18 oct 2008

Quelques éléments de sécurité autour du protocole de routage BGP

Catégorie : Administration réseau Tags : misc

~~Retrouvez cet article dans : **Misc 21**~~

L’une des problématiques récurrentes des réseaux est de faire transiter des données le plus rapidement et le plus sûrement possible. 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.

1. Introduction

Le déploiement de réseaux IP de grande taille a rapidement nécessité la mise au point de protocoles de routage dynamique chargés de déterminer le plus efficacement possible la meilleure route pour atteindre une destination donnée. Par ailleurs, il a aussi été nécessaire de découper le réseau en différents systèmes autonomes (ou Autonomous System) afin de réduire cette complexité en taille. Il doit être noté que les systèmes autonomes du cœur de réseau Internet sont gérés par les opérateurs de télécommunications.
Ces considérations ont donné lieu à une classification des protocoles de routage dynamique en deux grandes familles. Les protocoles de routage dynamique IGP (Interior Gateway Protocol) qui permettent d’échanger des informations d’accessibilité au sein d'un système autonome. Les protocoles de routage dynamique EGP (Exterior Gateway Protocol) qui permettent d’échanger des informations d’accessibilité entre systèmes autonomes.
Le protocole BGP s’appuie sur la couche TCP (port 179) et fait partie de la famille des protocoles EGP. Le mode de fonctionnement du protocole BGP entre deux routeurs consiste à établir une connexion TCP et à échanger d’une manière dynamique les annonces de routes [RFC1771].
Le protocole BGP est basé sur l'algorithme de Bellman-Ford et consiste à optimiser de manière itérative la distance de x à y en passant par un voisin z. C'est un algorithme à correction d'étiquettes (label correcting algorithms) pouvant affiner à chaque itération le coût associé à une distance. Dans un tel contexte de routage, le protocole BGP n'a pas de vision globale de la topologie de routage et envoie donc uniquement à ses voisins les annonces de routes. Pour éviter tout bouclage de routes, le protocole BGP gère un attribut contenant l'ensemble des AS traversés. De plus, les sessions iBGP ne redistribuent par les routes apprises en iBGP pour éviter les phénomènes de bouclage.
Les connexions établies entre des routeurs appartenant à des AS distincts sont qualifiées de type eBGP

(external BGP), alors que les connexions établies entre des routeurs BGP appartenant au même AS sont qualifiées de type iBGP (internal BGP) comme l'illustre la figure 1.

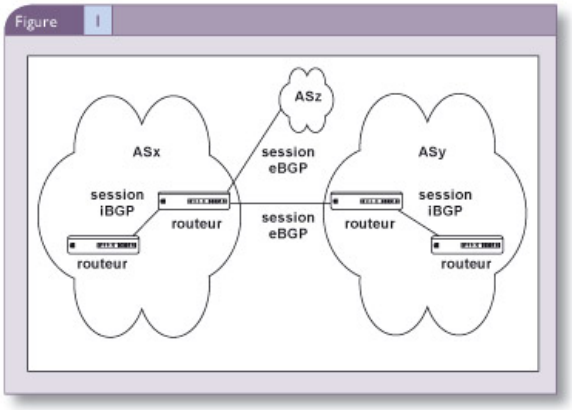

Fig. 1

Sachant que toute attaque ou perturbation du routage peut impacter directement la disponibilité du réseau et de ses services, il est primordial de considérer les protocoles de routage comme des éléments-clés de la sécurité d'un réseau. Il doit être noté qu'il est aussi possible de détourner du trafic par le routage à des fins de vol d'information.

2. Les mécanismes de sécurité

2.1. Le contrôle des topologies de routage iBGP

Les routes apprises par les sessions eBGP d'un système autonome doivent être propagées au sein du système autonome par le biais de sessions iBGP. Il s'agit effectivement de maintenir une vue cohérente de l'ensemble des routes externes au système autonome pour l'ensemble des routeurs.
La spécification initiale de BGP suppose qu'un graphe complet (modèle « complet ») de sessions iBGP soit configuré au sein du système autonome pour distribuer les routes inter-domaines. Par conséquent, il doit y avoir $\frac{n*(n-1)}{2}$ sessions iBGP au sein d'un système autonome si n est le nombre de routeurs. La raison est que les sessions iBGP ne redistribuent pas les routes apprises en iBGP pour éviter les phénomènes de bouclage.
Par exemple, pour un réseau contenant 100 routeurs, il serait nécessaire de configurer de l'ordre de 5000 sessions iBGP au total dans les configurations des routers. Deux modèles ont alors été proposés pour résoudre la problématique des configurations des sessions iBGP. Le modèle des confédérations (que nous ne détaillerons pas) [RFC3065] et celui des réflecteurs de routes que nous détaillons ci-après [RFC2796].
Un réflecteur de route est un routeur BGP qui peut redistribuer sur des sessions iBGP les routes qu'il a apprises d'autres sessions iBGP. Un réflecteur de routes a des voisins clients et des voisins non-clients (les voisins non-clients sont considérés ici comme des réflecteurs de routes). Un réflecteur de routes reçoit des routes de tous ses voisins iBGP et utilise son processus de décision BGP afin de déterminer les meilleures routes pour joindre chaque destination. Si la meilleure route a été reçue sur une session iBGP avec un voisin client, le réflecteur de route ré-annoncera cette route à tous ses voisins iBGP. En revanche, si la route a été reçue d'un voisin non-client, alors la route ne sera annoncée qu'aux voisins clients comme l'illustre la figure suivante :

