Mécanisme d'observation d'attaques sur Internet avec rebonds

Éric Alata, Ion Alberdi, Vincent Nicomette, Philippe Owezarski et Mohamed Kaâniche

Mercredi 30 Mai 2007

Plan

Plan

Plan

Plan

Motivations

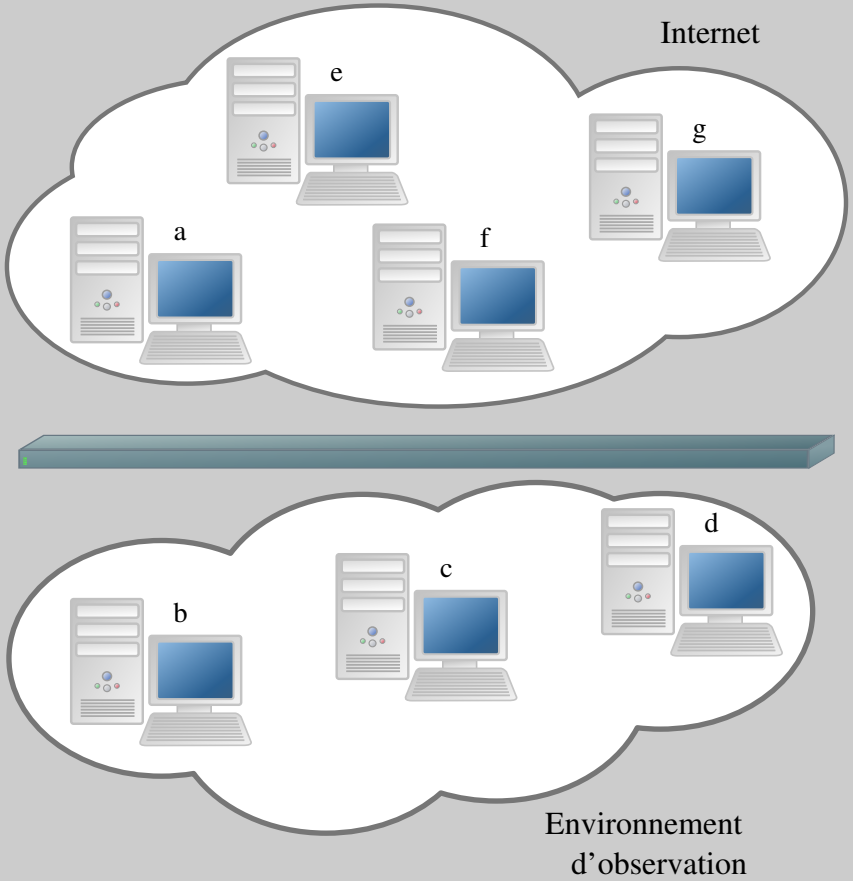
- observation des comportements malveillants
  - Pots de miel / sandbox
- délicat : rebonds
  - la cible peut être utilisée comme tremplin pour accéder à d'autres cibles
- solutions
  - interdire : masque des informations intéressantes
  - autoriser : hors-la-loi et dangeureux
    - Rate-Limiting

Motivations

- observation des comportements malveillants
  - Pots de miel / sandbox
- délicat : rebonds
  - la cible peut être utilisée comme tremplin pour accéder à d'autres cibles
- solutions
  - interdire : masque des informations intéressantes
  - autoriser : hors-la-loi et dangeureux
    - Rate-Limiting

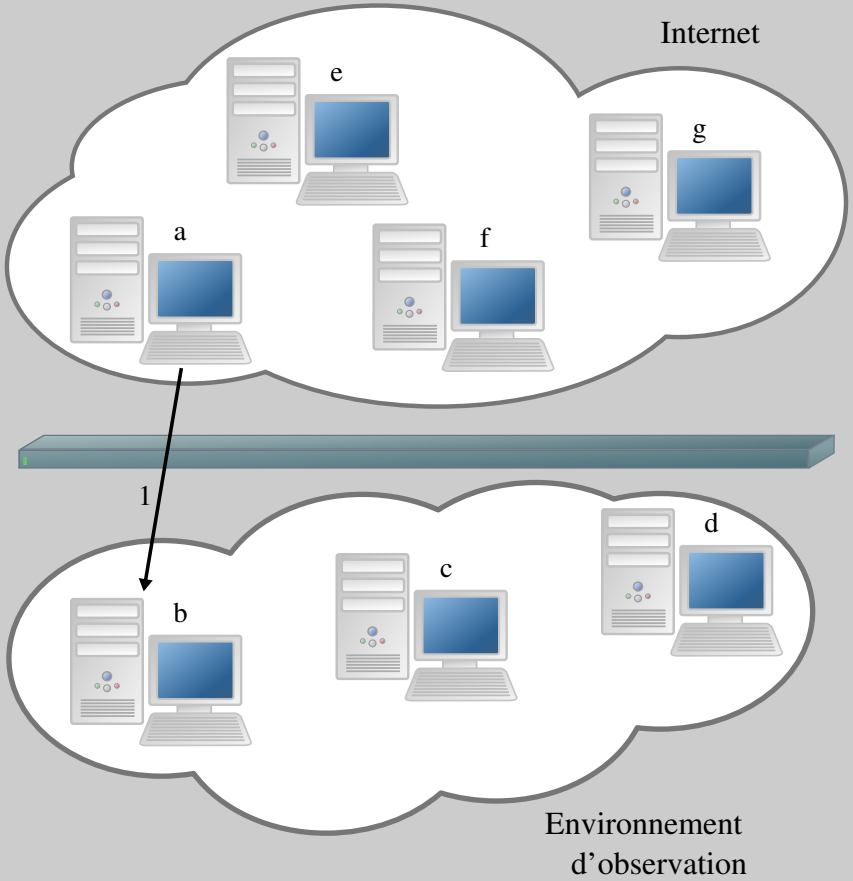
$\Rightarrow$ Besoin d'un environnement permettant l'analyse des rebonds "sans" risques.

