Les systèmes de détection d'intrusions réseaux

Claude Duvallet

Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon
76058 LE HAVRE CEDEX
Courriel : Claude.Duvallet@gmail.com

Plan de la présentation

Introduction

- Détection d'attaques : afin de détecter les attaques que peut subir un système, il est nécessaire de disposer d'un logiciel spécialisé dont le rôle sera de surveiller les données qui transistent sur ce système et qui serait capable de réagir si des données semblent suspectes.
- Les logiciels qui sont les plus à même d'effectuer cette tâche sont les systèmes de détection d'intrusions : les IDS.

Définition Un système de détection d'intrusions (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une action de prévention sur les risques d'intrusions (source : wikipédia).

Historique

- Les premiers systèmes de détection d'instrusions ont été initiés par l'armée américaine puis par des entreprises.
- Plus tard, des projets open-source ont été lancés comme Snort ou Prelude.
- Des produits commerciaux ont aussi vu le jour par le biais d'entreprises spécialisées en sécurité informatique : Internet Security Systems, Symantec, Cisco Systems, ...

Les différents types d'IDS

À cause de la diversité des attaques que mettent en œuvre les pirates, la détection d'instrusions doit se faire à plusieurs niveaux.
Il existe donc différents types d'IDS :

- Les systèmes de détection d'intrusions (IDS)
- Les systèmes de détection d'intrusions "réseaux" (NIDS)
- Les systèmes de détection d'intrusions de type hôte (HIDS)
- Les systèmes de détection d'intrusions hybrides
- Les systèmes de prévention d'intrusions (IPS)
- Les systèmes de prévention d'intrusions "noyau" (KIDS/KIPS)
- Les pare-feux

Les systèmes de détection d'intrusions (IDS)

- Définition : ensemble de composants logiciels et matériels dont la fonction principale est de détecter et d'analyser toute tentative d'effraction (volontaire ou non).
- Fonctions de détection :
  - des techniques de sondage (balayage de ports, fingerprinting),
  - des tentatives de compromission de systèmes,
  - d'activités suspectes internes,
  - des activités virales,
  - ou encore des audits de fichiers journaux (logs).
- Deux notions fondamentales :
  - Faux positifs : une alerte provenant d'un IDS mais qui ne correspond pas à une attaque réelle.
  - Faux négatifs : une intrusion réelle qui n'a pas été détectée.

Les systèmes de détection d'intrusions "réseaux" (NIDS)

- Objectif : analyser de manière passive les flux en transit sur le réseau et détecter les intrusions en temps réel.
- Un NIDS écoute donc tout le trafic réseau, puis l'analyse et génère des alertes si des paquets semblent dangereux.
- Les NIDS sont les IDS plus intéressants et les plus utiles du fait de l'omniprésence des réseaux dans notre vie quotidienne.

Les systèmes de détection d'intrusions de type hôte (HIDS)

- Un HIDS se base sur une unique machine, n'analysant cette fois plus le trafic réseau mais l'activité se passant sur cette machine.
- Il analyse en temps réel les flux relatifs à une machine ainsi que les journaux.
- Un HIDS a besoin d'un système sain pour vérifier l'intégrité des données.
- Si le système a été compromis par un pirate, le HIDS ne sera plus efficace.
- Pour parer à ces attaques, il existe des KIDS (Kernel Intrusion Detection System) et KIPS (Kernel Intrusion Prevention System) qui sont fortement liés au noyau.

Les systèmes de détection d'intrusions "hybrides"

- Généralement utilisés dans un environnement décentralisé, ils permettent de réunir les informations de diverses sondes placées sur le réseau.
- Leur appellation « hybride » provient du fait qu'ils sont capables de réunir aussi bien des informations provenant d'un système HIDS qu'un NIDS.
- L'exemple le plus connu dans le monde Open-Source est Prelude.
  - Ce framework permet de stocker dans une base de données des alertes provenant de différents systèmes relativement variés.
  - Utilisant Snort comme NIDS, et d'autres logiciels tels que Samhain en tant que HIDS, il permet de combiner des outils puissants pour permettre une visualisation centralisée des attaques.

Les systèmes de prévention d'intrusions (IPS)

- Définition : ensemble de composants logiciels et matériels dont la fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système.
- Les IPS sont des outils aux fonctions « actives », qui en plus de détecter une intrusion, tentent de la bloquer.
- Les IPS ne sont pas la solution parfaite.
- Plusieurs stratégies de prévention d'intrusions existent :
  - **host-based memory and process protection :** surveille l'exécution des processus et les tue s'ils ont l'air dangereux (buffer overflow). Cette technologie est utilisée dans les KIPS (Kernel Intrusion Prevention System).
  - **session interception / session sniping :** termine une session TCP avec la commande TCP Reset : « RST ». Ceci est utilisé dans les NIPS.
  - **gateway intrusion detection :** si un système NIPS est placé en tant que routeur, il bloque le trafic ; sinon il envoie des messages à d'autres routeurs pour modifier leur liste d'accès.