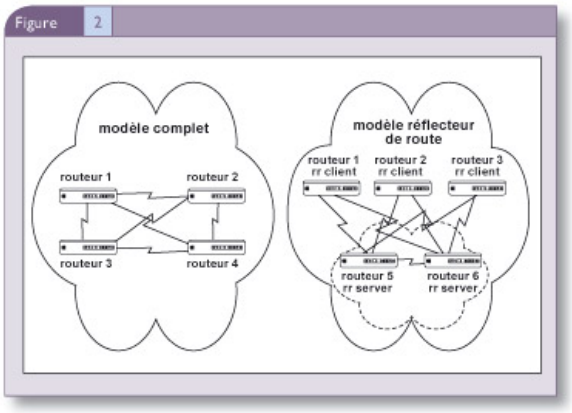

Fig.2

Le modèle réflecteur de routes permet donc de réduire le nombre de configurations nécessaires.

Sachant que le sous-graphe associé aux réflecteurs de routes doit est complet, il doit y avoir $\frac{n*(n-1)}{2}$ sessions iBGP entre les réflecteurs de routes. Cependant, le nombre de réflecteurs de routes nécessaires est par architecture très inférieur comparé au nombre de routeurs dans le système autonome.
Dans ces deux types de topologies, le contrôle des topologies de routage est primordial afin d'assurer la disponibilité du réseau et de ses services comme nous le verrons par la suite. Enfin, il doit être noté qu'une combinaison des deux modèles est possible afin d'éviter d'avoir uniquement un modèle « complet », très consommateur en termes de mémoire et de temps processeur, mais aussi d'avoir un modèle « Réflecteur de route », apportant des problématiques de routage sous-optimales.

2.2. Le contrôle par les secrets partagés

Le contrôle d'une session de routage BGP entre deux routeurs peut être réalisé par l'option d'empreinte MD5 véhiculée dans les paquets TCP. Il s'agit alors de vérifier en point à point les annonces de routes échangées entre deux routeurs à l'aide d'un secret partagé ou clé secrète [RFC2385]. Ce contrôle doit être mis en œuvre en priorité sur les sessions eBGP qui sont le plus à risque pour un AS.
Sachant que les deux routeurs possèdent un secret partagé, une empreinte basée sur une fonction de hachage (MD5, SHA1, ..) est alors générée pour contrôler les échanges de routes. Plus précisément, quand un routeur émet un paquet IP contenant des données BGP, une empreinte est calculée et insérée dans le paquet TCP, puis vérifiée par l'autre routeur BGP comme l'illustre la figure suivante :

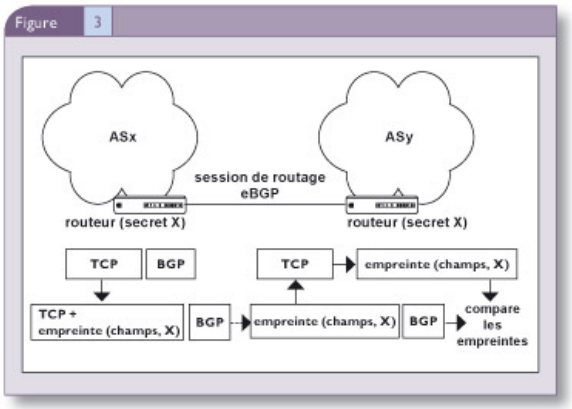

Fig.3

Cette empreinte est calculée à partir de la clé secrète et de champs constants qui n'ont pas été modifiés par le processus d'acheminement du paquet tels que :

- L'adresse IP source ;
- L'adresse IP destination ;
- ....
- L'en-tête TCP sans les options avec un checksum à 0 ;
- Les données du segment TCP ;
- Le secret partagé ou clé secrète (qui aura été distribué par un canal sécurisé).

Cette empreinte est alors insérée dans le champ option du paquet TCP et permet de mettre en œuvre un mécanisme de contrôle d'une session de routage BGP. En revanche, elle ne permet pas d'authentifier le chemin pris par une route ainsi que l'origine de la route.
Enfin, différents secrets partagés permettent aussi de créer des groupes distincts ou périmètres de sécurité entre les sessions iBGP et les diverses sessions eBGP.