Principe



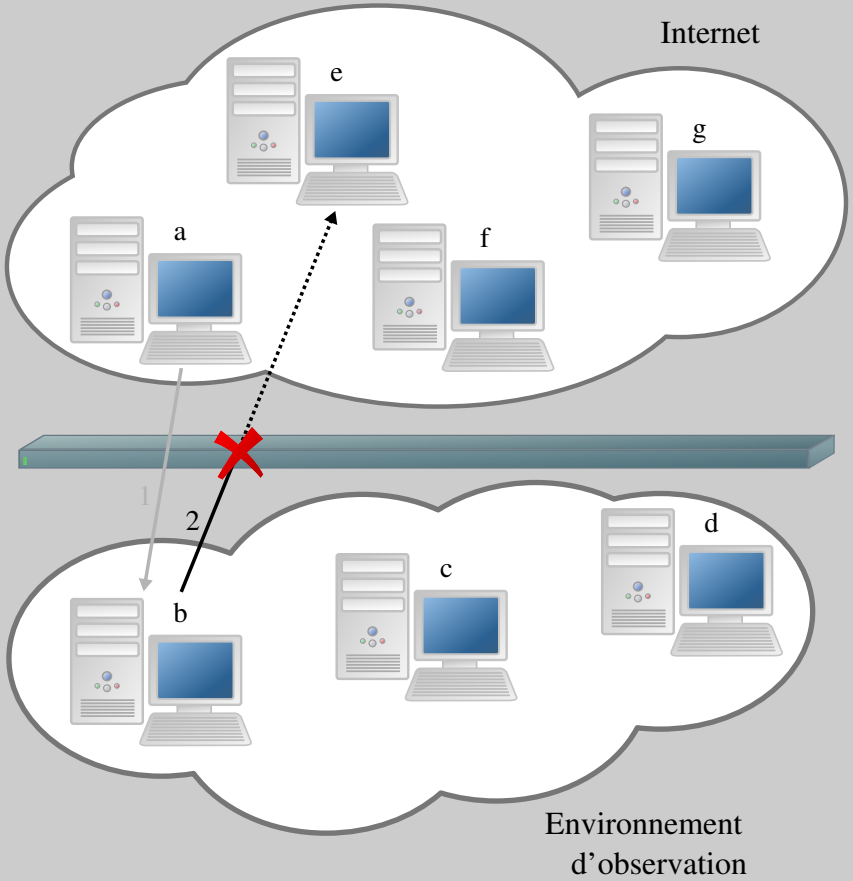
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



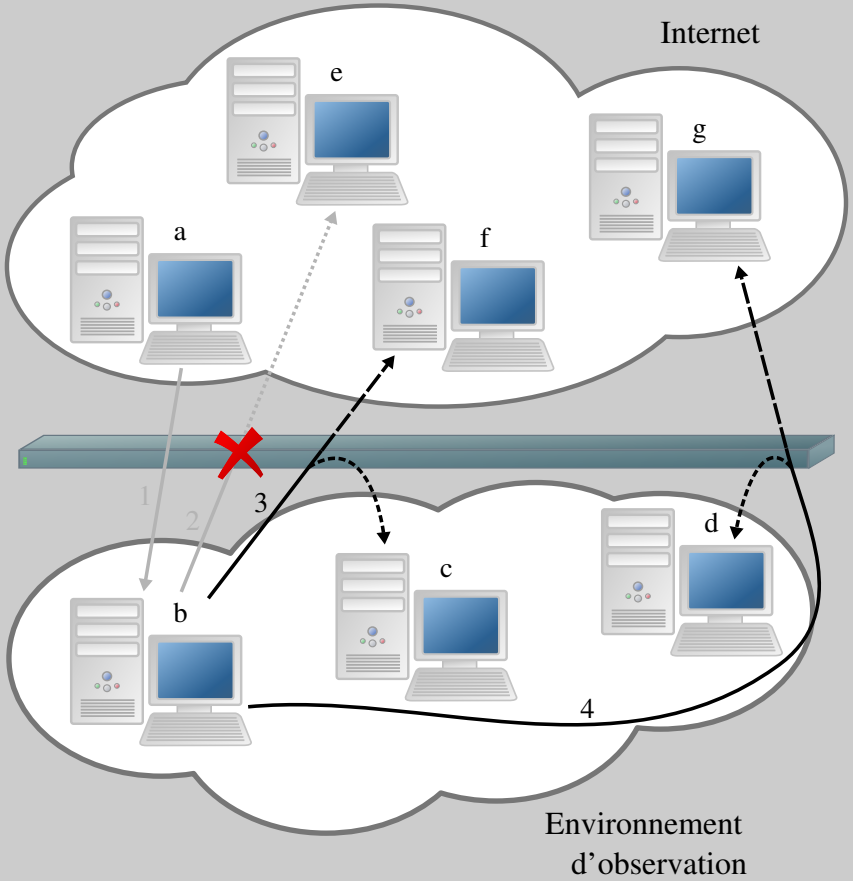
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