Les inconvénients des IPS (1/2)

Un IPS possède de nombreux inconvénients :

- Le premier est qu'il bloque toute activité qui lui semble suspecte.
  - Or, il est impossible d'assurer une fiabilité à 100% dans l'identification des attaques.
  - Un IPS peut donc malencontreusement bloquer du trafic inoffensif !
  - Exemple : un IPS peut détecter une tentative de déni de service alors qu'il s'agit d'une période chargée en trafic.
  - Les faux positifs sont donc très dangereux pour les IPS.
- Le deuxième inconvénient est qu'un pirate peut utiliser sa fonctionnalité de blocage pour mettre hors service un système.
  - Prenons l'exemple d'un individu mal intentionné qui attaque un système protégé par un IPS, tout en spoofant son adresse IP. Si l'adresse IP spoofée est celle d'un nœud important du réseau, les conséquences seront catastrophiques.
  - Pour palier ce problème, de nombreux IPS disposent de « white lists », c-à-d des listes d'adresses réseaux qu'il ne faut en aucun cas bloquer.

Les inconvénients des IPS (2/2)

- Et enfin, le troisième inconvénient et non le moindre : un IPS est peu discret.
  - En effet, à chaque blocage d'attaque, il montre sa présence.
  - Cela peut paraître anodin, mais si un pirate remarque la présence d'un IPS, il tentera de trouver une faille dans celui-ci afin de réintégrer son attaque... mais cette fois en passant inaperçu.
  - Voilà pourquoi les IDS passifs sont souvent préférés aux IPS.
  - Cependant, il est intéressant de noter que plusieurs IDS (Ex : Snort, RealSecure, Dragon, ...) ont été dotés d'une fonctionnalité de réaction automatique à certains types d'attaques.

Les IPS "noyau" (KIDS/KIPS) (1/2)

- Dans le cadre du HIDS, l'utilisation d'un détecteur d'intrusions au niveau noyau peut s'avérer parfois nécessaire pour sécuriser une station.
- Exemple d'un serveur web : il serait dangereux qu'un accès en lecture/écriture dans d'autres répertoires que celui consultable via http, soit autorisé. Cela pourrait nuire à l'intégrité du système.
- Grâce à un KIPS, tout accès suspect peut être bloqué directement par le noyau, empêchant ainsi toute modification dangereuse pour le système.
- Le KIPS peut reconnaître des motifs caractéristiques du débordement de mémoire, et peut ainsi interdire l'exécution du code.

Les IPS "noyau" (KIDS/KIPS) (2/2)

Kernel Intrusion Detection Systems/Kernel Intrusion Prevention Systems

- Le KIPS peut également interdire l'OS d'exécuter un appel système qui ouvrirait un shell de commandes.
- Puisqu'un KIPS analyse les appels systèmes, il ralentit l'exécution.
- C'est pourquoi, ce sont des solutions rarement utilisées sur des serveurs souvent sollicités.
- Exemple de KIPS : SecureIIS, qui est une surcouche du serveur IIS de Microsoft.

Les pare-feux

- Les pare-feux ne sont pas des IDS à proprement parler mais ils permettent également de stopper des attaques.
- Les pare-feux sont basés sur des règles statiques afin de contrôler l'accès des flux.
- Ils travaillent en général au niveau des couches basses du modèle OSI (jusqu'au niveau 4), ce qui est insuffisant pour stopper une intrusion.
- Par exemple, lors de l'exploitation de la faille d'un serveur Web, le flux HTTP sera autorisé par le pare-feu puisqu'il n'est pas capable de vérifier ce que contiennent les paquets.

Les différents types de pare-feux

Il existe trois types de pare-feux :

- Les systèmes à filtrage de paquets sans état : ils analysent les paquets les uns après les autres, de manière totalement indépendante.
- Les systèmes à maintien d'état (stateful) : ils vérifient que les paquets appartiennent à une session régulière.
  - Ce type de pare-feu possède une table d'états où est stocké un suivi de chaque connexion établie, ce qui permet au pare-feu de prendre des décisions adaptées à la situation.
  - Ces pare-feux peuvent cependant être outrepassés en faisant croire que les paquets appartiennent à une session déjà établie.
- Les pare-feux de type proxy : Le pare-feu s'intercale dans la session et analyse l'information afin de vérifier que les échanges protocolaires sont conformes aux normes.

Les méthodes de détections

- Comprendre les systèmes de détection d'intrusions nécessite de se poser les questions suivantes :
  - Comment une intrusion est-elle détectée par un tel système ?
  - Quel critère différencie un flux contenant une attaque d'un flux normal ?
- Á partir de ces questions, nous pouvons étudier le fonctionnement interne d'un IDS.
- Deux techniques principales sont mises en place dans la détection d'attaques :
  - La première consiste à détecter des signatures d'attaques connues dans les paquets circulant sur le réseau.
  - La seconde consiste, quant à elle, à détecter une activité suspecte dans le comportement de l'utilisateur.

$\Longrightarrow$ Ces deux techniques, aussi différentes soient-elles, peuvent être combinées au sein d'un même système afin d'accroître la sécurité.

L'approche par scénario

- Cette technique s'appuie sur la connaissance des techniques utilisées par les attaquants pour en déduire des scénarios typiques.
- Elle ne tient pas compte des actions passées de l'utilisateur et utilise des signatures d'attaques :
  - ensemble de caractéristiques permettant d'identifier une activité intrusive :
    - une chaîne alphanumérique,
    - une taille de paquet inhabituelle,
    - une trame formatée de manière suspecte, ...