2.3. Le contrôle par les TTL

Une autre méthode pour contrôler une session de routage BGP consiste à mettre en place un contrôle du TTL (Time To Live) contenu dans les paquets IP échangés par la session de routage BGP. Ce contrôle doit être mis en œuvre en priorité sur les sessions eBGP qui sont le plus à risque pour un AS. En effet, partant du principe que les sessions de routage BGP entre deux routeurs sont généralement directes, les paquets IP contenant des informations de routage BGP émis par un routeur doivent arriver à l'autre routeur avec un TTL = TTL -1 comme l'illustre la figure suivante :

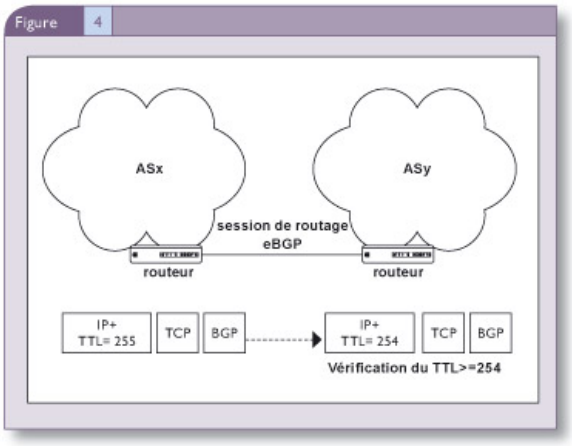

Fig.4

Comme une annonce de routes entre deux routeurs correspond à chaque fois à un nouveau paquet IP, le TTL du paquet IP émis sera par défaut égal à 255. Ainsi, si l'autre routeur reçoit des annonces de routes ayant un TTL qui n'est pas égal à 254, il peut en conclure que ce n'est pas le routeur avec lequel il a une session de routage qui a émis cette annonce.
Ce contrôle permet de mettre en œuvre un mécanisme de contrôle d'une session de routage BGP. En revanche, il ne permet pas d'authentifier le chemin annoncé par une route ainsi que l'origine de la route.

2.4. Le contrôle des annonces

de routes eBGP
Les annonces de routes peuvent être soumises à une réelle politique de routage définie par l'administrateur d'un système autonome (opérateur de télécommunications). Cette politique peut à la fois s'appliquer aux annonces de routes émises vers un système autonome (routes transmises à l'intérieur d'un AS) ainsi qu'aux annonces de routes qu'émet le système autonome (routes émises à l'extérieur d'un AS) comme l'illustre la figure 5.
Cette politique de routage définit des règles de contrôle ou de filtrage basées sur (la liste n'est pas exhaustive) :

- Des listes de filtrages associées aux valeurs des systèmes autonomes. Par exemple, telle route ne peut être annoncée que par la liste des systèmes autonomes suivants.
- Des listes de filtrages associées aux préfixes annoncés ou émis. Par exemple, certains préfixes ne doivent pas être annoncés [RFC1918].
- Des contrôles de l'instabilité des routes. Par exemple, si un préfixe fait l'objet de mises à jour incessantes, il peut être alors mis en quarantaine afin de protéger le processus de routage BGP.

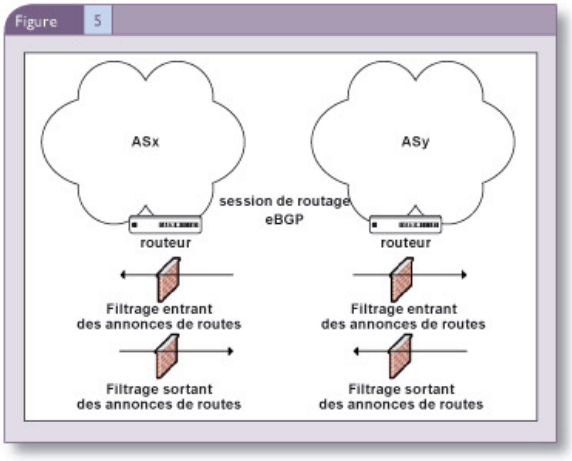

Fig.5

Ces mesures de sécurité ont pour objectif de protéger le réseau d’éventuelles attaques de routage qui pourraient impacter sa disponibilité. En revanche, elles ne permettent pas d'authentifier le chemin pris par une route ainsi que l'origine de la route.

2.5. Le contrôle des attaques

de type Déni de Service
Le protocole BGP ne constitue pas une contre-mesure aux attaques de type Déni de Service, mais peut aider à anticiper et limiter leurs effets [RFC3882]. Dans le cas d’une attaque de type Déni de Service, ce ne sont pas les équipements réseau contenus au sein d'un AS qui sont généralement visés, mais plutôt les équipements (serveurs web, de mail, etc.) de ses clients. Une telle attaque peut alors générer un trafic important qui, dans le meilleur des cas, écroulera seulement le lien d’accès du client et, dans le pire des cas, écroulera un ou des liens d’infrastructure de l’opérateur de télécommunications.

2.5.1 BGP et le routage complet

Sachant qu'Internet n’est qu’une interconnexion de réseaux, les routeurs d'un système autonome connaissent (après la convergence des tables de routage) l’ensemble des routes annoncées au sein

