirer de la liste
  }

}

/* Exécuter en fin du programme générique rta */
void rta1_postprocess(...)
{
  Imprime le rapport des connexions actives
  Libération de la mémoire
}
```

Considérons les entrées syslog suivantes, incluant des sessions multiples pour les utilisateurs dv@tdbsr.fr et cl@tdbsr.fr :

```
margot/rta$ more ./rta1.test1
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.107 User=dv@tdbsr.fr
➥ Groups=secu TunIP=10.22.46.87
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.110 User=dv@tdbsr.fr
➥ Groups=secu
AddressAssigned Vpn=extranet Method=ipsec SrcIp=83.197.89.110 User=dv.tdbsr.fr
➥ TunIP=10.22.46.90
Logout Vpn=extranet SrcIp=83.197.89.107 User=dv@tdbsr.fr
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.110 User=cl@tdbsr.fr
➥ Groups=secu
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.111 User=cl@tdbsr.fr
➥ Groups=secu
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.112 User=cl@tdbsr.fr
➥ Groups=secu
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.113 User=cl@tdbsr.fr
➥ Groups=secu
LoginSucceeded Vpn=extranet Method=ipsec SrcIp=83.197.89.114 User=cl@tdbsr.fr
➥ Groups=secu
```

Nous obtenons les alertes suivantes (une pour l'utilisateur dv et quatre pour l'utilisateur cl) si l'on exécute le programme RTA, embarquant rta1 sur ces entrées :

```
margot/rta$ make test1
```

```
{ sleep 2; logger -i -f ./rta1.test1 -t VPN ;} &
./rta -f /var/log/rta

2009-07-20@10:01:03(GMT) RTA1 ALERT: dv@tdbsr.fr from 83.197.89.110 via margot at
➥ Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.107 via margot since Mon Jul 20 10:01:03 2009

2009-07-20@10:01:03(GMT) RTA1 ALERT: cl@tdbsr.fr from 83.197.89.111 via margot at
➥ Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.110 via margot since Mon Jul 20 10:01:03 2009

2009-07-20@10:01:03(GMT) RTA1 ALERT: cl@tdbsr.fr from 83.197.89.112 via margot at
➥ Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.111 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.110 via margot since Mon Jul 20 10:01:03 2009

2009-07-20@10:01:03(GMT) RTA1 ALERT: cl@tdbsr.fr from 83.197.89.113 via margot at
➥ Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.112 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.111 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.110 via margot since Mon Jul 20 10:01:03 2009

2009-07-20@10:01:03(GMT) RTA1 ALERT: cl@tdbsr.fr from 83.197.89.114 via margot at
➥ Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.113 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.112 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.111 via margot since Mon Jul 20 10:01:03 2009
    already logged in from 83.197.89.110 via margot since Mon Jul 20 10:01:03 2009
```

Cette règle de corrélation peut être renforcée ou faire l'objet d'une nouvelle règle en fonction des besoins de l'équipe sécurité.

Règle de corrélation sur les traces des accès à un système réseau

Les connexions d'accès à un système réseau sont critiques et peuvent avoir des conséquences sur la sécurité du réseau interne de l'entreprise. Il est donc nécessaire d'analyser en temps réel les informations de traces disponibles.

Dans notre exemple, le format syslog pour l'analyse des connexions et déconnexions *via* le protocole SSH est donné par les expressions régulières suivantes :

```
/*pour la connexion */
^(WARNING|INFO): DNS lookup( failed)? for \"" RE_IPADDR "\"

/* pour la déconnexion */
^(Local|Remote host) disconnected:
```

Nous allons écrire un programme (rta2) dont l'objectif est de détecter les attaques en force brute sur un compte utilisateur *via* le protocole SSH. Sachant que le programme est exécuté pour chaque nouvelle entrée syslog, il doit obligatoirement gérer les contextes de session des utilisateurs. Il est possible d'ajouter une clé permettant de définir une session

SSH *via* les champs *nom du système* et le numéro du processus ssh attribué par le système d'exploitation (pid).

Le pseudo-code des trois fonctions rta2, nécessaires pour écrire une corrélation d'événements par le biais de l'outil maison RTA illustrent ce codage :

```
/* Exécuter au démarrage du programme générique rta */
void rta2_preprocess(...)
{
  Initialisation
}

/* Exécuter pour chaque entrée syslog traitée par le programme générique rta */
void rta2_process(...)
{

  Extraction/décodage des champs du syslog

  Si il s'agit d'une connexion
  {
       Insérer la nouvelle connexion dans la liste
  }
  Sinon si il s'agit d'un problème d'authentification
  {
       Rechercher la session dans la liste
           Si la session est trouvée
           {
              Imprimer une alerte avec les informations stockées dans la liste
           }
           Sinon imprimer une alerte sans information de contexte
  }
  Sinon il s'agit d'une déconnexion
  {
       Retirer de la liste
  }

}

/* Exécuter en fin du programme générique rta */
void rta2_postprocess(...)
{
  Libération de la mémoire
}
```

Considérons les entrées syslog suivantes :

```
margot/rta$ logger -i -f ./rta2.test2a -t sshd ./rta2.test2a
Jul 20 12:40:27 margot sshd[8718]: WARNING: DNS lookup failed for "10.239.141.172".
Jul 20 12:40:27 margot sshd[8718]: Wrong password given for user 'denis'.
Jul 20 12:40:27 margot sshd[8718]: User denis, coming from 10.239.141.172,
➥ authenticated
```

Nous obtenons une alerte (pour l'utilisateur denis) sur l'exécution de RTA :

```
margot/rta$ make test2
{ sleep 2; \
logger -i -f ./rta2.test1a -t sshd; \
logger -i -f ./rta2.test1b -t PAM_pwdb; \
logger -f ./rta2.test1c -t su; \
logger -i -f ./rta2.test2a -t sshd; \
logger -i -f ./rta2.test2b -t sshd ;} &
./rta -f /var/log/rta

...

2009-07-20@12:53:38(GMT) RTA2 ALERT: SSH authentication failure from 10.239.141.172
    Jul 20 12:53:38 margot sshd[8718]: Wrong password given for user 'denis'
...
```

Remarquons que l'alerte donne le lien entre l'adresse IP et le problème d'authentification grâce à une clé commune basée sur *nom du système = margot* et *numéro de processus sshd = 8718*.

D'autres jeux d'exemples liés à rta2 sont fournis dans le package de l'outil RTA, illustrant d'autres types de détection. Nous laissons le soin au lecteur de les découvrir.

Analyse de risques

Un de nos objectifs majeurs est de protéger le réseau interne (réseau situé derrière le pare-feu), ainsi que les systèmes appartenant au réseau externe (routeur, pare-feu, serveur de messagerie entrant, etc.). Il est aussi primordial de déterminer un niveau de risque pour le réseau, correspondant aux vulnérabilités de sécurité détectées. Rappelons qu'il s'agit de déterminer le risque pris si ces vulnérabilités de sécurité ne sont pas corrigées.

Nous utilisons notre outil BAYES afin de connaître le niveau de risque du réseau interne et externe. La modélisation pour notre calcul de risque est la suivante. Pour chaque objet, trois tests sont possibles, pouvant référencer une ou plusieurs vulnérabilités. De plus, il y a six impacts possibles, comme le montrent les tableaux 20.5 et 20.6.

Tableau 20.5 Répartition des tests et des impacts pour le réseau externe

Objet	Test	Impact
Routeur	1	1 (impact faible)
	2	2 (impact moyen)
	3	3 (impact fort)
Pare_feu	4	1 (impact faible)
	5	2 (impact moyen)
	6	3 (impact fort)

Objet	Test	Impact
Boitier_ipsec	7	1 (impact faible)
	8	2 (impact moyen)
	9	3 (impact fort)
Serveur_mail_entrant	10	1 (impact faible)
	11	2 (impact moyen)
	12	3 (impact fort)

Tableau 20.6 Répartition des tests et des impacts pour le réseau interne

Objet	Test	Impact
Reseau_interne	13	4 (impact faible)
	14	5 (impact moyen)
	15	6 (impact fort)

Dans ce modèle, si nous tenons compte de la topologie réseau et si nous considérons que les attaques viennent uniquement de l'extérieur, les règles de propagation sont les suivantes :

```
margot/20.3/bayes$ cat dmz.rule
0 routeur routeur 1 2 3    # règles de propagation à la racine
0 parefeu parefeu 4 5 6
0 boitier_ipsec boitier_ipsec 7 8 9
0 serveur_mail_entrant serveur_mail_entrant 10 11 12
1 routeur routeur 1        # règles de propagation hors racine
2 routeur routeur 2
3 routeur routeur 3
3 routeur parefeu 4 5 6
3 routeur boitier_ipsec 7 8 9
3 routeur serveur_mail_entrant 10 11 12
4 parefeu parefeu 4
5 parefeu parefeu 5
6 parefeu parefeu 6
7 boitier_ipsec boitier_ipsec 7
8 boitier_ipsec boitier_ipsec 8
9 boitier_ipsec boitier_ipsec 9
6 parefeu reseau_interne 13 14 15
6 parefeu boitier_ipsec 7 8 9
6 parefeu serveur_mail_entrant 10 11 12
6 parefeu routeur 1 2 3
10 serveur_mail_entrant serveur_mail_entrant 10
11 serveur_mail_entrant serveur_mail_entrant 11
12 serveur_mail_entrant serveur_mail_entrant 12
```

```
12 serveur_mail_entrant parefeu 4 5 6
12 serveur_mail_entrant boitier_ipsec 7 8 9
13 reseau_interne reseau_interne 13
14 reseau_interne reseau_interne 14
15 reseau_interne reseau_interne 15
```

Si nous prenons en compte les fichiers de conséquences et de probabilités suivants :

```
margot/20.3/bayes $ cat dmz.cons
0   /* pas d'impact */
10  /* impact faible : réseau externe */
25  /* impact moyen : réseau externe */
50  /* impact fort : réseau externe */
10  /* impact faible : réseau interne */
25  /* impact moyen : réseau interne */
50  /* impact fort : réseau interne */

margot/20.3/bayes$ cat dmz.proba
0.1  /* pas d'impact */
0.3  /* impact faible : réseau externe */
0.3  /* impact moyen : réseau externe */
0.8  /* impact fort : réseau externe */
0.3  /* impact faible : réseau interne */
0.3  /* impact moyen : réseau interne */
0.8  /* impact fort : réseau interne */
```

Nous pouvons exécuter le programme BAYES pour chacun des fichiers de vulnérabilités détectés par les contrôles internes et externes.

Le Makefile suivant permet de lancer une simulation composée de six fichiers en considérant les mêmes paramètres de règles, conséquences et probabilités :

```
margot/20.3/bayes$ cat Makefile
PGM=bayes

dmz:
        normalise dmz.rule dmz.proba dmz.txt dmz.cons
        $(PGM) dmz.txt.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt1 dmz.cons
        $(PGM) dmz.txt1.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt2 dmz.cons
        $(PGM) dmz.txt2.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt3 dmz.cons
        $(PGM) dmz.txt3.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt4 dmz.cons
        $(PGM) dmz.txt4.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt5 dmz.cons
        $(PGM) dmz.txt5.ref.dat[1234] 100
```

Nous exécutons le programme BAYES sur les différents fichiers contenant les vulnérabilités de sécurité :

```
margot/20.3/bayes$ gmake dmz

normalise dmz.rule dmz.proba dmz.txt dmz.cons
bayes dmz.txt.ref.dat[1234] 100

---------------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:6 | nb de vulnérabilités:1 | impact:3
    test:13 | nb de vulnérabilités:1 | impact:4
    test:14 | nb de vulnérabilités:3 | impact:5
    test:15 | nb de vulnérabilités:2 | impact:6
nb_impacts (7) =  0  0  0  0  0  0  3
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01
➥ 2.500000e+01  5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
---------------------------------------------------------------------
distribution des probabilités (impacts): 2.226400e-01  0.000000e+00  0.000000e+00
➥ 7.200000e-01  4.500000e-03  2.646000e-02  2.640000e-02 / somme=1.000000e+00
risque : 3.802650e+01
nombre de noeuds de l'arbre : 8.000000e+00
nombres de feuilles par impact: 8.000000e+00  0.000000e+00
➥ 0.000000e+00  1.000000e+00  1.000000e+00  3.000000e+00
➥ 2.000000e+00 / somme=1.500000e+01
profondeur de l'arbre : 4
---------------------------------------------------------------------

normalise dmz.rule dmz.proba dmz.txt1 dmz.cons
bayes dmz.txt1.ref.dat[1234] 100

---------------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:4 | nb de vulnérabilités:1 | impact:3
    test:7 | nb de vulnérabilités:1 | impact:1
    test:8 | nb de vulnérabilités:1 | impact:2
    test:13 | nb de vulnérabilités:1 | impact:4
    test:14 | nb de vulnérabilités:3 | impact:5
    test:15 | nb de vulnérabilités:2 | impact:6
nb_impacts (7) =  0  0  0  0  3  0  0
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01  2.500000e+01
➥ 5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
---------------------------------------------------------------------
distribution des probabilités (impacts): 5.800000e-01  9.000000e-02  9.000000e-02
➥ 2.400000e-01  0.000000e+00  0.000000e+00  0.000000e+00 / somme=1.000000e+00
risque : 1.515000e+01
nombre de noeuds de l'arbre : 4.000000e+00
```

```
nombres de feuilles par impact: 4.000000e+00  1.000000e+00  1.000000e+00
➥ 1.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 / somme=7.000000e+00
profondeur de l'arbre : 1
-----------------------------------------------------------------

normalise dmz.rule dmz.proba dmz.txt2 dmz.cons
bayes dmz.txt2.ref.dat[1234] 100

-----------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:10 | nb de vulnérabilités:5 | impact:1
    test:13 | nb de vulnérabilités:1 | impact:4
    test:14 | nb de vulnérabilités:3 | impact:5
    test:15 | nb de vulnérabilités:2 | impact:6
nb_impacts (7) =  0  0  0  0  0  0  0
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01  2.500000e+01
➥ 5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
-----------------------------------------------------------------
distribution des probabilités (impacts): 3.774880e-01  6.225120e-01  0.000000e+00
➥ 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 / somme=1.000000e+00
risque : 6.225120e+00
nombre de noeuds de l'arbre : 6.000000e+00
nombres de feuilles par impact: 6.000000e+00  5.000000e+00  0.000000e+00
➥ 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 / somme=1.100000e+01
profondeur de l'arbre : 5
-----------------------------------------------------------------

normalise dmz.rule dmz.proba dmz.txt3 dmz.cons
bayes dmz.txt3.ref.dat[1234] 100

-----------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:6 | nb de vulnérabilités:5 | impact:3
    test:7 | nb de vulnérabilités:1 | impact:1
    test:13 | nb de vulnérabilités:8 | impact:4
nb_impacts (7) =  0  0  0  0  0  0  3
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01  2.500000e+01
➥ 5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
-----------------------------------------------------------------
distribution des probabilités (impacts): 2.990784e-01  4.679007e-02  0.000000e+00
➥ 6.189316e-01  3.520001e-02  0.000000e+00  0.000000e+00 / somme=1.000000e+00
risque : 3.176648e+01
nombre de noeuds de l'arbre : 5.200000e+01
nombres de feuilles par impact: 5.200000e+01  6.000000e+00  0.000000e+00
```

```
➥ 5.000000e+00  4.000000e+01  0.000000e+00  0.000000e+00 / somme=1.030000e+02
profondeur de l'arbre : 13
-----------------------------------------------------------------

normalise dmz.rule dmz.proba dmz.txt4 dmz.cons
bayes dmz.txt4.ref.dat[1234] 100

-----------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:6 | nb de vulnérabilités:1 | impact:3
    test:13 | nb de vulnérabilités:1 | impact:4
    test:14 | nb de vulnérabilités:3 | impact:5
    test:15 | nb de vulnérabilités:2 | impact:6
nb_impacts (7) =  0  0  0  0  0  0  3
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01  2.500000e+01
➥ 5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
-----------------------------------------------------------------
distribution des probabilités (impacts): 2.226400e-01  0.000000e+00  0.000000e+00
➥ 7.200000e-01  4.500000e-03  2.646000e-02  2.640000e-02 / somme=1.000000e+00
risque : 3.802650e+01
nombre de noeuds de l'arbre : 8.000000e+00
nombres de feuilles par impact: 8.000000e+00  0.000000e+00  0.000000e+00
➥ 1.000000e+00  1.000000e+00  3.000000e+00  2.000000e+00 / somme=1.500000e+01
profondeur de l'arbre : 4
-----------------------------------------------------------------

normalise dmz.rule dmz.proba dmz.txt5 dmz.cons
bayes dmz.txt5.ref.dat[1234] 100

-----------------------------------------------------------------
nb_vulnerabilités par test :
    test:0 | nb de vulnérabilités:1 | impact:0
    test:1 | nb de vulnérabilités:7 | impact:1
    test:2 | nb de vulnérabilités:7 | impact:2
    test:14 | nb de vulnérabilités:1 | impact:5
    test:15 | nb de vulnérabilités:2 | impact:6
nb_impacts (7) =  0  1  2  0  0  0  0
nb_probabilités (7 impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
➥ 8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (7 impacts) =  0.000000e+00  1.000000e+01  2.500000e+01
➥ 5.000000e+01  1.000000e+01  2.500000e+01  5.000000e+01
-----------------------------------------------------------------
distribution des probabilités (impacts): 3.438957e-01  3.280522e-01  3.280522e-01
➥ 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 / somme=1.000000e+00
risque : 1.148183e+01
nombre de noeuds de l'arbre : 1.500000e+01
nombres de feuilles par impact: 1.500000e+01  7.000000e+00  7.000000e+00
```