Recherche de motifs (pattern matching)

- La méthode la plus connue et la plus à facile à comprendre.
- Elle se base sur la recherche de motifs (chaînes de caractères ou suite d'octets) au sein du flux de données.
- L'IDS comporte une base de signatures où chaque signature contient le protocole et le port utilisé par l'attaque ainsi que le motif qui permettra de reconnaître les paquets suspects.
- Le principal inconvénient de cette méthode est que seules les attaques reconnues par les signatures seront détectées.
  - Il est donc nécessaire de mettre à jour régulièrement le base de signatures.
- Un autre inconvénient est que les motifs sont en général fixes.
  - Or, une attaque n'est pas toujours identique à 100%.
  - Le moindre octet différent par rapport à la signature provoquera la non détection de l'attaque.
- Pour les IDS utilisant cette méthode, il est nécessaire d'adapter la base de signatures en fonction du système à protéger.

Recherche de motifs dynamiques

- Le principe de cette méthode est le même que précédemment mais les signatures des attaques évoluent dynamiquement.
- L'IDS est de ce fait doté de fonctionnalités d'adaptation et d'apprentissage.

Analyse de protocoles

- Cette méthode se base sur une vérification de la conformité (par rapport aux RFC) des flux, ainsi que sur l'observation des champs et paramètres suspects dans les paquets.
- Cependant, les éditeurs de logiciels et les constructeurs respectent rarement à la lettre les RFC et cette méthode n'est pas toujours très performante.
- L'analyse protocolaire est souvent implémentée par un ensemble de préprocesseurs, où chaque préprocesseur est chargé d'analyser un protocole particulier (FTP, HTTP, ICMP, ...).
- Du fait de la présence de tous ces préprocesseurs, les performances dans un tel système s'en voient fortement dégradées.
- L'intérêt fort de l'analyse protocolaire est qu'elle permet de détecter des attaques inconnues, contrairement au pattern matching qui doit connaître l'attaque pour pouvoir la détecter.

Analyse heuristique et détection d'anomalies

- Le but de cette méthode est, par une analyse intelligente, de détecter une activité suspecte ou toute autre anomalie.
- Par exemple : une analyse heuristique permet de générer une alarme quand le nombre de sessions à destination d'un port donné dépasse un seuil dans un intervalle de temps prédéfini.

L'approche comportementale

- Cette technique consiste à détecter une intrusion en fonction du comportement passé de l'utilisateur.
- Pour cela, il faut préalablement dresser un profil utilisateur à partir de ses habitudes et déclencher une alerte lorsque des événements hors profil se produisent.
- Cette technique peut être appliquée non seulement à des utilisateurs mais aussi à des applications et services.
- Plusieurs métriques sont possibles : la charge CPU, le volume de données échangées, le temps de connexion sur des ressources, la répartition statistique des protocoles et applications utilisés, les heures de connexion, ...

Inconvénients de l'approche comportementale

- Peu fiable : tout changement dans les habitudes de l'utilisateur provoque une alerte.
- Elle nécessite une période de non fonctionnement pour mettre en œuvre les mécanismes d'auto-apprentissage :
  - si un pirate attaque pendant ce moment, ses actions seront assimilées à un profil utilisateur, et donc passeront inaperçues lorsque le système de détection sera complètement mis en place.
- L'établissement du profil doit être souple afin qu'il n'y ait pas trop de fausses alertes :
  - le pirate peut discrètement intervenir pour modifier le profil de l'utilisateur afin d'obtenir après plusieurs jours ou semaines, un profil qui lui permettra de mettre en place son attaque sans qu'elle ne soit détectée.

Différentes méthodes dans l'approche comportementale

- Approche probabiliste :
  - Des probabilités sont établies permettant de représenter une utilisation courante d'une application ou d'un protocole.
  - Toute activité ne respectant pas le modèle probabiliste provoquera la génération d'une alerte.
  - Exemple : Avec le protocole HTTP, il y a une probabilité de 0.9 qu'une commande GET soit faite après une connexion sur le port 80. Il y a ensuite une probabilité de 0.8 que la réponse à cette commande GET soit « HTTP/1.1 200 OK ».
- Approche statistique :
  - Le but est de quantifier les paramètres liés à l'utilisateur : taux d'occupation de la mémoire, utilisation des processeurs, valeur de la charge réseau, nombre d'accès à l'Intranet par jour, vitesse de frappe au clavier, sites les plus visités, ...
  - Elle n'est actuellement présente que dans le domaine de la recherche, où les chercheurs utilisent des réseaux neuronaux et la fouille de données pour tenter d'avoir des résultats convaincants.

Les méthodes répandues

Les IDS mélangent généralement les différents méthodes de détection d'intrusions.

Pattern Matching	→algorithmes de recherche de motifs
	→algorithmes de comptage
	→algorithmes génétiques
Analyse Protocolaire	→conformité aux RFC
Détection d'anomalies	→méthodes heuristiques
Analyse statistique	→modèles statistiques
Analyse probabiliste	→réseaux bayésiens
Autres méthodes	→réseaux de neurones
	→systèmes experts
	→fouille de données
	→immunologie
	→graphes

Déploiement d'un NIDS

- Un NIDS n'est pas suffisant pour assurer la sécurité.
- Il faut aussi effectuer les actions habituelles :
  - les systèmes et applications doivent être mises à jour régulièrement (mises à jour de sécurité).
  - les systèmes utilisant internet doivent être dans un réseau isolé (DMZ).
  - chaque utilisateur doit être averti de 'importance de la sécurité de ses mots de passe.
  - les services qui ne sont pas utilisés doivent être désactivés.
- Le déploiement d'un IDS doit tenir compte du système d'exploitation et les règles doivent être correctement configurées par rapport à celui-ci.
- L'emplacement des sondes est très important : i.e. l'endroit où l'on capture le traffic réseau.
- Il faut penser à sécuriser les sondes et les logs d'alerte.

