Le protocole spanning-tree

Cédric Baillet

Les réseaux LAN sont aujourd'hui considérés comme des environnements relativement stables, connus et bien sécurisés. Cependant c'est une allégation bien légère et très dangereuse, la corruption d'un réseau permet de remonter dans les couches applicatives et d'accéder à de nombreuses données.

Heureusement, les protocoles couramment usités sur les LAN possèdent la plupart du temps des fonctionnalités permettant de les sécuriser plus fortement qu'en implémentant la simple configuration par défaut.

Un réseau sécurisé sera donc composé de périphériques totalement dédiés à la sécurité comme les firewalls ou les IPS (système de prévention d'intrusion), mais aura aussi été pensé pour mettre en place tous les éléments de sécurité des différents protocoles implémentés. C'est à cette seule exigence que des conditions de sécurité optimales seront réunies.

La création d'une entreprise et de son réseau à partir du néant est de plus en plus rare. La plupart du temps, les administrateurs réseaux doivent gérer un existant plus ou moins obsolète ou encore la réunion de différentes infrastructures nées au gré de la vie de l'entreprise. C'est dans ce genre de situation que la notion d'audit devient intéressante pour répondre à cette angoissante question : l'ensemble de mon infrastructure réseau est-il bien sécurisé ?

Notez bien que cet article ne vous permettra pas de réaliser un audit réseau complet. Sa vocation est seulement de mettre en évidence la nécessité d'utiliser toutes les fonctions destinées à sécuriser les protocoles réseau au travers d'un exemple : le spanning-tree. Il sera nécessaire de démultiplier cette démarche sur l'ensemble des protocoles utilisés au sein d'un réseau pour commencer à avoir une procédure d'audit techniquement exhaustive.

Les réseaux LAN ont aujourd'hui montré leur valeur et sont couramment utilisés touchant à la

Ce qu'il faut savoir...

Idéalement, le lecteur connaîtra les différents types de périphériques réseaux, les bases des protocoles qu'ils utilisent ainsi que les grands concepts de l'IP.

Cet article explique...

- Les grandes lignes du fonctionnement du protocole spanning-tree.
- Les faiblesses potentiellement exploitables par un attaquant.
- Les contre-mesures pouvant être mises en place.

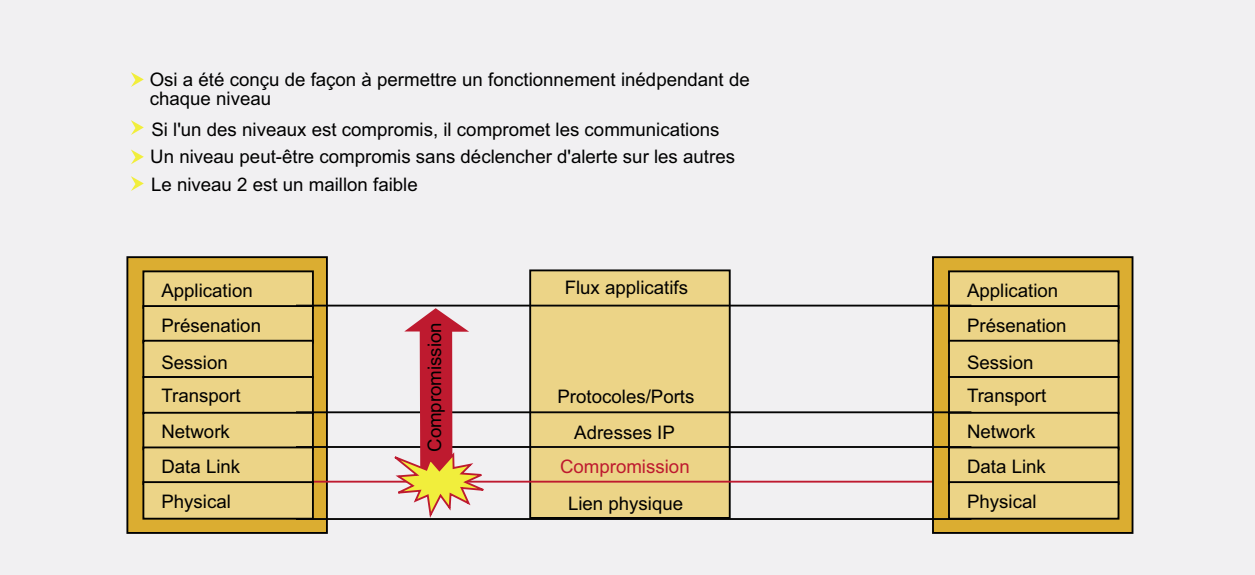

Figure 1. *Introduction*

fois les particuliers et les entreprises. Face à une utilisation de plus en plus répandue, il est donc nécessaire de se demander s'ils sont suffisamment sécurisés pour ne pas représenter un risque pour l'ensemble des données transitant sur leurs infrastructures. Question épineuse à laquelle les audits LAN sont censés répondre.

Cet article tentera donc d'apporter des éléments de réponse au travers de l'analyse de l'un des protocoles les plus couramment utilisés dans ce type d'environnement : le *spanning-tree*.

Le protocole spanning-tree

Un commutateur est un pont multiport (un pont est un dispositif matériel permettant de relier des réseaux travaillant avec le même protocole. Ainsi, contrairement au répéteur, qui travaille au niveau physique, le pont travaille également au niveau logique), c'est-à-dire qu'il s'agit d'un élément actif agissant au niveau 2 du modèle OSI.