```
➥ 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 / somme=2.900000e+01
profondeur de l'arbre : 7
---------------------------------------------------------------------
```

Exemple de tableau de bord de sécurité réseau

Le tableau 20.7 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 20.7 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et des routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et des routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, analyse des configurations des VLAN, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et les systèmes connectés
Internet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateurs, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)

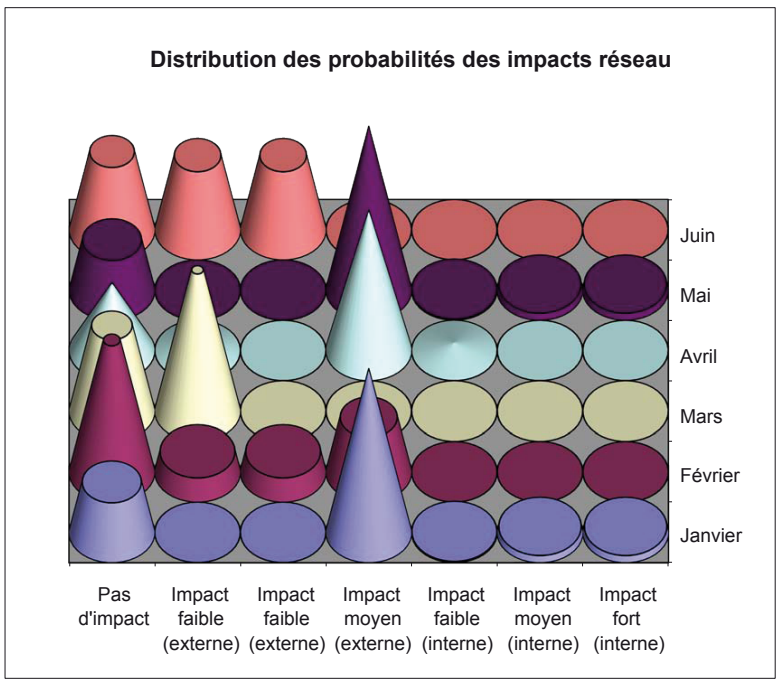
Sous-réseau	Catégorie	Élément
Tierce partie (suite)	Événement réseau	Des commutateurs (accès non autorisés, etc.), modems (routeurs accès non autorisés, etc.), boîtier IPsec (sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous calculons tous les scénarios d'événements possibles par le biais d'un arbre probabiliste (fondé sur les faiblesses de sécurité préalablement détectées), il est possible de déterminer les probabilités associées pour chaque niveau d'impact, comme l'illustre la figure 20.18.

Figure 20.18

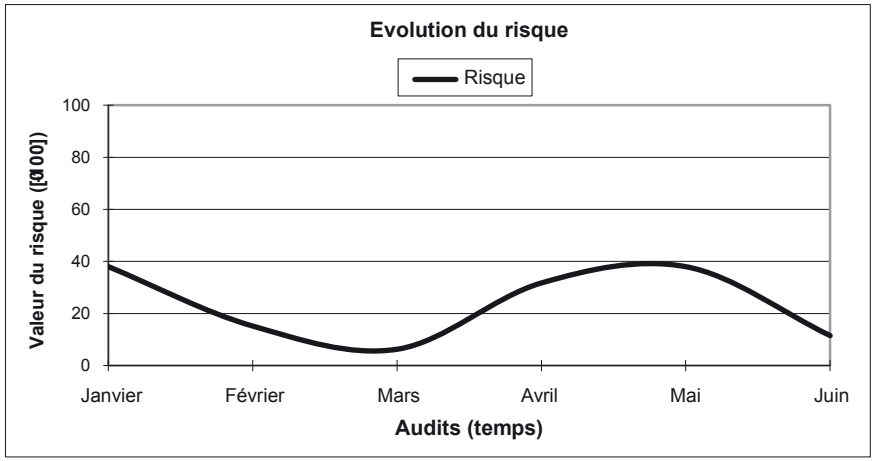
Distribution des probabilités des impacts réseau pour les trois derniers mois



Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées à ces impacts pour calculer le risque associé à la non-application de la politique de sécurité. Le risque est alors calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau, comme l'illustre la figure 20.19.

Figure 20.19

Évolution du risque dans le temps



En résumé

Le réseau de RadioVoie ainsi que les règles de sécurité ont évolué dans le temps avec les besoins de l'entreprise. Ces évolutions montrent que la politique de sécurité réseau et les solutions techniques doivent être remises en cause afin de s'adapter à chaque nouvelle contrainte.

Les solutions et architectures mises en œuvre intègrent des équipements de sécurité tels que des pare-feu, des boîtiers de chiffrement, etc. Nous avons détaillé pour chaque solution proposée les risques couverts et les risques restants.

Des contrôles de sécurité ont été également proposés afin de valider les points critiques de chaque solution technique. De manière plus générale, ces contrôles doivent être simples et facilement automatisables.

Le chapitre suivant détaille l'évolution du réseau de RadioVoie vers une structure de type multinationale.

21

RadioVoie étend son réseau

Ce dernier chapitre couvre la prise en compte d'une extension classifiée Secret Défense, ainsi que l'extension du réseau RadioVoie à l'international.

Pour chaque évolution du réseau de RadioVoie, nous détaillons l'analyse des besoins, la définition de la politique de sécurité, les solutions techniques et les contrôles de sécurité, les risques couverts et non couverts par la solution technique proposée, ainsi que l'établissement d'un tableau de bord de sécurité.

RadioVoie négocie un contrat militaire

L'entreprise a convaincu le ministère de la Défense que son émetteur-récepteur pouvait, par des modifications mineures, offrir à l'armée une meilleure efficacité sur le terrain, grâce des connexions permanentes entre chaque soldat et le commandement.

Ce marché stratégique pour le développement de l'entreprise a pu être obtenu parce que RadioVoie a accepté les contraintes draconiennes de l'armée. Ces contraintes exigent que RadioVoie fabrique elle-même ses produits, au lieu de les sous-traiter. L'entreprise choisit donc d'installer une unité de production ainsi qu'une équipe de recherche et développement sur son site de Mouans-Sartoux, déjà utilisé par le personnel administratif.

Une équipe de spécialistes militaires en sécurité des communications (chiffrement) est hébergée chez RadioVoie. Cette équipe a pour mission d'effectuer la recherche et développement de l'unité qui assure le chiffrement des données. Souveraine sur son périmètre, c'est elle qui décide qui peut accéder à ses locaux.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- Fournir un local répondant aux spécifications des militaires concernant la production.
- Fournir un local répondant aux spécifications des militaires pour l'hébergement de son équipe de spécialistes.
- Fournir un local pour l'équipe de recherche et développement de RadioVoie et s'assurer que les réseaux au sein des unités de recherche de RadioVoie communiquent *via* des flux chiffrés.
- S'assurer que le réseau recherche et développement de Mouans-Sartoux accède aux autres réseaux de RadioVoie et à Internet *via* le site de Paris.
- Prévoir un emplacement réseau pour le serveur d'authentification des contrôles d'accès physiques aux locaux classés Secret Défense. Les labels sont CD (Confidentiel Défense), SD (Secret Défense), et TSD (Top Secret Défense).

Étude de risques

Les contraintes physiques appliquées aux locaux classés Secret Défense ont été fournies par l'armée. Cette dernière a effectué des audits de sécurité et prévu d'auditer régulièrement par la suite afin de valider l'application de ces contraintes.

Les risques liés aux contraintes physiques (épaisseur des murs, résistance des portes, protection incendie, etc.) étant hors du propos de cet ouvrage, ils ne sont pas détaillés à une exception près : le serveur d'authentification pour l'accès aux locaux, lequel ne peut être unique et situé physiquement au sein du local dont il est chargé de protéger l'accès. En cas de refus de service de ce serveur, les locaux deviendraient en effet inaccessibles.

L'unité de production doit être isolée physiquement de tout autre réseau. Si le bâtiment dispose d'une infrastructure physique globale, il existe un risque que des connexions physiques soient établies ultérieurement, au niveau des armoires de brassage, par exemple. On peut aussi considérer un piratage physique des connexions si elles sont accessibles depuis l'extérieur de l'unité de production ou du local des spécialistes. Cela vaut également pour le contrôle d'accès au local des spécialistes.

Les machines utilisées par les spécialistes militaires devant être hors de portée de l'entreprise, il n'est pas facile de s'assurer que ces machines respectent les standards, alors même qu'elles ont la possibilité d'accéder aux autres réseaux de l'entreprise puisque la solution protégeant leur réseau est sous leur contrôle.

En cas de déni de service du lien entre les sites de Paris et Mouans-Sartoux, le lien entre les unités de recherche de RadioVoie deviendrait également hors service.

Politique de sécurité réseau

La politique de sécurité de l'armée édicte les règles suivantes :

- *« La fabrication des produits est réalisée dans une unité de production spécifique classée Secret Défense. »*

- *« Le réseau de l'unité de production est isolé physiquement de tout autre réseau. »*
- *« Tout réseau au sein d'un local classé Secret Défense ou lui-même classé comme tel est isolé logiquement de tout autre réseau. »*
- *« Le contrôle d'accès à tout réseau au sein d'un local classé Secret Défense ou lui-même classé comme tel est sous l'autorité militaire. »*
- *« Le contrôle d'accès physique à tout local classé Secret Défense est sous l'autorité militaire. »*
- *« Un local classé Secret Défense est mis à disposition de l'équipe de militaires spécialistes en communication. Ce local ne peut être situé au sein de l'unité de production. »*

RadioVoie ajoute par ailleurs des contraintes à sa politique existante :

- *« Les communications réseau entre les unités de recherche sont chiffrées. »*
- *« Le réseau recherche et développement de Mouans-Sartoux accède aux réseaux de RadioVoie par l'interconnexion du réseau recherche et développement de Paris. »*

Solution de sécurité

La satisfaction des besoins d'interconnexion des unités de recherche est simple à satisfaire. Il suffit de créer un réseau virtuel au-dessus du réseau bureautique (tunnel IPsec dans le tunnel IPsec utilisé pour les communications entre sites). Cela donne l'architecture illustrée à la figure 21.1.

Les flux permettant d'atteindre un quelconque réseau depuis le réseau recherche et développement de Mouans-Sartoux sont routés *via* le boîtier IPsec qui les chiffre.

Si le flux n'est pas pour le réseau recherche et développement de Mouans-Sartoux, il passe par le boîtier IPsec de Mouans-Sartoux (si le flux est autorisé) pour être chiffré. Il traverse ensuite le réseau bureautique, rejoint Paris par l'interconnexion intersite, traverse le pare-feu du réseau recherche et développement de Paris et entre dans le boîtier IPsec de ce dernier, où il est déchiffré.

Si les flux sont destinés au réseau recherche et développement de Paris, le trafic s'arrête là. S'ils sont destinés au réseau bureautique, à Internet ou au réseau de recherche et développement militaire, ils passent par le pare-feu de recherche et développement de Paris (cette fois en clair) pour aller vers leur destination. La politique de filtrage du pare-feu Internet décide si le trafic peut atteindre Internet.

Nous avons donc un tunnel pour l'interconnexion entre les unités de recherche et développement au sein du tunnel d'interconnexion entre les sites, comme illustré à la figure 21.2.

L'unité de production doit satisfaire les contraintes militaires de sécurité physique. Pour la partie réseau, RadioVoie se contente de s'assurer que la connectivité physique du réseau de production est bien localisée au sein des locaux, et donc inaccessible de l'extérieur, et qu'elle dépend d'une armoire de brassage dédiée, également située au sein du local de production. Ces mêmes contraintes s'appliquent au local de recherche et développement réservé aux spécialistes militaires.

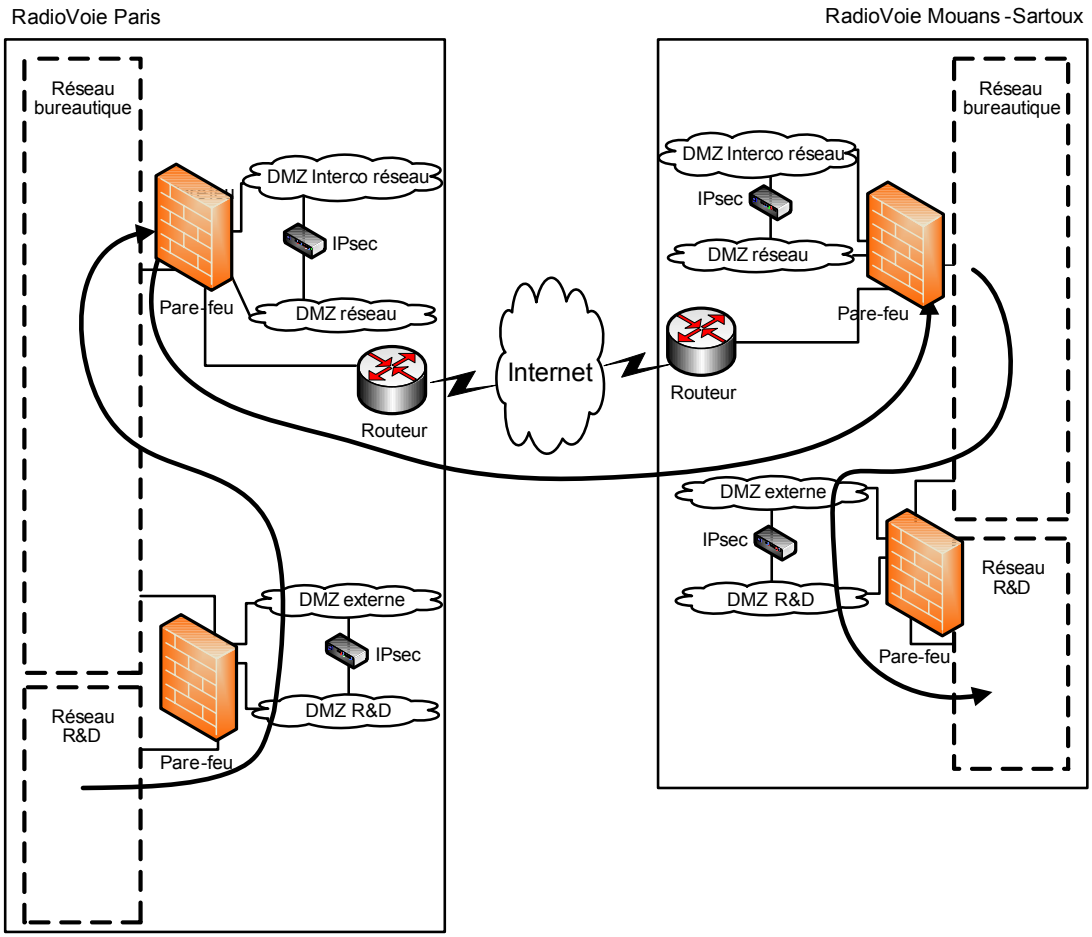
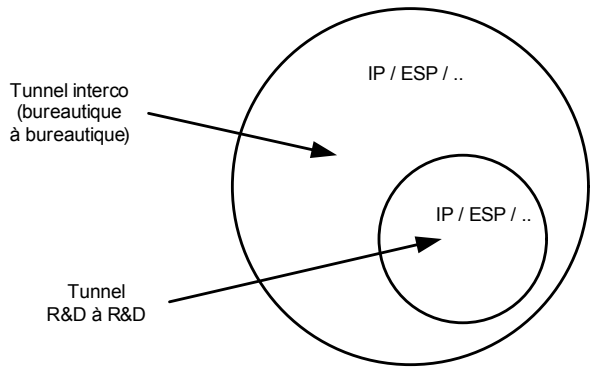


Figure 21.1

Architecture sécurisée d'interconnexion entre les sites de RadioVoie

Figure 21.2

Les différents niveaux de tunnel



L'architecture physique illustrée à la figure 21.3 est proposée aux militaires pour prévenir le risque de pénétration physique des locaux de production ou du local de recherche et développement militaire. C'est l'autorité militaire qui est responsable du contrôle des systèmes d'accès.

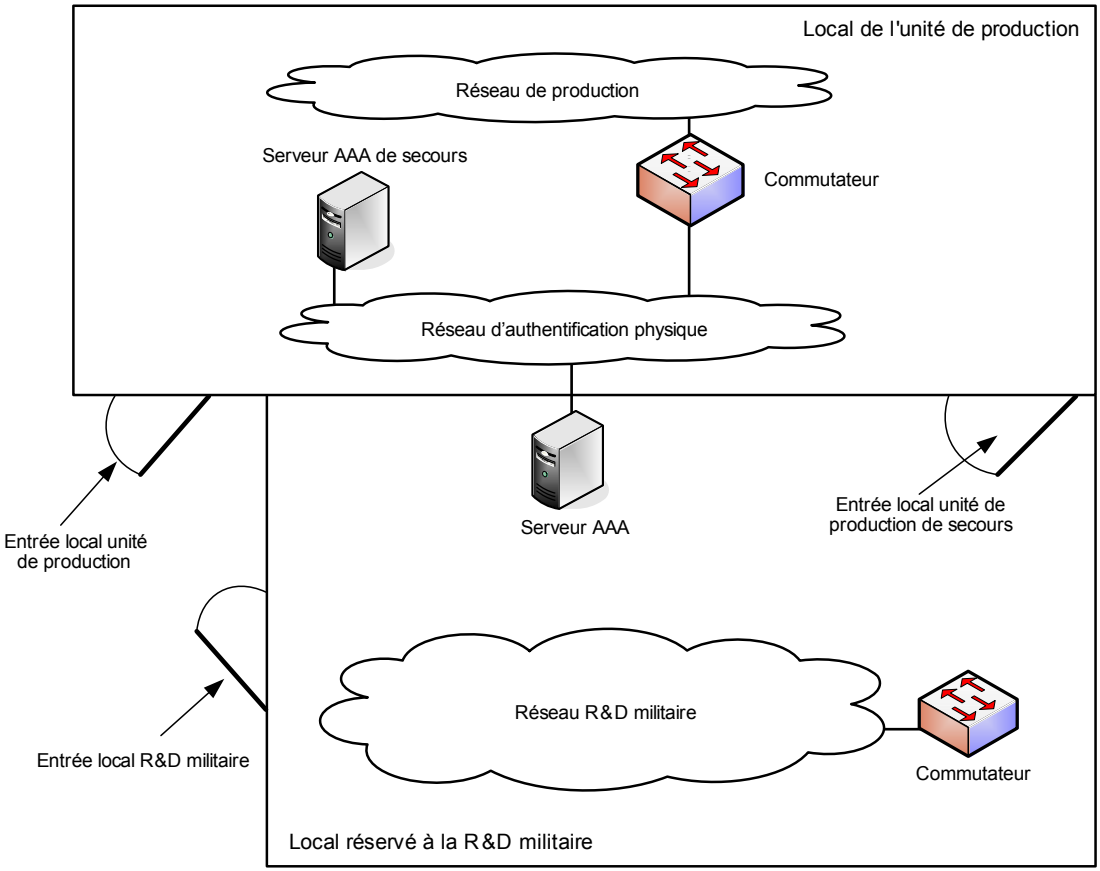


Figure 21.3

Sécurisation physique du site de Mouans-Sartoux

Les équipements réseau tels que les armoires de brassage et les commutateurs sont séparés et enfermés dans une pièce sécurisée au sein de l'unité de production.

Au commutateur du réseau de production est raccordé un réseau d'authentification, qui connecte les serveurs d'authentification, de contrôle d'accès et de surveillance. Le commutateur implémente le contrôle d'accès au niveau MAC sur le VLAN d'authentification.

Afin de limiter le risque de ne pouvoir entrer dans l'unité de production en cas de panne d'un serveur, ce sont deux serveurs d'authentification qui sont installés, dont l'un est un

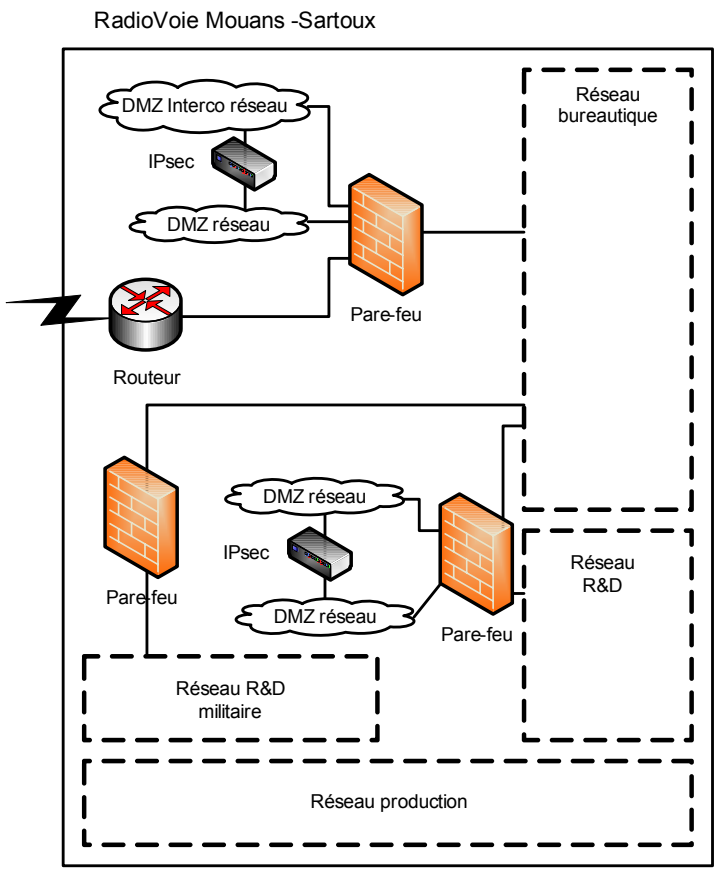
serveur de secours répliquant les données depuis le primaire. Ces serveurs sont situés dans les deux locaux dont l'accès est sous le contrôle des militaires. Il existe une entrée de secours entre le local de recherche et développement militaire et l'unité de production en cas d'ultime besoin.

À l'analyse de cette architecture, nous constatons que l'accès est bien sous le contrôle des militaires dans les deux cas. De plus, le réseau d'authentification est indépendant du réseau de production afin de limiter les risques d'attaque vers les équipements de contrôle et de surveillance des accès. Enfin, le réseau de recherche et développement militaire reste un réseau indépendant (jusqu'à son commutateur) et ne peut être utilisé pour atteindre le réseau de production.

La connexion du réseau de recherche et développement militaire sur le site de Mouans-Sartoux est des plus simple, comme l'illustre la figure 21.4.

Figure 21.4

Architecture sécurisée du réseau recherche et développement



Un pare-feu situé physiquement au sein du local de recherche et développement des militaires et sous son contrôle isole logiquement le réseau militaire du reste de l'entreprise. Ce pare-feu est relié au commutateur de recherche et développement militaire, d'une part, et à celui de l'entreprise, d'autre part, limitant ainsi les risques associés aux attaques de commutateur.

Risques réseau couverts

RadioVoie s'est assuré de l'isolation physique des réseaux classés Secret Défense. Le risque de détournement des liens physiques tend donc vers zéro.

Le contrôle d'accès physique est réalisé en deux endroits distincts disposant d'accès différents. Le risque de ne pouvoir accéder aux locaux suite à un refus de service des serveurs d'authentification d'accès est donc minimal.

Tous les équipements en charge de l'authentification d'accès étant isolés logiquement, le risque de piratage par le réseau est minimal.

Le réseau de recherche et développement militaire étant logiquement isolé des autres réseaux et la solution d'isolation sous le contrôle militaire, le risque de pénétration est minimal.

Les réseaux de recherche et développement de RadioVoie s'échangent bien des informations de manière chiffrée *via* IPsec, et il existe un goulet d'étranglement pour les échanges non chiffrés entre le réseau global de recherche et développement de Paris et les autres réseaux.

Risques réseau non couverts

Hormis les risques associés à une intrusion physique permettant de compromettre un équipement réseau, sujet que nous ne traitons pas dans le contexte de cet ouvrage, plusieurs risques demeurent.

La perte du commutateur de l'unité de production peut engendrer un refus de service global et une impossibilité d'accéder aux locaux par les moyens normaux. Ce risque peut être pallié en doublant le VLAN d'authentification et en reliant les cœurs des commutateurs entre eux *(backpane).*

Dans ce cas, il est nécessaire que chaque équipement associé à l'authentification d'accès soit connecté à chacun des VLAN d'authentification, comme illustré à la figure 21.5.

Un autre risque est la possibilité d'accéder au commutateur par le biais de son interface d'administration à distance. Cette interface n'existe que sur le VLAN d'authentification.

Compte tenu de l'isolation du réseau d'administration, il n'y a pas de traçabilité des flux réseau sur ce VLAN. Si un équipement vient à être compromis, il peut être impossible de déterminer la source de l'intrusion. Bien sur, il est possible de placer un équipement réseau pour, par exemple, collecter les traces et les stocker, telle une sonde d'intrusion, par exemple.

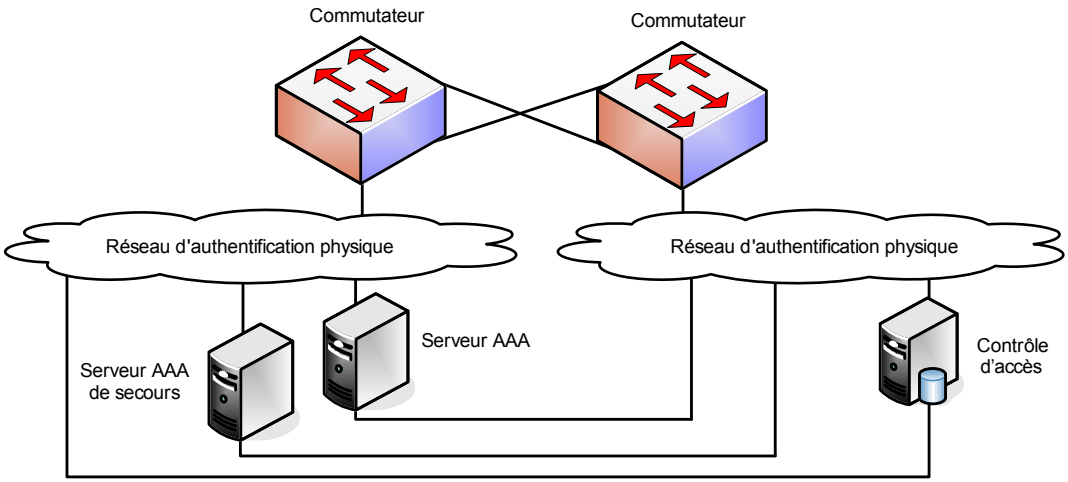


Figure 21.5

Architecture de haute disponibilité des serveurs AAA

Tableau de bord de sécurité

Cette section détaille les principaux contrôles de sécurité à mettre en place et fournit des éléments de vérification fondés sur nos outils maison, ainsi qu'un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le contrôle de sécurité le plus délicat se situe au niveau des systèmes du réseau de recherche et développement militaire. Techniquement, ce réseau est une menace pour l'entreprise puisqu'il échappe à son contrôle.

Les militaires peuvent installer des modems entrants, faire entrer des virus, etc., sur le réseau bureautique. Il faut dès lors s'appuyer sur une politique de contrôle acceptée par les militaires. Une telle politique peut prévoir des contrôles de sécurité effectués par RadioVoie sous la tutelle de l'autorité militaire du site ainsi que l'engagement des militaires de respecter les standards de l'entreprise en matière de protection antivirus et d'accès à Internet.

Si nécessaire, RadioVoie peut placer en frontal du pare-feu de recherche et développement militaire un pare-feu vérifiant les flux sortants du pare-feu militaire.

Comme pour tous les commutateurs, la configuration doit être régulièrement vérifiée afin de s'assurer qu'elle respecte le standard.

Les traces collectées par tous les équipements (contrôles d'accès, serveurs d'authentification, commutateur, pare-feu, etc.) sont analysées de manière humaine ou automatisée afin de détecter les comportements déviants.

Sous l'autorité des militaires, des audits réguliers peuvent être effectués sur les équipements de contrôle d'accès et les pare-feu afin de s'assurer de l'application de la politique de sécurité.

Mise en œuvre des outils maison

Cette section décrit la mise en œuvre de nos outils maison afin de répondre aux besoins de sécurité de RadioVoie. Elle détaille dans ce contexte la vérification des configurations des VPN IPsec et la vérification des périmètres réseau correspondants.

Corrélation d'événements

Non seulement les traces provenant de systèmes critiques doivent être analysées, mais on doit aussi s'assurer de leur disponibilité.

Nous allons écrire un programme (rta3) dont l'objectif est de détecter si les systèmes critiques émettent bien des syslog. La liste des systèmes critiques doit être définie au niveau du programme, de même que la période maximale autorisée pendant laquelle un système peut ne pas recevoir de syslog. On lancera les processus syslogd sur les systèmes critiques avec l'option *-m 1* indiquant que le processus doit générer un événement par minute afin d'indiquer qu'il est vivant.

En conséquence, le programme rta3 sera exécuté au pire des cas toutes les minutes si aucune autre entrée syslog n'est reçue (la période maximale autorisée étant obligatoirement supérieure à une minute).

Le pseudo-code des trois fonctions rta3, nécessaires pour écrire une corrélation d'événements, illustrent un tel codage :