Principe de précaution

- Il faut configurer correctement l'IDS pour qu'il n'inonde pas les rapport d'alertes avec des faux positifs.
  - ils peuvent rendre les rapports d'alertes long à analyser.
  - les administrateurs passeront beaucoup de temps à distinguer les faux positifs des véritables intrusions.
- Il faut aussi veiller à ce que l'IDS ne génère pas de faux négatifs.
- Les débits actuels des réseaux augementent de plus en plus et donc les IDS ont de plus en plus de paquets à traiter et à analyser.

Les étapes de fonctionnement d'un IDS

1. capture de la trame par l'interface en mode promiscuité (promiscous mode).
2. analyse de la trame et filtrage éventuel en bas niveau.
3. détection de la présence de fragments ou non et passage éventuel à un moteur de reconstruction.
4. transfert de la trame vers le système d'exploitation.
5. filtrage éventuel.
6. applications de divers préprocesseurs en fonction du type de requête afin de contrer des techniques d'évasion d'attaques (voir plus loin).
7. passage vers le moteur d'analyse (protocole, pattern matching, statistique, ...).

- Répartition de charges et amélioration des performances
  $\Rightarrow$ séparer les flux et mettre en place plusieurs sondes.
- L'utilisation de plusieurs sondes nécessite la corrélation des informations par le biais de plusieurs actions :
  - agrégation : rassembler les informations des différentes sondes.
  - fusion : fusionner en supprimant les doublons (même attaque détectée par plusieurs sondes).
  - corrélation : définir un motif commun, c'est-à-dire interpréter une suite d'événements et les résumer.

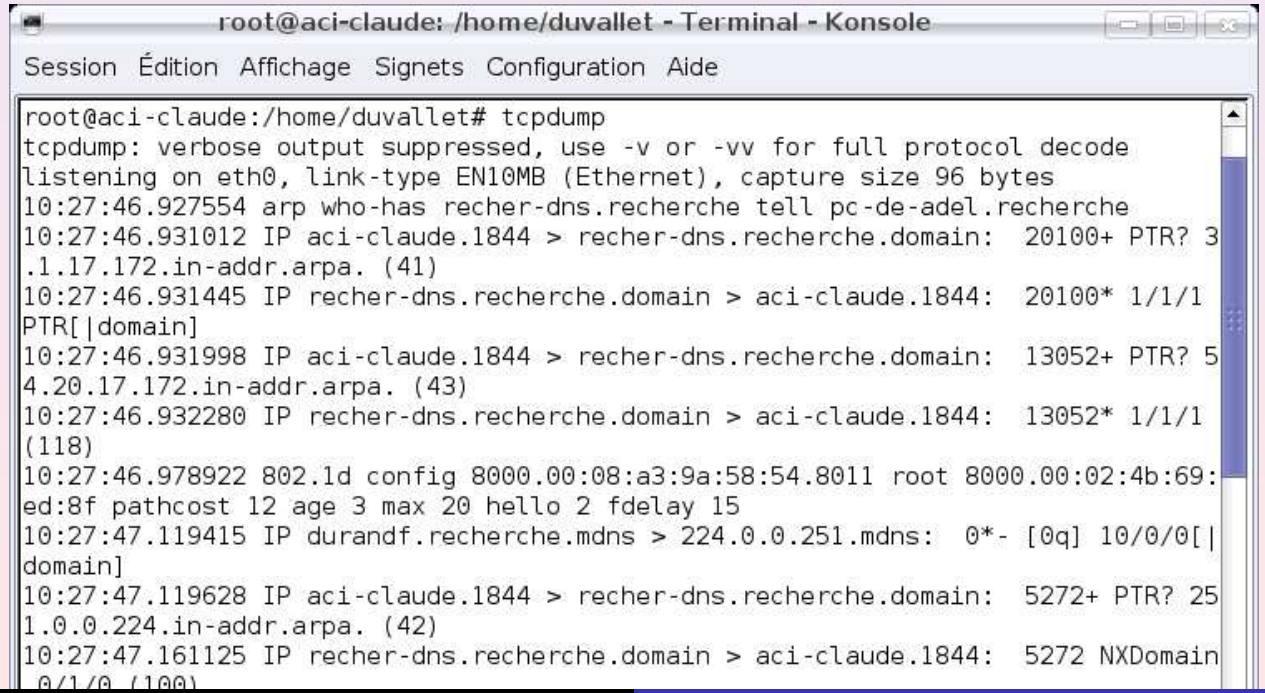
$\Longrightarrow$ Une corrélation intéressante serait de ne garder que les alertes qui concernent une faille probable du système.