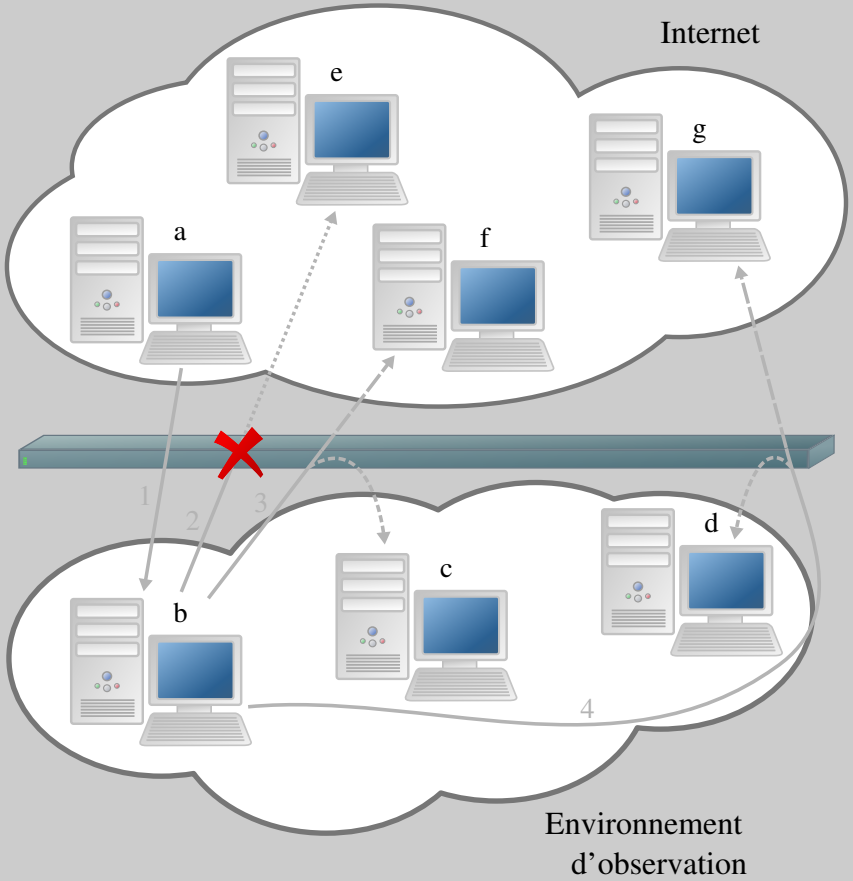
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



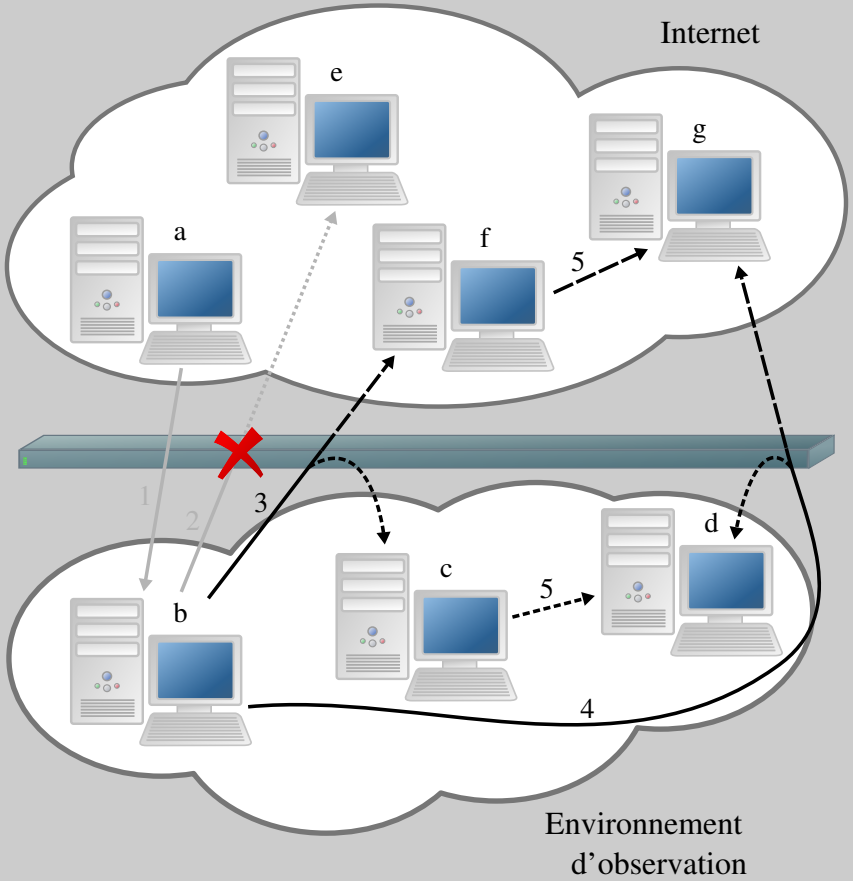
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



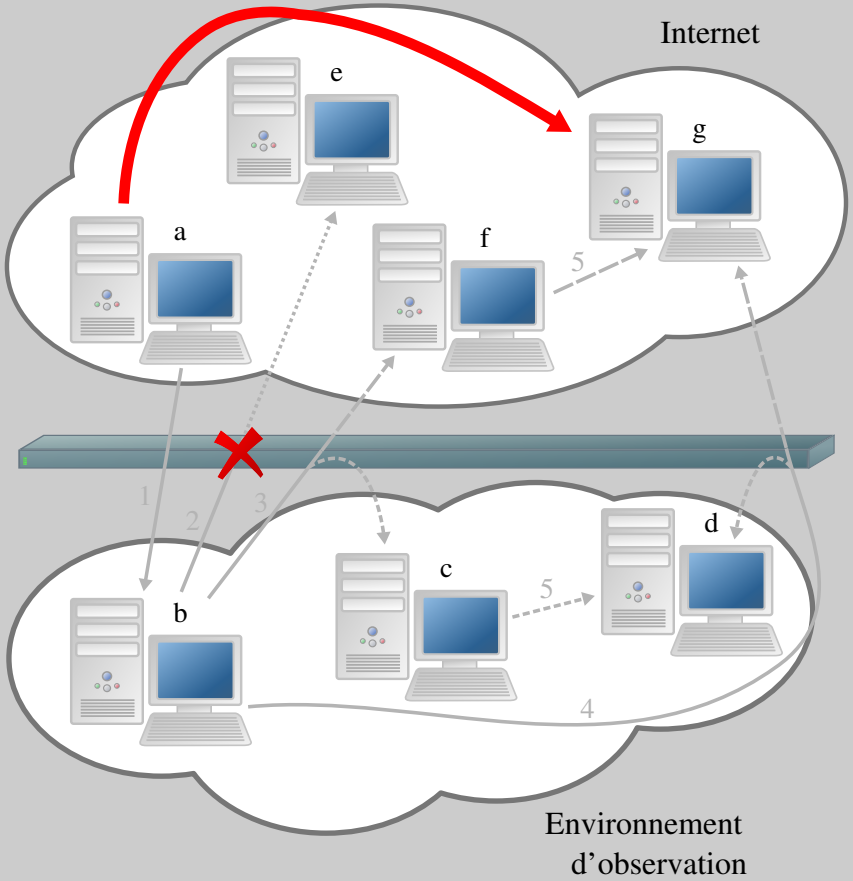
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