```
/* Exécuter au démarrage du programme générique rta */
void rta3_preprocess(...)
{
  Initialisation de la liste des systèmes à contrôler, ainsi que
  les temps de non réception des traces

  Initialisation du temps max d'inactivité
}

/* Exécuter pour chaque entrée syslog traitée par le programme générique rta */
void rta3_process(...)
{

  Extraction/décodage des champs du syslog

  Pour chaque système de la liste
  {
      Si le nom_du_système est égal au nom_du_système
      émetteur du syslog
      {
         Mise à zéro du temps de non réception des traces
         pour ce système
```

```
      }
      Sinon Si le temps de non réception des traces dépasse
      le temps max
      {
          Imprimer une alerte de non réception de syslog
          pour ce système
      }

  }

}

/* Exécuter en fin du programme générique rta */
void rta3_postprocess(...)
{
  Libèrer la memoire
}
```

Considérons la liste des systèmes suivants nécessitant d'être contrôlés :

```
margot/rta$ more ./rta3.c
...
rta3_systems[] =
{
 { "langevin",  NULL },
 { "margot",  NULL },
 { "gw1.tdbsr.fr", NULL },
 { "gw2.tdbsr.fr", NULL },
 { "gw3.tdbsr.fr", NULL },
 { "www.tdbsr.fr", NULL }

...
```

On obtient alors les alertes suivantes (par rta3) si aucune entrée n'est détectée et si seul le daemon « syslogd » ne s'exécute sur le système margot :

```
margot/rta$ make test3
./rta -f /var/log/rta

2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring gw1.tdbsr.fr
2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring gw2.tdbsr.fr
2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring gw3.tdbsr.fr
2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring langevin
2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring margot
2009-07-20@15:39:00(GMT) RTA3 INFO: monitoring www.tdbsr.fr

2009-07-20@15:42:57(GMT) RTA3 ALERT: system gw1.tdbsr.fr inactive since Mon Jul 20
➥ 15:39:00 2009
2009-07-20@15:42:57(GMT) RTA3 ALERT: system gw2.tdbsr.fr inactive since Mon Jul 20
➥ 15:39:00 2009
```

```
2009-07-20@15:42:57(GMT) RTA3 ALERT: system gw3.tdbsr.fr inactive since Mon Jul 20
➥ 15:39:00 2009
2009-07-20@15:42:57(GMT) RTA3 ALERT: system langevin inactive since Mon Jul 20
➥ 15:39:00 2009
2009-07-20@15:42:57(GMT) RTA3 ALERT: system www.tdbsr.fr inactive since Mon Jul 20
➥ 15:39:00 2009
2009-07-20@15:42:57(GMT) RTA3 ALERT: system 10.237.0.166 inactive since Mon Jul 20
➥ 15:39:00 2009
```

Remarquons une alerte sur presque tous les systèmes à contrôler au bout de deux minutes (durée fixée pour la non-activité maximale des syslogs), excepté pour le système margot, sur lequel s'exécute le daemon syslogd générant par lui-même un événement chaque minute :

```
Margot/rta$ ps -aux | grep syslogd
root     30875  0.0  0.5  1428  732 ?        S    Jul10   0:37 syslogd -r -m 2
```

Cette règle de corrélation pourra donc être renforcée ou faire l'objet d'une nouvelle règle en fonction des besoins de l'équipe sécurité.

Analyse des configurations

Les configurations IPsec doivent être analysées afin de détecter toute mauvaise configuration à l'aide du patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration Cisco suivant :

```
margot/21.1/hawk$ cat conf.txt
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
!
```

La justification des éléments de configuration est fournie à la partie IV de l'ouvrage, relative à la configuration des équipements réseau.

Pour analyser ces configurations, nous utilisons l'outil HAWK avec le patron de sécurité suivant en mode strict :

```
margot/21.1/hawk$ cat ipsec.tp
DECL {
   str strict_state;
}
```

```
BEGIN {
      strict_state = "crypto isakmp key";
}

*!:crypto isakmp key

+ (
    [e]:crypto isakmp key [0-9a-z]+ address [0-9]{1,3}(\.[0-9]{1,3}){3}
)
{ strict_state = "crypto isakmp policy"; }

*!:crypto isakmp policy

+ (
    [e]:crypto isakmp policy [0-9]+
    (
            : encr 3des
            : hash md5
            : authentication pre-share
            : group 2
    )
)

*:.*

SUCCESS {
}

FAILURE {
printf("%s;%s (line %d);(erreur) configuration non conforme au mode strict;%s\n",
➥ FILENAME, LINE, LINENO, strict_state);
}
```

Si nous exécutons le programme HAWK sur une configuration qui ne respecte pas le patron de sécurité (sur la clé), nous obtenons les résultats suivants :

```
margot/21.1/hawk$ hawk -f ./ipsec.tp ./conf1.txt
./conf.txt;crypto isakmp key G4cewao6wcbw83 address 192.168.1.154 (line 3);(erreur)
➥ configuration non conforme au mode strict;crypto isakmp key
```

Cet exemple illustre une erreur sur le format de la clé qui contient en fait un G interdit par le patron de sécurité.

Si nous exécutons le programme HAWK sur une configuration qui ne respecte pas le patron de sécurité (sur la politique), nous obtenons le résultat suivant :

```
margot/21.1/hawk$ hawk -f ./ipsec.tp ./conf2.txt
./conf2.txt; hash md4 (line 7);(erreur) configuration non conforme au mode strict;
➥ crypto isakmp policy
```

Cet exemple illustre une erreur sur la politique qui contient en fait une ligne de configuration *hash md4* interdite par le patron de sécurité.

Si nous exécutons maintenant le programme HAWK sur une configuration qui respecte le patron de sécurité, nous obtenons le résultat suivant :

```
margot/21.1/hawk$ hawk -f ./ipsec.tp ./conf.txt
```

Il est ainsi possible avec l'outil HAWK de contrôler en profondeur les configurations IPsec et de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de périmètres

Bien qu'il soit important de contrôler les configurations des équipements réseau, il est non moins primordial de valider les périmètres IPsec implémentés. Pour y parvenir, nous utilisons l'outil GRAPH, ainsi qu'un script d'extraction utilisé pour déterminer les nœuds et les arcs de notre graphe IPsec VPN.

Avant d'utiliser l'outil GRAPH, il nous faut définir les nœuds et arcs de notre graphe. Pour cela, nous considérons tout d'abord que le nom d'une cryptomap suit la règle de configuration suivante :

```
IPsec_X_Y
   X : identifiant unique d'un VPN IPsec
   Y:  instance d'une nouvelle politique pour un VPN IPsec
```

Par exemple, la cryptomap IPsec_1_1 correspond au VPN 1 et à la politique de sécurité 1. De même, IPsec_1_2 correspond au VPN 1 et à la politique de sécurité 2.

Si, pour chaque configuration, nous arrivons à renseigner les champs de la table IPsec suivante (il peut y avoir plusieurs enregistrements par configuration de routeur), il est possible de construire le graphe IPsec VPN :

```
table IPsec
      champ : NomRouteur : nom du routeur
      champ : CryptoMapId : identifiant unique d'un VPN IPsec
      champ : IpAdr : adresse ip de l'interface ou est appliquée une cryptomap
      champ : IpAdrDest : adresse ip destinatrice du tunnel IPsec
```

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, sous réserve que l'adresse IP de l'interface (où est appliquée une cryptomap) soit égale à l'adresse IP destinatrice du tunnel IPsec et que les CryptoMapId soient identiques, donne tous les arcs de notre graphe IPsec, comme l'illustre la requête SQL suivante :

```
SELECT
        Ipsec.NomRouteur, Ipsec.CryptoMapId, Ipsec.IpAdr,
        Ipsec.IpAdrDest,
              Ipsec_1.NomRouteur, Ipsec_1.CryptoMapId, Ipsec_1.IpAdr,
        Ipsec_1.IpAdrDest
```

```
FROM
        Ipsec, Ipsec AS Ipsec_1
WHERE
             Ipsec.IpAdrDest=Ipsec_1.IpAdr 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. Par ailleurs, l'asymétrie de configuration d'un tunnel IPsec indique que le graphe IPsec construit est dirigé.

Une fois les nœuds et les arcs extraits de la ou des configurations, nous fournissons ces données à l'outil GRAPH, lequel calcule les composantes connexes (s'il existe un chemin entre toute paire de sommets (*x,y*) de la composante) et 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 VPN.

Les nœuds contenus dans une composante connexe impliquent qu'ils communiquent entre eux. Si les composantes connexes ne sont pas égales aux composantes fortement connexes, c'est qu'il y a des inconsistances de configuration. De même, toute configuration non bidirectionnelle entre deux sommets révèle des inconsistances de configuration.

Si nous appliquons cette méthode à l'exemple suivant, composé de deux configurations (conf.txt1 et conf.txt2), nous obtenons les résultats suivants :

```
margot/21.1/graph_ipsec$ ./ipsec_graph.sh
<stdin>: 4 nodes, 4 edges, 1644 bytes
# nodes = 4
# edges = 4
#
N       ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1
N       ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1
N       ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2
N       ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2
#
U       ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1 ./conf2.
        ➥ txt-192.168.1.154/192.168.1.1-ipsec1
U       ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1 ./conf1.
        ➥ txt-192.168.1.1/192.168.1.154-ipsec1
U       ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2 ./conf3.
        ➥ txt-192.165.1.154/192.165.1.1-ipsec2
U       ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2 ./conf1.
        ➥ txt-192.165.1.1/192.165.1.154-ipsec2
connected component (2 nodes):
{ ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1
➥ ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1 }
connected component (2 nodes):
{ ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2
➥ ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2 }
```

Les résultats de l'outil GRAPH indiquent que les deux composantes fortement connexes suivantes ont été trouvées, comme l'illustre la figure 21.6 :

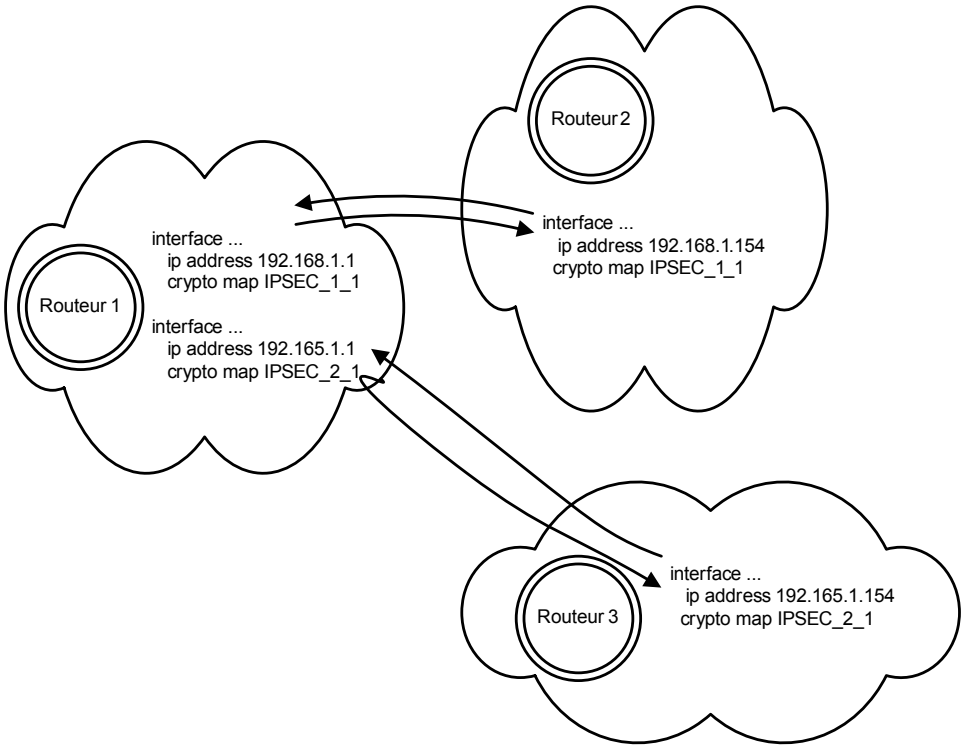


Figure 21.6

Composantes fortement connexes IPsec

- Composante 1 : le VPN ipsec1 référencé par ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1 est connecté au VPN ipsec1 référencé par ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1.
- Composante 2 : le VPN ipsec2 référencé par ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2 est connecté au VPN ipsec2 référencé par ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2.