d'Internet. A l'heure actuelle, un routeur d'infrastructure d'un opérateur de télécommunications peut apprendre jusqu'à 200.000 routes et sait donc comment atteindre toutes les adresses disponibles sur Internet [ROUTES].
Partant de la constatation que les vers/virus se propagent généralement en commençant par l'adresse 0.0.0.0 et en incrémentant de 1 jusqu'à l'adresse 255.255.255.255 ou utilisent des modes de distributions différents (toujours en incrémentant mais en partant de certaines classes C, tirages « aléatoires » d'adresses, algorithmes d'incrémentations avec des pas différents de 1, etc.), le protocole BGP va permettre d'agir ici comme un signal d'alarme.
En effet, si on connaît toutes les routes de l'Internet et qu'on reçoit un paquet qu'on ne sait pas router (i. e. un paquet en adresse privée ou réservée RFC[1918]), il est alors fort probable qu'il s'agisse d'une attaque de type Déni de Service.
La mise en œuvre d'un tel mécanisme consiste à mettre au sein de son infrastructure un système appelé « puits », de lui donner une adresse IP et d'annoncer dans BGP cette adresse comme la route par défaut. Ainsi, quand un routeur ne saura pas comment router un paquet, c'est-à-dire qu'il n'a pas appris en BGP vers où envoyer ce paquet, il va donc l'envoyer vers la route par défaut. On est alors vite alerté sur les vers/virus en temps réel. Les avantages sont les suivants :

- L'alerte est quasi-temps réel, bien avant les annonces des organismes officiels ou des éditeurs.
- On dispose aussitôt du pattern de l'attaque, ce qui permet d'adapter sa politique de filtrage.
- On peut alerter rapidement ses clients.

Bien que cela semble idéal en théorie, la pratique nécessite quelques réglages. Ce système « puits » va non seulement recevoir des attaques réelles, mais aussi de fausses attaques dues simplement à des erreurs de configuration de routage. Il convient donc de ne pas considérer l'arrivée d'un paquet comme une attaque réelle. En revanche, en cas d'arrivée massive de paquets, il est fort probable qu'on soit en présence d'un vers/virus. Moyennant de fixer de bons seuils, cette solution peut donner des informations très intéressantes avec très peu de faux positifs.
Enfin, quand on déclare au sein de son réseau une route par défaut, il convient de s'assurer de ne pas annoncer cette route par défaut à d'autres systèmes autonomes sous peine de faire face à des conséquences fâcheuses.

2.5.2 BGP et puits de routage (black hole)

Dans le cas d'un système autonome avec de multiples points d'accès vers d'autres AS, l'attaque peut venir de différentes sources et il n'est pas envisageable d'intervenir sur tous les routeurs d'interconnexion. Il suffit donc pour l'adresse visée par ces attaques de l'annoncer dans BGP avec comme chemin le système puits de routage. Ce puits de routage met alors à la poubelle systématiquement tout le trafic qu'il reçoit.
Il est cependant possible de faire plus simple et d'éviter de transporter ce flux inutile. Dans BGP, on va annoncer que le chemin pour atteindre l'adresse IP visée par l'attaque est une interface poubelle du routeur lui-même (null0 pour CISCO par exemple). Une fois que BGP aura propagé cette information, dès qu'un routeur recevra un paquet à destination de l'adresse attaquée, il le détruira.
On notera que du côté du client visé et de l'attaquant, l'attaque a parfaitement réussi puisque ce client a été inaccessible le temps de l'attaque. Bien que cette solution permette à l'opérateur de protéger son cœur de réseau, cette solution n'est donc pas très satisfaisante. Il faut donc trouver une solution qui protège le cœur de réseau de l'opérateur et qui garantit un minimum de service au client visé par un Déni de Service.

2.5.3 BGP et puits de filtrage (sink hole)

Les équipements réseau n'ont pas forcément la capacité à analyser et à filtrer le trafic pour séparer le trafic légitime de celui de l'attaque. L'idée est donc de rediriger le trafic vers un équipement dédié, qui lui aura cette capacité.
Dans ce cas, BGP ne propage pas l'adresse du « puits de routage », mais celle du « puits de filtrage ». Ainsi, tous les paquets à destination de l'adresse IP attaquée vont passer par cet équipement filtrant. Le système « puits de filtrage » permettra alors de déterminer exactement l'attaque à l'aide d'outils embarqués (Snort, Radware Defense Pro, etc.).

Une fois les données analysées, le trafic épuré de l'attaque sera alors envoyé vers l'adresse IP destinatrice. D'un point de vue du routage, on a tout d'abord routé du trafic externe vers le puits au sein de l'AS par un protocole EGP. On a ensuite injecté le trafic épuré à partir du puits et routé ce trafic vers l'adresse IP destinatrice par un protocole de routage IGP (IGP achemine alors ce trafic vers l'adresse IP destinatrice dont il connaît le chemin pour l'atteindre au sein de son AS) comme l'illustre la figure suivante :