Le commutateur analyse les trames arrivant sur ses ports d'entrée et filtre les données afin de les aiguiller uniquement sur les ports adéquats (on parle de commutation ou de réseaux commutés – le principe est illustré sur la Figure 2). Le commutateur allie donc les propriétés du pont en matière de filtrage et du concentrateur (*hub*) en matière de connectivité.

Connaissant le port du destinataire, le commutateur ne transmettra le message que sur le port adéquat, les autres ports restants dès lors libres pour d'autres transmissions susceptibles de se produire simultanément. Il en résulte que chaque échange peut s'effectuer à débit nominal (plus de partage de la bande passante), sans collision, avec pour conséquence une augmentation très sensible de la bande passante du réseau (à vitesse nominale égale).

Le protocole spanning-tree (ou STP pour spanning-tree protocol) a été créé pour prévenir la formation de boucles sur les réseaux de niveau 2 lors des interconnectxions multiples mises en place entre commutateurs pour assurer la redondance du réseau (Figure 3).

Ceci est réalisé en rendant les commutateurs *conscients* de l'existence de leurs voisins et de l'état de la bande passante sur les liens communs. Chaque commutateur sélectionne alors un seul parcours, sans boucle et avec une bande passante disponible maximale par rapport à un point de référence commun à l'ensemble des membres du domaine de niveau 2 (sera appelé domaine de niveau 2 l'ensemble des commutateurs travaillant sur un même subnet

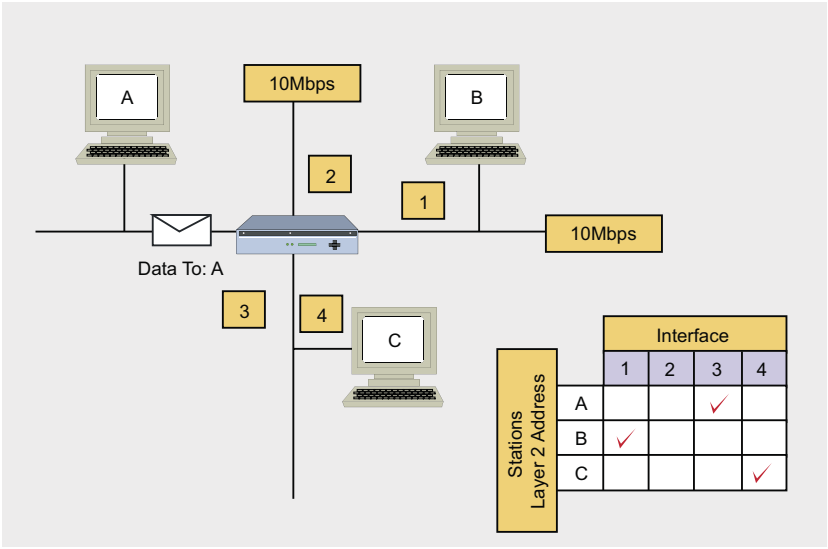

Figure 2. *Apprentissage du commutateur*

IP que ce soit au niveau physique ou logique).

Si le phénomène de boucle n'était pas maîtrisé au travers de ce protocole, le réseau pourrait être victime du phénomène appelé tempête de broadcast (Figure 4). Celui-ci se produit lorsque des trames de type broadcast (à destination de l'ensemble des éléments actifs du réseau) sont émises. Les commutateurs les renvoient alors sur l'ensemble des ports, les trames continuant ainsi à circuler en boucle et à se multiplier jusqu'à expiration du TTL (*Time To Live* – temps de vie maximum autorisé pour une trame sur le réseau).

Le point de référence permettant la mise en place d'itinéraires uniques et sans boucle sur le réseau se nomme le *root bridge*.

Ce dernier est choisi parmi les différents commutateurs au travers d'un processus d'élection qui permet d'arriver à un résultat unique. U ne fois le *root bridge* désigné, chaque commutateur définit un port *root* représentant le chemin à emprunter pour accéder à ce dernier. L'ensemble des parcours possibles pour atteindre chaque point du réseau est défini et peut être représenté comme un arbre de possibilités ayant le *root bridge* pour origine.

La fin de ce processus, le STP est construit et le domaine de niveau 2 dans lequel se trouve notre commutateur ne doit posséder aucune boucle.

Les ports des commutateurs (ceux reliant les différents commutateurs entre eux) ne participant pas au processus STP sont bloqués. Un port dans cet état particulier n'émet ni ne reçoit de données. La seule action autorisée est l'écoute des BPDU (STP *Bridge Protocol Data Unit* – trame permettant de gérer les processus STP sur le réseau). C'est la présence de cette fonction particulière qui permet d'élimer les boucles sur le domaine de niveau 2. Lors d'un recalcul de l'arbre du STP, il est possible que les ports bloqués changent d'état pour prendre en compte une modification de l'architecture du réseau. Les ports passeront alors dans les états listening, learning, pour enfin atteindre l'état forwarding permettant au trafic de données de s'établir. Ce processus prend de 30 à 50 secondes par défaut (une norme plus récente appelée RSTP, pour *Rapid Spanning-Tree Protocol*, permet une convergence plus rapide pour les réseaux qui la supportent).

Maintenant que nous comprenons les grands processus utilisés par le protocole spanning-tree, il sera plus simple de définir les différents axes d'attaques possibles. Tous mettent en jeu les BPDU évoquées précédemment pour usurper une fonction (au hasard le *root bridge*) ou provoquer un déni de service (calcul récurrent de l'arbre du STP par exemple).