Le contrôle de sécurité consiste donc à vérifier si les périmètres sont bien en ligne avec ce qui aurait dû être configuré. En cas d'erreur, c'est que l'isolation des périmètres n'est plus assurée. Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Exemple d'un tableau de bord de la sécurité réseau

Le tableau 21.1 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 21.1 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Recherche	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, etc.) et pare-feu (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMR R&D) et systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateurs (vérification VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateurs, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateurs (vérification VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), modems, routeurs (accès non autorisés, etc.), boîtier IPsec (sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés

Sous-réseau	Catégorie	Élément
Production	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.) et serveurs dédiés d'authentification (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.) et serveurs dédiés d'authentification (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

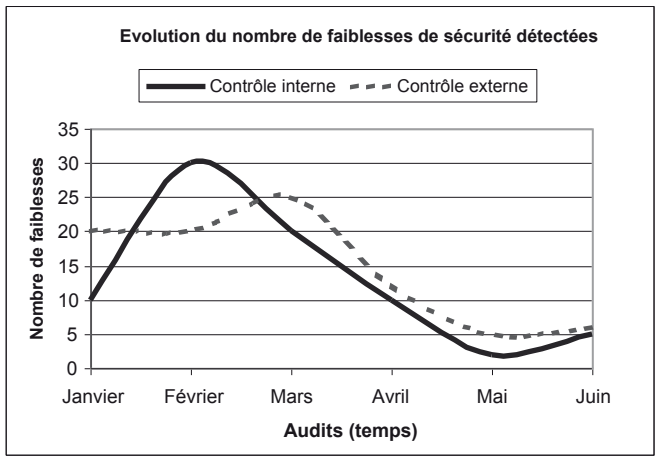
Le tableau de bord de la sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, l'évolution dans le temps du nombre de faiblesses de sécurité détectées par les contrôles interne et externe permet de donner une mesure de l'application de la politique de sécurité réseau.

La figure 21.7 illustre le fait que les faiblesses détectées par le contrôle interne de sécurité sont les mêmes que celles détectées par le contrôle externe au mois de février. Après correction des faiblesses de sécurité, nous observons une baisse commune des deux courbes de mars à mai. Si la courbe du contrôle externe ou interne ne décroissait pas, une investigation de sécurité devrait être menée afin de trouver et de clarifier la cause de ces faiblesses de sécurité.

Figure 21.7

Évolution du nombre de faiblesses de sécurité détectées



RadioVoie étend son réseau à l'international

Les parts de marché acquises et le succès des produits de RadioVoie permettent à l'entreprise d'étendre sa base de clients et son ambition de croissance.

De nombreux prospects, tant militaires que du domaine public, originaires des États-Unis, d'Europe et d'Asie, sont devenus des clients.

Besoins à satisfaire

Afin de faire face à cette forte croissance d'activité, RadioVoie décide de créer des agences satellites dans les pays où le nombre de ses clients est important, afin de répondre à la demande croissante de support et de service.

Chaque agence est autonome, financièrement et opérationnellement, mais doit suivre les règles de sécurité communes définies par le DSSI de l'entreprise. Toutes les agences sont considérées comme des entreprises de la multinationale RadioVoie.

Les sites de production disposent d'un réseau de recherche et développement, d'un réseau bureautique et d'un réseau de production. En dehors des sites de production, les agences ne disposent que de réseaux bureautiques. Le réseau global interconnectant les différentes agences doit offrir des garanties de qualité de service ainsi que des mécanismes d'isolation de trafic offrant un premier niveau de sécurité réseau.

Les militaires de chacun des pays où RadioVoie est implantée ont des exigences de sécurité draconiennes. Tous exigent qu'une unité de production soit implémentée dans leur pays respectif selon des contraintes de sécurité particulières. L'entreprise réussit à limiter ces contraintes en regroupant les pays appartenant à l'OTAN dans une même unité de production située à Bruxelles. Ce site obéit aux mêmes contraintes que Mouans-Sartoux.

Compte tenu des enjeux stratégiques et financiers liés au développement de l'entreprise, RadioVoie décide d'investir à la fois dans le réseau et dans les solutions de sécurité qui seront retenues.

Étude de risques

Plusieurs acteurs jouent un rôle clé dans le réseau interconnectant les entreprises de la multinationale. Ces acteurs sont l'opérateur de télécommunications, l'entité de sécurité de la multinationale et les équipes de recherche-développement, de production et de bureautique. Chacun de ces rôles est associé à une responsabilité de sécurité, qui peut être d'ordre physique ou logique. Un des risques majeurs encouru par RadioVoie est que ces responsabilités ne soient pas clairement définies, introduisant de fait une mauvaise compréhension de la sécurité et des failles résultantes.

L'architecture technique mise en place distingue donc des périmètres de sécurité ainsi que des objectifs de sécurité à mettre en place. Il s'agit de limiter les risques d'infiltration par une séparation des éléments de sécurité par périmètre.

Reste la solution d'interconnexion réseau, qui doit fournir une isolation du trafic et des options de qualité de service afin de diminuer les risques associés à l'intégrité et à la disponibilité des services réseau.

Politique de sécurité réseau

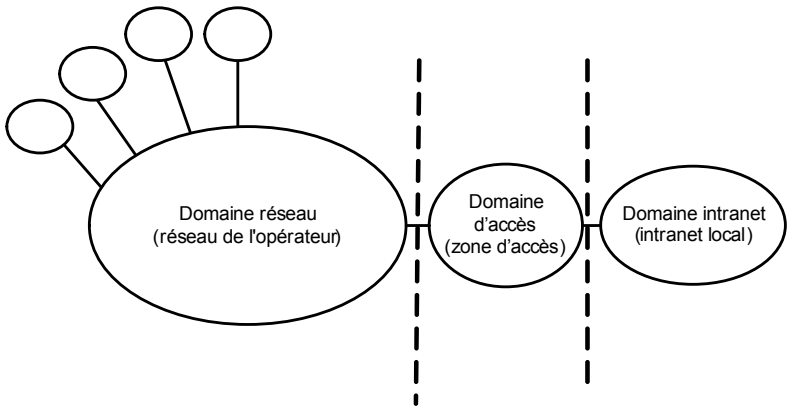
D'après les besoins à satisfaire et l'étude de risques, RadioVoie définit une politique de sécurité minimale, qui s'appuie sur des domaines de sécurité pour attribuer aux acteurs des responsabilités d'ordre physique et logique.

Comme expliqué précédemment, les acteurs de la politique de sécurité sont l'opérateur de télécommunications, qui offre la connectivité réseau aux sites des entreprises de la multinationale, l'entité de sécurité de la multinationale, qui a en charge la définition de la politique de sécurité réseau et la gestion des équipements de sécurité connectés au réseau, et les équipes de recherche et développement, de production et de bureautique, qui doivent se conformer à la politique de sécurité réseau et être les correspondants sécurité de leur réseau respectif.

Le modèle sécuritaire proposé repose sur l'architecture illustrée à la figure 21.8.

Figure 21.8

Séparation logique des périmètres de sécurité



Le service d'interconnexion des entreprises de la multinationale est réalisé au travers du réseau de l'opérateur de télécommunications. Il s'agit du premier domaine de sécurité, identifié sous le nom « domaine réseau ».

Pour chaque entreprise de la multinationale, une zone d'accès est définie, dont la fonction consiste à interconnecter le réseau interne intranet de l'entreprise au réseau interentreprise. Cette zone a en outre pour rôle d'établir une première zone de sécurité entre ces réseaux. Il s'agit du deuxième domaine de sécurité, identifié sous le nom « domaine d'accès ».

Pour chaque entreprise, une zone intranet local connecte le réseau d'entreprise à la zone d'accès. Cette zone devant évidemment être sécurisée, un deuxième niveau de sécurité

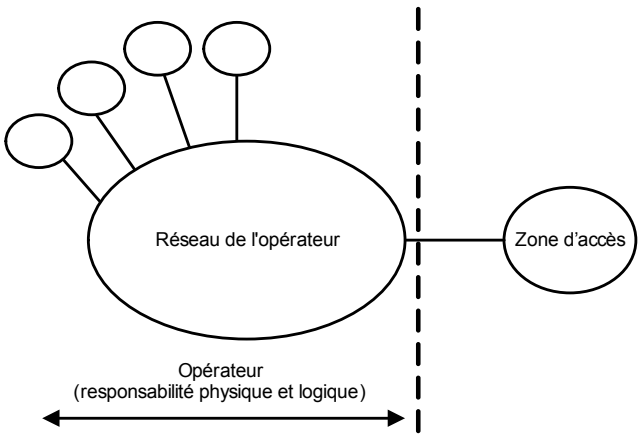
du réseau interne intranet de l'entreprise est défini. Il s'agit du troisième domaine de sécurité, identifié sous le nom « domaine intranet ».

Politique de sécurité du domaine réseau

Le domaine de sécurité relatif au réseau interconnectant les différentes entreprises de RadioVoie est sous la responsabilité de l'opérateur de télécommunications, comme l'illustre la figure 21.9.

Figure 21.9

Périmètre du domaine réseau



La politique de sécurité minimale relative au domaine réseau édicte les règles de sécurité suivantes :

- *« L'opérateur de télécommunications offre un service de réseau privé virtuel non accessible depuis Internet. »*
- *« L'opérateur de télécommunications explique les mécanismes de sécurité mis en œuvre sur son réseau et ses services de réseau privé virtuel. »*
- *« L'opérateur de télécommunications garantit et démontre que le périmètre logique de sécurité du réseau privé virtuel offert est limité au réseau privé virtuel de la multinationale RadioVoie. »*
- *« L'opérateur de télécommunications mène des contrôles de sécurité logique sur les configurations des équipements offrant le service de réseau privé virtuel et diffuse les rapports du réseau privé virtuel à la multinationale RadioVoie. »*
- *« L'opérateur de télécommunications s'engage à donner toutes les informations relatives à des incidents de sécurité qui mettraient en péril l'isolation du réseau privé virtuel de la multinationale RadioVoie. »*
- *« Un point de contact sécurité avec l'opérateur de télécommunications est établi, incluant les procédures de réponse aux incidents. »*

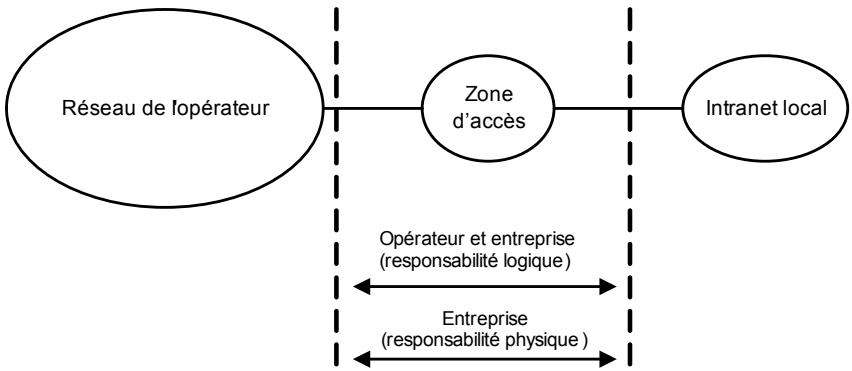
Politique de sécurité du domaine d'accès

Le domaine de sécurité relatif à la zone d'accès entre le réseau intranet d'une des entreprises de RadioVoie et le domaine réseau est sous la responsabilité physique de l'entreprise concernée et sous la responsabilité logique de l'opérateur de télécommunications pour les équipements offrant le service de connexion au domaine réseau, comme illustré à la figure 21.10.

Tout équipement n'appartenant pas à l'opérateur de télécommunications est sous la responsabilité logique de l'entreprise.

Figure 21.10

Périmètre du domaine accès



La politique de sécurité minimale relative au domaine d'accès édicte les règles de sécurité suivantes :

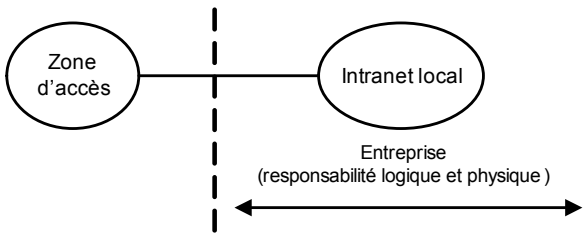
- *« L'opérateur de télécommunications démontre que l'équipement de connexion au réseau privé virtuel installé dans un site physique d'une entreprise de RadioVoie ne permet pas d'accéder au réseau privé virtuel de la multinationale. »*
- *« L'opérateur de télécommunications permet, si nécessaire, d'ajouter aux équipements de connexion au réseau privé virtuel de la multinationale des filtrages sur protocoles. »*
- *« Les échanges réseau transitant sur le réseau privé virtuel de la multinationale sont chiffrés. »*
- *« L'établissement de tunnels chiffrés entre deux sites est authentifié à l'aide de certificats électroniques. »*
- *« Les certificats sont utilisés pour authentifier les sessions réseau. Ces certificats électroniques ne sont pas fournis par l'opérateur de télécommunications mais par une infrastructure à clés publiques propre à la multinationale et à ses entreprises. »*
- *« Les équipements réseau et de chiffrement sont hébergés dans une salle informatique protégée des menaces physiques (humidité, feu, chaleur, etc.) et à accès restreint et contrôlé. »*
- *« Des procédures d'incident de sécurité de la zone d'accès sont définies par l'entreprise ayant la gestion de la zone d'accès. »*

Politique de sécurité du domaine intranet

Le domaine de sécurité relatif au réseau intranet de l'entreprise est sous la responsabilité physique et logique de l'entreprise, comme illustré à la figure 21.11.

Figure 21.11

Périmètre du domaine intranet



La politique de sécurité minimale relative au domaine intranet édicte les règles de sécurité suivantes :

- *« Un système de filtrage du trafic échangé entre le domaine d'accès et le domaine intranet est mis en place. »*
- *« Un système de translation d'adresse et de port de services est implémenté afin de cacher le plan d'adressage interne du réseau d'une entreprise. »*
- *« Tout incident de sécurité détecté est répertorié et fait l'objet d'une enquête. De plus, tout incident de sécurité est aussitôt connu de toutes les entreprises de la multinationale. »*
- *« Les journaux d'activité des systèmes de filtrage, translation et détection d'intrusion sont centralisés et soumis à des logiciels de corrélation afin de détecter ou de confirmer des incidents de sécurité. Les journaux d'activité sont archivés et sauvegardés sur un support physique. »*
- *« Des procédures d'incident de sécurité de la zone intranet sont définies par l'entreprise concernée. »*

Solution de sécurité

Le réseau de RadioVoie comprend désormais les sites illustrés à la figure 21.12.

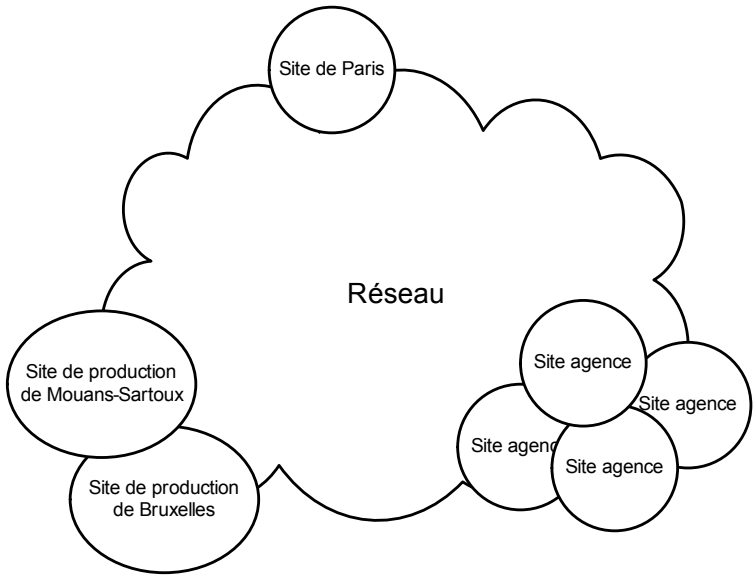
RadioVoie mettant en œuvre une solution technique pour chaque domaine de sécurité réseau, nous présentons ces solutions pour chacun de ces domaines. Nous détaillerons ensuite les risques couverts et les risques restants à couvrir.

Solution de sécurité pour le domaine réseau

L'une des problématiques récurrentes des réseaux est de faire transiter des données le plus rapidement et le plus sûrement possibles. La disponibilité des services réseau est généralement couverte par la topologie du réseau. Quant à l'intégrité des services réseau, elle est généralement couverte par les protocoles réseau.

Figure 21.12

Les sites du réseau RadioVoie



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 dans 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. Dans la mesure où le protocole MPLS est devenu un standard IETF (Internet Engineering Task Force) et qu'il s'est imposé face à l'ATM, nous nous penchons davantage sur ce protocole et ses fonctionnalités.

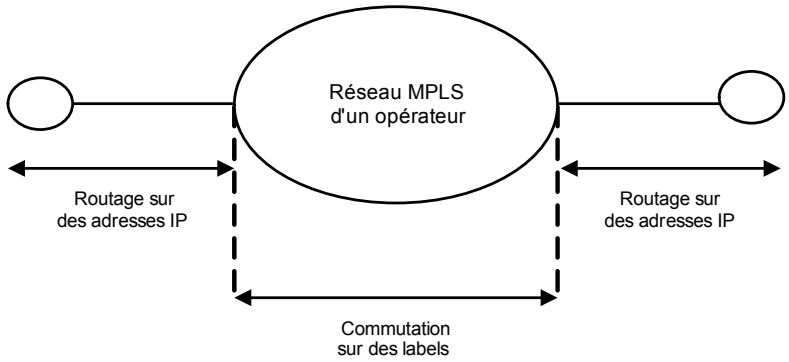
Plutôt que de décider du routage des paquets dans le réseau à partir des adresses IP, MPLS s'appuie sur des *labels,* ou étiquettes. La commutation de paquets se réalise sur ces labels et ne consulte plus les informations relatives au niveau 3, incluant les adresses IP. En d'autres termes, l'acheminement ou le routage des paquets est fondé sur les labels et non plus sur les adresses IP, comme sur le réseau Internet. La figure 21.13 illustre ce mode de fonctionnement.

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 offre des avantages notables par rapport au protocole IP. Il est, par exemple, possible de créer des 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, qui est une extension du protocole de routage BGP (Border Gateway Protocol) pour les réseaux MPLS.