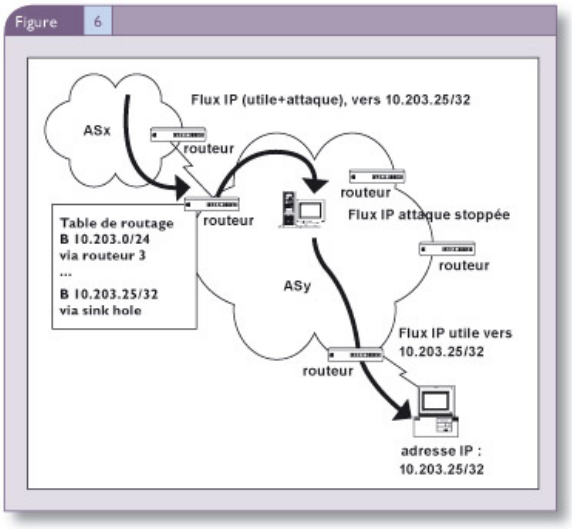

Fig.6

Pour le client, cette solution est bien plus efficace que la précédente, car son trafic n'a pas été complètement coupé. Cependant, cette solution a aussi ses limites s'il s'agit par exemple d'une attaque vers le port HTTP provenant d'une multitude de différentes sources, le filtre serait alors « on interdit le trafic HTTP vers cette adresse IP ». Si le client est un hébergeur web, il appréciera tout de même de pouvoir conserver son trafic email. Il doit être cependant noté que des règles de filtrage basées sur les données applicatives peuvent être aussi définies.
Une fois le déni de services arrêté, le retour à la normale consiste à arrêter d'annoncer des routes spécifiques, les paquets utiliseront alors automatiquement le chemin standard.

2.6. Vers le contrôle

de l'authentification des routes
Bien qu'il existe un certain nombre d'éléments de configuration permettant de renforcer la sécurité des sessions BGP, deux problèmes fondamentaux subsistent. Le premier consiste à authentifier l'origine d'une route et le second à authentifier le chemin pris par une route. Quelques initiatives ont vu le jour pour répondre à ces problématiques.
La première initiative est sBGP (secure-BGP) et consiste à déployer un système à clé publique où chaque système autonome possède alors un certificat électronique. De plus, les sessions de routage BGP s'établissent via le protocole IPsec. Enfin, lors d'une annonce d'une route, chaque système autonome vérifie le chemin émis et signe à son tour avec sa clé privée le chemin s'il doit l'annoncer à un autre système autonome comme l'illustre la figure suivante (les signatures s'empilent comme les couches d'un oignon) [sBGP] :

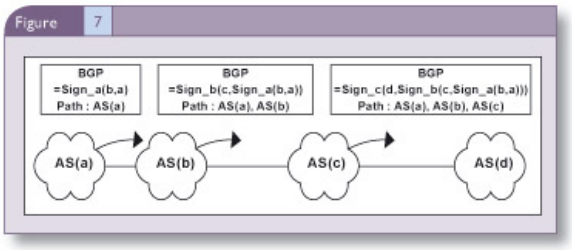

Fig.7

La deuxième initiative exploite le fait que le déploiement d'un système à clé publique ainsi que les impacts cryptographiques sur les processeurs des routeurs limitent une mise en œuvre rapide d'un tel système [WHISPER]. « Listen and Whisper » propose notamment une méthode de contrôle des annonces de routes en limitant au maximum les impacts sur le temps processeur des routeurs. L'idée consiste à fournir un mécanisme permettant de vérifier la consistance des annonces de routes. Par exemple, l'AS(e) reçoit deux annonces de routes par deux chemins différents comme l'illustre la figure suivante :

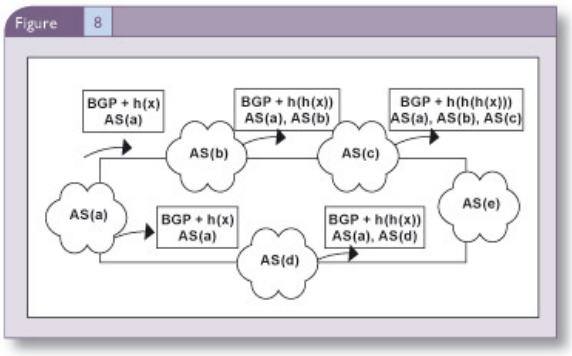

Fig.8

A l'initialisation d'une annonce d'une route, l'AS(a) génère un secret X et utilise alors une fonction de hachage pour rajouter une empreinte à ses annonces de routes. Chaque AS traversé génère une nouvelle empreinte basée sur l'empreinte précédente. Ainsi, si l'AS(e) reçoit deux annonces de routes "r" et "s", de longueurs respectives ~~"k"~~ et ~~"l"~~ (représentant le nombre d'AS traversés, ~~k>l~~) et d'empreintes ~~"yr"~~ et ~~"Ys"~~, il peut alors vérifier la consistance de la route en réalisant le calcul suivant ~~"hk-l(Ys) = yr"~~. Il doit être noté que cette solution, si elle n'impacte que faiblement les temps processeurs des routeurs, ne permet pas d'authentifier de manière sûre l'origine d'une route.
La troisième initiative SoBGP (Secure origin BGP) de CISCO veut répondre aux mêmes besoins de sécurité que la solution sBGP, mais avec une approche différente qui nécessite néanmoins de déployer une nouvelle couche de serveurs pour contrôler les certificats et les chemins associés aux routes [soBGP].
Enfin, l'initiative IRV (Interdomain Routing Validation) consiste à ne pas modifier le protocole BGP et à proposer une architecture de serveurs spécifiques permettant de valider les informations de routage inter-domaine hors-bande [IRV]. Malgré ces différentes initiatives, aucunes de ces solutions ne sont actuellement mises en œuvre.