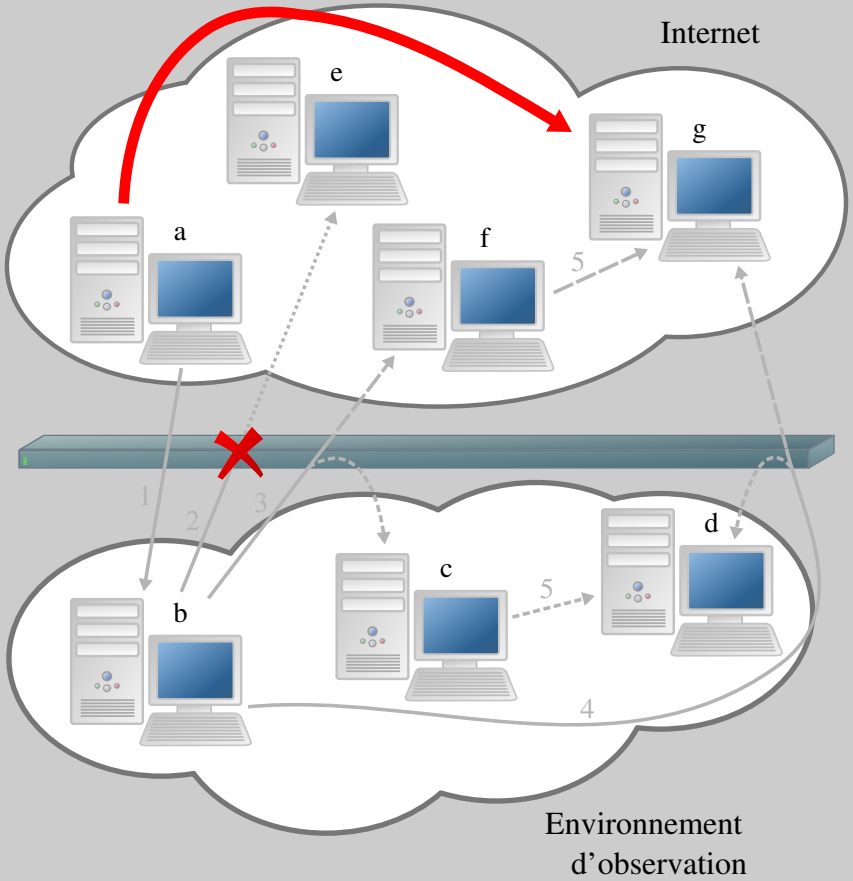
- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

Principe



- ✔ Permettre les connexions depuis Internet
- ✔ Interdire les connexions vers Internet
- ✔ Permettre la redirection d'une connexion, voire de plusieurs
- ✔ Assurer un minimum la cohérence des redirections
- ✔ Minimiser la latence induite
- ✘ Limite

⇒ Modification à apporter au niveau de la passerelle

Besoins

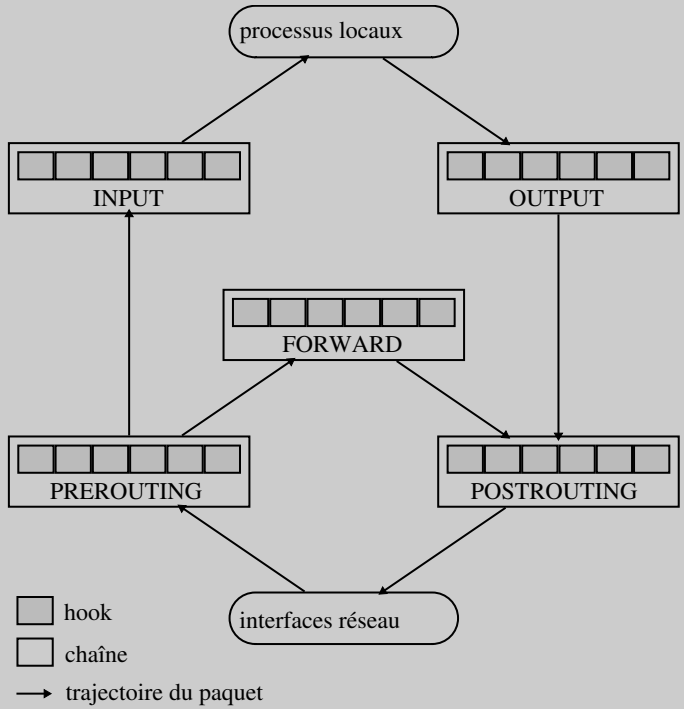
- Traiter le premier paquet de chaque communication
  Ex. TCP : SRC :X(A) DST :Y(B)
- Définir une règle de traitement pour chaque communication, en dynamique
  Ex. TCP : SRC :X(A) DST :Y(B) $\Rightarrow$ ACCEPT DST :Z(B)
  Ex. TCP : SRC :X(A) DST :Y(B) $\Rightarrow$ DROP
- Appliquer la même règle à tous les paquets d'une même communication
- Permettre de définir des règles s'appliquant à un ensemble de communications
  Ex. TCP : SRC :X(A) DST :*(B) $\Rightarrow$ ACCEPT DST :Z(C)