Figure 21.13

Commutation des paquets dans un réseau MPLS



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

Chaque VPN peut faire transiter les classes d'adresses IP qu'il désire sans qu'il y ait de conflit d'adresse 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. Pour cela, un identifiant, appelé RD (Route Distinguisher), est accolé à chaque sous-réseau IPv4 afin de créer une route VPNv4.

Un réseau MPLS/VPN est composé de routeurs P (Provider), dédiés à la commutation, ou LSR (Label Switch Router), de routeurs PE (Provider Edge), dédiés à la création des MPLS/VPN ainsi qu'a la connectivité avec les équipements localisés chez les clients, ou LER (Label Edge Router), et de routeurs CE (Customer Edge), installés chez les clients et connectés aux routeurs PE.

Seuls les routeurs PE contiennent la définition effective des MPLS/VPN, les routeurs P et CE n'ayant aucune connaissance de la configuration des MPLS/VPN. Les routeurs P commutent des labels MPLS, tandis que les routeurs CE commutent des adresses IP, comme l'illustre la figure 21.14.

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/VPN, prenons l'exemple de deux VPN (rouge, bleu), que nous allons définir afin de relier deux sites différents pour chacun des VPN, comme illustré à la figure 21.15.

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

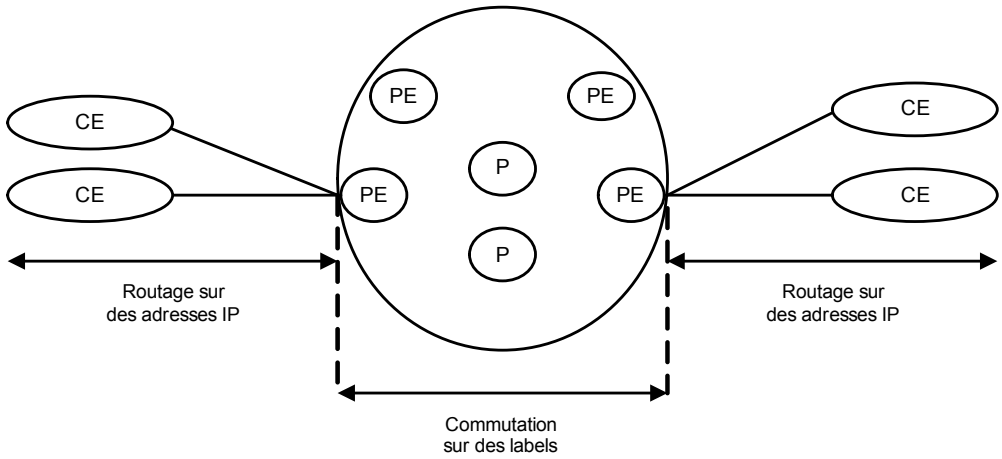


Figure 21.14

Connexions à un réseau MPLS

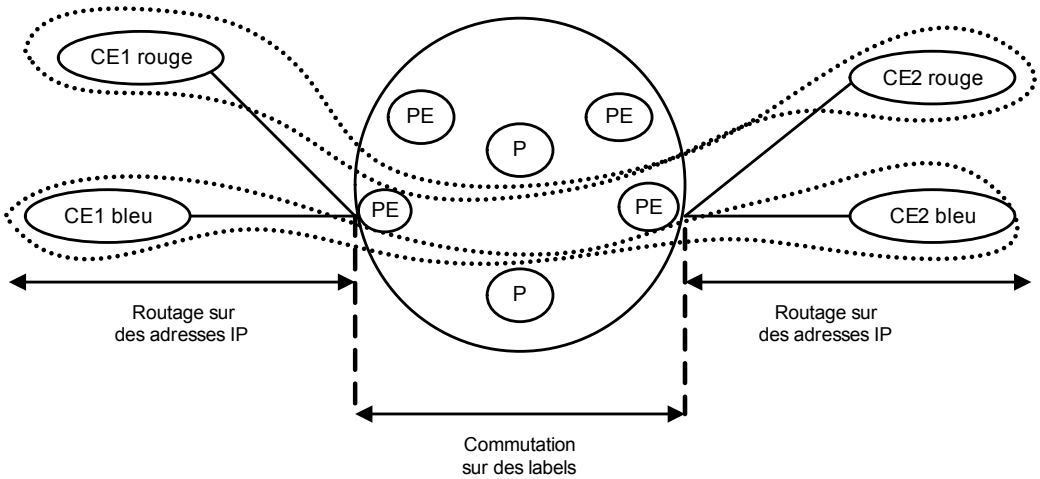


Figure 21.15

Réseaux privés virtuels sur un réseau MPLS

Les routeurs CE1 et CE2 rouge appartiennent au MPLS/VPN rouge et les routeurs CE1 et CE2 bleu au MPLS/VPN bleu. La configuration des routeurs PE permet de créer ces VPN sur le réseau par les configurations décrites ci-dessous des deux PE (implémentation Cisco).

Configuration du routeur PE, connecté à CE1 rouge et CE1 bleu :

```
# Définition du MPLS/VPN rouge :
ip vrf rouge
```

```
# La valeur du rd (route distinguisher) permet d'isoler les routes échangées entre
➥ les PE routeurs pour chaque MPLS/VPN :
    rd x1
# Les valeurs des route-targets permettent de définir le MPLS/VPN par le fait que
➥ le MPLS/VPN rouge importe toutes les routes véhiculant la route-target 100:1 et
➥ exporte les routes apprises de son côté au réseau MPLS en insérant la route-
➥ target 100:1 :
    route-target import 100 : 1
    route-target export 100 : 1

# Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x2
   route-target import 100 : 2
   route-target export 100 : 2

# Connexion de CE1 rouge au PE :
interface ...

# Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   ...

# Connexion de CE1 bleu au PE :
interface ...

# Cette connexion appartient au MPLS/VPN bleu :
   ip vrf forwarding bleu
   ...
```

Configuration du routeur PE, connecté à CE2 rouge et CE2 bleu :

```
# Définition du MPLS/VPN rouge :
ip vrf rouge

# La valeur du rd (route distinguisher) permet d'isoler les routes échangées entre
➥ les PE routeurs pour chaque MPLS/VPN :
   rd x3

# Les valeurs des route-target permettent de définir le MPLS/VPN par le fait que
➥ le MPLS/VPN rouge importe toutes les routes véhiculant la route-target 100:1 et
➥ exporte les routes apprises de son côté au réseau MPLS en insérant la route-
➥ target 100:1 :
   route-target import 100 : 1
   route-target export 100 : 1

# Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x4
   route-target import 100 : 2
   route-target export 100 : 2
```

```
# Connexion de CE2 rouge au PE :
interface ...

# Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   ...

# Connexion de CE2 bleu au PE :
interface ...

# Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    ...
```

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

La sécurité du réseau MPLS ainsi que la configuration logique des MPLS/VPN sont sous la responsabilité du fournisseur de services réseau. Ce dernier doit clairement expliquer dans sa politique de sécurité comment il remplit ses obligations et ses responsabilités de sécurité.

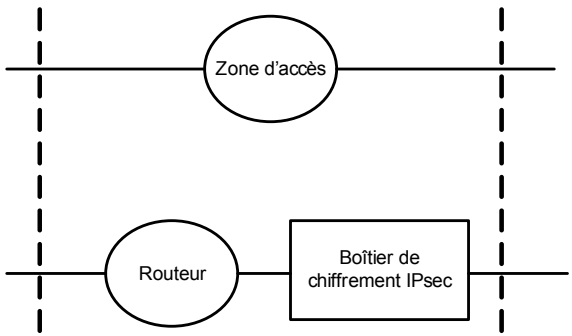
Solution de sécurité pour le domaine d'accès

De nombreuses contraintes de sécurité pèsent sur le domaine d'accès. Elles imposent notamment la séparation des fonctions de sécurité sur des équipements dédiés.

Le domaine d'accès est constitué d'un routeur dédié à la connexion au routeur PE et au routage et d'un boîtier de chiffrement spécifique, qui n'implémente que la fonction IPsec, comme l'illustre la figure 21.16.

Figure 21.16

Solution technique pour la zone d'accès



Le routeur

Le routeur est géré par l'opérateur de télécommunications. Il se connecte au réseau MPLS et est donc le premier équipement traversé. La sécurité physique de cet équipement demeure cependant sous la responsabilité de l'entreprise.

Pour renforcer la sécurité du réseau privé virtuel, une plage d'adresses IP est spécifiquement définie afin d'attribuer ces adresses aux routeurs et aux boîtiers de chiffrement côté réseau du réseau privé virtuel. Comme nous le verrons, le reste du trafic est caché par les tunnels IPsec établis sur le réseau privé virtuel. On appelle cette plage d'adresses IP ip_vpn_adresses.

Quelle que soit la marque du routeur fourni par l'opérateur de télécommunications (Cisco, Bay Networks, etc.), des filtrages fondés sur des ACL (Access Control List) permettent de dresser une première barrière de sécurité au niveau du routeur.

Dans notre cas, deux ACL peuvent être définies sur l'interface WAN du routeur :

```
# Filtrage du trafic du WAN vers le routeur :
ip access-list extended site-acl-in

# Filtrage du trafic IPsec appartenant au réseau privé virtuel :
permit udp ip_vpn_adresses ip_vpn_adresses eq isakmp
permit udp ip_vpn_adresses eq isakmp ip_vpn_adresses
permit esp ip_vpn_adresses ip_vpn_adresses
permit ahp ip_vpn_adresses ip_vpn_adresses

# On laisse passer du trafic ICMP :
permit icmp ip_vpn_adresses ip_vpn_adresses echo
permit icmp ip_vpn_adresses ip_vpn_adresses echo-reply

# Destruction du reste du trafic et génération des logs :
deny ip any any log

# Filtrage du trafic du routeur vers le WAN :
ip access-list extended site-acl-out

# Filtrage du trafic IPsec appartenant au réseau privé virtuel :
permit udp ip_vpn_adresses eq isakmp ip_vpn_adresses
permit udp ip_vpn_adresses ip_vpn_adresses eq isakmp
permit esp ip_vpn_adresses ip_vpn_adresses
permit ahp ip_vpn_adresses ip_vpn_adresses

# On laisse passer du trafic ICMP :
permit icmp ip_vpn_adresses ip_vpn_adresses echo
permit icmp ip_vpn_adresses ip_vpn_adresses echo-reply

# Destruction du reste du trafic et génération des logs :
deny ip any any log

# Filtrage du trafic du routeur de et vers le WAN et application des ACL sur
➥ l'interface WAN du routeur :
(Interface ATM) :
ip access-group site-acl-in
ip access-group site-acl-out
      ...
```

Les boîtiers de chiffrement IPsec

Pour les boîtiers de chiffrement, RadioVoie opte pour une solution entièrement dédiée au chiffrement IPsec et n'offrant pas d'autres options, tel le routage ou le filtrage du trafic. L'idée est de suivre la règle de séparation logique et physique des fonctions de sécurité.

Les nombreux boîtiers IPsec disponibles implémentent de plus en plus d'options. Le tableau 21.2 récapitule ces options pour les différentes offres du marché.

Tableau 21.2 Types de boîtiers IPsec

Service	AEP Net series	Evidian/Bull TrustWay VPN series	Thalès Datacryptor series	CheckPoint IPsec VPN series
Protocole IPsec	Oui	Oui	Oui	Oui
Algorithmes de chiffrement	3DES, AES, etc.	3DES, AES, etc.	3DES, AES, etc.	3DES, AES, etc.
Authentification	Certificat x509	Certificat x509, utilisateur/mot de passe, etc.	Certificat x509	Certificat x509, utilisateur/mot de passe, etc.
NAT/PAT	Oui	Non	Non	Non
Filtres IP	Non	Oui	Non	Pare-feu stateful
Protocoles de routage	Non	Non	Non	Oui (RIP, OSPF, etc.)
Architecture haute disponibilité	Oui	Oui	Oui	Oui

RadioVoie opte pour les solutions de type AEP ou Bull ou Thalès, limitées volontairement à la fonction de gestion de tunnels IPsec.

Ces boîtiers couvrent bien le principe de ségrégation des fonctions de sécurité et répondent donc aux règles édictées par la politique de sécurité réseau.

Solution de sécurité pour le domaine intranet

Le domaine intranet correspond aux réseaux internes des sites de la multinationale RadioVoie. D'après les règles définies par la politique de sécurité réseau, un pare-feu est implémenté pour prendre en charge le filtrage des trafics réseau mais aussi la translation d'adresses et de port afin que le trafic soit émis vers les boîtiers de chiffrement IPsec.

L'architecture adoptée est illustrée à la figure 21.21.

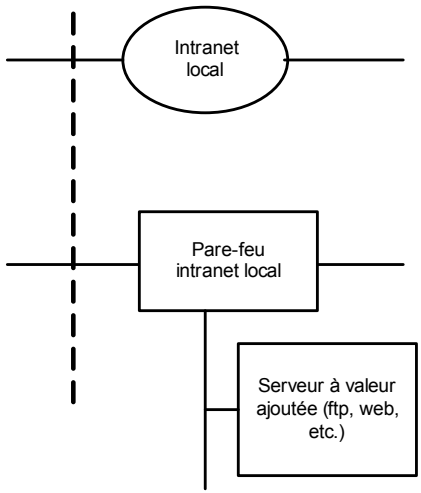
Les nombreux produits de pare-feu disponibles sur le marché répondent souvent à un besoin de sécurité spécifique.

Parmi les pare-feu les plus courants, citons les suivants :

- Alcatel-Lucent : Lucent Firewall VPN family
- CheckPoint Software : Secure Platform NG

Figure 21.17

Solution technique pour le domaine intranet



Les besoins de sécurité de ce domaine visent avant tout les fonctions de filtrage du trafic réseau de l'entreprise. Le choix se porte donc naturellement sur un produit conçu à la base dans cette optique, le pare-feu CheckPoint.

Solution de sécurité pour l'interconnexion des sites

L'architecture réseau du domaine d'interconnexion des sites de Paris et Mouans-Sartoux est illustrée à la figure 21.18.

Du point de vue purement réseau, le routeur d'interconnexion est remplacé par un routeur CE connecté au routeur PE de l'opérateur de télécommunications.

Pour le domaine d'accès intranet, nous reconnaissons le routeur CE connecté au pare-feu et le boîtier IPsec connecté par deux interfaces pare-feu. Rappelons que cette architecture permet de filtrer et de tracer toutes les connexions IPsec avant et après le passage par le boîtier IPsec à des fins d'investigation en cas d'incident de sécurité.

Pour le domaine d'accès recherche et développement, nous reconnaissons un boîtier IPsec connecté au pare-feu. Ce pare-feu comporte deux autres interfaces, l'une connectée à une zone DMZ externe et l'autre connectée à une DMZ recherche et développement. De même que pour le domaine d'accès recherche et développement, cette architecture permet de filtrer et de tracer toutes les connexions IPsec avant et près le passage par le boîtier IPsec.

En application de la règle de variété des fonctions de sécurité, RadioVoie choisit des équipements de marque différente pour les boîtiers IPsec et les pare-feu, afin d'assurer une sécurité des protections en profondeur.

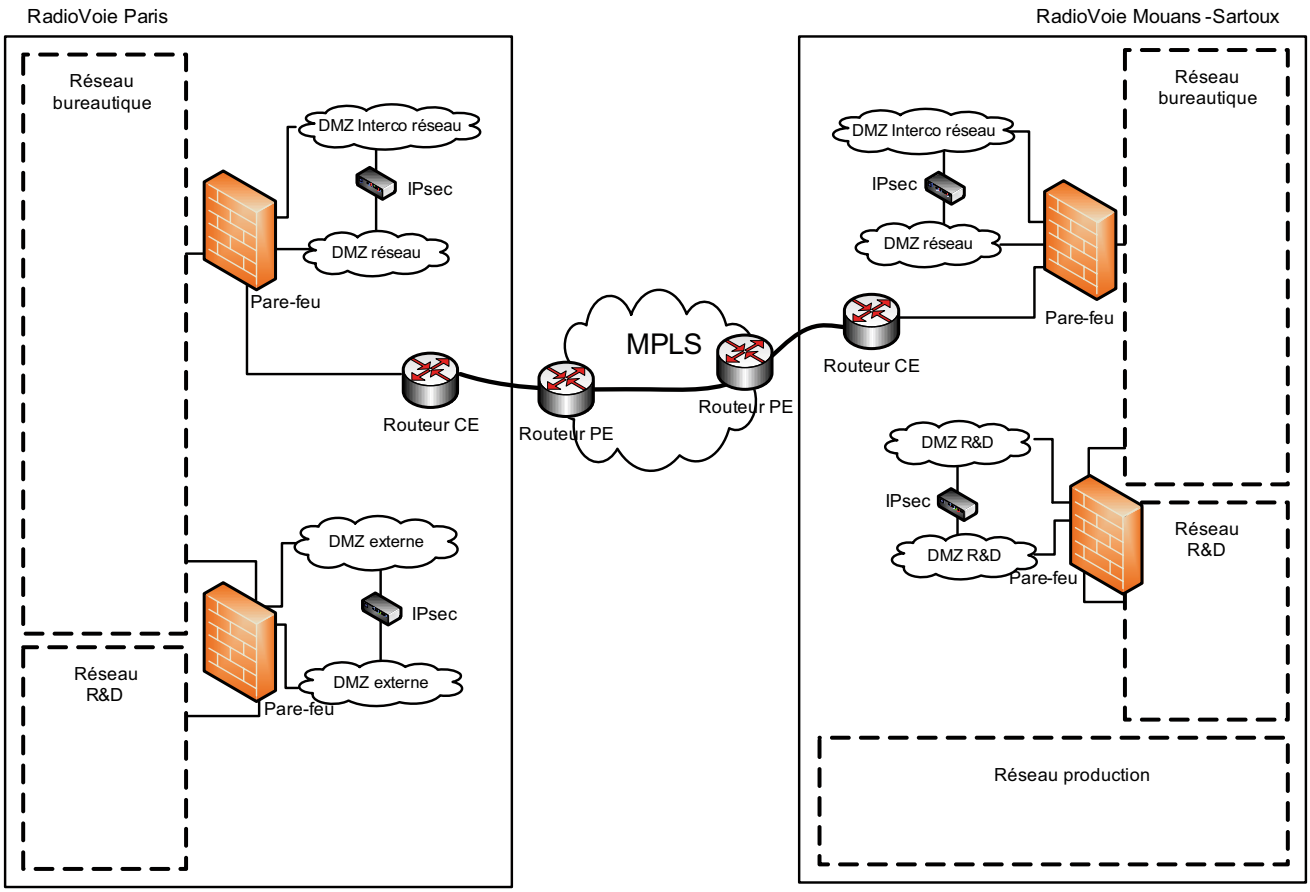


Figure 21.18

Interconnexion des sites de Paris et Mouans-Sartoux

Solution de sécurité pour les accès à Internet

Comme illustré à la figure 21.19, la séparation physique et logique entre le réseau privé virtuel MPLS et le réseau Internet de ce domaine garantit une ségrégation complète des accès réseau.

