Modélisation, conception et implémentation d'une porte dérobée à canaux cachés ICMP

Yann Tourdot
yann@tourdot.fr

15 Février 2010

Résumé

Dans cet article nous présentons la modélisation et la conception d'une porte dérobée permettant le contrôle de machines à distance. Des contraintes sont définies afin que la porte dérobée soit la plus efficace possible : utilisation de canaux cachés et des flux a priori inoffensifs (ICMP [1]) pour limiter la détection, injection « à la main » possible sur la machine cible, aucun port ouvert sur la machine cible, non-nécessité de privilèges pour l'implantation ou l'exécution de la porte dérobée sur la machine cible... Une implémentation de la porte dérobée modélisée dans cet article est disponible [2].

Mots-clé : sécurité, porte dérobée, backdoor, canaux cachés, ICMP.

1 Introduction

Une porté dérobée (ou *backdoor* en anglais) permet à un attaquant prendre le contrôle d'une machine afin d'y réaliser des opérations non autorisées (récupération d'informations, pénétration en profondeur du réseau, déni de service…) à l'insu du propriétaire légitime de cette dernière.

Face à la difficulté pour un attaquant de pénétrer un système informatique, il est souvent préférable pour lui de laisser sur une ou plusieurs machines du réseau qu'il a pénétré un programme qui lui permettra d'accéder de nouveau au système sans avoir à réitérer son exploit (correction de la vulnérabilité entre temps, détection de l'attaque…).

Pour être efficace, une porte dérobée doit être discrète pour une raison simple : si les propriétaires légitimes de la machine piégée soupçonnent l'existence de la porte dérobée, ils vont mener des investigations (analyse des programmes exécutés, analyse des flux émis par la machine piégée, vérification d'intégrité…) et l'attaquant perdra ainsi tout le bénéfice de sa porte dérobée. La porte dérobée doit donc être suffisamment discrète pour ne pas éveiller de soupçons de la part des propriétaires légitimes de la machine piégée.

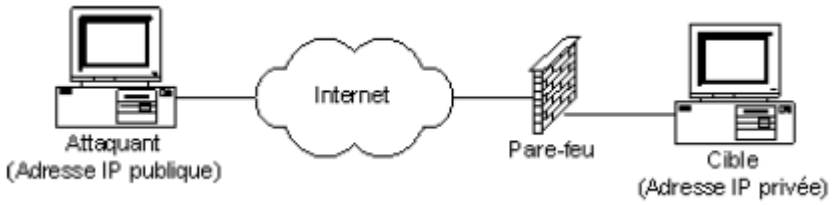

Figure 1 : Utilisation traditionnelle d'une porte dérobée permettant le contrôle d'une machine cible à distance.

2 Définitions

2.1 Définition des objets

M_c	La machine cible.
P_c	Le programme exécuté sur M_c.
M_a	La machine de l'attaquant.
P_c	Le programme serveur exécuté sur la machine M_a.
$F_{a \to c}$	Les flux émis de la machine M_a vers la machine M_c.
$F_{c \to a}$	Les flux émis de la machine M_c vers la machine M_a.

2.2 Définition des contraintes

Les contraintes sur les objets sont les suivantes :

$C(P_c)_1$	P_c doit pouvoir être exécuté sur M_c quels que soient son système d'exploitation et son architecture.
$C(P_c)_2$	P_c ne doit ouvrir aucun port TCP ou UDP en écoute sur M_c.
$C(P_c)_3$	P_c doit pouvoir être exécuté sur M_c sans aucun droit d'administration.
$C(P_c)_4$	P_c doit ne doit utiliser aucun programme tiers [1] installé sur M_c.
$C(F_{a \to c})_1$	Les flux $F_{a \to c}$ doivent être possibles.
$C(F_{a \to c})_2$	Les flux $F_{a \to c}$ doivent être transportés via un protocole standard et a priori inoffensif.

1. Le seul programme tiers utilisé par la porte dérobée est la commande native PING de la machine cible M_c.

$C(F_{c\to a})_1$	Les flux $F_{c\to a}$ doivent être possibles.
$C(F_{c\to a})_2$	Les flux $F_{c\to a}$ doivent être transportés via un protocole standard et a priori inoffensif.
$C(F_{c\to a})_3$	Les flux $F_{a\to c}$ ne doivent pas être de type CONNECT afin traverser les pare-feux protégeant M_c.
$C(F)$	Les fréquences des flux $F_{a\to c}$ et $F_{c\to a}$ doivent être minimales afin de ne pas éveiller les soupçons.