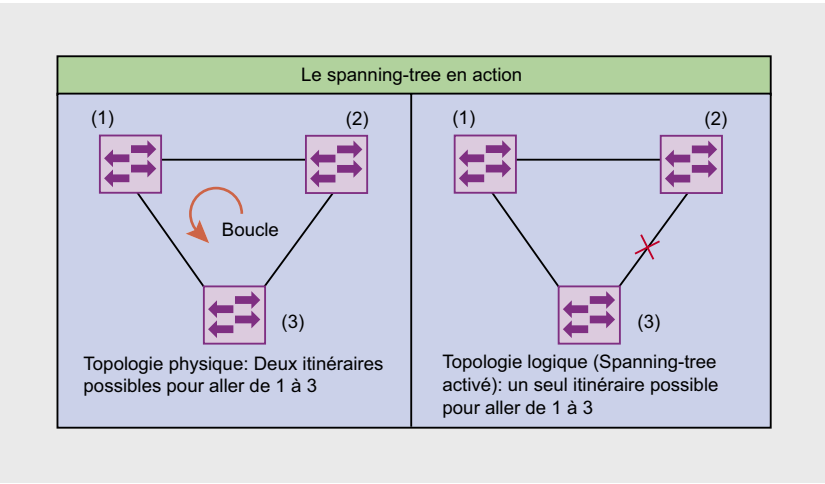

Figure 3. *L'action du spanning-tree sur un réseau Ethernet*

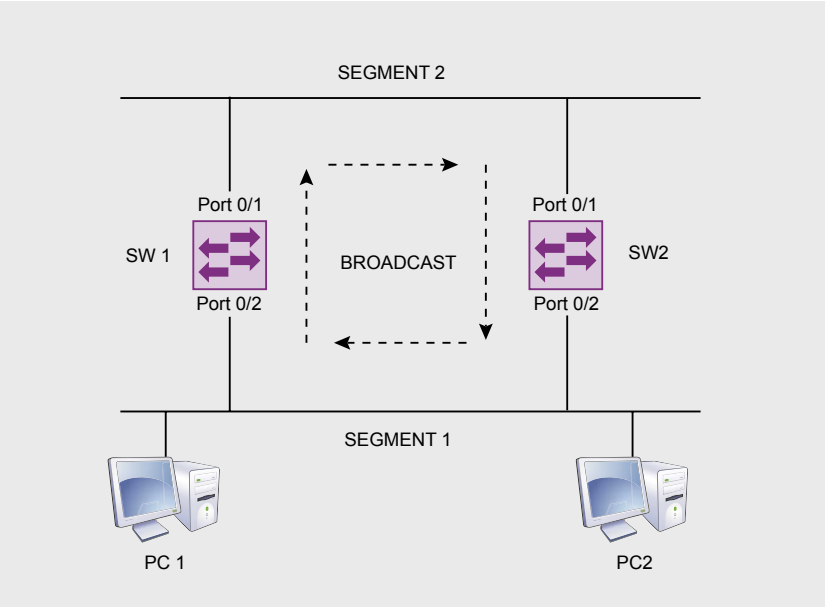

Figure 4. *Tempête de broadcast*

Récupération du rôle root bridge par un attaquant

Ici nous traiterons de la théorie, de la pratique ainsi que les contre-mesures possibles.

Récupération du rôle de root bridge : la théorie

Le *root bridge* est sélectionné en comparant l'identifiant du commutateur contenu dans un champ des trames BPDU. La valeur par défaut de cet identifiant pour les commutateurs Cisco Catalyst est de 32768

et est donc utilisée comme telle lors de l'élection du *root bridge* si l'administrateur réseau ne la modifie pas. Lorsque deux commutateurs ont un identifiant avec la même valeur, leurs adresses MAC sont alors utilisées pour les départager, l'adresse la plus faible remportant l'élection.

Pour récupérer le rôle du *root bridge*, un attaquant devra donc envoyer des BPDU forgées avec une priorité plus faible que celle du *root bridge* actuel (Figure 5). Si le *root bridge* légitime a une priorité à zéro, il sera alors nécessaire de modifier l'axe d'attaque en travaillant en plus avec les adresses MAC (pour rappel, le but sera de forger une adresse MAC plus basse que celle du *root* légitime).

Récupération du rôle de root bridge : la pratique

Comme évoqués en début d'article, nous allons utiliser le logiciel Yersinia pour mettre en œuvre cette attaque. Vous verrez que ce dernier est d'une simplicité extrême et permet de mettre en pratique toute la théorie en trois clics.

Yersinia possède désormais une interface graphique (Figure 6), encore instable, c'est vrai, mais simplifiant énormément les choses.