$\Rightarrow$ Développement d'un nouveau mécanisme

Plan

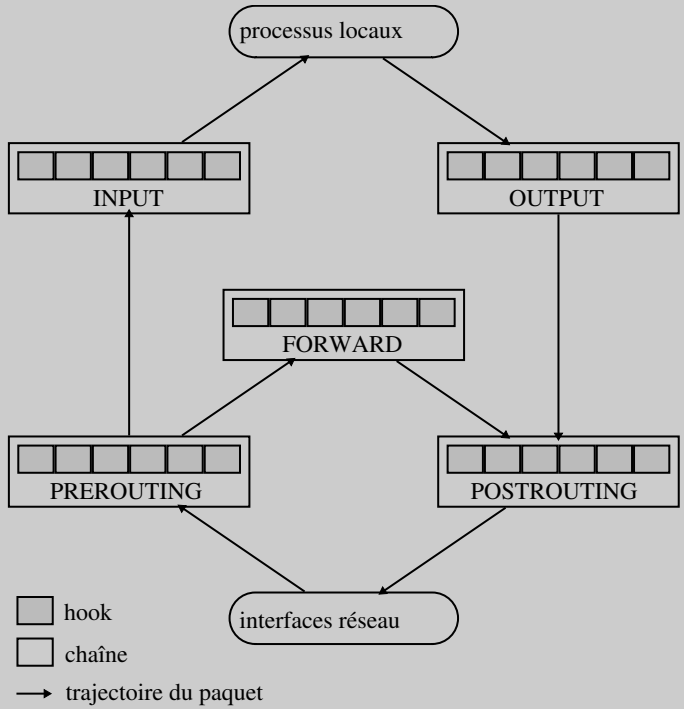
Système cible

- Système Gnu/Linux 2.6.18
- Pare-feu Netfilter
  - 3 tables principales...
    - filter, nat, mangle
  - ... contenant des chaînes...
    - INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING
  - ... qui permettent d'intercepter les paquets.
    - point d'accroche (hook)



Système cible

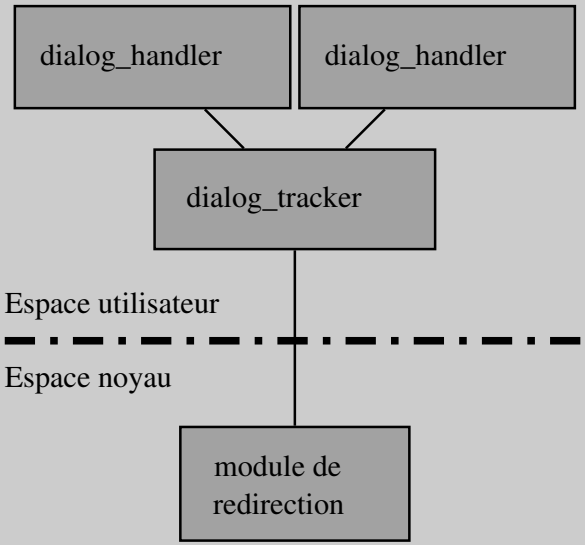
- Système Gnu/Linux 2.6.18
- Pare-feu Netfilter
  - 3 tables principales...
    - filter, nat, mangle
  - ... contenant des chaînes...
    - INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING
  - ... qui permettent d'intercepter les paquets.
    - point d'accroche (hook)



⇒ Création de points d'accroche dans Netfilter

Architecture

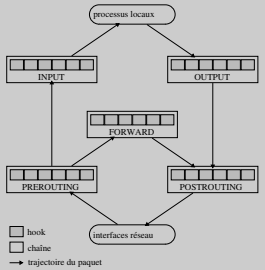
- 4 composantes :
  - *dialog_handler* : crée des règles correspondant à des communications
  - *dialog_tracker* : joint le *dialog_handler* au noyau
  - module noyau :
    1. si besoin, récupère les règles auprès du *dialog_tracker*
    2. marque les communications en fonction des règles correspondantes
  - des règles iptables : rendent les redirections effectives, en fonction de la marque




```
iptables –t nat –A PREROUTING –m connmark
   –m mark XXX –j DNAT ––to–destination HP1
iptables –t nat –A PREROUTING –m connmark
   –m mark YYY –j DNAT ––to–destination HP2
iptables –t nat –A PREROUTING –m connmark
   –m mark ZZZ –j DNAT ––to–destination HP3
.....
```

