Capturer le trafic réseau au niveau utilisateur avec Wireshark

Philippe Latu
philippe.latu(at)inetdoc.net

https://www.inetdoc.net

Résumé

Cet article est un plagiat éhonté de deux autres billets publiés par Jeremy Stretch sur le blog du site PacketLife.net et par Gerald Combs sur le blog du site Wireshark.org. Le contenu de ces billets est tellement intéressant que je me permets de les reprendre ici avec quelques adaptations vraiment mineures.

Table des matières

1. Copyright et Licence

```
Copyright (c) 2000,2024 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included
in the section entitled "GNU Free Documentation License".
```

```
Copyright (c) 2000,2024 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon
les termes de la Licence de Documentation Libre GNU (GNU Free Documentation
License), version 1.3 ou toute version ultérieure publiée par la Free Software
Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture,
et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est
incluse dans la section intitulée « Licence de Documentation Libre GNU ».
```

Méta-information

Cet article est écrit avec DocBook XML sur un système Debian GNU/Linux. Il est disponible en version imprimable au format PDF : `wireshark-as-user.pdf`.

2. Pourquoi capturer le trafic au niveau utilisateur ?

Pour effectuer une capture de trafic au niveau le plus proche possible du matériel, il faut disposer de droits d'accès particuliers sur les interfaces réseau. Historiquement, cette contrainte impose d'exécuter le programme Wireshark avec les droits du super-utilisteur. Comme il s'agit d'une application graphique assez lourde, il est justement vivement déconseillé de l'utiliser au niveau super-utilisteur ou root. Voilà, les termes de la contradiction sont posés et trouver un moyen d'exécuter l'analyseur au niveau utilisateur devient un exercice très intéressant.

Un premier niveau de réponse assez répandu consiste à utiliser l'outil sudo qui attribue les droits du super-utilisateur «à l'acte» pour une application donnée. C'est ce que j'ai proposé dans le support Introduction à l'analyse réseau. Ce n'est pas vraiment satisfaisant dans la mesure où on a toujours recours au niveau super-utilisteur.

Sur un système GNU/Linux comme sur tout autre système Unix, la règle de conception fondamentale veut que tous les droits d'accès soient basés sur les fichiers puisque l'on accède à tous les éléments du système comme s'ils étaient des fichiers. Depuis quelques années, une «méthode d'accès» baptisée

Linux Capabilities a été développée pour introduire davantage de granularité dans le «monde binaire» des droits d'accès limité aux deux seuls niveaux utilisateur et super-utilisateur.

3. Les aptitudes du système GNU/Linux

Au niveau du système Unix, les propriétés d'un utilisateur particulier sont représentées sous forme de fichiers marqués par un identifiant numérique de propriété appelé UID. Un système Linux gère de la même façon les accès au matériel via un niveau d'abstraction. Il offre ainsi un environnement dans lequel les programmes s'exécutent. Une partie du niveau d'abstraction fait respecter les droits de propriété des données. Pour respecter les droits des propriétaires des données en question, Linux se conforme à des règles spécifiques qui limitent les manipulations faites par les programmes sur les données. Les appels système (system-calls) effectués par les programmes utilisent des attributs tels que l'identifiant du propriétaire (UID).

Pour débuter l'exécution de programmes, un utilisateur passe par une authentification (login, ssh, etc.). À la suite de cette authentification, l'utilisateur dispose d'un contexte de travail pour appeler d'autres programmes et manipuler ses données. Les fonctions apportées par les Linux Capabilities ont justement pour but de modifier ce contexte de travail qui était jusque là figé après l'authentification.

Sans fonction offrant la capacité à modifier le contexte de travail d'un utilisateur, le seul recours possible est l'exécution en mode privilégié (avec l'UID 0) qui passe outre tous les contrôles de permissions du noyau. C'est la façon dont sont exécutés les programmes à l'aide de l'outil sudo.

Plus d'une trentaine d'***aptitudes*** ont été ajoutées au noyau Linux. Ces fonctions interviennent sur presque tous les aspects du contexte de travail de l'utilisateur. Pour la capture de trafic réseau, seules deux fonctions nous intéressent.

CAP_NET_ADMIN

> Effectuer différentes opérations réseau comme : obtenir des options privilégiées sur les sockets, activer le multicast, configurer les interfaces, modifier les tables de routage, etc.
>
> Avec Wireshark, on doit modifier l'état de l'interface de capture et la placer en mode promiscuous de façon à traiter toutes les trames vues par cette interface.

CAP_NET_RAW

> Utiliser des sockets RAW et PACKET.
>
> Avec Wireshark, on doit capturer le trafic au plus près de l'interface.

Le document Linux Capabilities: making them work donne des informations beaucoup plus détaillées sur les aptitudes du noyau Linux.