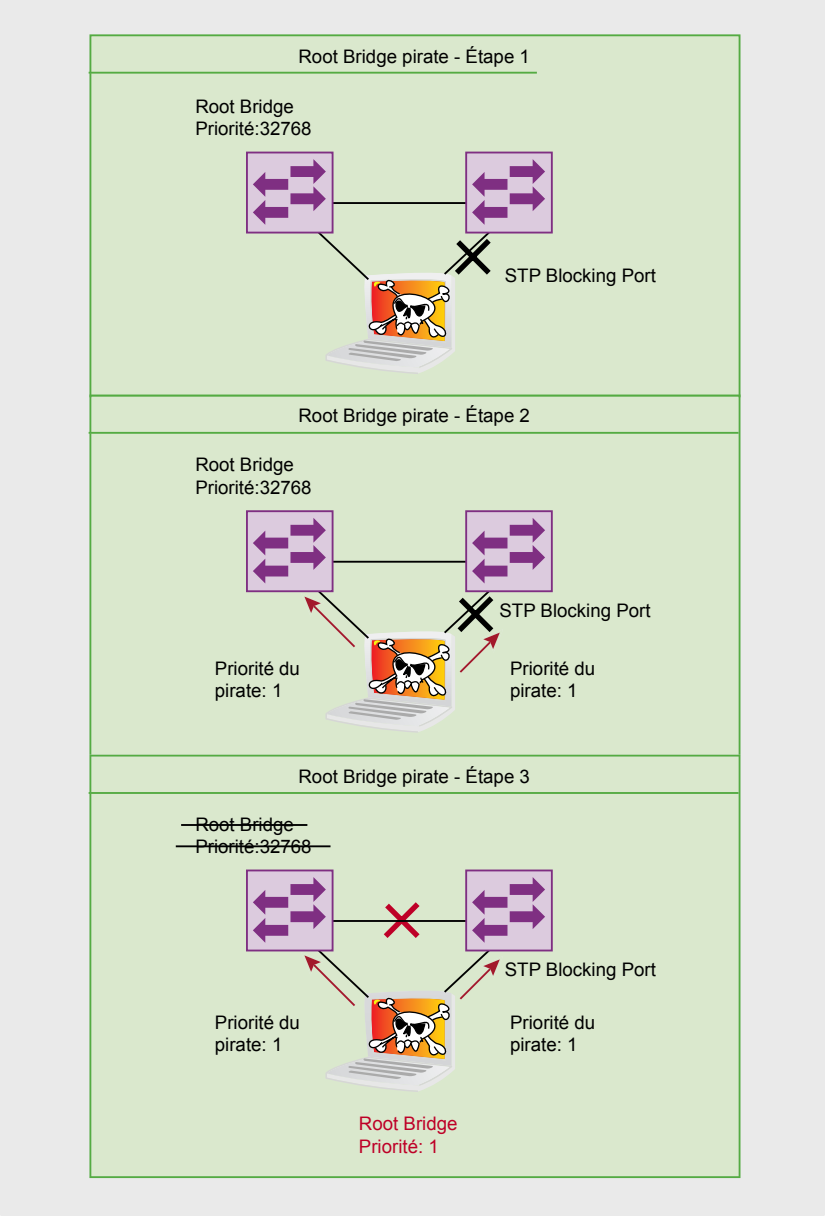

Figure 5. *Récupération du rôle root bridge par le pirate*

La partie gauche permet de voir les paquets reçus et envoyés par protocole, la partie droite nous donne le détail et permet de modifier certains paramètres pour lancer des attaques. Il est possible de modifier les principaux champs du protocole spanning-tree pour mettre en œuvre l'attaque décrite précédemment dans la partie inférieure droite. Le réseau utilisé pour la démonstration ayant été configuré avec les paramètres spanning-tree par défaut, nous n'aurons pas besoin de modifier les valeurs des différents champs. Il suffira de laisser Yersinia agir.

Pour lancer notre attaque, il suffit de cliquer sur le bouton *launch attack* dans la barre supérieure et la fenêtre présentée dans la Figure 7.

Listant toutes les attaques possibles pour le protocole spanning-tree apparaîtra. La dernière étape consiste simplement à sélectionner l'option *claiming root role* et à valider.

La Figure 8 provient du commutateur utilisé pour la démonstration. Comme vous le constatez, le rôle du *root bridge* est désormais attribué au port de notre PC pirate. L'attaque a donc réussi. C'était extrêmement simple n'est-ce pas ?

Récupération du root bridge : quel intérêt ?

Une fois que l'attaquant a réussi à récupérer le rôle de *root bridge*, deux options s'offrent à lui :

- La possibilité de voir passer beaucoup plus de trafic, ce qui permettra de récupérer des informations.
- La possibilité de créer un déni de service sur le réseau.

La première option est sans doute la plus intéressante puisqu'elle permet de récupérer de l'information pour continuer à pénétrer le réseau plus profondément.

Récupération du root bridge : contre-mesure

Comme nous l'avons vu précédemment, le spanning-tree assigne

différents rôles aux ports d'un commutateur. Le *root bridge* est supposé être atteint via le port *root*, ce dernier étant supposé représenter la voie la plus efficace.

Supposons que nous rajoutions un nouveau commutateur avec une priorité plus basse (donc plus intéressante dans le cadre de l'élection du *root bridge*). Ce nouveau commutateur a toutes les chances de devenir *root bridge* au terme de l'élection et la topologie du spanning-tree de reconverger sous une forme différente. Notez bien que jusqu'ici, tout ceci respecte parfaitement le processus normal. Cependant, l'administrateur du réseau ne souhaite pas forcément que ce cas de figure se réalise, la nouvelle topologie créée dynamiquement par le processus spanning-tree pouvant être totalement en inadéquation avec l'architecture du réseau (l'élection d'un commutateur avec peu de capacité et en bout de chaîne par exemple).

La fonctionnalité *root guard* a été développée comme un moyen de contrôler les candidats au rôle de *root bridge* connectés sur le réseau.

Si un autre commutateur émet une BPDU supérieure ou un meilleur bridge ID sur un port avec la fonctionnalité *root guard active*, le commutateur local refusera le nouvel arrivant en tant que *root*. De plus, tant que ces BPDU seront reçues sur ce port, celui-ci sera déclaré comme étant dans un état inconsistant pour le process spanning-tree. Aucune donnée ne sera envoyée ou reçue, seule sera permise l'écoute des BPDU arrivant sur ce port.

Nous pourrions dire que la fonctionnalité *root guard force* un port à se transformer en relais pour les BPDU mais ne lui permet pas de les recevoir directement. Cette fonctionnalité doit être configurée sur tous les ports qui ne sont pas destinés à devenir *root*.