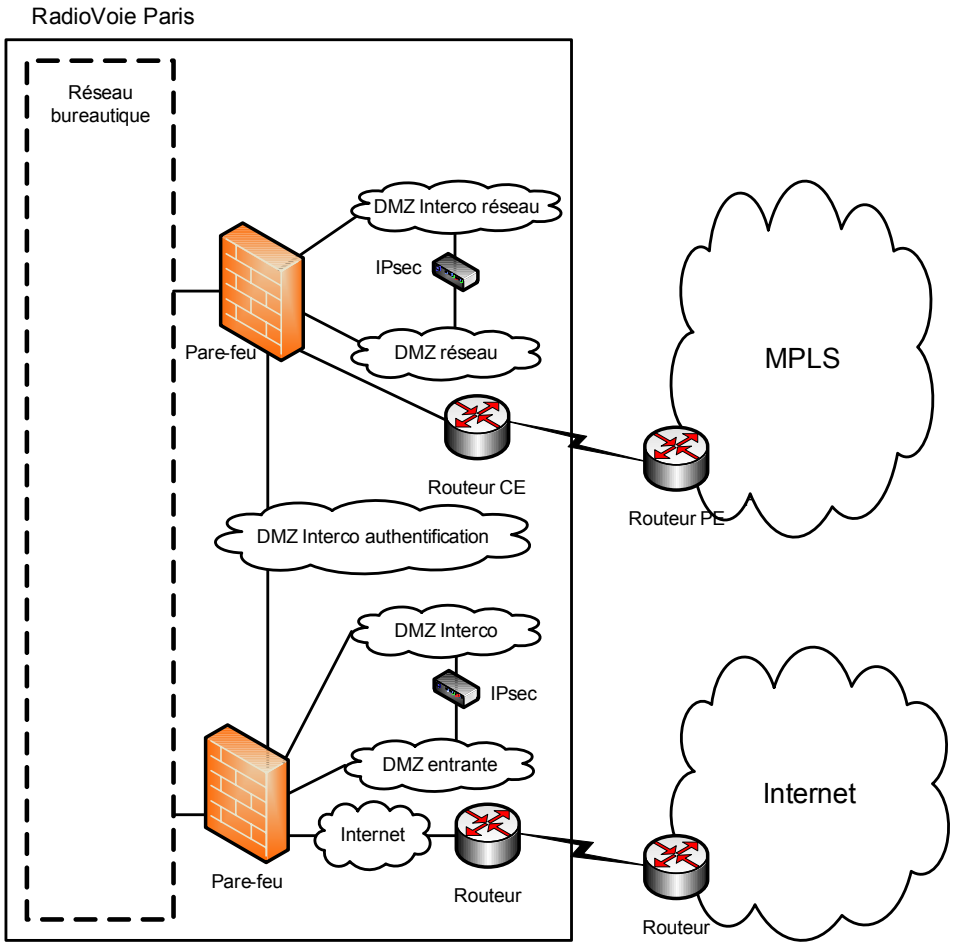


Figure 21.19

Solution d'accès à Internet

Une zone d'accès spécifique est créée pour Internet. De même, un commutateur non représenté sur la figure est dédié à la connexion Internet afin d'assurer une isolation physique et logique avec le réseau privé virtuel.

Une zone d'interconnexion, ou DMZ interco, est toutefois créée entre le pare-feu dédié à l'accès VPN et celui dédié à l'accès Internet afin d'y placer des serveurs spécifiques pour de futures évolutions ou de nouveaux services.

Que ce soit pour l'accès à Internet ou au VPN, l'architecture proposée offre une traçabilité importante des flux réseau transitant dans les zones d'accès. De plus, puisque RadioVoie dispose d'un routeur pour accéder à Internet, les capacités filtrantes de ce dernier assurent un premier nettoyage des flux en provenance d'Internet (principe du routeur

choke). Le routeur filtre ainsi les flux « polluants », tels que les connexions 137/UDP en provenance des stations de travail Windows mal configurées, les classes d'adresses IANA non attribuées, etc.

Solution de sécurité pour les accès à une tierce partie

De la même manière qu'un réseau privé virtuel est mis en place sur le réseau MPLS, un service d'accès aux serveurs RadioVoie est créé pour les tierces parties.

Si nous définissons un accès réseau pour une tierce partie au réseau privé virtuel de RadioVoie, il nous faut ajouter le routeur CE gris, dédié à l'accès de la tierce partie. Comme nous le verrons par la suite, ce routeur CE gris est logiquement connecté au routeur CE1 rouge *via* le réseau MPLS (connexion au site de Paris de l'entreprise RadioVoie), comme l'illustre la figure 21.20.

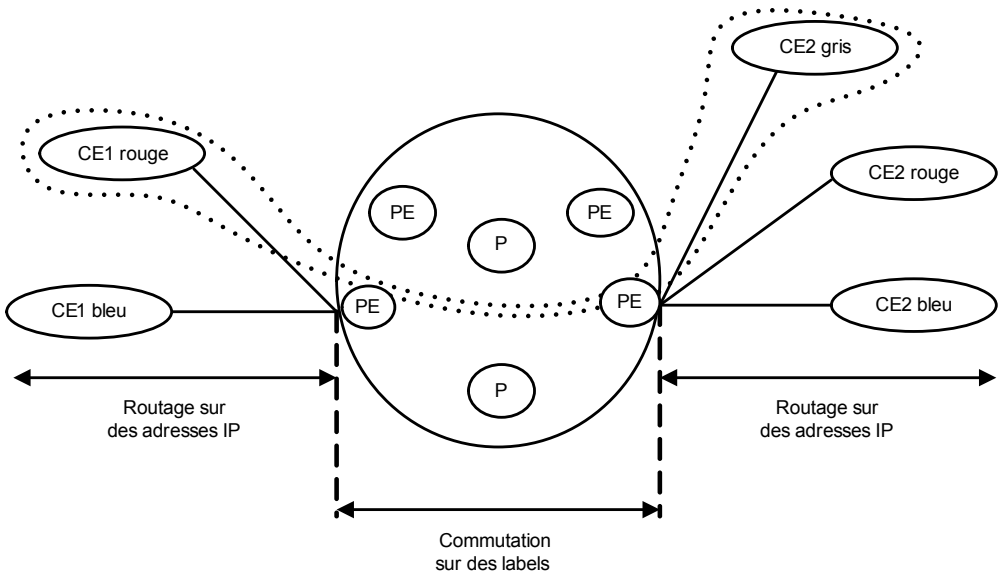


Figure 21.20

Solution d'accès à la tierce partie

Les routeurs CE1 et CE2 rouges appartiennent au MPLS/VPN rouge, et les routeurs CE1 et CE2 bleus au MPLS/VPN bleu. La configuration des routeurs PE permet de créer sur le CE1 rouge un accès de service par le biais des configurations suivantes sur les deux PE (implémentation Cisco).

Configuration du routeur PE, connecté à CE1 rouge et CE1 bleu :

```
# Définition du MPLS/VPN rouge :
ip vrf rouge
```

```
# La valeur du rd (route distinguisher) permet d'isoler les routes échangées entre
➥ les PE routeurs pour chaque MPLS/VPN :
   rd x1

# Les valeurs des route-target permettent de définir le MPLS/VPN par le fait que
➥ le MPLS/VPN rouge importe toutes les routes véhiculant la route-target 100:1 et
➥ exporte les routes apprises de son côté au réseau MPLS en insérant la route-
➥ target 100:1 :

   route-target import 100 : 1
   route-target export 100 : 1

# On ajoute les routes pour permettre au CE gris de se connecter à ce routeur CE1
➥ rouge :
   route-target import 100 : 4
   route-target export 100 : 5

# Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x2
   route-target import 100 : 2
   route-target export 100 : 2

# Connexion de CE1 rouge au PE :
interface ...

# Cette connexion appartient au MPLS/VPN rouge :
    ip vrf forwarding rouge
    ...

# Connexion de CE1 bleu au PE :
interface ...

# Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    ...
```

Configuration du routeur PE, connecté à CE2 rouge, CE2 bleu et CE2 gris :

```
# Définition du MPLS/VPN rouge :
ip vrf rouge

# La valeur du rd (route distinguisher) permet d'isoler les routes échangées entre
➥ les PE routeurs pour chaque MPLS/VPN :
     rd x3

# Les valeurs des route-target permettent de définir le MPLS/VPN par le fait que
➥ le MPLS/VPN rouge importe toutes les routes véhiculant la route-target 100:1 et
➥ exporte les routes apprises de son côté au réseau MPLS en insérant la route-
➥ target 100:1 :
```

```
    route-target import 100 : 1
    route-target export 100 : 1

# Définition du MPLS/VPN bleu :
ip vrf bleu
    rd x4
    route-target import 100 : 2
    route-target export 100 : 2

# Définition du MPLS/VPN gris :
ip vrf gris
    rd x5

# Les valeurs des route-target permettent de se connecter au CE1 rouge. Les valeurs
➥ des route-target sont importées et exportées de manière asymétrique comparées à
➥ celles configurées sur le PE connecté au CE1 rouge :
    route-target import 100 : 5
    route-target export 100 : 4

# Connexion de CE2 rouge au PE :
interface ...

# Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   ...

Connexion de CE2 bleu au PE :
interface ...

Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    ...

Connexion de CE2 gris au PE :
interface ...

Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding gris
    ...
```

La figure 21.21 montre que le service d'accès créé logiquement sur le réseau MPLS permet aux tierces parties d'accéder au site de Paris.

La configuration du service d'accès est donc réalisée. L'isolation d'un tel service repose sur la configuration logique des routeurs PE et, surtout, sur la sécurisation des accès externes à la fois de la tierce partie et de l'entreprise RadioVoie. RadioVoie n'a en effet aucune raison *a priori* de faire confiance à une tierce partie, et une tierce partie n'a aucune raison *a priori* de faire confiance à RadioVoie.

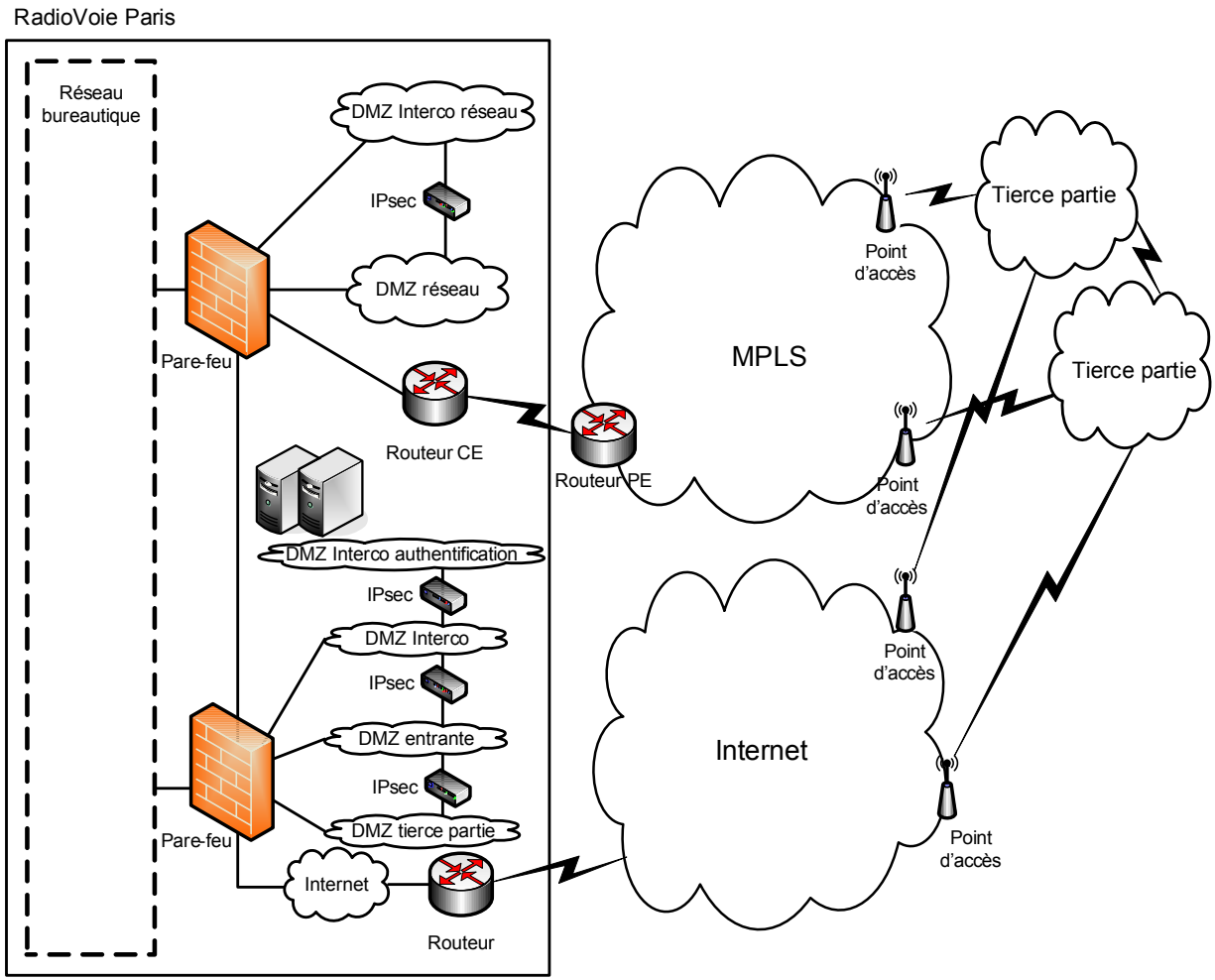


Figure 21.21

Solution d'accès de la tierce partie

La tierce partie accède à l'entreprise RadioVoie par le biais du réseau MPLS mais réclame un accès de secours en cas d'indisponibilité. Les accès distants des commerciaux s'effectuent par le biais du réseau Internet.

Les serveurs d'authentification sont placés dans la zone DMZ interco entre le pare-feu Internet et le pare-feu dédié au VPN. Ces serveurs servent à authentifier les utilisateurs provenant des deux types de réseau.

Solution de gestion des équipements de sécurité

La multinationale RadioVoie regroupe une centaine d'entreprises. Une zone d'administration dédiée à la gestion de l'ensemble des équipements de sécurité est mise en place.

Pour des raisons de redondance et de disponibilité, il s'agit en réalité de deux zones d'administration, l'une à Paris, l'autre à Mouans-Sartoux. Un VPN spécifique est créé sur le réseau MPLS afin que les deux zones d'administration puissent s'échanger des données. Ce VPN est différent du VPN de l'entreprise RadioVoie, et les deux VPN ne communiquent pas par défaut.

Deux modes d'administration sont possibles, le mode dit hors bande *(voir figure 21.22)* et le mode dans la bande.

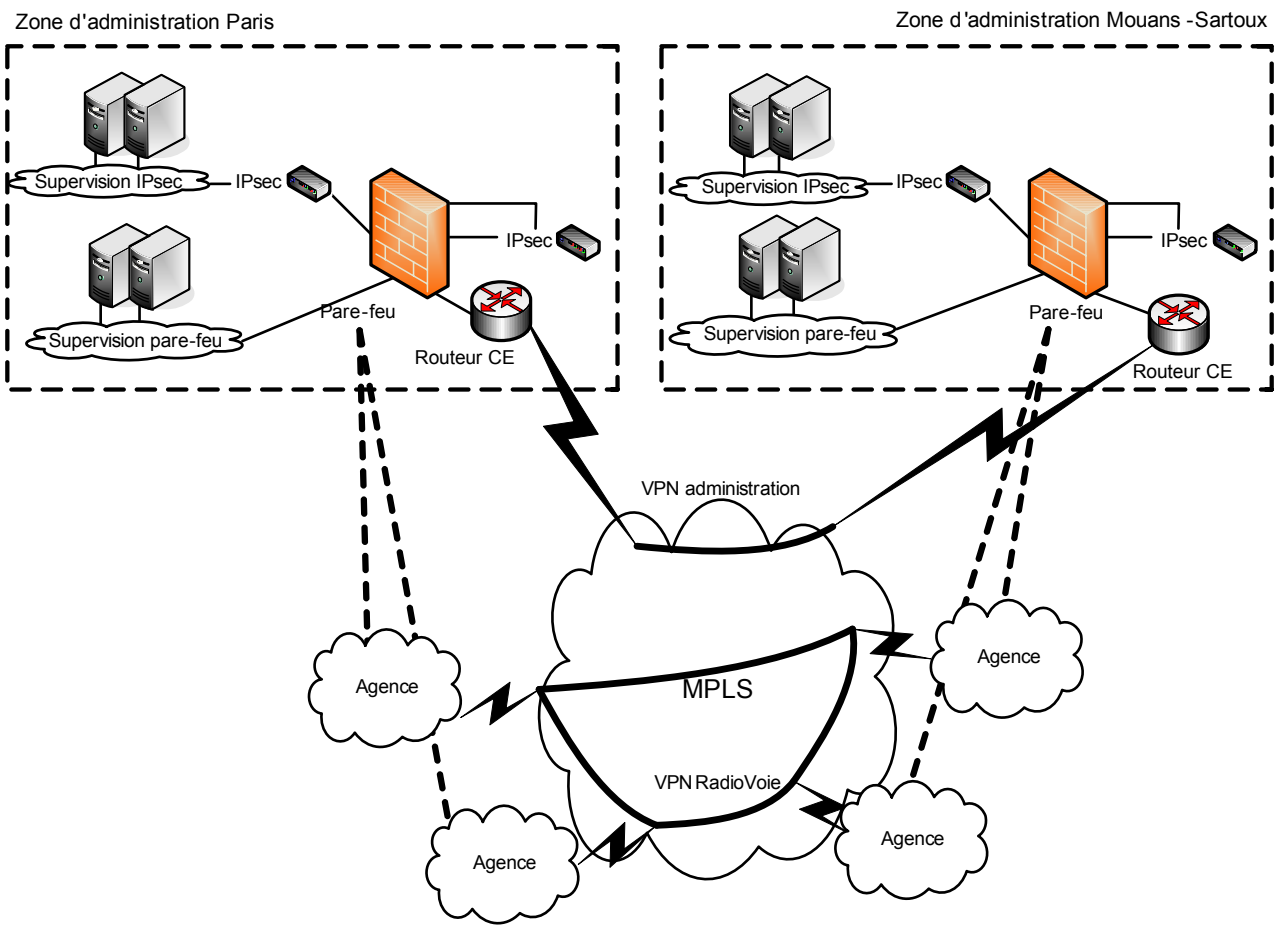


Figure 21.22

Gestion des équipements hors bande

Dans le mode d'administration hors bande, le VPN d'administration MPLS n'est pas connecté au VPN de RadioVoie. L'administration des équipements est réalisée par des connexions dédiées, offrant un unique chemin pour accéder en administration aux équipements. Ce mode est très sécurisé, car il y a isolation entre le réseau de RadioVoie et le réseau permettant de se connecter aux équipements administrés.

Les protocoles utilisés pour l'administration des équipements peuvent être rudimentaires, puisqu'on ne peut y accéder que si l'on pénètre dans la zone d'administration, laquelle n'est pas accessible depuis le réseau de RadioVoie.