Les outils

- Tcpdump : outil en ligne de commande permettant d'écouter le réseau.
- Ethereal ou Wireshark : logiciel open-source permettant la capture et l'analyse de traffic réseau en mode graphique.
- Snort : système de détection d'intrusion libre publié sous licence GNU GPL. À l'origine écrit par Martin Roesch, il appartient actuellement à Sourcefire.
- ACID (Analysis Console for Intrusion Databases) : outil d'aministration d'un IDS permettant de se connecter la base de données de SNORT.

Lancement de tcpdump

- Il est préférable de lancer TCPdump en mode super utilisateur car il passe votre interface réseau en "promiscuous mode" ce qui nécessite certain priviléges.
- "promiscuous mode" signifie que votre interface va accepter tous les paquets IP, même ceux qui ne lui sont pas destinés.
- Vous pouvez lancer tcpdump sans option :



Interprétation de tcpdump

TCPdump génére une ligne par paquet IP. Avec les options par défaut, une ligne ressemble à :

- **10 :27 :46.931012** IP aci-claude.1844 > recher-dns.recherche.domain
  → Heure d'arrivée du paquet sur l'interface réseau
- 10 :27 :46.931012 **IP** aci-claude.1844 > recher-dns.recherche.domain
  → Type de protocole (ici IP, peut être aussi ARP, IGMP, GRE, ...)
- 10 :27 :46.931012 IP **aci-claude**.1844 > recher-dns.recherche.domain
  → Adresse réseau source
- **10 :27 :46.931012** IP aci-claude.**1844** > recher-dns.recherche.domain
  → Port réseau source
- **10 :27 :46.931012** IP aci-claude.1844 > **recher-dns.recherche**.domain
  → Adresse réseau destination
- 10 :27 :46.931012 IP aci-claude.1844 > recher-dns.recherche.**domain**
  → Port réseau destination (domain = requête DNS UDP/53)

Les modes verbeux de tcpdump

Pour en savoir plus sur les paquets échangés, vous avez les options suivantes :

- **-v**

```
11:52:16.126791 IP (tos 0x0, ttl 64, id 0, offset 0, flags
[DF], proto: UDP (17), length: 146)
recher-dns.recherche.domain > aci-claude.1850: 38664* 1/1/1
165.30.17.172.in-addr.arpa.
```

- **-vv**

```
11:53:50.173426 IP (tos 0x0, ttl 64, id 0, offset 0, flags
[DF], proto: UDP (17), length: 146)
recher-dns.recherche.domain > aci-claude.1850: 27392* q: PTR?
203.30.17.172.in-addr.arpa. 1/1/1
203.30.17.172.in-addr.arpa.[|domain]
```

- **-vvv**

```
...
```

Pour avoir un dump de chaque paquet :

- **-A** : exemple tcpdump -v -A

```
13:43:28.699404 IP (tos 0x0, ttl 255, id 0, offset 0, flags
[DF], proto: UDP (17), length: 73) aci-claude.mdns >
224.0.0.251.mdns: 0 PTR? 255.255.17.172.in-addr.arpa. (45)
E..I..@....p...&.........5.0.............255.255.17.172.in-addr.arpa
```

Filtrer les paquests de tcpdump

- En fonction de l'adresse source ou destination :
```
tcpdump host www.google.fr
14:11:20.438496 IP aci-claude.45050 >
fx-in-f103.google.com.www: . ack 6753 win 5774
<nop,nop,timestamp 176188646 2732389773>
```
- En fonction de l'adresse de destination uniquement :
```
tcpdump dst www.google.fr
14:20:41.950306 IP aci-claude.46051 >
fx-in-f99.google.com.www: . ack 6836 win 5774
<nop,nop,timestamp 176329020 3755286292>
```
- En fonction de port utilisé :
```
tcpdump port http
14:30:18.440969 IP aci-claude.40621 >
sebulba.privatedns.com.www: P 4245:4958(713) ack 100806 win
16022 <nop,nop,timestamp 176473139 2858750850
```
- En fonction de protocole utilisé :
```
tcpdump proto TCP
14:38:52.774942 IP aci-claude.30961 >
scott.univ-lehavre.fr.ssh: . 18488273:18489721(1448) ack 10896
win 2516 <nop,nop,timestamp 176601720 707031894>
```

Quelques options supplémentaires de tcpdump

- **-n** : ne pas effectuer de résolution de nom et donc afficher directement les adresses IP

```
14:46:50.672227 IP 172.17.20.38.30961 >
193.52.167.215.22: . 14770137:14771585(1448) ack 8832
win 2516 <nop,nop,timestamp 176721191 707509879>
```