3. La vérification

des configurations de routage

3.1. Le contrôle de la consistance

des configurations BGP
Sachant que les inconsistances des configurations BGP peuvent engendrer des problèmes de sécurité (isolation, intégrité, etc.), nous considérerons qu'une configuration est consistante si les deux conditions suivantes (ou invariants) sont remplies :

- Tous les éléments de routage définis doivent être référencés.
- Tous les éléments de routage référencés doivent être définis.

Si nous prenons la configuration CISCO ~~conf_test~~ suivante :

```
router bgp 1
 neighbor 1.14.2.2 remote-as 1
 neighbor 1.14.2.2 password 7 xxxxxxxxxxxxxxxxx
 neighbor 2.125.252.53 remote-as 2
 neighbor 2.125.252.53 password 7 xxxxxxxxxxxxxxxxx
 neighbor 2.125.252.53 prefix-list bgp-deny-in in
 neighbor 2.125.252.53 prefix-list bgp-deny-out out
 neighbor 2.125.252.53 route-map bgp-neighbor-2-in in
 neighbor 2.125.252.53 route-map bgp-neighbor-2-out out
!
ip prefix-list bgp-deny-in description ingress filtering peering
ip prefix-list bgp-deny-in seq 15 deny 10.0.0.0/8 le 32
ip prefix-list bgp-deny-in seq 20 deny 172.16.0.0/12 le 32
ip prefix-list bgp-deny-in seq 25 deny 192.168.0.0/16 le 32
ip prefix-list bgp-deny-in seq 999 permit 0.0.0.0/0 le 24
!
ip prefix-list bgp-deny-out description egress filtering peering
ip prefix-list bgp-deny-out seq 15 deny 10.0.0.0/8 le 32
ip prefix-list bgp-deny-out seq 20 deny 172.16.0.0/12 le 32
ip prefix-list bgp-deny-out seq 25 deny 192.168.0.0/16 le 32
ip prefix-list bgp-deny-out seq 999 permit 0.0.0.0/0 le 24
!
route-map bgp-neighbor-2-in deny 10
 match community 1
!
route-map bgp-neighbor-2-in permit 100
 set community x:4
!
route-map bgp-neighbor-2-out deny 10
 match community 1
!
route-map bgp-neighbor-2-out permit 100
 set community x:4
!
ip community-list 1 permit y:1
!
```

Le script ~~bgp_check.sh~~ (http://www.miscmag.com/articles/21-MISC/conf-bgp/) vérifie la consistance d'implémentation de routage. Ce script est un exemple non exhaustif et devra donc être complété. Par ailleurs, il est écrit en langage AWK et s'exécute sur une configuration CISCO. Si on exécute ce script sur la configuration ~~conf_test~~, on obtient alors le résultat suivant (aucune inconsistance n'a été détectée) :

```
bash$ awk -f ./bgp_check.sh ./conf_test
bash$
```

Modifions la configuration afin d'introduire des inconsistances comme l'illustre la commande UNIX diff entre les deux fichiers conf_test et conf_test1 :

```
bash$ diff ./conf_test ./conf_test1
7c7
<  neighbor 2.125.252.53 prefix-list bgp-deny-out out
---
>  neighbor 2.125.252.53 prefix-list bgp-deny-1-out out
25c25
```

```
< route-map bgp-neighbor-2-in deny 10
---
> route-map bgp-neighbor-3-in deny 10
bash$
```

Si on exécute ce script sur la configuration ~~conf_test1~~, on obtient alors le résultat suivant pointant les inconsistances de configuration :

```
bash$ awk -f ./bgp_check.sh ./conf_test1
./conf_test1;déf/non réf;bgp-neighbor-3-in;route-map bgp-neighbor-3-in deny 10;line 23
./conf_test1;déf/non réf;bgp-deny-out;ip prefix-list bgp-deny-out description egress filtering;line 17
./conf_test1;réf/not déf;bgp-deny-1-out; neighbor 2.125.252.53 prefix-list bgp-deny-1-out out;line 7
bash$
```

Enfin, bgp_check.l (http://www.miscmag.com/articles/21-MISC/conf-bgp/) est le même contrôle écrit en FLEX. Il utilise les fonctions de gestion d'arbre de search.h (tfind, tsearch, twalk) pour stocker et faire des recherches sur les éléments de routage. Ce programme est plus performant pour des configurations contenant un nombre important d'éléments de routage. Les options de compilation du programme sont indiquées dans l'en-tête du programme.

3.2. Le contrôle des topologies de routage iBGP et eBGP

Les topologies de routage iBGP et eBGP sont présentes dans les configurations des équipements réseau. Nous pouvons donc extraire ces informations 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 topologie structuré par les champs suivants :

- ~~<router_name>~~ extrait de la commande ~~hostname name~~
- ~~<bgp_as_id>~~ extrait de la commande ~~router bgp autonomous-system~~
- ~~<bgp_ip_address>~~ extrait de la commande ~~neighbor ip-address~~

et le fichier ~~adresse_ip~~ structuré par les champs suivants :