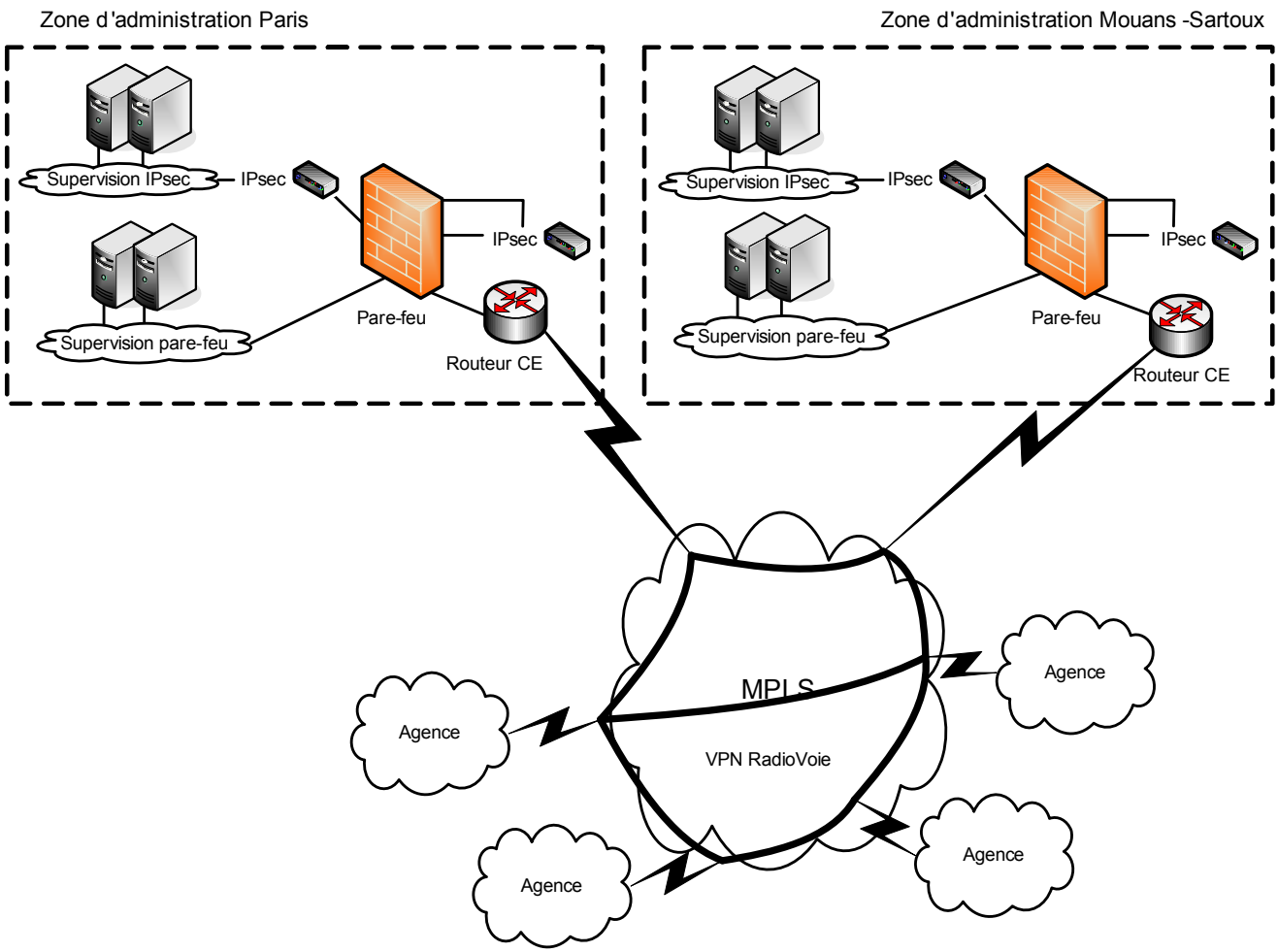


Figure 21.23

Gestion des équipements dans la bande

Ce mode d'administration est en revanche très coûteux, car les connexions dédiées à chaque équipement représentent un coût supplémentaire pour l'entreprise en plus des coûts des VPN d'administration et de RadioVoie.

Dans le mode d'administration dit dans la bande, le VPN d'administration MPLS est connecté au VPN de RadioVoie et utilise les connexions réseau du VPN pour ses sessions d'administration *(voir figure 21.23).*

Le mode dans la bande est évidemment moins sécurisé puisqu'il n'offre pas d'isolation native avec le réseau de RadioVoie. Les protocoles utilisés pour l'administration des équipements doivent être particulièrement sécurisés en terme d'authentification et de chiffrement, puisqu'on pourrait y accéder théoriquement à partir de n'importe quel point du réseau RadioVoie.

La zone d'administration est en outre plus exposée que dans le mode d'administration hors bande. De surcroît, l'intérêt d'avoir un réseau VPN d'administration spécifique est moins évident, et l'on pourrait considérer les deux zones d'administration comme des connexions spécifiques du VPN de RadioVoie.

Isolation logique des trafics réseau

L'architecture adoptée pour l'isolation logique des trafics réseau permet leur isolation complète, comme l'illustre la figure 21.24.

La première isolation est réalisée par le protocole MPLS, qui permet de créer des réseaux privés virtuels par routage. Les différents VPN n'ayant pas accès aux tables de routage des autres VPN, une première isolation logique est appliquée au niveau du réseau.

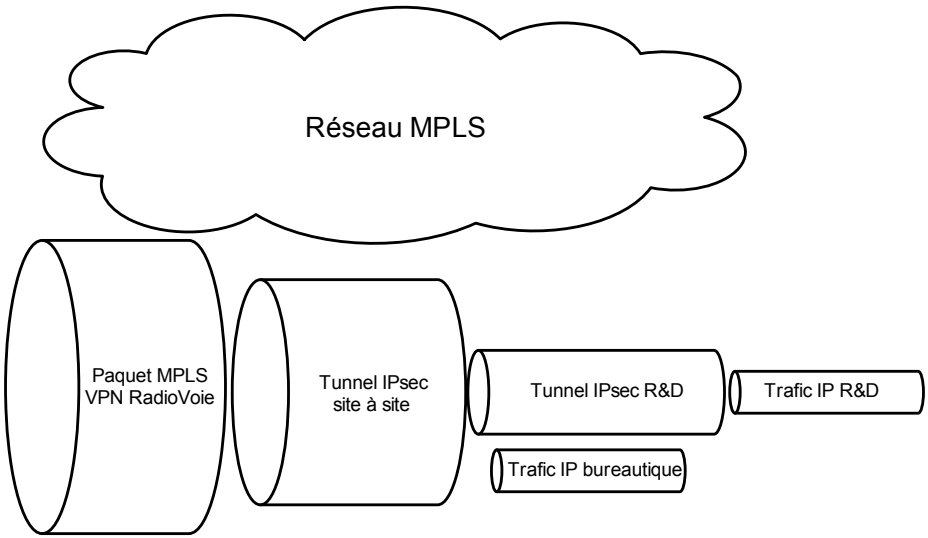


Figure 21.24

Niveaux d'isolation des trafics réseau

La deuxième isolation logique est réalisée par les tunnels IPsec établis entre les sites dans le but de les authentifier et de chiffrer les trafics réseau.

La troisième isolation logique est réalisée par les tunnels IPsec établis entre les sous-réseaux des sites pour les authentifier et offrir un deuxième niveau de chiffrement des trafics réseau.

Risques réseau couverts

Comme expliqué en début de chapitre, l'un des aspects critiques de la politique de sécurité de la multinationale RadioVoie est la définition des responsabilités entre les différents acteurs.

Le tableau 21.3 donne la matrice de l'ensemble de ces responsabilités.

Tableau 21.3 Matrice des responsabilités

	Opérateur de télécommunications	Équipe sécurité	Équipe sécurité militaire
Domaine réseau	Physique et logique	-	-
Zone d'accès routeur	Logique	Physique	-
Zone d'accès boîtier IPsec	-	Physique et logique	-
Intranet local	-	Physique et Logique	-
Intranet local réseau R&D		Physique et logique	-
Intranet local réseau R&D militaire	-	-	Physique et logique
Intranet local réseau bureautique	-	Physique et logique	-
Intranet local réseau production	-	Physique et logique	-
Réponse aux incidents	Non pour la partie cliente Oui pour la partie réseau	Oui	Oui

Du fait de l'isolation logique du réseau de RadioVoie en plusieurs niveaux, une sécurité en profondeur des flux de trafic est assurée. Le risque de pénétration directe du réseau RadioVoie est donc minime.

L'architecture mise en place produit des traces importantes des flux réseau transitant sur le réseau de RadioVoie, permettant ainsi d'analyser de manière fine tout incident de sécurité.

Risques réseau non couverts

Aucune architecture de haute disponibilité des accès réseau n'étant définie, le trafic réseau serait nécessairement impacté si l'un des équipements venait à défaillir.

Pour remédier à ce risque, les connexions réseau doivent être redondantes. Une architecture d'accès au réseau MPLS doit pour cela être mise en œuvre par l'opérateur de télécommunications, généralement au moyen de mécanismes de routage pour la perte d'une connexion réseau. Pour les autres équipements, tels les boîtiers IPsec et les pare-feu, des architectures spécifiques doivent être mises en œuvre localement afin de garantir l'acheminement du trafic.

La solution réseau repose sur l'offre d'un seul opérateur de télécommunications. Si le réseau de celui-ci vient à être attaqué par des moyens non connus de RadioVoie, les communications du réseau privé virtuel peuvent être rendues indisponibles ou, pire, injectées dans d'autres VPN ou sur Internet. Bien que ces cas de figure semblent peu probables, le transport des flux réseau de RadioVoie par des tunnels IPsec offre une garantie de sécurité logique suffisante.

Comme édicté dans la politique de sécurité réseau, l'opérateur de télécommunications doit toutefois garantir la sécurité des configurations des VPN au travers de rapports de contrôle réalisés sur les configurations des équipements du réseau MPLS.

Le risque le plus important reste la sécurité des secrets associés aux clés de chiffrement IPsec. Ces secrets doivent disposer d'une protection maximale et de procédures strictes afin d'éviter toute pénétration.

Tableau de bord de la sécurité

Cette section détaille les principaux contrôles à mettre en place et fournit des éléments de vérification fondés sur les outils maison ainsi qu'un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le contrôle de sécurité peut se réaliser à plusieurs niveaux :

- Le premier niveau de contrôle consiste à demander à l'opérateur de télécommunications de fournir des rapports de sécurité sur la configuration du réseau privé virtuel. Cela permet de garantir les bonnes pratiques de configuration des routeurs mais aussi de valider l'isolation logique du réseau privé virtuel de la multinationale.
- Le deuxième niveau de contrôle consiste à superviser les équipements de chiffrement et de filtrage de premier niveau permettant l'accès à un site de la multinationale. Des tableaux de bord peuvent être définis afin de suivre toute évolution des éléments de sécurité. Ce travail incombe à l'équipe de sécurité de la multinationale.
- Le troisième niveau de contrôle consiste à superviser les équipements de chiffrement et de filtrage du deuxième niveau permettant l'accès à un sous-réseau d'un site de la multinationale. Des tableaux de bord peuvent être aussi définis afin de suivre toute évolution des éléments de sécurité.

Mise en œuvre des outils maison

Cette section décrit la mise en œuvre des outils maison afin de répondre aux besoins de sécurité de RadioVoie. Elle détaille dans ce contexte la vérification des configurations

des MPLS/VPN, la vérification des périmètres réseau des MPLS/VPN et une analyse de risques du réseau.

Analyse des configurations

Comme indiqué précédemment, les configurations du ou des MPLS/VPN doivent être analysées afin de détecter toute mauvaise configuration par rapport au patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration Cisco suivant :

```
ip vrf A                        # nom de la vrf
 rd 1:1                         # route distinguisher associé à la vrf
 route-target export 1:1        # export de routes
 route-target import 1:1        # import de routes
 maximum routes 1000 10         # limitation du nombre de routes
!
```

La justification des éléments de configuration est fournie à la partie IV de l'ouvrage, relative à la configuration des équipements réseau.

Pour analyser les configurations de MPLS/VPN, nous utilisons notre l'outil HAWK avec le patron de sécurité suivant, qui implémente des contrôles en mode strict et laxiste :

```
margot/21.2/hawk$ cat vpn.tp
DECL {
   str this_vrf, vrf[], i, strict_state;
   int nb_vrf, check_rd[], check_route[];
}

BEGIN {
   nb_vrf = 0;
   strict_state = "no ip vrf";
}

*!:^ip vrf

# verification de la presence de vrf en mode strict
+ (
     :^ip vrf
     { vrf[++nb_vrf] = LINE; }

     # verification des lignes suivantes en mode laxiste
     *(
         [e]:^ rd [A-Z0-9]+:[0-9]+
         { check_rd[nb_vrf] = 1; }
         |
         :^ maximum routes 100 10
         { check_route[nb_vrf] = 1; }
         |
         :^[ ]
```

```
     )

     # poursuite de l'analyse
     *!:^ip vrf
)

SUCCESS {
forall(this_vrf = vrf[i])
   {
      if (check_rd[i] == 0)
          printf("%s;%s;(erreur) configuration n'implemente
          pas de RD conforme\n", FILENAME, this_vrf);
      if (check_route[i] == 0)
          printf("%s;%s;(erreur) configuration n'implemente
          pas de maximum-routes conforme\n", FILENAME, this_vrf);
   }
}

FAILURE {
printf("%s;%s (line %d);(erreur) configuration non conforme au mode strict;%s\n",
➥ FILENAME, LINE, LINENO, strict_state);
}
margot/21.2$
```

Si nous exécutons le programme HAWK sur une configuration Cisco qui ne respecte pas le patron de sécurité sur des contrôles codés en mode laxiste, nous obtenons le résultat suivant :

```
margot/21.2/hawk$ hawk -f ./vpn.tp vpn.txt
vpn.txt;ip vrf E;(erreur) configuration n'implemente pas de RD conforme
vpn.txt;ip vrf E;(erreur) configuration n'implemente pas de maximum-routes conforme
margot/21.2$
```

Cet exemple illustre en première erreur qu'un MPLS/VPN *vrf E* n'implémente pas un RD conforme avec le template de sécurité *rd AS:A5*. La seconde erreur est qu'un MPLS/VPN *vrf E* n'implémente pas de « maximum routes ».

Il est donc possible avec l'outil HAWK de contrôler en profondeur les configurations des MPLS/VPN et de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de périmètres MPLS/VPN

S'il est important de contrôler les configurations des équipements réseau, il est non moins primordial de valider les périmètres MPLS/VPN implémentés. Pour y parvenir, nous utilisons l'outil GRAPH ainsi qu'un script d'extraction utilisé pour déterminer les nœuds et les arcs de notre graphe MPLS/VPN.

Si nous désirons vérifier les périmètres de configuration des réseaux privés virtuels MPLS/VPN, l'approche consiste à analyser le graphe VPN engendré par les configurations des MPLS/VPN (seules les configurations des routeurs PE nous intéressent). Nous

définissons alors le périmètre de sécurité d'un MPLS/VPN comme étant l'ensemble des interconnexions autorisées de ce VPN avec d'autres VPN.

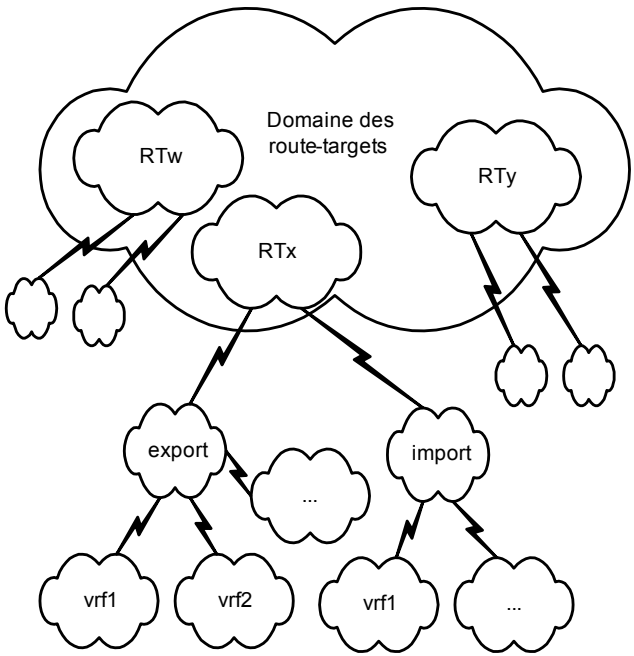
Si, pour chaque configuration PE, nous arrivons à renseigner les champs de la table VPN suivante, il est possible de construire le graphe MPLS/VPN :

```
table VPN
      champ : NomPEVrf : nom de PE concaténé avec le nom du VPN
      champ : I/E: définit l'action associée au route-target,
 [soit "import" (j'apprends les routes), soit "export" (j'exporte les routes)]
      champ : RT : définit la valeur du route-target
```

L'idée consiste ensuite à construire, pour chaque route-target, l'ensemble des VRF qui réalisent un « import » et l'ensemble des VRF qui réalisent un « export », comme l'illustre la figure 21.25.

Figure 21.25

Hiérarchie des routes-targets



Nous pouvons en déduire les liens de connectivité entre les VRF. Ainsi, pour une route-target donnée, chaque VRF appartenant à la liste des « exports » est connectée à toutes les VRF appartenant à la liste des « imports ». Nous pouvons donc construire un graphe VPN, où un sommet est représenté par une VRF et un arc par une connexion entre deux VRF différentes.

Une fois la table VPN construite à partir de l'extraction des informations contenues dans les configurations, le produit cartésien de la table VPN par elle-même sous les

conditions suivantes donne tous les arcs de notre graphe VPN, comme l'illustre la requête SQL suivante :

```
SELECT
          Vpn.NomPEVrf, Vpn.I/E, Vpn.Rt, Vpn_1.NomPEVrf, Vpn_1.I/E, Vpn_1.Rt
FROM
      Vpn, Vpn AS Vpn_1
WHERE
      Vpn.Rt = Vpn_1.Rt and
          Vpn.IE = "export" and Vpn_1.IE = "import" and
          Vpn.NomPEVrf != Vpn_1.NomPEVrf
```

Un sommet du graphe VPN est représenté par *NomPEVrf* et un arc par un enregistrement trouvé par le produit cartésien précédemment décrit. L'asymétrie de configuration d'un VPN indique que le graphe VPN construit est dirigé.

Le calcul des composantes connexes (s'il existe un chemin entre toute paire de sommets (*x*,*y*) de la composante) et 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*) permet de déterminer les périmètres de sécurité des VPN.

Une fois les nœuds et les arcs extraits des configurations, nous fournissons ces données à l'outil GRAPH afin qu'il calcule les composantes connexes du graphe MPLS/VPN. Les nœuds contenus dans une composante fortement connexe impliquent donc qu'ils communiquent entre eux. En revanche, si les composantes connexes ne sont pas égales aux composantes fortement connexes, c'est qu'il existe des inconsistances de configuration. De même, toute configuration non bidirectionnelle entre deux sommets montre des inconsistances de configuration.