Aucune contrainte n'est imposée sur P_a et M_a car on suppose que l'attaquant dispose d'un contrôle total de sa machine M_a.

La solution retenue afin de répondre aux contraintes identifiées ci-dessus est :

1. La machine cible M_c et la machine de l'attaquant M_a communiquent via le protocole ICMP.
2. La machine cible M_c envoie les données à la machine de l'attaquant M_a via les messages ICMP_ECHO_REQUEST du protocole ICMP.
3. La machine de l'attaquant M_a envoie les données à la machine cible M_c via les messages ICMP_ECHO_REPLY du protocole ICMP.
4. L'identification de la machine cible M_c auprès de la machine de l'attaquant M_a se fait via la taille de la charge utile du message ICMP_ECHO_REQUEST envoyé par M_c à M_a, autrement dit, via le champ LENGTH du message ICMP_ECHO_REQUEST.
5. Le transport des données entre la machine cible M_c et la machine de l'attaquant M_a se fait via le champ TTL des messages ICMP_ECHO_REQUEST.
6. Le transport des données entre la machine de l'attaquant M_a et la machine cible M_c se fait via le champ TTL des messages ICMP_ECHO_REPLY.
7. Le programme P_c est un simple fichier texte (script) n'utilisant que la commande PING offerte par le système d'exploitation de la machine cible M_c et aucun autre programme.

Le choix des champs LENGTH et TTL des messages ICMP_ECHO_REQUEST et ICMP_ECHO_REPLY comme vecteurs de données a été motivé par la possibilité de définir et / ou de visualiser les valeurs de ces deux champs sur les principaux systèmes d'exploitation [2] à l'aide de la commande PING comme le montre la Figure 1.

	Option	Paramètre	Valeur
1	-t (Linux) -i (Windows)	TTL	$1 \leq TTL \leq 255$
2	-s (Linux) -l (Windows)	Size	$1 \leq Size \leq 65500$
3	-Q (Linux) -v (Windows)	ToS	$1 \leq ToS \leq 255$

2. Windows et Linux.

Figure 1 : Options de la commande PING sur Windows et Linux.

Les paramètres des options 1 et 2 peuvent être définis et affichés sur la sorite standard de la machine cible M_c alors que le paramètre de l'option 3 peut être défini mais n'est pas affiché sur la sortie standard de M_c.

2.3 Noeuds et réseaux

Noeud Un noeud X, noté N_X , est un équipement faisant partie d'un réseau.

Réseau Un réseau est un ensemble fini de noeuds.

Poids d'un noeud Chaque noeud N_X possède un poids noté $W(N_X)$ dont la valeur est un entier naturel. $W(N_X) \in \mathbb{N}$.

Noeud passif Un noeud N_X est qualifié de passif lorsque son poids $W(N_X)$ vaut 0. $W(N_X) = 0$.

Noeud actif Un noeud N_X est qualifié d'actif lorsque son poids $W(N_X)$ est strictement supérieur à 0. $W(N_X) > 0$.

Chemin entre deux noeuds Le chemin parcouru par une donnée pour aller du noeud N_A au noeud N_B est noté $P_{N_A \to N_B}$. $P_{N_A \to N_B}$ est un arrangement fini de noeuds : $P_{N_A \to N_B} = (N_A, N_i, N_{i+1}, ..., N_B)$.

Nombre de noeuds d'un chemin On note $Card(P_{N_A \to N_B})$ le nombre de noeuds (actifs et passifs) composant le chemin $P_{N_A \to N_B}$. $Card(P_{N_A \to N_B}) \geq 2$.

Nombre de noeuds actifs d'un chemin On note $A(P_{N_A \to N_B})$ le nombre de noeuds actifs composant le chemin $P_{N_A \to N_B}$. $A(P_{N_A \to N_B}) \geq 0$.

Nombre de noeuds passifs d'un chemin On note $P(P_{N_A \to N_B})$ le nombre de noeuds passifs composant le chemin $P_{N_A \to N_B}$. $P(P_{N_A \to N_B}) \geq 0$.