La fonctionnalité *root guard* n'est pas activée par défaut. Elle doit être configurée sur chaque interface à l'aide de la commande ci-dessous (en environnement Cisco IOS).

```
Switch(config-if)# spanning-tree
      guard root
```

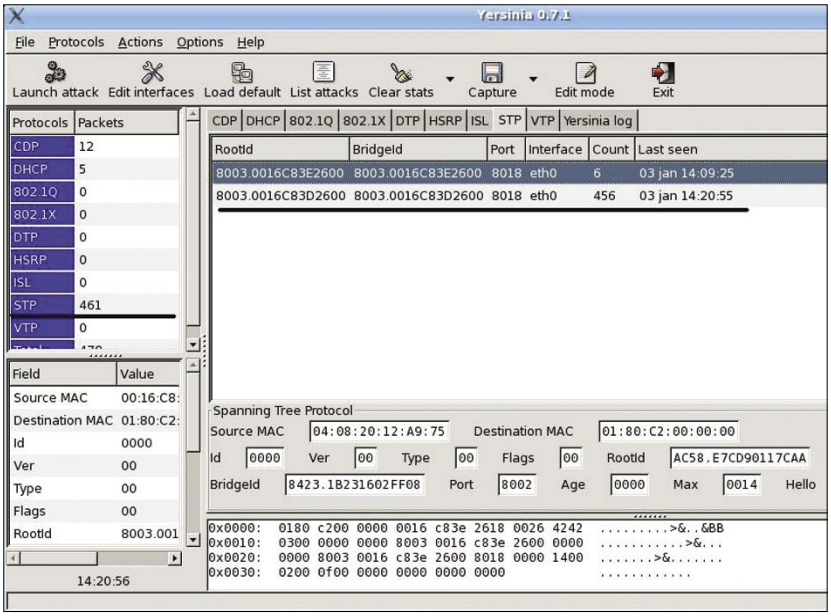

Figure 6. *Le logiciel Yersinia*

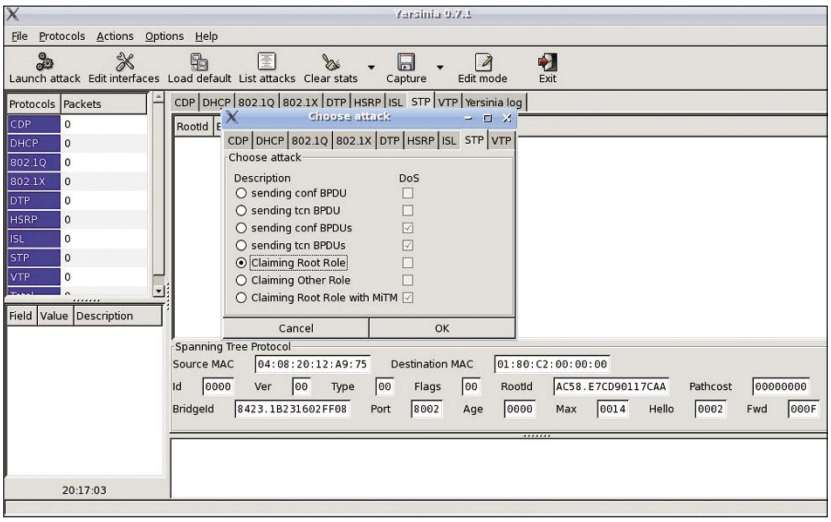

Figure 7. *Les attaques disponibles contre le protocole spanning-tree*

Réalisation d'un déni de service en utilisant le spanning-tree

Nous allons réaliser à l'aide de spanning-tree un déni de service, sans omettre les éléments théoriques et pratiques.

Réalisation d'un déni de service : la théorie

La table CAM d'un commutateur lui permet de garder en mémoire l'ensemble des adresses MAC des machines clientes et ainsi que les ports physiques correspondants. Il s'agit donc d'un élément essentiel pour le bon fonctionnement du système.

Lors de la réinitialisation du processus spanning-tree (nouveau calcul de l'arbre suite à une modification dans le réseau), la table CAM est réinitialisée pendant les quinze premières secondes.

Le temps de validité d'une entrée dans la table CAM d'un commutateur étant de 300 secondes par défaut, la mise à jour du spanning-tree réduit la durée de vie d'une entrée d'un coefficient estimé à vingt. Pendant cette période, le commutateur quitte le mode commuté normal et se met à reproduire le trafic réseau sur tous

ses ports jusqu'à ce que la table MAC soit à nouveau à jour.

Ce type de fonctionnement nous offre plusieurs approches pour attaquer le réseau via le *spanning-tree* qui pourraient être résumées ainsi : si un calcul récurent du spanning-tree peut être provoqué, il devrait être possible de récupérer une partie du trafic réseau transitant par ce commutateur (réplication du trafic sur tous les ports pendant les quinze premières secondes) et provoquer ultimement un déni de service sur le réseau (Figure 10).

Bien qu'une attaque par déni de service ne semble pas aussi intéressante qu'une redirection et/ou une modification du trafic, un déni de service fondé sur le spanning-tree peut rendre inutilisable un réseau de grande taille et sera difficilement identifiable. De plus, le but de l'attaquant est probablement de segmenter le réseau et non de provoquer sa chute complète.

Ainsi, un attaquant cherchant à contourner un IDS/IPS ou un pare-feu pour atteindre une partie du réseau contenant des données sensibles pourra utiliser une technique comme celle-ci.

S'il s'agit d'un IDS/IPS à part entière connecté sur le réseau, cette attaque ne sera pas toujours couronnée de succès, par contre, s'il s'agit d'une carte placée dans le commutateur censé être le *root bridge*, isoler ce dernier permettra d'éliminer la protection mise en place par l'administrateur et d'accéder à une nouvelle partie du réseau.

