tcpdump

Scénario

Vous êtes un analyste de sécurité, et devez capturer et analyser en temps réel le trafic réseau d'une machine virtuelle Linux à l'aide de `tcpdump`.

Au début de l'atelier, votre compte utilisateur `analyst` est déjà connecté au terminal Linux.

Le répertoire d'accueil lié à votre compte utilisateur Linux comporte un exemple de fichier de capture de paquets que vous utiliserez à la fin de l'atelier pour répondre à quelques questions sur le trafic réseau qu'il contient.

Voici comment procéder : **premièrement**, identifiez les interfaces réseau pour capturer les données des paquets réseau. **Deuxièmement**, utilisez `tcpdump` pour filtrer le trafic réseau en temps réel. **Troisièmement**, capturez le trafic réseau avec `tcpdump`. **Enfin**, filtrez les données de paquets capturées.

Tâche 1 : Identifier les interfaces réseau

Dans cette tâche, vous allez identifier les interfaces réseau qui vous serviront à capturer les données des paquets réseau.

1. Utilisez `ifconfig` pour identifier les interfaces disponibles :

L'interface réseau Ethernet est identifiée par l'entrée portant le préfixe `eth`.

Dans les tâches suivantes de l'atelier, vous allez donc capturer les données des paquets réseau en utilisant l'interface `eth0`.

```
analyst@4eba2ae2fb83:~$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1460
        inet 172.18.0.2  netmask 255.255.0.0  broadcast 172.18.255.255
        ether 02:42:ac:12:00:02  txqueuelen 0  (Ethernet)
        RX packets 770  bytes 13968959 (13.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 434  bytes 39386 (38.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 100  bytes 12780 (12.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 100  bytes 12780 (12.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

analyst@4eba2ae2fb83:~$
```

2. Utilisez `tcpdump` afin d'identifier les options d'interface disponibles pour la capture des paquets :

```
analyst@4eba2ae2fb83:~$ sudo tcpdump -D
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
analyst@4eba2ae2fb83:~$
```

Tâche 2 : Inspecter le trafic réseau d'une interface réseau avec tcpdump

Dans cette tâche, vous allez filtrer en temps réel le trafic des paquets réseau sur une interface à l'aide de `tcpdump`.

- Filtrez en temps réel les données des paquets réseau de l'interface `eth0` avec `tcpdump` :

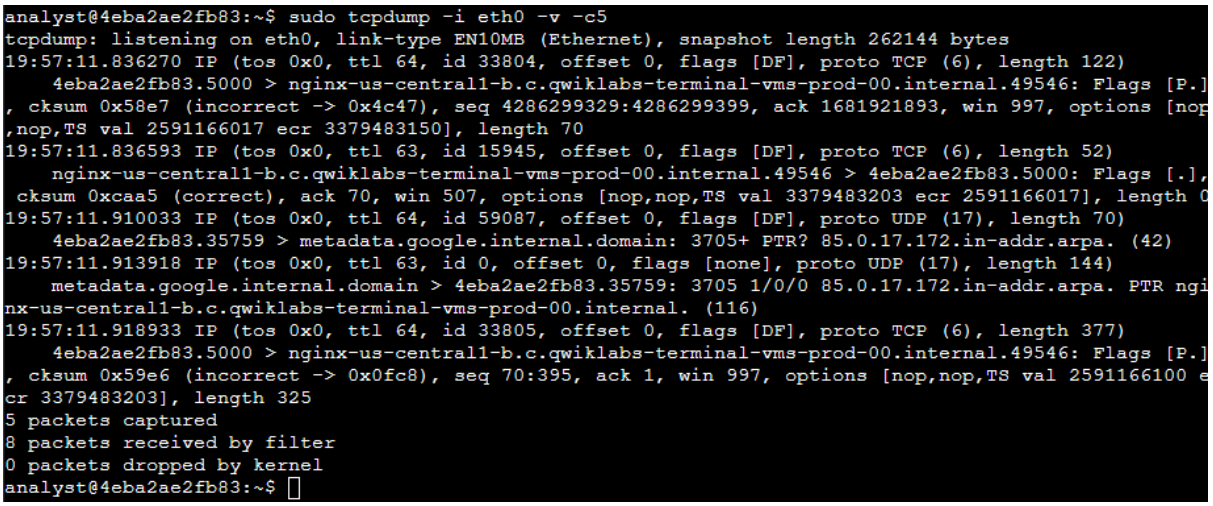
  La commande est la suivante : sudo tcpdump -i eth0 -v -c5

Cette commande exécute `tcpdump` avec les options suivantes :

- `-i eth0` : capture les données associées à l'interface `eth0`.
- `-v` : affiche les données détaillées des paquets.
- `-c5` : capture cinq paquets de données.

Explorer les détails des paquets réseau

Dans cet exemple, vous allez identifier certaines propriétés renvoyées par `tcpdump` pour les données de capture des paquets que nous venons de voir.