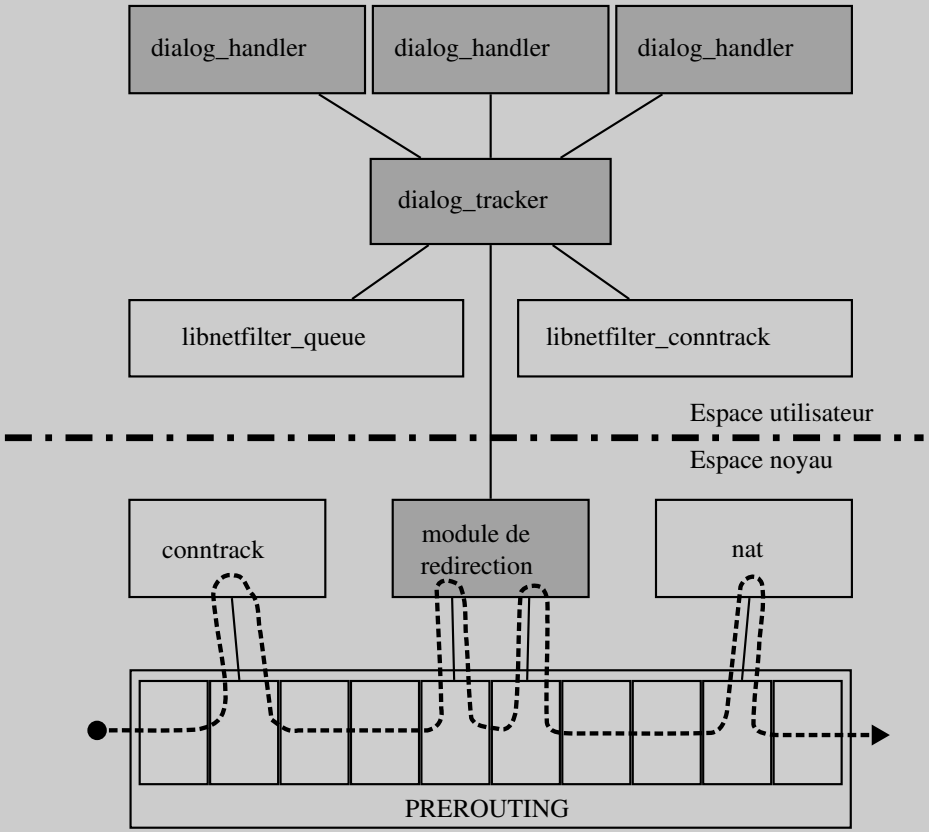
Méthodologie

- Traiter les paquets avant leur routage
  - ⇒ renseigner la chaîne PREROUTING
- Utiliser les capacités de translation de Netfilter
  - ⇒ se placer avant le point d'accroche **nat**
- Appliquer les mêmes changements à tous les paquets d'une même communication
  - ⇒ se placer après le point d'accroche **conntrack**
- Mémoriser les changements – via une base de données – pour les appliquer dans le futur
  - ⇒ passer au niveau de l'espace utilisateur (**libnetfilter_queue**)



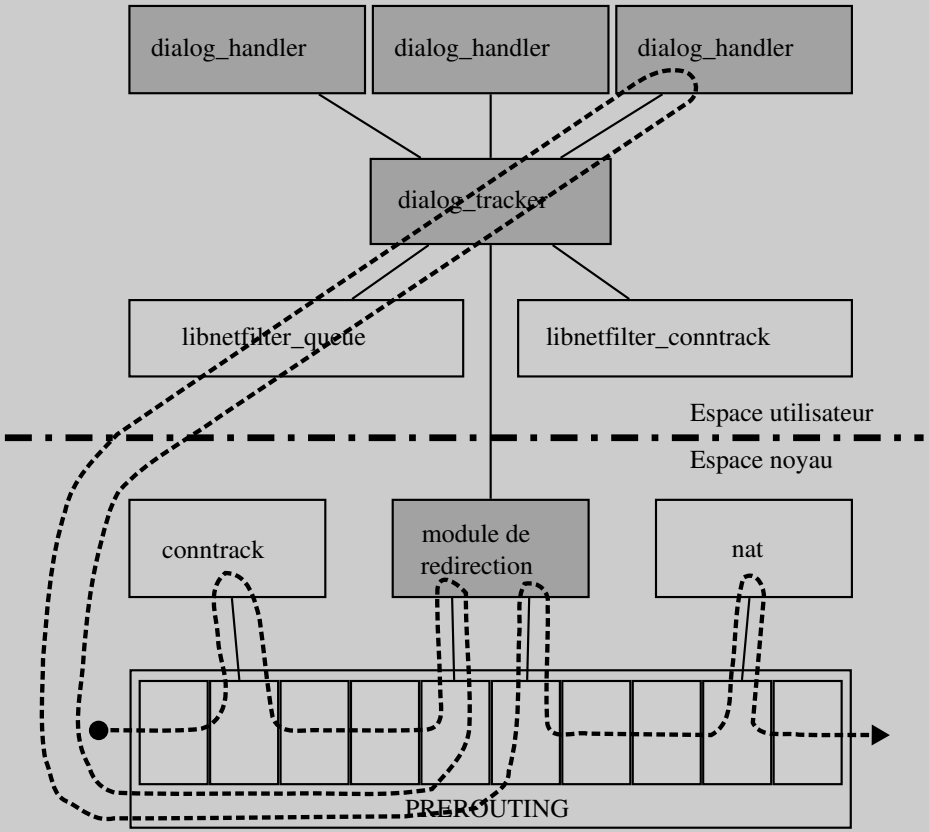
Scénario 1/2

Traitement d'une nouvelle communication associée à une règle existante



Scénario 2/2

Traitement d'une nouvelle communication associée à aucune règle existante



Plan

Nécessité d'évaluer la latence

L'implémentation du mécanisme de redirection a impliqué :

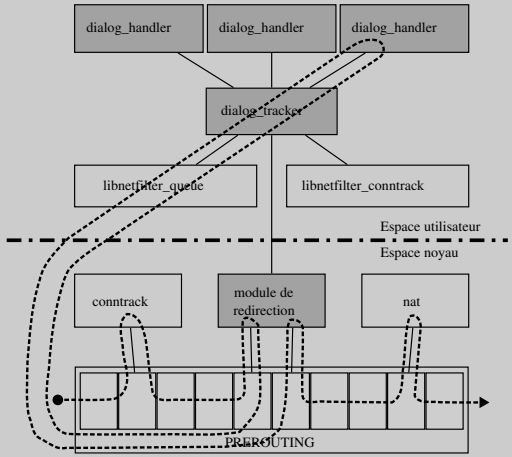
- Modification de *netfilter* .
- Communications espace utilisateur $\leftrightarrow$ espace noyau.

Ce qui implique :

- Ralentissement des connexions.
- Mécontentement et suspicions de l'intrus.

Implémentation du scan réseau

- Le mécanisme réflechit le plus lors des demandes d'établissement de nouvelles sessions.
- Regardons comment il réagit aux scans réseau.