Un deuxième exemple serait celui d' En revanche, s'il s'agit d'une carte placée dans le commutateur censé être le *root bridge*, isoler ce dernier permettra d'éliminer la protection mise en place par l'administrateur et d'accéder à une nouvelle partie du réseau. Un deuxième exemple serait celui d'un attaquant isolant des machines clientes des serveurs pour usurper leurs identités et avoir ainsi accès à des informations.

Réalisation d'un déni de service : la pratique

Comme nous venons de le voir, le déni de service pouvant être provoqué via le *spanning-tree* ne doit pas être sous-estimé et devrait toujours être pris en compte lors du design et de la configuration d'un réseau commuté, d'autant plus qu'un nombre important de variations sur ce thème sont envisageables. L'une des approches possibles est de provoquer l'élection et la disparition du *root bridge* de façon perpétuelle.

Une attaque comme celle-ci se réalise de deux façons :

- Cas 1, l'attaquant envoie sur le réseau des BPDU pour récupérer le rôle du *root* en décrémentant la

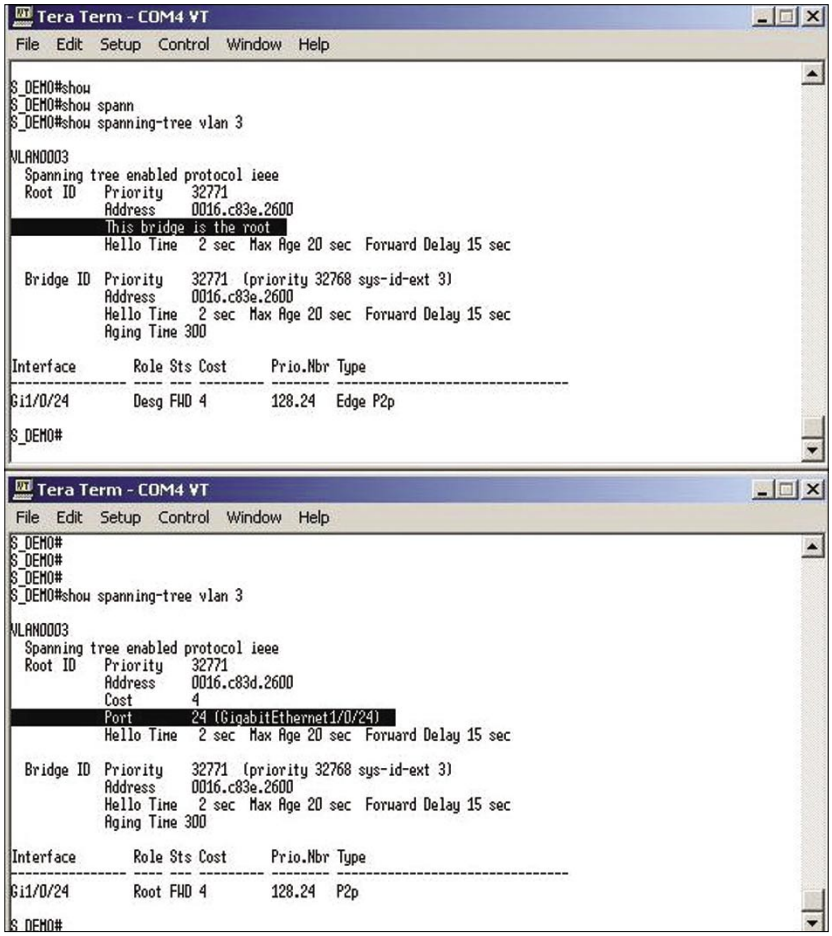

Figure 8. *Root bridge du VLAN 3 Avant/Après*

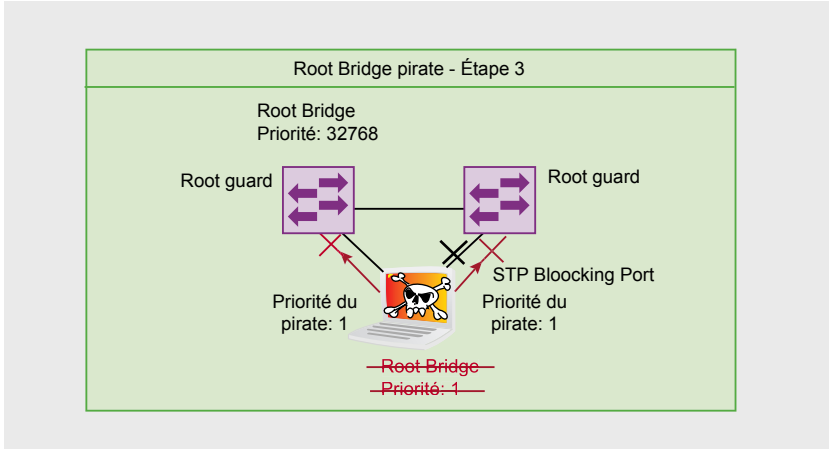

Figure 9. *Mise en échec du pirate grâce à la fonction root guard*

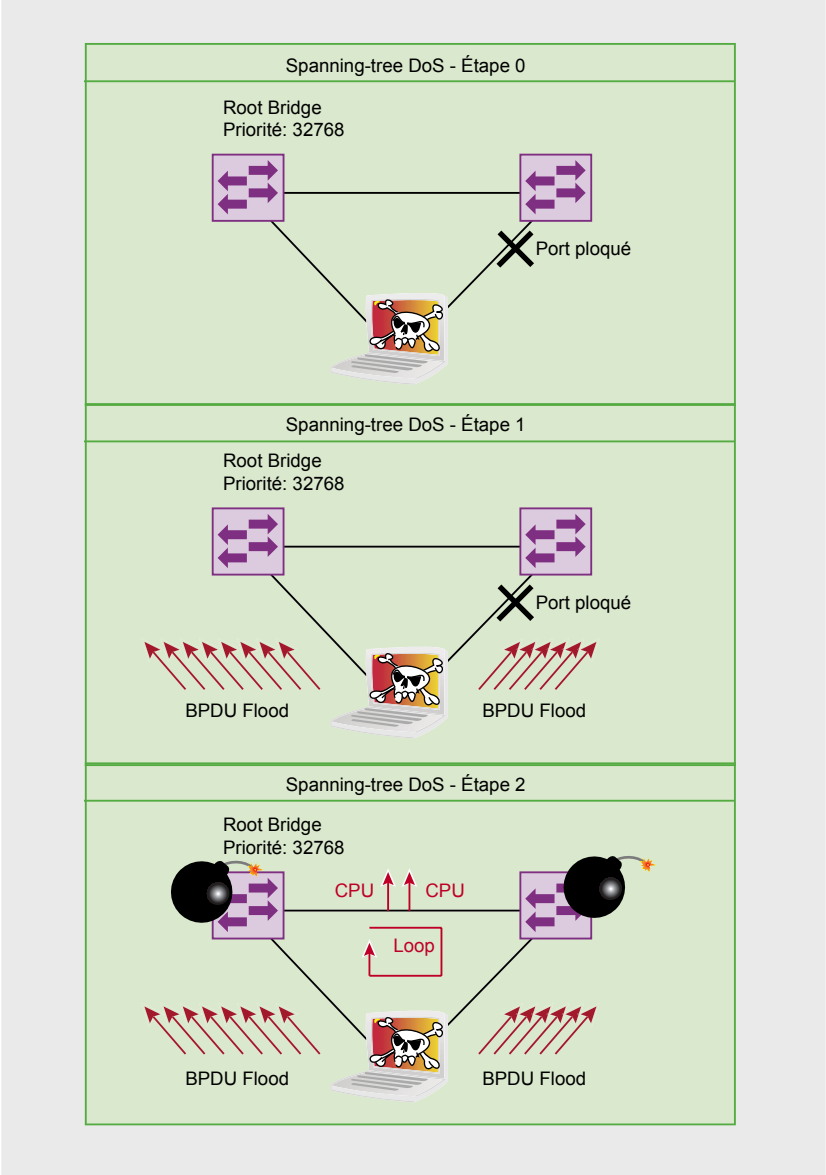