Comme tous les outils de ce genre sur les systèmes GNU/Linux, les fonctions sont réparties entre l'espace noyau (kernelspace) et l'espace utilisateur (userspace). On suppose que les Linux Capabilities sont disponibles au niveau du noyau. C'est vrai pour toutes les distributions contemporaines. Pour ce qui est des outils utilisateurs, on doit contrôler leur installation. Ici, il s'agit du paquet `libcap2-bin`.

```
$ aptitude search ~ilibcap2-bin
i   libcap2-bin   - basic utility programs for using capabilities
```

4. Configuration de la capture de trafic au niveau utilisateur

Debian GNU/Linux

Les paquets de la distribution Debian GNU/Linux intègrent cette fonctionnalité de délégation des droits de capture de paquets. Pour l'activer, il suffit de reconfigurer le paquet wireshark-common.

```
# dpkg-reconfigure wireshark-common
```

Comme indiqué dans la copie d'écran ci-dessus, l'utilisateur doit appartenir au groupe système `wireshark` pour bénéficier de la fonctionnalité. Par exemple, l'ajout de l'utilisateur `etu` au groupe via la commande adduser donne le résultat suivant :

```
# adduser etu wireshark
Ajout de l'utilisateur « etu » au groupe « wireshark »...
Ajout de l'utilisateur etu au groupe wireshark
Fait.
```

Lors de la connexion suivante avec ce compte utilisateur il sera possible d'utiliser directement les outils wireshark ou tshark.

Détail des manipulations

1. Création d'un groupe système dédié à la capture de trafic réseau.

   ```
   # addgroup --system pcap
   Adding group `pcap' (GID 136) ...
   Done.
   ```

2. Ajout d'un utilisateur au groupe système.

   ```
   # adduser phil pcap
   Adding user `phil' to group `pcap' ...
   Adding user phil to group pcap
   Done.
   ```

   > **Avertissement**
   >
   > Cette nouvelle attribution n'est valable qu'après une nouvelle authentification. Nous sommes encore dans le cas classique de l'évaluation du contexte de travail utilisateur au moment de l'authentification.

3. Modification des propriétés du programme dumpcap.

   Avant modification du groupe propriétaire :

   ```
   # ls -lh `which dumpcap`
   -rwxr-xr-x 1 root root 62K  4 mars  18:04 /usr/bin/dumpcap
   ```

   Changement de groupe propriétaire et nouveau masque de permission. Une fois cette opération faite, les membres du groupe système `pcap` seront les seuls utilisateurs à pouvoir exécuter le programme en mode non privilégié.

   ```
   # chgrp pcap /usr/bin/dumpcap
   # chmod 750 /usr/bin/dumpcap
   # ls -lh /usr/bin/dumpcap
   -rwxr-x--- 1 root pcap 62K  4 mars  18:04 /usr/bin/dumpcap
   ```

   Mémorisation de ces nouvelles propriétés par le gestionnaire de paquets Debian.

   ```
   # dpkg-statoverride --add root pcap 750 /usr/bin/dumpcap
   # dpkg-statoverride --list /usr/bin/dumpcap
   root pcap 750 /usr/bin/dumpcap
   ```

   Avec cette dernière manipulation, toutes les mises à jour de paquets conserveront les changements de propriétés du programme en l'état.

4. Modification du contexte de travail pour le programme `dumpcap`.

   ```
   # setcap cap_net_raw,cap_net_admin=eip /usr/bin/dumpcap
   # getcap /usr/bin/dumpcap
   /usr/bin/dumpcap = cap_net_admin,cap_net_raw+eip
   ```

   Les bits `eip` correspondent aux attributs effective, inheritable et permitted.

   Avec l'attribut effective, le noyau ne vérifie pas si l'UID vaut 0 (mode privilégié) si le programme nécessite une opération en mode privilégié.

   L'attribut inheritable transmet les aptitudes du processus actuel aux autres processus enfants.

   L'attribut permitted indique que le processus peut utiliser les aptitudes étendues du noyau Linux.

   La documentation sur les Linux Capabilities est disponible à partir de la page Not needing root to administer Linux.

Il ne reste plus qu'à tester les nouvelles fonctionnalités attribuées à l'utilisateur non privilégié membre du groupe système `pcap`.

```
phil@0wnB0x:~$ id
uid=1000(phil) gid=1000(phil) groupes=1000(phil),4(adm),20(dialout),<snip/>,128(tftp),136(pcap)
phil@0wnB0x:~$ tshark -i wlan0
Capturing on wlan0
<snip>
  0.577380  192.168.1.9 -> 192.168.1.1  SSH Encrypted response packet len=352
  0.578481  192.168.1.1 -> 192.168.1.9  TCP 46722 > ssh [ACK] Seq=1 Ack=353 Win=30 Len=0 TSV=7711965 TS
  1.582427  192.168.1.9 -> 192.168.1.1  SSH Encrypted response packet len=240
  1.583462  192.168.1.1 -> 192.168.1.9  TCP 46722 > ssh [ACK] Seq=1 Ack=593 Win=33 Len=0 TSV=7712217 TS
</snip>
```

5. Pour conclure

On dispose maintenant de la capacité à capturer et analyser le trafic réseau en tant qu'utilisateur non privilégié. Cette modification du contexte de travail de l'utilisateur normal peut être particulièrement intéressante sur des postes de travaux pratiques sur lesquels on ne souhaite pas que les étudiants aient un accès au mode privilégié.

C'est aussi la première fois que j'ai eu l'occasion de voir une application concrète des Linux Capabilities même si le sujet n'a été qu'effleuré.