Cas d'étude choisi

Scan de $N_{adresse}$ sur un port TCP donné, n_{hote} offrent un service sur ce port. Envoi d'un seul segment SYN par addresse, réponse attendue avant $T_{timeout}$ s.

Mesure de la latence engendrée

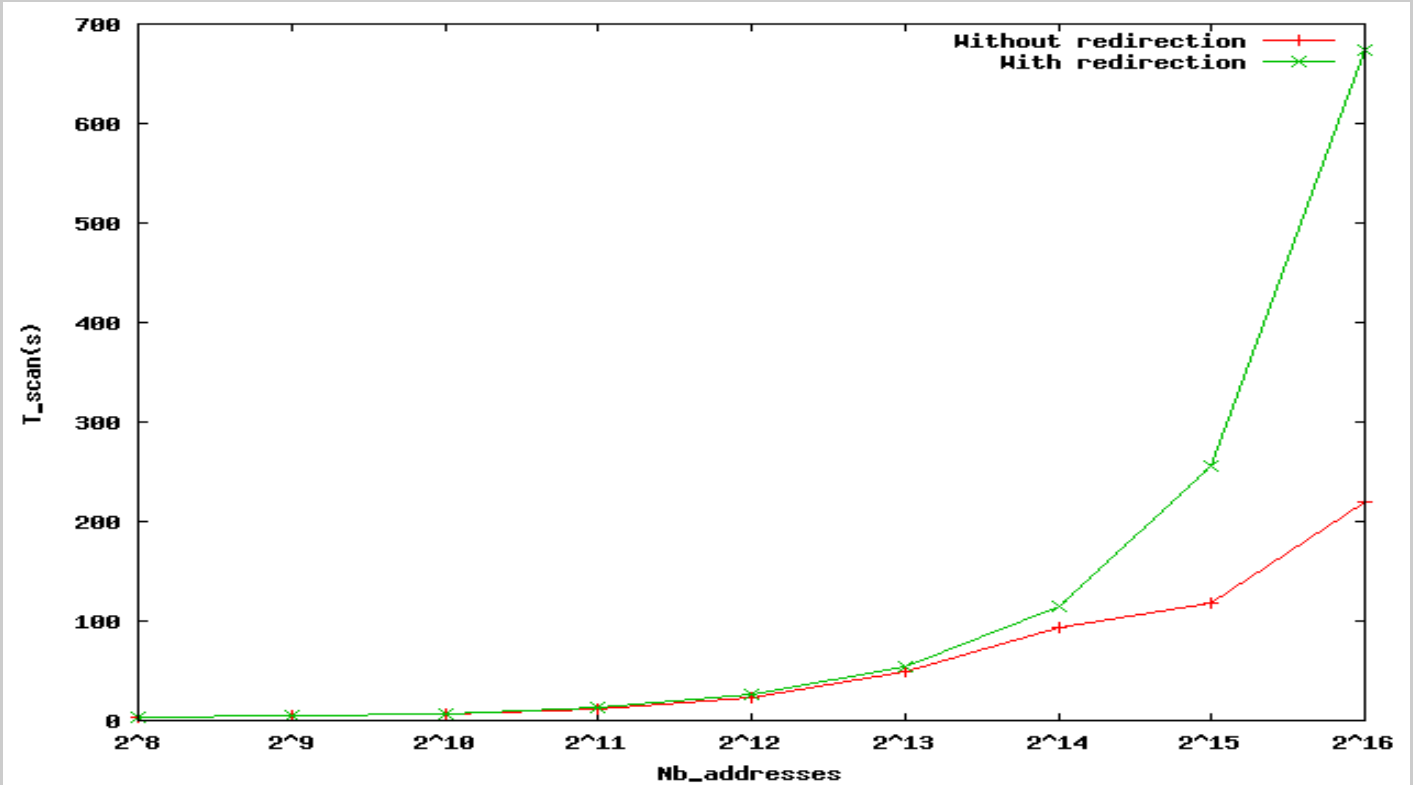


FIG.: Latence engendrée par notre mécanisme

Après avoir identifié les canaux C&C d'un botnet :