Poids d'un chemin entre deux noeuds Le poids d'un chemin entre deux noeuds N_A et N_B est égal à la somme des poids des noeuds constituant le chemin $P_{N_A \to N_B}$, N_A et N_B exclus. On a donc :

$$\begin{aligned} W(P_{N_A \to N_B}) &= \sum_{i=1}^{i=Card(P_{N_A \to N_B})} W(N_i) - W(N_A) - W(N_B) \\ &= A(P_{N_A \to N_B}) + P(P_{N_A \to N_B}) \end{aligned}$$

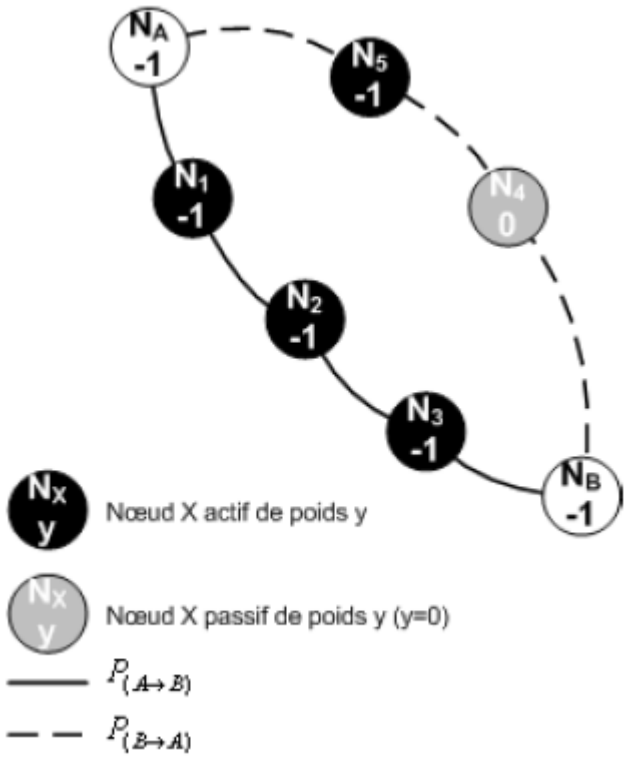

Figure 3 : Communication bi-directionnelle
entre deux noeuds N_A et N_B.

Par exemple, la Figure 3 montre que :
$P_{N_A \to N_B} = (N_1, N_2, N_3)$
$P_{N_B \to N_A} = (N_4, N_5)$
$Card(P_{N_A \to N_B}) = 3$
$Card(P_{N_B \to N_A}) = 2$
$W(P_{N_A \to N_B}) = W(N_1) + W(N_2) + W(N_3) = -1 - 1 - 1 = -3$
$W(P_{N_B \to N_A}) = W(N_4) + W(N_5) = 0 - 1 = -1$

Chemin nominal Le chemin nominal est le chemin emprunté par une information entre deux noeuds N_A et N_B (Voir Figure 4).

Chemin alternatif non altérant Un chemin alternatif non altérant est un chemin emprunté par une information entre deux noeuds N_A et N_B , différent du chemin nominal, mais dont le poids est identique au chemin nominal entre les deux noeuds N_A et N_B (Voir Figure 4).

Chemin alternatif altérant Un chemin alternatif altérant est un chemin emprunté par une information entre deux noeuds N_A et N_B , différent du chemin nominal, et dont le poids est différent du chemin nominal entre les deux noeuds N_A et N_B (Voir Figure 4).

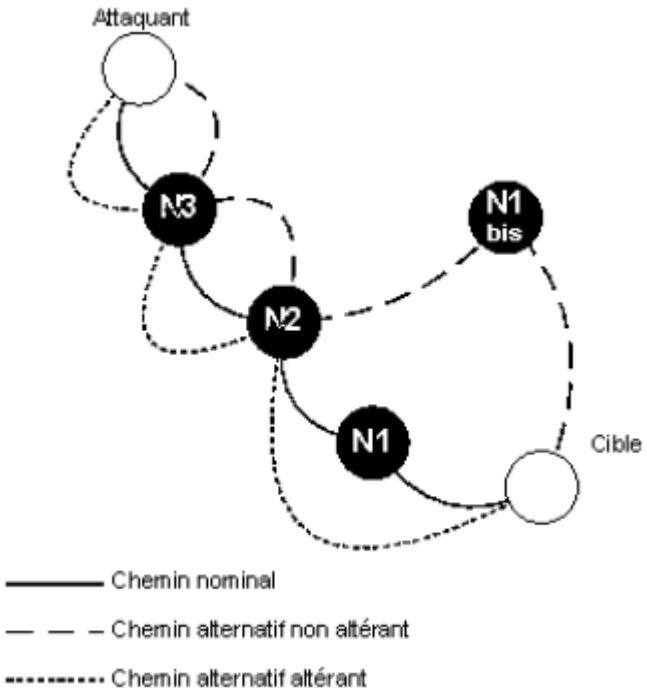

Figure 4 : Chemins nominal, alternatif et altérant entre la machine de l'attaquant M_a et la machine cible M_c.