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

Ces informations sont alors utilisées afin de déduire les topologies de routage BGP par une jointure algébrique entre les fichiers ~~topologie~~ et ~~adresse_ip~~. Il doit être noté que la symétrie des sessions de routage est possible lorsqu'il s'agit de sessions de routage internes. Dans le cas de sessions de routage externes, les lignes non résolues par l'opération de jointure signifieront qu'il s'agit de sessions eBGP. Si on considère les données contenues dans les fichiers ~~topologie~~ et ~~adresse_ip~~, on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe pour les sessions eBGP entre les systèmes autonomes :

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

    /* Liste sessions de routage entre les routeurs */
    topologie[bgp_as_id] as a join adresse_ip as b join topologie[bgp_as_id] as c
               on a[bgp_ip_address] = b[ip_address] and
               b[router_name] =c[router_name]
```

```
    where
                a[bgp_as_id] = valeur and c[bgp_as_id] != valeur

 FinFaire
```

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

La vérification de la topologie de routage eBGP consiste à valider que chaque session de routage avec d'autres réseaux est résiliente ou doublée.
Si on considère les données contenues dans les fichiers ~~topologie~~ et ~~adresse_ip~~, on peut déduire par la requête algébrique suivante les sommets et les arcs du graphe pour les sessions iBGP au sein d'un système autonome :

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

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

 FinFaire
```

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

La vérification de la topologie de routage iBGP consiste à valider que le graphe est complet pour le modèle « complet ». Pour le modèle « Réflecteur de route », il s'agit de vérifier que le graphe est connexe et sans point d'articulation. Rappelons que l'extraction de toutes les composantes fortement connexes d'un graphe et le calcul des points d'articulation sont des problèmes faciles [BRASSARD].

3.3. Le contrôle de la politique de routage

Comme nous l'avons détaillé, une politique de routage peut se baser sur différents mécanismes de sécurité. Le contrôle de cette politique dans les configurations des équipements réseau est fondamental afin de s'assurer qu'elle est définie et appliquée.
Si nous définissons la politique de routage suivante :

- Sous-politique de routage eBGP :
  - Un mot de passe doit être défini pour chaque session BGP ;
  - Des filtrages des préfixes reçus et émis doivent être actifs ;
  - Des filtrages des attributs étendus reçus et émis doivent être actifs ;
- Sous-politique de routage iBGP :
  - Un mot de passe doit être défini pour chaque session BGP.

Si nous prenons la configuration CISCO ~~conf_test~~ suivante :

```
router bgp 1
 neighbor 10.10.15.65 remote-as 1
 neighbor 10.10.15.65 password 7 011E57
 neighbor 10.10.15.66 remote-as 1
 neighbor 172.100.61.1 remote-as 2
 neighbor 172.100.61.1 password 7 011E54
 neighbor 172.100.61.1 prefix-list p2-in in
 neighbor 172.100.61.1 prefix-list p2-out out
 neighbor 172.100.61.1 route-map r2-in in
 neighbor 172.100.61.1 route-map r2-out out
 neighbor 172.100.61.2 remote-as 3
 neighbor 172.100.61.2 password 7 011E55
 neighbor 172.100.61.2 prefix-list p2-in in
 neighbor 172.100.61.2 route-map r2-out out
!
```

Le script ~~bgp_contol.sh~~ (http://www.miscmag.com/articles/21-MISC/conf-bgph/) contrôle cette politique

de routage dans les configurations. Ce script est un exemple non exhaustif et devra donc être complété. Par ailleurs, il est écrit en langage AWK et s'exécute sur une configuration CISCO. Si on exécute ce script sur la configuration ~~conf_test~~, on obtient alors le résultat suivant pointant les inconsistances de configuration :

```
bash$ awk -f ./bgp_control.sh ./conf_test
./bgp_conf_test;eBGP;1;3;172.100.61.2;n'a pas de prefix-list out
./bgp_conf_test;eBGP;1;3;172.100.61.2;n'a pas de route-map in
./bgp_conf_test;iBGP;1;10.10.15.66;n'a pas de mot de passe
bash$
```

Enfin, ~~bgp_control.l~~ (http://www.miscmag.com/articles/21-MISC/conf-bgph/) est le même contrôle écrit en FLEX. Il utilise les fonctions de gestion d'arbre de ~~search.h (tfind, tsearch, twalk)~~ pour stocker et faire des recherches sur les éléments de routage. Ce programme est plus performant pour des configurations contenant un nombre important d'éléments de routage. Les options de compilation du programme sont indiquées dans l'en-tête du programme.

4. Conclusion

Les protocoles de routage sont devenus un élément-clé de la disponibilité d'un réseau. Nous avons abordé dans cet article la famille des protocoles de routage EGP, il doit être cependant noté que la famille des protocoles de routage IGP (comme ISIS, OSPF) est tout aussi importante pour assurer la disponibilité d'un réseau. Enfin, les évolutions de services basées sur les protocoles de routage multicast renforcent encore l'importance et l'enjeu stratégique de la sécurité des protocoles de routage.