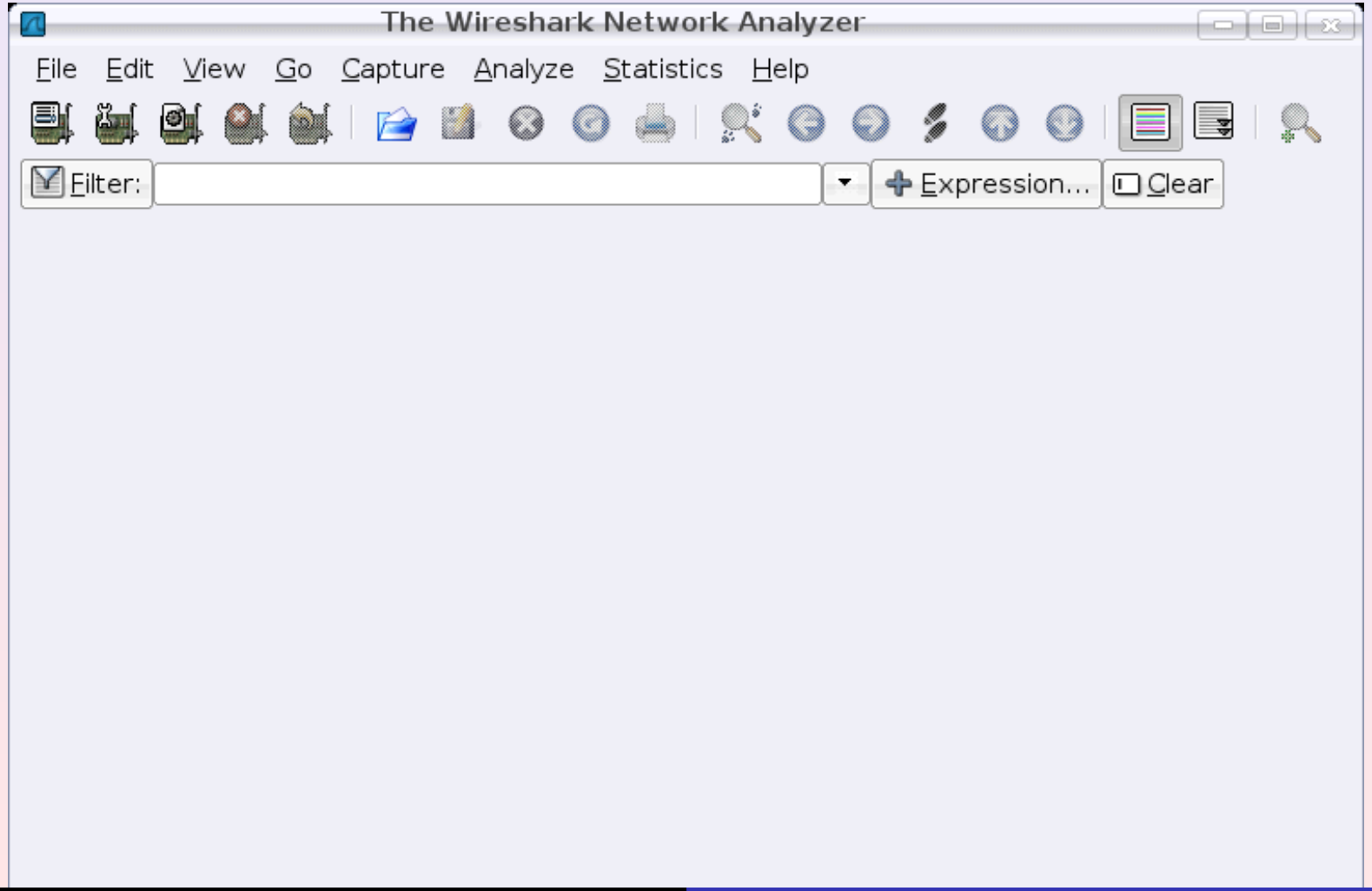
- **-i** : force l'utilisation d'une interface (par exemple : -i eth0)

```
...
```

- **man tcpdump** : pour obtenir plus d'information sur les différentes options de tcpdump.

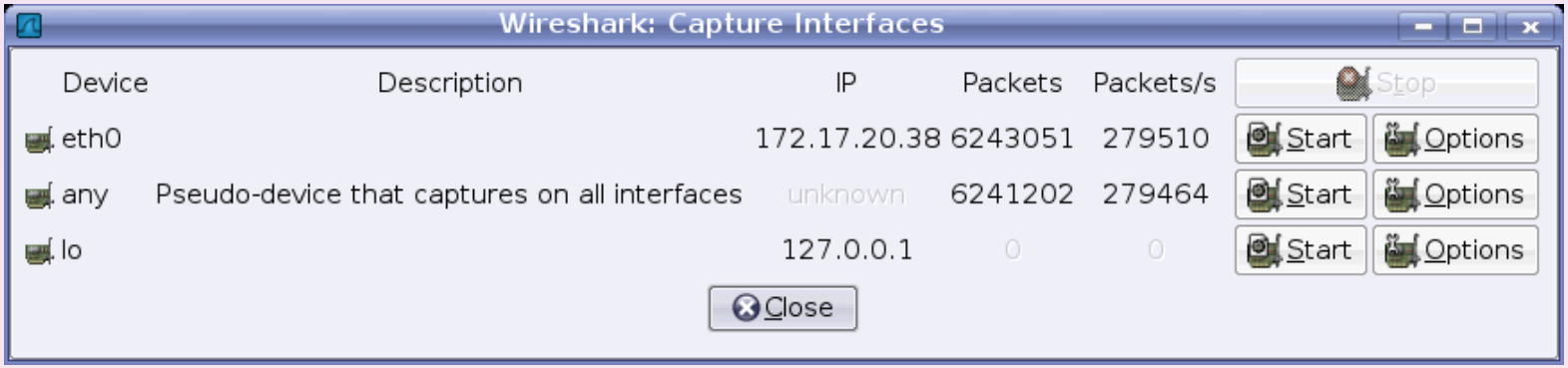
Lancement de Wireshark

Au moment du lancement de Wireshark, vous obtenez l'écran suivant :