Dans l'exemple de données au début du résultat des paquets, il est précisé l'interface sur laquelle `tcpdump` écoute (`eth0`), ainsi que des informations sur le type de liaison et la taille de la capture en octets.

Sur la ligne suivante, le premier champ correspond au code temporel du paquet, suivi du type de protocole (IP) .

L'option de verbosité, `-v`, a fourni plus d'informations sur les champs du paquet IP, y compris les valeurs TOS et TTL, le décalage, les flags, le type de protocole interne (ici, TCP [6]) et la longueur du paquet IP externe en octets .

Dans la section suivante, les données indiquent les systèmes qui communiquent entre eux .

Les données restantes filtrent les données d'en-tête du paquet TCP interne

Le champ "flags" identifie les flags TCP. Dans cet exemple, la lettre "P" représente le flag push et le point signifie qu'il s'agit d'un flag ACK. Autrement dit, le paquet transfère des données.

Le champ suivant correspond à la somme de contrôle TCP, qui sert à détecter les erreurs dans les données.

Cette section inclut aussi les numéros de séquence et d'accusé de réception, la taille de la fenêtre et la longueur du paquet TCP interne en octets.

Tâche 3 : Capturer le trafic réseau avec tcpdump

Dans cette tâche, vous allez enregistrer les données réseau capturées dans un fichier de capture de paquets à l'aide de `tcpdump`.

Dans la commande précédente, vous avez filtré tout le trafic réseau avec `tcpdump`. Dans cette tâche, vous allez utiliser un filtre et d'autres options de configuration de `tcpdump` pour enregistrer un petit échantillon contenant uniquement les données des paquets réseau Web (TCP sur port 80).

1. Capturez les données des paquets dans un fichier `capture.pcap` :

La commande utilisée est la suivante :**sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &**

Cette commande exécute `tcpdump` en arrière-plan avec les options suivantes :

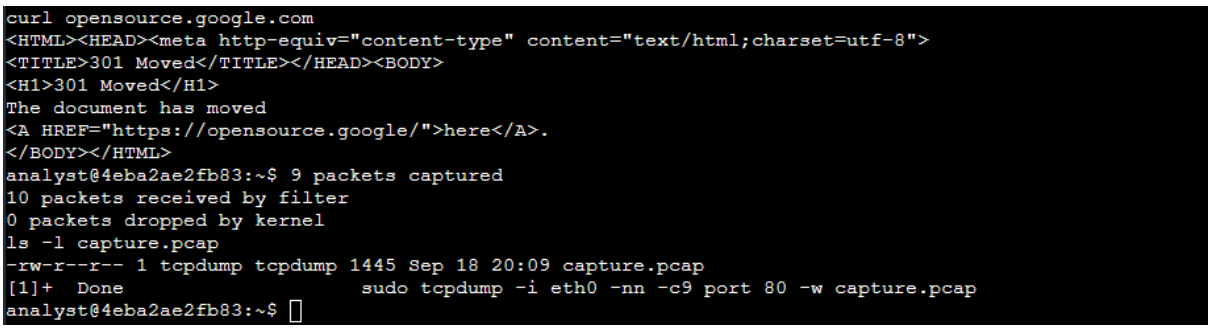
- `-i eth0` : capture les données de l'interface `eth0`.
- `-nn` : bloque la résolution des ports et adresses IP en noms. Cette pratique est conseillée en termes de sécurité, car les données recherchées ne sont peut-être pas valides. Cela évite aussi d'alerter des individus malintentionnés qu'une enquête est en cours.
- `-c9` : capture neuf paquets de données, puis quitte.
- `port 80` : filtre uniquement le trafic sur le port 80 (port HTTP par défaut).
- `-w capture.pcap` : enregistre les données capturées dans le fichier indiqué.
- `&` : demande au shell bash d'exécuter la commande en arrière-plan.

```
analyst@4eba2ae2fb83:~$ sudo tcpdump -i eth0 -nn -c9 port 80 -w capture.pcap &
[1] 13516
analyst@4eba2ae2fb83:~$ tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 26214
4 bytes
```

2. Utilisez `curl` pour générer du trafic HTTP (port 80) :

Lorsque la commande `curl` est utilisée de la sorte pour ouvrir un site Web, elle génère du trafic HTTP (TCP sur port 80) qui peut être capturé.

3. Vérifiez que les données des paquets ont été capturées.



Tâche 4 : Filtrer les données de paquets capturées

Dans cette tâche, vous allez filtrer à l'aide de `tcpdump` les données du fichier de capture de paquets enregistré précédemment.