Si nous appliquons cette méthode à l'exemple de configuration présenté précédemment, nous obtenons les résultats suivants :

```
margot/21.2/graph_vpn$ ./vpn_graph.sh
<stdin>: 7 nodes, 10 edges, 3578 bytes
# nodes = 7
# edges = 10
#
N       pe1-A
N       pe2-A
N       pe1-B
N       pe1-C
N       pe1-D
N       pe3-D
N       pe1-E
#
D       pe1-A pe2-A
D       pe2-A pe1-A
D       pe1-D pe3-D
D       pe3-D pe1-D
D       pe1-E pe1-A
```

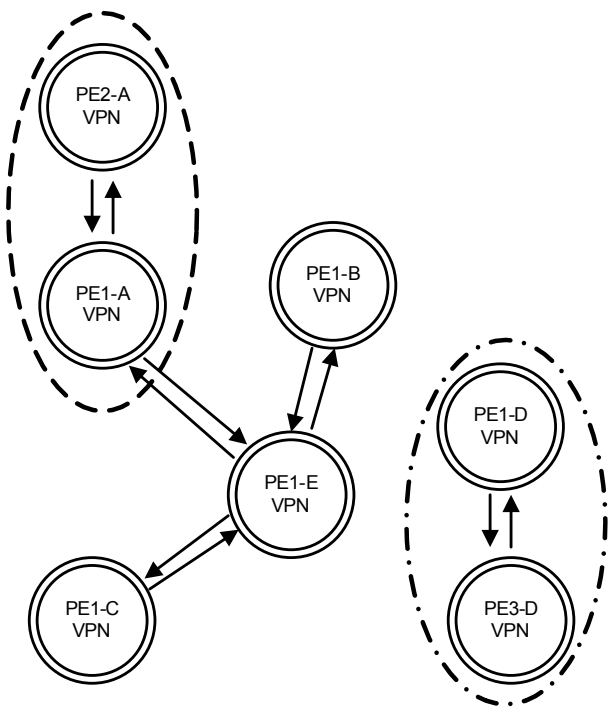
```
D       pe1-E pe1-B
D       pe1-E pe1-C
D       pe1-A pe1-E
D       pe1-B pe1-E
D       pe1-C pe1-E
connected component (5 nodes):
{ pe1-A pe2-A pe1-B pe1-C pe1-E }
articulation point: pe1-A
node partition: { pe2-A }
node partition: { pe1-B pe1-C pe1-E }
articulation point: pe1-E
node partition: { pe1-A pe2-A }
node partition: { pe1-B }
node partition: { pe1-C }
connected component (2 nodes):
{ pe1-D pe3-D }
```

Les résultats de l'outil GRAPH indiquent que les deux composantes connexes suivantes ont été trouvées, comme illustré à la figure 21.26 :

- Composante 1 : les nœuds A, B, C et E peuvent communiquer entre eux. Cependant, le nœud E est un point d'articulation pour la composante fortement connexe.
- Composante 2 : le nœud D est isolé.

Figure 21.26

Interconnexions entre les VPN



Le contrôle de sécurité consiste donc à vérifier si les périmètres de sécurité trouvés correspondent bien aux périmètres de sécurité demandés. En cas d'erreur de configuration, l'isolation n'est plus assurée. Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de périmètres BGP

S'il est important de contrôler les configurations des équipements réseau, il est non moins primordial de valider les périmètres BGP implémentés. Pour y parvenir, nous utiliserons l'outil GRAPH ainsi qu'un script d'extraction utilisé pour déterminer les nœuds et les arcs de notre graphe BGP.

Si nous désirons vérifier les périmètres de configuration du routage BGP, l'approche consiste à analyser le graphe BGP engendré par les configurations des équipements réseau. Nous détaillerons uniquement le graphe BGP associé aux sessions eBGP (c'est-à-dire appartenant à des AS différents).

Nous pouvons extraire les informations des sessions BGP 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 utilisée pour
➥ définir les sessions de routage.
router bgp autonomous-system: définit le système autonome du processus BGP.
neighbor ip-address ...: définit les sessions de routage.
```

De manière plus précise, il nous faut extraire les informations de routage BGP à partir des configurations des équipements réseau afin de créer le fichier bgp.ne trié sur le champ neighbor ip-address et le fichier bgp.ip trié sur le champ ip-address. Ces informations sont alors utilisées afin de déduire le graphe BGP par une jointure algébrique entre les fichiers bgp.ne et bgp.ip, comme l'illustre la requête SQL suivante :

```
SELECT
    bgp.ne.hostname, bgp.ne.as, bgp.ip.hostname, bgp.ip.as
FROM
    bgp.ne, bgp.ip
WHERE
    bgp.ne.neighbor.ip = bgp.ip.ip
    and bgp.ne.hostname != ngp.ip.hostname
    and bgp.ne.as != ngp.ip.as
```

Un sommet du graphe BGP est représenté par bgp.ne.hostname, bgp.ne.as, et un arc par un enregistrement trouvé par le produit cartésien précédemment décrit.

Le calcul des composantes connexes, s'il existe un chemin entre toute paire de sommets (*x*,*y*) de la composante, permet de déterminer les périmètres de sécurité BGP, comme dans la commande suivante :

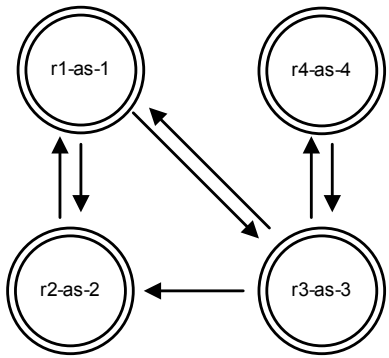
```
margot/21.2/graph_bgp$ ./bgp_graph.sh
```

```
<stdin>: 4 nodes, 7 edges, 1536 bytes
# nodes = 4
# edges = 7
#
N       r1-as-1
N       r2-as-2
N       r3-as-3
N       r4-as-4
#
D       r1-as-1 r2-as-2
D       r1-as-1 r3-as-3
D       r2-as-2 r1-as-1
D       r3-as-3 r1-as-1
D       r3-as-3 r2-as-2
D       r3-as-3 r4-as-4
D       r4-as-4 r3-as-3

connected component (4 nodes):
{ r1-as-1 r2-as-2 r3-as-3 r4-as-4 }
paths:

r1-as-1 <-> r2-as-2
asymetric paths:
cost: 1 < (r1-as-1 -> r2-as-2) >
cost: 1 < (r2-as-2 -> r1-as-1) >

r1-as-1 <-> r3-as-3
asymetric paths:
cost: 1 < (r1-as-1 -> r3-as-3) >
cost: 1 < (r3-as-3 -> r1-as-1) >

r1-as-1 <-> r4-as-4
asymetric paths:
cost: 2 < (r1-as-1 -> r3-as-3) (r3-as-3 -> r4-as-4) >
cost: 2 < (r4-as-4 -> r3-as-3) (r3-as-3 -> r1-as-1) >

r2-as-2 <-> r3-as-3
asymetric paths:
cost: 2 < (r2-as-2 -> r1-as-1) (r1-as-1 -> r3-as-3) >
cost: 1 < (r3-as-3 -> r2-as-2) >

r2-as-2 <-> r4-as-4
asymetric paths:
cost: 3 < (r2-as-2 -> r1-as-1) (r1-as-1 -> r3-as-3) (r3-as-3 -> r4-as-4) >
cost: 2 < (r4-as-4 -> r3-as-3) (r3-as-3 -> r2-as-2) >

r3-as-3 <-> r4-as-4
asymetric paths:
cost: 1 < (r3-as-3 -> r4-as-4) >
cost: 1 < (r4-as-4 -> r3-as-3) >
```

Les résultats de l'outil GRAPH indiquent qu'il n'y a qu'un seul composant connexe, comme l'illustre la figure 21.27 :

Figure 21.27

Graphe iBGP



La sortie de l'outil GRAPH illustre aussi les chemins du graphe avec l'option -p pointant la non-symétrie de configuration BGP entre les routeurs r2-as-2 et r3-as-3.

Le contrôle de sécurité consiste donc à vérifier si les périmètres de sécurité trouvés correspondent bien aux périmètres de sécurité demandés. En cas d'erreur de configuration, l'isolation n'est plus assurée. Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de risques

Il est primordial de déterminer un niveau de risque pour le réseau correspondant aux vulnérabilités de sécurité détectées. Rappelons qu'il s'agit de déterminer le risque pris si ces vulnérabilités de sécurité ne sont pas corrigées.

Nous utilisons notre outil BAYES afin de connaître le niveau de risque des réseaux interne et externe. La modélisation pour notre calcul de risque est la suivante : pour chaque objet, il y a 3 tests possibles, pouvant référencer une ou plusieurs vulnérabilités. De plus, il y a 10 impacts possibles (pas d'impact, 3 impacts pour le VPN vert, 3 impacts pour le VPN bleu et 3 impacts pour le cœur de réseau), comme le résume le tableau 21.4.

Tableau 21.4 Répartition des tests et des impacts

Objet	Test	Impact
Cœur de réseau (routeurs PE, P)	1	1 (faible impact réseau)
	2	2 (moyen impact réseau)
	3	3 (fort impact réseau)
CE vert	4	4 (faible impact VPN vert)
	5	5 (moyen impact VPN vert)
	6	6 (fort impact VPN vert)

Objet	Test	Impact
CE bleu	7	7 (faible impact VPN bleu)
	8	8 (moyen impact VPN bleu)
	9	9 (fort impact VPN bleu)

Dans ce modèle, si nous tenons compte uniquement de la topologie réseau, les règles de propagation sont les suivantes :

```
margot/21.2/bayes$ cat dmz.rule
0 ce_bleu ce_bleu 4 5 6        # règles de propagation à la racine
0 ce_vert ce_vert 7 8 9
0 pe pe 1 2 3
0 p p 1 2 3
1 pe pe 1                      # règles de propagation hors racine
2 pe pe 2
3 pe pe 3
1 p p 1
2 p p 2
3 p p 3
4 ce_bleu ce_bleu 4
5 ce_bleu ce_bleu 5
6 ce_bleu ce_bleu 6
7 ce_vert ce_vert 7
8 ce_vert ce_vert 8
9 ce_vert ce_vert 9
4 ce_bleu ce_bleu 4
5 ce_bleu ce_bleu 4 5
6 ce_bleu ce_bleu 4 5 6
6 ce_bleu pe 3
7 ce_vert ce_vert 7
8 ce_vert ce_vert 7 8
9 ce_cert ce_vert 7 8 9
9 ce_vert pe 3
1 pe pe 1
2 pe pe 1 2
3 pe pe 1 2 3
1 pe p 1
2 pe p 1 2
3 pe p 1 2 3
1 pe p 1
2 pe p 1 2
3 pe p 1 2 3
1 p p 1
2 p p 1 2
3 p p 1 2 3
1 p pe 1
2 p pe 1 2
3 p pe 1 2 3
```

Si nous prenons aussi en compte les fichiers de conséquences et de probabilités suivants :

```
margot/21.2/bayes$ cat dmz.cons
0     /* aucun impact */
6     /* impact faible : réseau */
30    /* impact moyen : réseau */
60    /* impact fort : réseau */
2     /* impact faible : ce_bleu */
10    /* impact moyen : ce_bleu */
20    /* impact fort : ce_bleu */
2     /* impact faible : ce_vert */
10    /* impact moyen : ce_vert */
20    /* impact fort : ce_vert */

margot/21.2/bayes$ cat dmz.proba
0.1   /* pas d'impact */
0.3   /* impact faible : réseau */
0.3   /* impact moyen : réseau */
0.8   /* impact fort : réseau */
0.3   /* impact faible : ce_bleu */
0.3   /* impact moyen : ce_bleu */
0.8   /* impact fort : ce_bleu */
0.3   /* impact faible : ce_vert */
0.3   /* impact moyen : ce_vert */
0.8   /* impact fort : ce_vert */
```

il est possible d'exécuter le programme BAYES pour chacun des fichiers de vulnérabilités détectés par les contrôles internes et externes.

Le Makefile suivant permet de lancer une simulation composée de six fichiers en considérant les mêmes paramètres de règles, conséquences et probabilités :

```
margot/21.2/bayes$ cat Makefile
PGM=bayes

mpls:
        normalise mpls.rule mpls.proba mpls.txt mpls.cons
        $(PGM) mpls.txt.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt1 mpls.cons
        $(PGM) mpls.txt1.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt2 mpls.cons
        $(PGM) mpls.txt2.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt3 mpls.cons
        $(PGM) mpls.txt3.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt4 mpls.cons
        $(PGM) mpls.txt4.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt5 mpls.cons
        $(PGM) mpls.txt5.ref.dat[1234] 1000
```

Nous exécutons alors le programme BAYES sur les différents fichiers contenant les vulnérabilités de sécurité :

```
margot/21.2/bayes$ gmake mpls | grep distribution

distribution des probabilités (impacts): 4.676320e-01  0.000000e+00  0.000000e+00
➥ 0.000000e+00  6.528000e-02  1.368000e-01  0.000000e+00  3.002880e-01
➥ 3.000000e-02  0.000000e+00 / somme=1.000000e+00

distribution des probabilités (impacts): 4.195909e-01  0.000000e+00  0.000000e+00
➥ 0.000000e+00  4.632369e-02  9.707538e-02  5.538462e-02  2.499762e-01
➥ 7.626462e-02  5.538462e-02 / somme=1.000000e+00

distribution des probabilités (impacts): 3.133001e-01  3.749214e-02  0.000000e+00
➥ 1.070575e-01  1.344220e-02  2.816932e-02  1.473982e-01  3.188917e-01
➥ 1.859661e-02  1.565217e-02 / somme=1.000000e+00

distribution des probabilités (impacts): 3.121092e-01  5.179567e-02  3.855578e-02
➥ 1.454619e-01  1.113643e-02  2.333737e-02  1.222497e-01  2.667094e-01
➥ 1.555353e-02  1.309091e-02 / somme=1.000000e+00

distribution des probabilités (impacts): 3.374458e-01  5.940910e-02  4.422309e-02
➥ 0.000000e+00  1.411175e-02  2.957243e-02  1.542497e-01  3.259782e-01
➥ 1.900987e-02  1.600000e-02 / somme=1.000000e+00

distribution des probabilités (impacts): 4.098510e-01  0.000000e+00  0.000000e+00
➥ 0.000000e+00  3.857143e-02  0.000000e+00  0.000000e+00  5.515776e-01
➥ 0.000000e+00  0.000000e+00 / somme=1.000000e+00

margot/21.2/bayes$ gmake mpls|grep risque
risque : 2.399136e+00
risque : 4.541384e+00
risque : 1.104174e+01
risque : 1.384658e+01
risque : 6.254144e+00
risque : 1.180298e+00
```

Exemple de tableau de bord de la sécurité réseau

Le tableau 21.5 récapitule les éléments de l'architecture réseau qui permettent d'établir des tableaux de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 21.5 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés

Sous-réseau	Catégorie	Élément
Recherche	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, etc.) et pare-feu (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMR R&D) et systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateurs (vérification VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateurs, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateurs (vérification VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), modems (routeurs accès non autorisés, etc.), boîtier IPsec (sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés
Production	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), serveurs dédiés d'authentification (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.) et serveurs dédiés d'authentification (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés

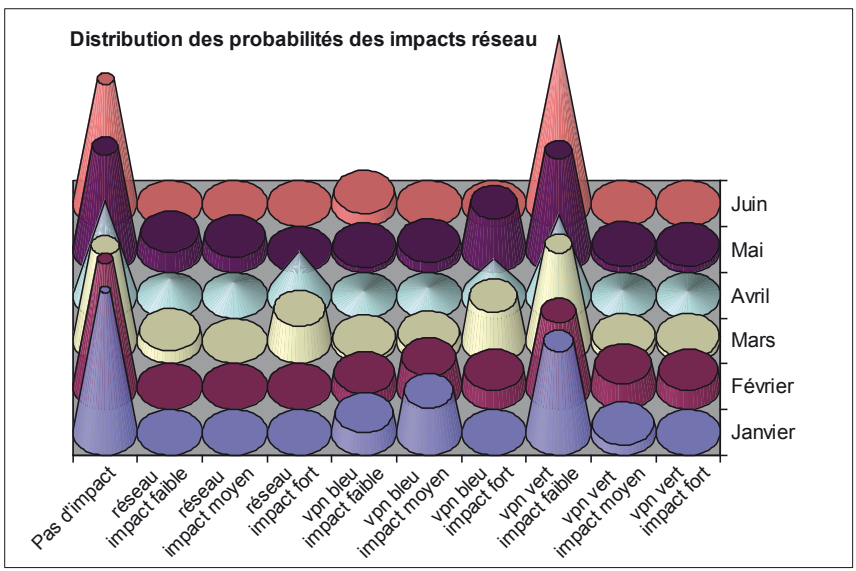
Sous-réseau	Catégorie	Élément
Administration	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.), pare-feu (vérification des règles, etc.) et serveurs d'administration (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.), pare-feu (sessions échouées, sessions intranet, etc.) et serveurs d'administration (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (supervision IPsec, supervision pare-feu, supervision équipement réseau) et systèmes connectés
Administration production	Configuration	Des commutateurs (vérification VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés

Le tableau de bord de la sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous calculons tous les scénarios d'événements possibles par le biais d'un arbre probabiliste (fondé sur les faiblesses de sécurité préalablement détectées), il est possible de déterminer les probabilités associées pour chaque niveau d'impact, comme l'illustre la figure 21.28 (les résultats correspondent à l'exemple détaillé précédemment).

Figure 21.28

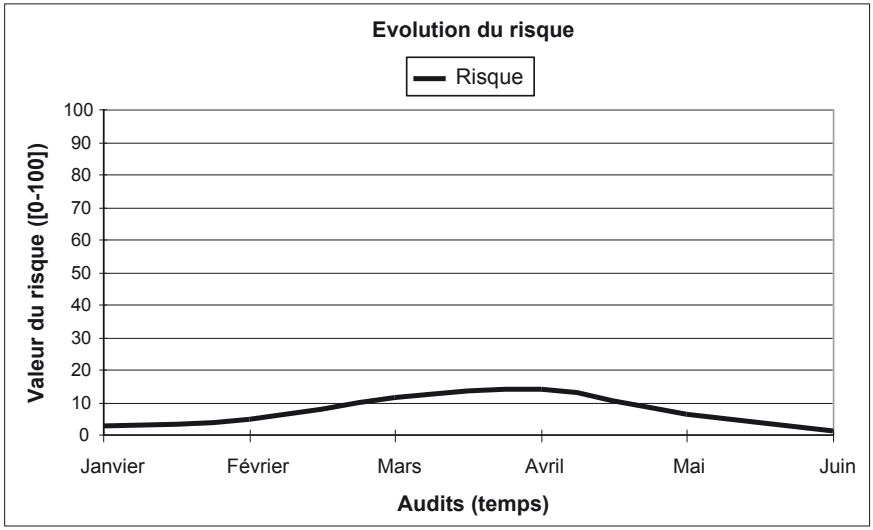
Distribution des probabilités des impacts réseau sur trois mois



Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées pour calculer le risque de 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 illustré à la figure 21.29 (les résultats correspondent à l'exemple détaillé précédemment).

Figure 21.29

Évolution du risque dans le temps



En résumé

Quelle que soit l'entreprise, l'analyse initiale des besoins de sécurité et la définition d'une politique de sécurité réseau sont les étapes capitales qui précèdent la mise en place d'architectures techniques et de solutions de sécurité.

Toute architecture ou solution de sécurité a ses forces et ses faiblesses, qu'il faut connaître et surveiller. En cas d'incident de sécurité, une alerte du niveau de sécurité approprié doit être déclenchée.

Des contrôles périodiques et en profondeur doivent être menés afin de vérifier l'application de la politique de sécurité réseau. Dans ce contexte, des tableaux de bord de sécurité peuvent être définis, suivis et réactualisés régulièrement afin de tenir compte de toute évolution des architectures et des mécanismes de sécurité.

Au travers de cette étude de cas, les principes présentés dans l'ensemble de l'ouvrage, sans lesquels aucune stratégie de sécurité ne saurait réussir, ont été méthodiquement appliqués : définition des besoins de sécurité de l'entreprise, définition d'une politique de sécurité réseau, mise en œuvre des solutions techniques adaptées, mise en place d'un contrôle de sécurité et établissement d'un tableau de bord de la sécurité afin de vérifier que la politique de sécurité définie est appliquée.

Index