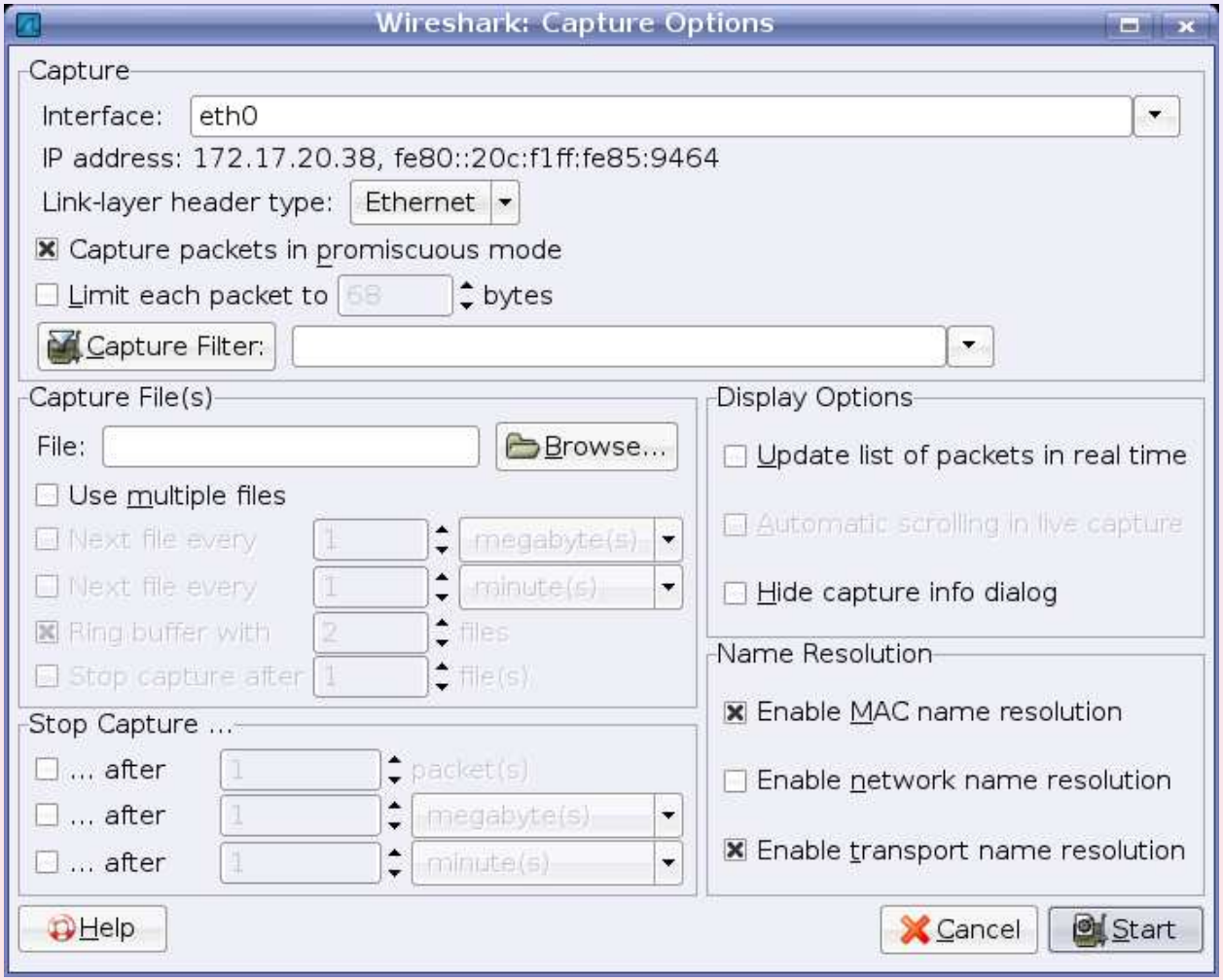
Choix de l'interface de capture

Pour démarrer la capture, il vous faut sélectionner d'abord une interface dans le menu `Capture/Interface`.



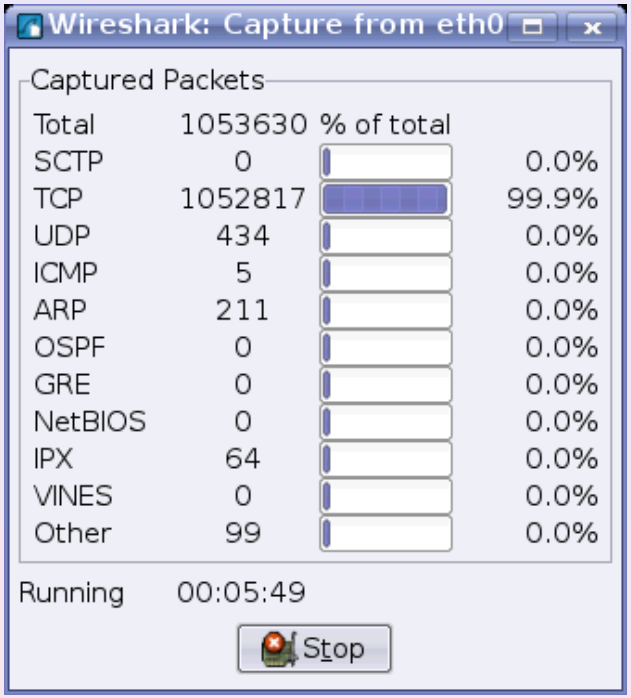
Options de capture

Vous pouvez modifier les options de capture en pressant le bouton **Option** :