Références

- [BRASSARD] Brassard (G.), Bratley (P.), Fundamentals of algorithmics, Prentice Hall, ASIN : 0133350681, 1995.
- [IRV] http://www.patrickmcdaniel.org/pubs/ccs03a.pdf
- [RFC1771] Rekhter (Y.), Li (T)., A Border Gateway Protocol 4 (BGP-4), IETF, 1995.
- [RFC1918] Rekhter (Y.), Moskowitz (B.), Karrenberg (D.), de Groot (G.J.), Lear (E.), Address Allocation for Private Internets, IETF, 1996.
- [RFC2796] Bates (T.), Chandra (R.), Chen (E.), BGP Route Reflection – An Alternative to Full Mesh IBGP, IETF, 2000.
- [RFC2385] Heffernan (A.), Protection of BGP Sessions via the TCP MD5 Signature Option, IETF, 1998.
- [RFC3065] Traina (P.), McPherson (D.), Scudder (J.), Autonomous System Confederations for BGP, IETF, 2001.
- [RFC3882] Turk (D.), Configuring BGP to Block Denial-of-Service Attacks, IETF, 2004.
- [ROUTES] http://bgp.potaroo.net
- [sBGP] http://www.net-tech.bbn.com/sbgp/sbgp-index.html
- [soBGP] http://www.nanog.org/mtg-0306/pdf/alvaro.pdf
- [WHISPER] http://www.nanog.org/mtg-0402/pdf/subramanian.pdf

~~Retrouvez cet article dans : **Misc 21**~~

Posté par (La rédaction) | Signature : Cédric Llorens, Fabrice Bruel | Article paru dans 

Laissez une réponse

Vous devez avoir ouvert une session pour écrire un commentaire.

« Précédent Aller au contenu »
Identifiez-vous
Inscription

Articles de 1ère page

- Problématique de consolidation et atteinte des objectifs de niveau de service (SLO) avec Xen
- GNU/Linux Magazine N°113 - Février 2009 - Chez votre marchand de journaux
- Linux Pratique Essentiel N°6 - Février/Mars 2009 - Chez votre marchand de journaux
- Le pavage façon Aqua
- Un peu plus loin avec Linux vserver
- A la découverte du protocole de routage OSPF
- GNU/Linux Magazine HS N°40 - Janvier/Février 2009 - Chez votre marchand de journaux
- Le noyau Linux et Debian
- Yafray, le moteur de rendu photoréaliste libre :maîtriser les shaders et les propriétés matériau
- FUSE, développez vos systèmes de fichiers dans l'espace utilisateur

Catégories

- Administration réseau
- Administration système
- Agenda-Interview
- Audio-vidéo

- Bureautique
- Comprendre
- Distribution
- Embarqué
- Environnement de bureau
- Graphisme
- Jeux
- Matériel
- News
- Programmation
- Réfléchir
- Sécurité
- Utilitaires
- Web

Articles secondaires

- 30/10/2008
  Google Gears : les services de Google offline

  Lancé à l'occasion du Google Developer Day 2007 (le 31 mai dernier), Google Gears est une extension open source pour Firefox et Internet Explorer permettant de continuer à accéder à des services et applications Google, même si l'on est déconnecté....

  Voir l'article...
  7/8/2008
  Trois questions à...

  Alexis Nikichine, développeur chez IDM, la société qui a conçu l'interface et le moteur de recherche de l'EHM....

  Voir l'article...
  11/7/2008
  Protéger une page avec un mot de passe

  En général, le problème n'est pas de protéger une page, mais de protéger le répertoire qui la contient. Avec Apache, vous pouvez mettre un fichier ~~.htaccess~~ dans le répertoire à protéger....

  Voir l'article...
  6/7/2008
  hypermail : Conversion mbox vers HTML

  Comment conserver tous vos échanges de mails, ou du moins, tous vos mails reçus depuis des années ? mbox, maildir, texte... les formats ne manquent pas. ...

  Voir l'article...
  6/7/2008
  iozone3 : Benchmark de disque

  En fonction de l'utilisation de votre système, et dans bien des cas, les performances des disques et des systèmes de fichiers sont très importantes....

  Voir l'article...
  1/7/2008
  Augmentez le trafic sur votre blog !

  Google Blog Search (http://blogsearch.google.fr/) est un moteur de recherche consacré aux blogs, l'un des nombreux services proposés par la célèbre firme californienne....

Voir l'article...

- **GNU/Linux Magazine**

  - GNU/Linux Magazine N°113 - Février 2009 - Chez votre marchand de journaux
  - Édito : GNU/Linux Magazine 113
  - Un petit sondage pour améliorer nos magazines
  - GNU/Linux Magazine HS N°40 - Janvier/Février 2009 - Chez votre marchand de journaux
  - Edito : GNU/Linux Magazine HS 40

- **GNU/Linux Pratique**

  - Linux Pratique Essentiel N°6 - Février/Mars 2009 - Chez votre marchand de journaux
  - Édito : Linux Pratique Essentiel N°6
  - Un petit sondage pour améliorer nos magazines
  - Linux Pratique N°51 - Janvier/Février 2009 - Chez votre marchand de journaux
  - Édito : Linux Pratique N°51

- **MISC Magazine**

  - Un petit sondage pour améliorer nos magazines
  - MISC N°41 : La cybercriminalité ...ou quand le net se met au crime organisé - Janvier/Février 2009 - Chez votre marchand de journaux
  - Édito : Misc 41
  - MISC 41 - Communiqué de presse
  - Les Éditions Diamond adhèrent à l'APRIL !