Délai d'accusé de réception Temps au-delà duquel un noeud émetteur N_A considère qu'une donnée envoyée au noeud récepteur N_B n'a pas été reçue s'il n'a pas reçu d'accusé de réception du noeud récepteur N_B.

3 Langage et alphabet

Afin d'assurer la cohérence de la communication entre deux noeuds, ces derniers doivent utiliser un langage commun noté L. Pour concevoir ce langage L, nous utiliserons les concepts de la théorie de l'information [3].

Rappels :

1. L'alphabet d'un langage L, noté $A(L)$ est un ensemble non vide et fini de lettres l.

2. Une lettre l est un élément de l'alphabet $A(L)$ du langage L.
3. Un mot est une suite finie de lettres.

Nous devons définir l'alphabet $A(L)$ du langage L de communication entre les noeuds. On rappelle que les noeuds communiquent via les champs TTL des messages :

1. ICMP_ECHO_REQUEST envoyés par les noeuds émetteurs.
2. ICMP_ECHO_REPLY envoyés par les noeuds récepteurs.

Or, conformément à la RFC ICMP [1], le champ TTL des messages ICMP_ECHO_REQUEST et ICMP_ECHO_REPLY ne peuvent prendre que des valeurs entières comprises dans l'intervalle $[MIN\_TTL_{Theorical}; MAX\_TTL_{Theorical}]$ avec $MIN\_TTL_{Theorical} = 0$ et $MAX\_TTL_{Theorical} = 255$.

Le langage L est donc constitué d'un alphabet théorique, noté $A(L)_{Theorical}$ de 256 lettres comprises dans l'intervalle des entiers : $A(L)_{Theorical} = \{0, 1, 2, 3, ...255\}$.

On a donc $Card(A(L)_{Theorical}) = MAX\_TTL_{Theorical} + 1 = 256$.

Mais il ne s'agit là que d'un alphabet théorique. En effet, conformément à [1] la valeur du champ TTL d'un message ICMP_ECHO_REQUEST (respectivement ICMP_ECHO_REPLY) est par définition décrémentée d'une unité chaque fois que le message ICMP_ECHO_REQUEST (respectivement ICMP_ECHO_REPLY) transite via un noeud actif [3]. Il y a donc une modification [4] potentielle de l'information durant son transport.

Le sous-ensemble des lettres de l'alphabet théorique $A(L)_{Theorical}$ réellement exploitables est donc un autre alphabet, dit alphabet réel, noté $A(L)_{Real}$, constitué des entiers compris dans l'intervalle $[MIN\_TTL_{Real}; MAX\_TTL_{Real}]$ avec $MIN\_TTL_{Real} = MIN\_TTL_{Theorical}] = 0$.

L'alphabet réel $A(L)_{Real}$ dépend donc du nombre maximal de noeuds actifs entre deux noeuds. Afin que le langage L soit utilisable, quel que soient deux noeuds N_A et N_B, il faut déterminer la valeur maximale réelle de $W(P_{N_A \to N_B})$, notée MAX_ACTIVE_NODES.

Ainsi la relation suivante est vraie :

$$\forall (A, B), W(P_{N_A \to N_B}) \leq MAX\_ACTIVE\_NODES$$

Des études statistiques montrent que le poids moyen du chemin entre deux noeuds ne dépasse pas 15 en général et rarement plus de 30. Afin que le langage L puisse être utilisé dans la grande majorité des cas, nous allons prendre un cas encore plus défavorable est poser l'hypothèse suivante :

3. "Time to live in seconds; as this field is decremented at each machine in which the datagram is processed, the value in this field should be at least as great as the number of gateways which this datagram will traverse."