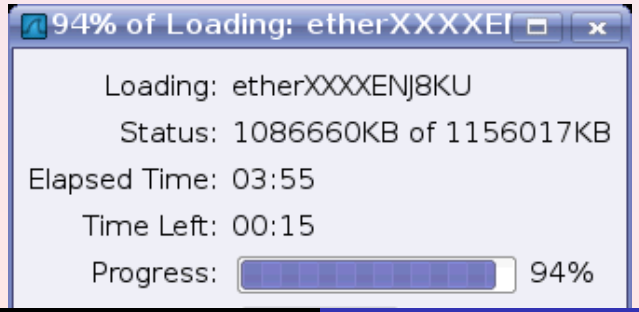


Lancement de la capture

Pour lancer la capture, il suffit de presser **Start** :

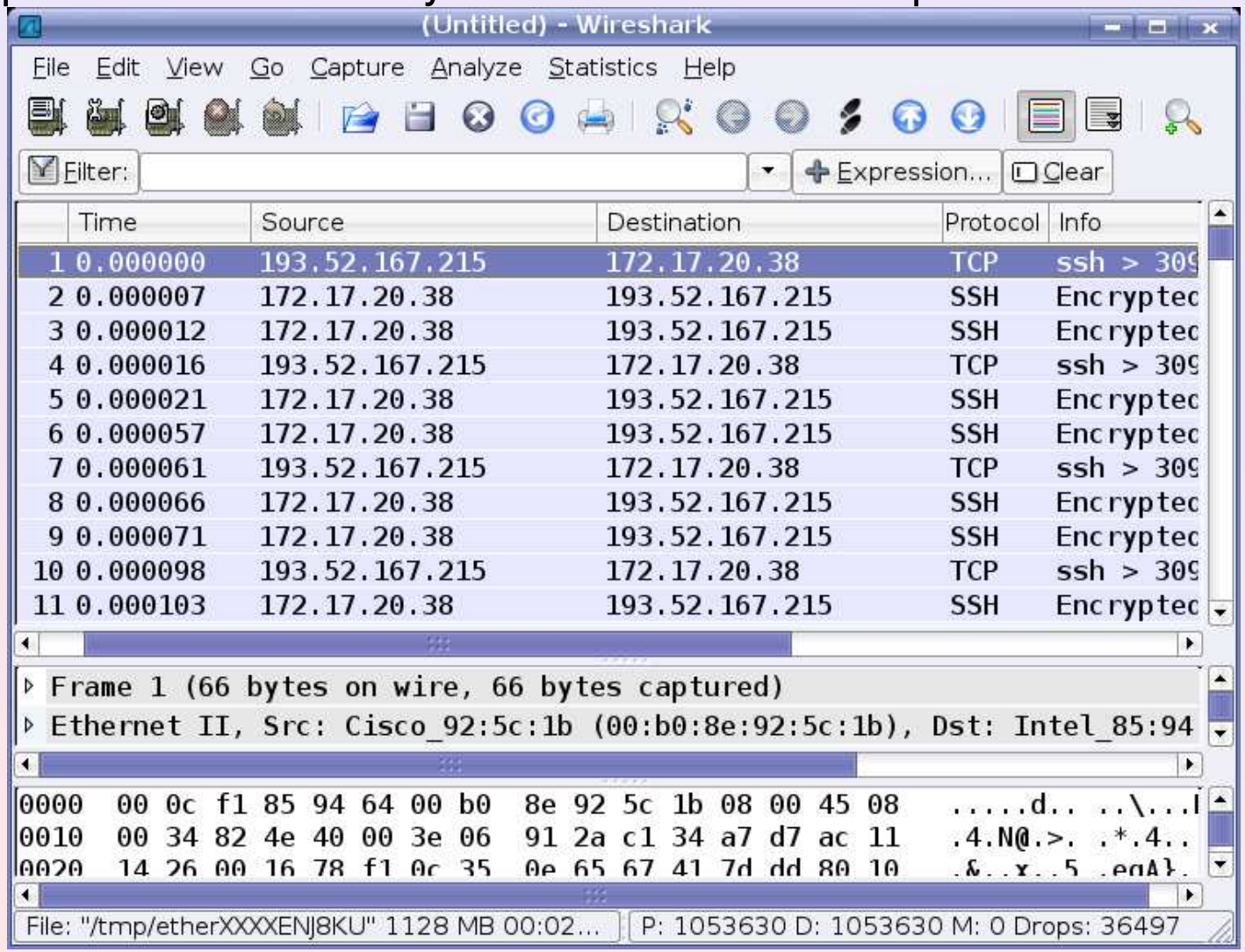


Pour arrêter la capture et charger celle-ci dans Wireshark, il suffit de presser **Stop** :



Analyse de la capture

Vous pouvez ensuite analyser le résultat de la capture :



Installation de Snort

- Installer les paquets Snort.
- Installer le support mysql pour Snort.
- Installer ACID.
- Configurer Snort.
- Simuler une attaque avec nmap.
- Analyser les logs.