Figure 10. *Réalisation d'un DoS via le spanning-tree*

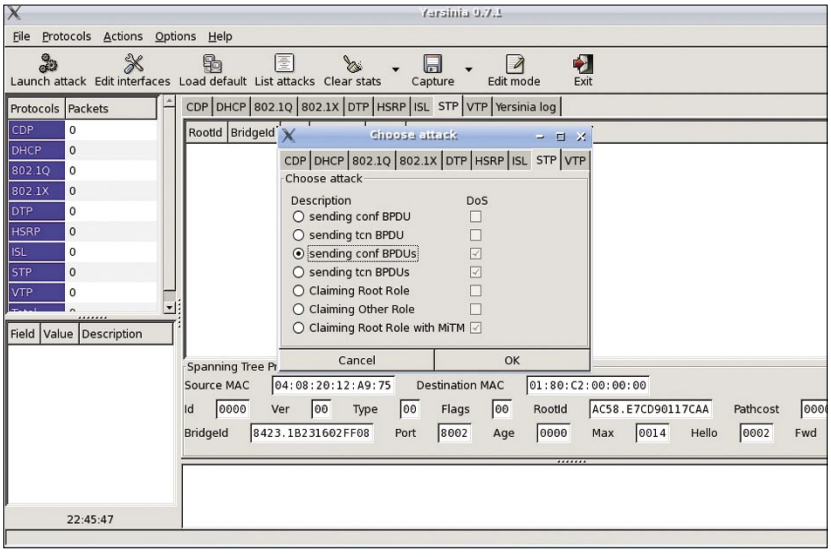

Figure 11. *Lancement d'un DoS sur le spanning-tree à l'aide de Yersinia*

priorité à chaque fois. Un recalcul du processus spanning-tree est lancé à chaque nouvelle séquence de BPDU provoquant ainsi le déni de service recherché.

- Cas 2, l'attaquant envoie sur le réseau des BPDU pour récupérer le rôle du *root* et fixe l'âge maximum à son minimum. L'élection devra alors recommencer quasiment immédiatement. Il suffira donc de faire tourner ce processus en boucle pour obtenir le déni de service recherché.

Une fois de plus, Yersinia nous permettra de réaliser cette attaque très simplement. Comme précédemment, il faut choisir l'onglet STP et cliquer sur le bouton *Launch attack* présent dans la barre supérieure. Nous obtenons alors la fenêtre listant toutes les attaques disponibles pour le protocole *spanning-tree* présenté dans la Figure 11. Il suffit alors de sélectionner l'option *sending conf BPDUs* et de valider.

Hormis constater le dysfonctionnement du réseau, il est possible de voir si l'attaque est effective en regardant le taux d'utilisation CPU du processus *spanning-tree* sur le commutateur avant et après avoir lancé l'attaque. C'est ce qui est présenté dans la Figure 12.