4. Le terme « modification » plutôt qu' « altération » est volontairement employé car la décrémentation de la valeur du champ TTL est une modification prédictible, contrairement à une altération qui ne l'est pas.

$$MAX\_ACTIVE\_NODES = 50$$

On a donc la relation suivante :

$$MAX\_TTL_{Real} = MAX\_TTL_{Theorical} - MAX\_ACTIVE\_NODES$$

Pour résumer, l'alphabet réel $A(L)_{Real}$:

1. Est un sous-ensemble de l'alphabet théorique $A(L)_{Theorical}$,
2. Est composé des entiers compris entre $MIN\_TTL_{Theorical}; MAX\_TTL_{Real}]$, soit $[0; 205]$,
3. Possède $MAX\_TTL_{Real} - MAX\_ACTIVE\_NODES$ lettres.

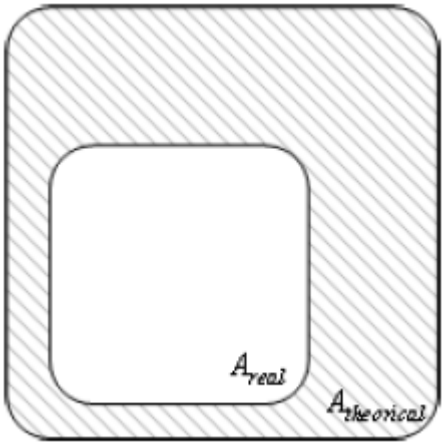

Figure 5 : Alphabet théorique et alphabet réel du langage L.

L'alphabet réel (A_{real}) du langage L est constitué de $MAX\_TTL + 1 - MAX\_ACTIVE\_NODES$ entiers à choisir dans l'intervalle $[MIN\_TTL; MAX\_TTL]$.

Attention, toutes les valeurs de cet ensemble ne sont pas utilisables pour l'alphabet de notre langage L. Par exemple, comment un noeud N_A peut-il faire parvenir une valeur TTL de MAX_TTL à un noeud N_B sachant que $W(P_{N_A \to N_B}) > 0$? C'est tout simplement impossible, car la valeur maximale de TTL que le noeud A puisse envoyer est $MAX\_TTL - W(P_{N_A \to N_B})$. L'alphabet du langage L doit donc être constitué des entiers compris dans l'intervalle $[MIN\_TTL; MAX\_TTL - MAX - ACTIVE\_NODES]$, à savoir $[0; 205]$.

Pour une lettre transmise du noeud émetteur N_A au noeud récepteur N_B par le chemin $P_{N_A \to N_B}$, la lettre à envoyer, notée $l(P_{N_A \to N_B})$, subit donc une modification, plus précisément une décrémentation de $W(P_{N_A \to N_B})$.

Il faut donc distinguer :

1. $l_s(P_{N_A \to N_B})$: La lettre à l'émission (au noeud N_A).
2. $l_r(P_{N_A \to N_B})$: La lettre à la réception (au noeud N_B).
3. $l(P_{N_A \to N_B})$: La lettre qui doit être transmise au noeud récepteur.

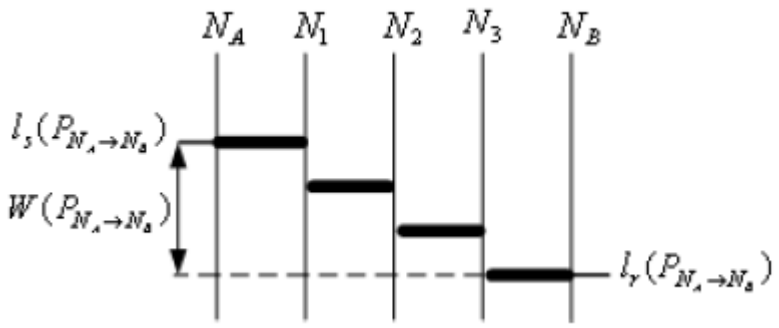

Figure 6 : Modification de la donnée pendant son transport du noeud N_A au noeud N_B.