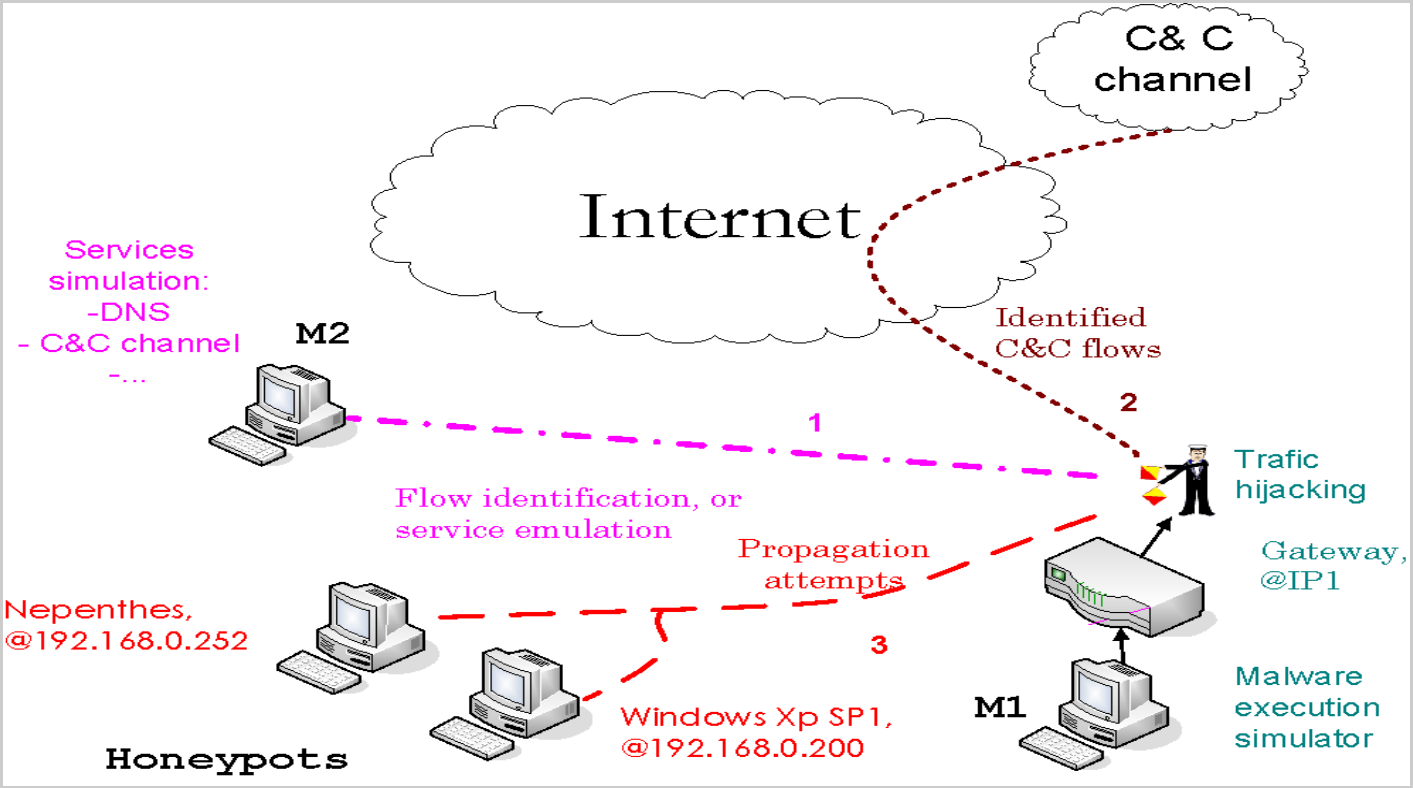


FIG.: Environnement d'expérimentation

Rapport d'activité du virus

1. Connexion à un serveur sur port TCP 5190.
2. Réception de l'ordre :
   :hub.24324.com 332 seivNbtC #last :=BGX5tCMl9HM
   uPlQRIfr7ZDvrWvjsrx3QcTGwmkNACosllT7o+6BL/FkEI1L
   zB/AkO7BSNYd1ycZi/zOu/AWHE5fJNT02YoaGogFZbH03O9Z/
   OVp4bDrWR3gJyIug2Eee3JVQHBn/fWG6ANIrYr0mZbtKuh

Rapport d'activité du virus

1. Connexion à un serveur sur port TCP 5190.
2. Réception de l'ordre :
   :hub.24324.com 332 seivNbtC #last :=BGX5tCMI9HM
   uPIQRIfr7ZDvrWvjsrx3QcTGwmkNACosIIT7o+6BL/FkEI1L
   zB/AkO7BSNYd1ycZi/zOu/AWHE5fJNT02YoaGogFZbH03O9Z/
   OVp4bDrWR3gJyIug2Eee3JVQHBn/fWG6ANIrYr0mZbtKuh

Rapport d'activité du virus

Conséquences :

1. Scan d'une plage generée à partir de @IP1 sur port TCP 135.
2. Envoi d'exploit sur les ip valides, honeypots simulent la réussitent.
3. Rapport d'activité envoyé au C&C :
   PRIVMSG #last :-04dcom2.04c-
   1. Raw transfer to 192.168.0.252 complete.
   PRIVMSG #last :-04dcom2.04c-
   2. Raw transfer to 192.168.0.200 complete.

Rapport d'activité du virus

Conséquences :

1. Scan d'une plage generée à partir de @IP1 sur port TCP 135.
2. Envoi d'exploit sur les ip valides, honeypots simulent la réussitent.
3. Rapport d'activité envoyé au C&C :
   PRIVMSG #last :-04dcom2.04c-
   1. Raw transfer to 192.168.0.252 complete.
   PRIVMSG #last :-04dcom2.04c-
   2. Raw transfer to 192.168.0.200 complete.

Rapport d'activité du virus

Conséquences :

1. Scan d'une plage generée à partir de @IP1 sur port TCP 135.
2. Envoi d'exploit sur les ip valides, honeypots simulent la réussitent.
3. Rapport d'activité envoyé au C&C :
   PRIVMSG #last :-04dcom2.04c-
   1. Raw transfer to 192.168.0.252 complete.
   PRIVMSG #last :-04dcom2.04c-
   2. Raw transfer to 192.168.0.200 complete.

Conclusion et perspectives

Le mécanisme de redirection dynamique de connexion nous a permis :

- Offrir un niveau d'interaction supérieur.
- Avoir des informations supplémentaires sur le comportement des intrus(pirate,malware).

Reste à améliorer :

- Les performances.
- Trouver des solutions aux différentes limites...

Expérimentations en cours pour avoir plus d'informations sur le comportement des intrus.

That's all folks !

Merci de votre attention.
Des questions ?

Stratégie synchrone

Stratégie de scan

- $N_{adresse} = q \times n_{thread} + r, 0 \leq r < n_{thread}$.
- *On attribue* $q + 1$ *adresses aux* r *premières, et* q *aux autres.*
- *On obtient* $T_{scan} = q \times T_{timeout}$, *avec* n_{thread}.

◂ Retour

Stratégie asynchrone

Stratégie de scan

- *Gestion des différents timers* $\Longrightarrow$ *complexité en mémoire et temps.*
- $T_{scan}(N_{adresses}) > 2 \times T_{scan}(\frac{N_{adresses}}{2})$, *pour* $N_{adresses} \geq 2^{12}$.
- 2^{12} *adresses scannées simultanément au max.*

◂ Retour