Réalisation d'un déni de service : contre-mesure

La fonctionnalité *BPDU Guard* a été développée pour protéger au mieux l'intégrité du protocole spanning-tree (Figure 13). Elle se configure directement sur les interfaces des commutateurs destinées à accueillir des clients (PC, serveurs, etc.)

Si des BPDU sont détectées sur des ports avec cette fonctionnalité activée, ces derniers passeront immédiatement en mode *errdisable*, ce qui correspond à une fermeture du port et une mise en mode erreur. Les ports ainsi fermés, pourront être à nouveau ouverts soit par une action directe de l'administrateur, soit après la fin du compteur de protection initié lors du passage en mode *errdisable*. La fonctionnalité BPDU

guard est facilement configurable par port et désactivée par défaut. La configuration demande donc une action volontaire de l'administrateur.

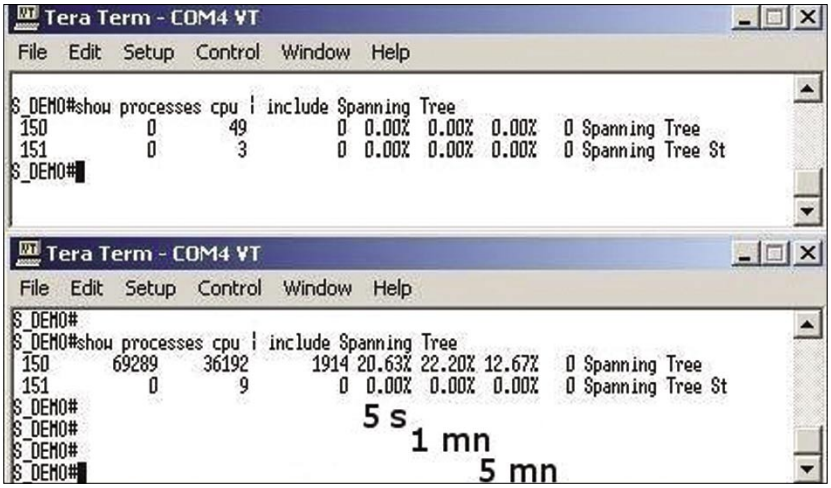

Figure 12. *Taux d'utilisation CPU du processus spanning-tree Avant/Après*

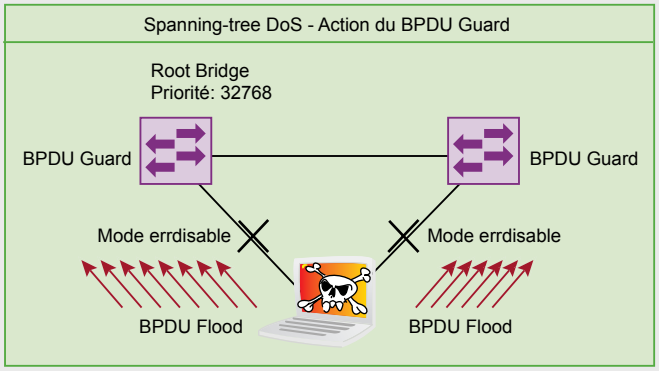

Figure 13. *Action du BPDU Guard*

La mise en place du *BPDU Guard* se fera sur les interfaces des commutateurs via la commande suivante :

```
Switch(config-if)# spanning-tree
                        bpduguard enable
```

Initialement, le BPDU Guard était prévu pour empêcher la mise en place d'un commutateur sur un port prévu pour un utilisateur lambda, que cela soit volontaire ou non.

L'action menée lors de l'attaque décrite précédemment correspond à ce cas de figure puisque l'attaquant se fait passer pour un commutateur émettant des BPDU. La contre-mesure est donc efficace.

Conclusion

Vous venez de voir comment mettre en évidence des défauts de configuration susceptibles de mener à des attaques dévastatrices pour des LAN sur le protocole *spanning-tree*.

Les contre-mesures présentées offrent une solution aux questions traitées, mais de nombreuses autres options permettant de renforcer ce protocole existent. Il est donc nécessaire de continuer à enquêter plus avant pour réussir à obtenir un réseau de niveau 2 fiable et solide.

Quelques pistes de recherches complémentaires dans un environnement Cisco seraient :

- *BPDU Loop Guard*
- *BPDU Skew Detection*
- *UDLD*

Enfin, n'oubliez pas que si la technique est importante dans un audit, cela n'est pas suffisant.

Il faudra par ailleurs être capable d'analyser correctement l'architecture globale du réseau, les différentes procédures d'exploitation ou encore la politique de sécurité de l'entreprise.

C'est seulement en prenant en compte les paramètres tant organisationnels que techniques que nous pourrons correctement sécuriser un LAN. ●

À propos de l'auteur

Après avoir travaillé pendant quatre années sur les technologies réseaux Cisco en tant qu'ingénieur de production, Cedric Baillet a été consultant sur les solutions de ToIP et les problématiques sécurité afférentes de 2004 à 2007. Il a aujourd'hui intégré une des équipes marketing d'Orange Business Services pour travailler sur les offres de services sécurité autour des nouvelles solutions de communications. L'auteur peut être contacté à l'adresse mail suivante : *cedric_baillet@yahoo.fr.*

Sur Internet

- *http://www.yersinia.net/* – site web de l'outil Yersinia.
- *http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cwsi2/cwsiug2/vlan2/stpapp.htm* – Une introduction au spanning-tree du constructeur Cisco (en anglais).
- *http://www.labo-cisco.com/fr/articles/theorie/spanning-tree-protocol.html* – Une introduction au protocole *spanning-tree* réalisée par les étudiants de *Sup info* (francais).