La figure ci-dessus illustre l'équation suivante :

$$l_s(P_{N_A \to N_B}) = l_r(P_{N_A \to N_B}) + W(P_{N_A \to N_B})$$

Afin de transmettre une lettre $l(P_{N_A \to N_B})$ du noeud émetteur N_A au noeud récepteur N_B par le chemin $P_{N_A \to N_B}$, il y a deux options possibles :

Option 1 : Le noeud émetteur N_A , connaissant $W(P_{N_A \to N_B})$, ajoute cette information à la lettre $l(P_{N_A \to N_B})$ à envoyer, ce qui se traduit par l'équation suivante : $l_s(P_{N_A \to N_B}) = l(P_{N_A \to N_B}) + W(P_{N_A \to N_B})$. Le noeud émetteur envoie $l_s(P_{N_A \to N_B})$ au noeud récepteur N_B. Le noeud récepteur N_B reçoit alors la lettre $l_r(P_{N_A \to N_B}) = l(P_{N_A \to N_B})$ [5].

Option 2 : Le noeud émetteur N_A ne connait pas $W(P_{N_A \to N_B})$, il émet la lettre $l(P_{N_A \to N_B})$, ce qui se traduit par l'équation $l_s(P_{N_A \to N_B}) = l(P_{N_A \to N_B})$. C'est le noeud N_B, qui connaissant $W(P_{N_A \to N_B})$, détermine $l(P_{N_A \to N_B})$ selon l'équation $l(P_{N_A \to N_B}) = l_r(P_{N_A \to N_B}) + W(P_{N_A \to N_B})$.

Les deux options aboutissent au même résultat, c'est-à-dire la capacité pour le noeud récepteur N_B à déterminer la lettre $l(P_{N_A \to N_B})$.

Le choix de l'une ou l'autre des deux options consiste à répondre à la question suivante : lequel des noeuds, émetteur ou récepteur, a le plus facilement connaissance de $W(P_{N_A \to N_B})$? La réponse est N_B comme le montre le protocole suivant :

5. On suppose que la lettre n'a pas été altérée, c'est-à-dire qu'elle n'a pas transité via un chemin alternatif altérant.

1. Les noeuds N_A et N_B connaissent une valeur commune notée C.
2. Le noeud N_A envoie la lettre $l_s(P_{N_A \to N_B}) = C$ au noeud N_B.
3. Le noeud N_B reçoit la lettre $l_r(P_{N_A \to N_B}) = C - W(P_{N_A \to N_B})$.
4. Le noeud N_B calcule $W(P_{N_A \to N_B}) = C - l_r(P_{N_A \to N_B})$.

Figure 7 : Protocole de calcul $W(P_{N_A \to N_B})$ par le noeud émetteur N_A.

4 Perte et altération des données

Trois évènements, notés E_1, E_2 et E_3 peuvent être à l'origine de la perte ou de l'altération d'une lettre $l_s(P_{N_A \to N_B})$ envoyée par le noeud N_A au noeud N_B.

Perte réelle de données :

E_1 La lettre $l_s(P_{N_A \to N_B})$ envoyée par le noeud N_A au noeud N_B n'a jamais été recue par le noeud N_B.

Perte supposée de données :

E_2 La lettre $l_s(P_{N_A \to N_B})$ envoyée par le noeud N_A au noeud N_B a bien été recue par le noeud N_B comme $l_r(P_{N_A \to N_B})$ mais l'accusé de réception envoyé par le noeud N_B au noeud N_A n'est pas parvenu au noeud au noeud N_A dans le délai d_{N_A} [6].

Altération de données :

E_3 La lettre $l_s(P_{N_A \to N_B})$ envoyée par le noeud $N_A$au noeud N_B a bien été recue par le noeud N_B, mais ne vaut pas $l_r(P_{N_A \to N_B})$ car elle a emprunté un chemin alternatif altérant entre les noeuds N_A et N_B.

5 Détection de la perte de données

Dans le cas d'une perte de données, qu'elle soit réelle (E_1) ou supposée (E_2), seul le noeud émetteur a conscience de cet « état de perte ».

En effet, dans le cas d'une perte réelle de données (E_1) le noeud N_B n'a reçu aucune donnée et ne peut donc savoir qu'une donnée qui lui était destinée a été perdue. Dans le cas d'une perte supposée de données (E_2), le noeud N_B reçoit la donnée qui lui était destinée, envoie l'accusé de réception à l'émetteur, c'est-à-dire au noeud N_A , mais n'est pas informé par le noeud N_A que son accusé de réception n'a pas été pris en compte pour cause de dépassement du délai d_{N_A} .

6. Voir définition de « délai d'accusé de réception » au chapitre 2 de cet article.

6 Détection de l'altération de données

Dans le cas de l'altération de données (E_3), seul le noeud récepteur peut avoir conscience de cet « état d'altération ». En effet, dans le cas d'une altération de données (E_3), le noeud récepteur reçoit une lettre $l_r(P_{N_A \to N_B})$ qui n'appartient pas à l'alphabet real $A(L)_{real}$. Il en déduit alors que la lettre a été altérée.

7 Traitement de la perte de données

La perte de données, qu'elle soit réelle (E_1) ou supposée (E_2) est relativement simple à traiter, il faut :

1. Déterminer si la perte de données est réelle ou supposée.
2. Renvoyer la même donnée si la perte est réelle (E_1) et ne rien faire si la perte sinon (E_2).

8 Traitement de l'altération des données

Il est possible d'intégrer un code correcteur d'erreurs au langage de communication entre les noeuds. En effet, l'altération d'une lettre échangée entre la cible N_A et l'attaquant N_B (respectivement entre l'attaquant et la cible) intervient lorsque le poids du chemin emprunté par le message ICMP_ECHO_REQUEST (respectivement ICMP_ECHO_REPLY) diffère du poids du chemin nominal entre les noeuds N_A et N_B . L'ajout (respectivement la suppression) d'au moins un noeud actif par rapport au chemin nominal, incrémenterait (respectivement décrémenterait) la valeur de $W(P_{N_A \to N_B})$.

Deux évènements peuvent alors se produire :

Altération détectable :

$E_{3.1}$ La lettre $l_r(P_{N_A \to N_B})$ n'appartient pas à l'alphabet réel $A(L)_{real}$.

Altération non détectable :

$E_{3.2}$ La lettre $l_r(P_{N_A \to N_B})$ appartient à l'alphabet réel $A(L)_{real}$.

Parce qu'il est possible pour un message ICMP_ECHO_REQUEST ou ICMP_ECHO_REPLY d'emprunter des chemins alternatifs altérants et qui altèrent donc les données durant leur transport, il faut différencier la lettre L_s envoyée par le noeud émetteur et la lettre L_r reçue par le noeud récepteur.

Si les données ne sont pas altérées durant leur transport, alors l'égalité $L_s = L_r$ est vraie (Cas 1 de la Figure 8). Sinon, si les données sont altérées durant leur transport, alors l'une des relations ci-dessous est vraie :

1. $L_s \neq L_r$ (Cas 2 de la Figure 8).
2. $L_s = L_r$ (Cas 3 de la Figure 8).

Soient $A(L)_{real\,s}$ les lettres de l'alphabet réel ($A(L)_{real}$) qui peuvent émises par le noeud émetteur.

Soient $A(L)_{real\,r}$ les lettres de l'alphabet réel ($A(L)_{real}$) qui peuvent reçues par le noeud récepteur.

On a :

$$\begin{cases} A(L)_{real\,s} \subseteq A(L)_{real} \\ A(L)_{real\,r} \subseteq A(L)_{real} \end{cases}$$

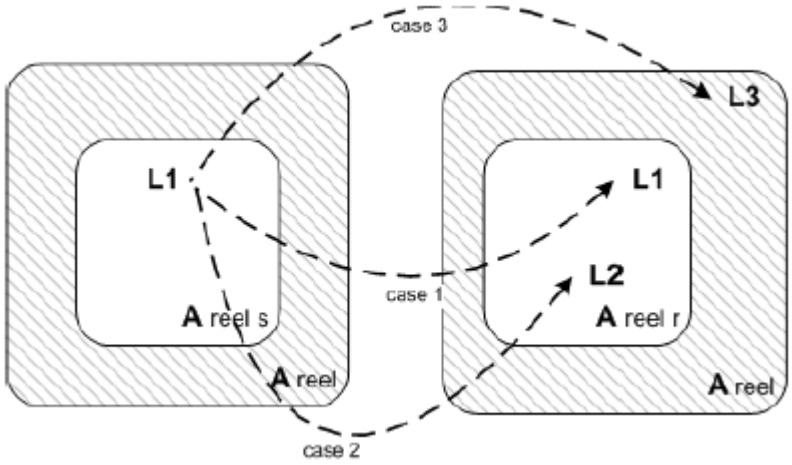

Figure 8 : Transmission des données.

La figure ci-dessus montre tous les cas qui peuvent se produire lors de la transmission des données :

Cas 1 (nominal) : La lettre L_1, $L_1 \in A(L)_{real\,s}$, envoyée par le noeud émetteur, est reçue L_1 (dans le même alphabet donc) par le noeud récepteur.

Cas 2 (altération détectable : $E_{3.1}$) : La lettre L_1, $L_1 \in A(L)_{real\,s}$, envoyée par le noeud émetteur, est reçue L_2 dans un alphabet différent de $A(L)_{real\,r}$. $L_2 \notin A(L)_{real\,r}$.

Cas 3 (altération non détectable : $E_{3.2}$) : La lettre L_1, $L_1 \in A(L)_{real\,s}$, envoyée par le noeud émetteur, est reçue L_3 dans le même alphabet. $L_3 \in A(L)_{real\,r}$.

Seize [7] lettres sont utilisées dans l'alphabet réel $A(L)_{real}$ pour la communication entre deux noeuds N_A et N_B.

7. Utilisation de l'encodage hexadécimal.

Le code correcteur d'erreur consiste à reconnaitre comme lettre L_x toute lettre reçue par le noeud récepteur comprise dans l'intervalle $[L_x - \triangle; L_x + \triangle]$.

Autrement dit, entre deux noeuds N_A et N_B, le code correcteur d'erreur est capable de corriger une variation de $W(P_{N_A \to N_B})$ de $\pm\triangle$. Plus la valeur de $\triangle$ est importante, plus la détection et la correction des erreurs sera possible.

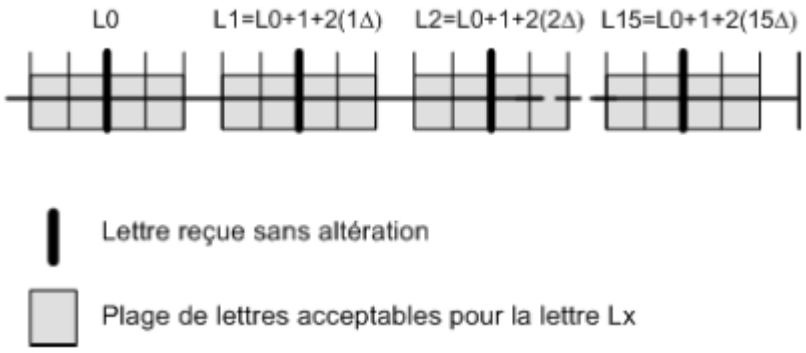

Figure 15 : Code correcteur d'erreurs.

9 Débit

Le temps en secondes nécessaire à l'envoi d'une lettre du noeud $N_A$vers le noeud N_B est noté $T(P_{(N_A \to N_B)})$.

Les lettres étant codées sur 4 bits dans Venik [2] (16 lettres), le débit en bits par seconde est donc de :

$$\frac{1}{T(P_{(N_A \to N_B)})} \times 4$$

En pratique, sur Internet, $T(P_{(N_A \to N_B)}) \approx 0.30$. Venik permet donc d'obtenir un débit approximatif de 13 bits par seconde :

$$\frac{4}{T(P_{(N_A \to N_B)})} = \frac{4}{0.30} = 13.33 \approx 13\text{bits}$$

10 Contre-mesures

Les canaux cachés sont très difficiles à détecter. Il est plus facile de mettre en place des mesures préventives destinées à empêcher l'utilisation d'un canal caché, que des mesures détectives destinées à détecter l'utilisation d'un canal caché. Les principales mesures préventives sont les suivantes :

Surveillance des flux réseau : Surveiller les flux réseau, aussi bien en termes de volume, que de fréquence.

Règles de filtrage : Appliquer la sacro-sainte règle de filtrage « Tout les flux qui ne sont pas strictement nécessaires sont interdits » [8].

8. Malheureusement, les flux ICMP sont souvent utiles dans les systèmes d'information.

Restriction de l'accès logique : Ne pas autoriser une personne potentiellement malveillante à accéder au système d'information, et ce, même de manière temporaire.

Plus de mesures préventives et détectives dans [4].

La plupart des normes de sécurité (TCSEC, ITSEC, Critères communs, …) prennent en compte l'étude des canaux cachés.

Références

[1] Internet Control Message Protocol, DARPA Internet Program, Protocol Specification.

[2] http ://www.tourdot.fr/projects/venik.

[3] Shannon, C.E., WEAVER, W, The mathematical theory of communication, Urbana, University of Illinois Press, 1949.

[4] Detecting Illicit ICMP Communication Channels, Scott Campbell.