1. Utilisez la commande `tcpdump` pour filtrer les données d'en-tête des paquets dans le fichier de capture `capture.pcap` :

La commande utilisée est : sudo tcpdump -nn -r capture.pcap -v

Cette commande exécute `tcpdump` avec les options suivantes :

- `-nn` : désactive la recherche des noms de port et de protocole.
- `-r` : lit les données de capture dans le fichier indiqué.
- `-v` : affiche les données détaillées des paquets.

Vous devez de nouveau indiquer l'option `-nn` pour vous assurer que `tcpdump` ne recherche pas de noms d'adresses IP ni de ports, ce qui risquerait d'alerter les individus malintentionnés.

Les données renvoyées se présentent comme suit :

```
analyst@4eba2ae2fb83:~$ sudo tcpdump -nn -r capture.pcap -v
reading from file capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
20:09:23.428159 IP (tos 0x0, ttl 64, id 11563, offset 0, flags [DF], proto TCP (6), length 60)
    172.18.0.2.49498 > 172.217.214.101.80: Flags [S], cksum 0x2f82 (incorrect -> 0xad26), seq 52037111
1, win 65320, options [mss 1420,sackOK,TS val 3731059148 ecr 0,nop,wscale 6], length 0
20:09:23.428775 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    172.217.214.101.80 > 172.18.0.2.49498: Flags [S.], cksum 0xf79c (correct), seq 1854070124, ack 520
371112, win 65535, options [mss 1420,sackOK,TS val 2843262775 ecr 3731059148,nop,wscale 8], length 0
20:09:23.428802 IP (tos 0x0, ttl 64, id 11564, offset 0, flags [DF], proto TCP (6), length 52)
    172.18.0.2.49498 > 172.217.214.101.80: Flags [.], cksum 0x2f7a (incorrect -> 0x2244), ack 1, win 1
021, options [nop,nop,TS val 3731059149 ecr 2843262775], length 0
20:09:23.428869 IP (tos 0x0, ttl 64, id 11565, offset 0, flags [DF], proto TCP (6), length 137)
    172.18.0.2.49498 > 172.217.214.101.80: Flags [P.], cksum 0x2fcf (incorrect -> 0x8ff7), seq 1:86, a
ck 1, win 1021, options [nop,nop,TS val 3731059149 ecr 2843262775], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.74.0
        Accept: */*

20:09:23.429216 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 52)
    172.217.214.101.80 > 172.18.0.2.49498: Flags [.], cksum 0x21d0 (correct), ack 86, win 1051, option
s [nop,nop,TS val 2843262776 ecr 3731059149], length 0
20:09:23.431855 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6), length 634)
    172.217.214.101.80 > 172.18.0.2.49498: Flags [P.], cksum 0xf86a (correct), seq 1:583, ack 86, win
1051, options [nop,nop,TS val 2843262779 ecr 3731059149], length 582: HTTP, length: 582
        HTTP/1.1 301 Moved Permanently
        X-Content-Type-Options: nosniff
        Cross-Origin-Resource-Policy: cross-origin
        Cache-Control: public, max-age=1800
        Expires: Thu, 18 Sep 2025 20:39:23 GMT
        Content-Type: text/html; charset=UTF-8
        Location: https://opensource.google/
        Date: Thu, 18 Sep 2025 20:09:23 GMT
        Server: sffe
        Content-Length: 223
        X-XSS-Protection: 0

        <HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
        <TITLE>301 Moved</TITLE></HEAD><BODY>
        <H1>301 Moved</H1>
        The document has moved
```

2. Utilisez la commande `tcpdump` pour filtrer les données étendues des paquets dans le fichier de capture `capture.pcap`.

La commande utilisée est : sudo tcpdump -nn -r capture.pcap -X

Cette commande exécute `tcpdump` avec les options suivantes :

- `-nn` : désactive la recherche des noms de port et de protocole.
- `-r` : lit les données de capture dans le fichier indiqué.
- `-X` : affiche les données de paquets renvoyées au format hexadécimal et ASCII. Les analystes de sécurité peuvent analyser le résultat au format hexadécimal et ASCII afin de détecter des schémas ou anomalies dans le cadre d'une analyse des logiciels malveillants ou d'une analyse forensique.

```
analyst@4eba2ae2fb83:~$ sudo tcpdump -nn -r capture.pcap -X
reading from file capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
20:09:23.428159 IP 172.18.0.2.49498 > 172.217.214.101.80: Flags [S], seq 520371111, win 65320, options
 [mss 1420,sackOK,TS val 3731059148 ecr 0,nop,wscale 6], length 0
        0x0000:  4500 003c 2d2b 4000 4006 de3d ac12 0002  E..<-+@.@..=....
        0x0010:  acd9 d665 c15a 0050 1f04 3ba7 0000 0000  ...e.Z.P..;.....
        0x0020:  a002 ff28 2f82 0000 0204 058c 0402 080a  ...(/...........
        0x0030:  de63 71cc 0000 0000 0103 0306            .cq.........
20:09:23.428775 IP 172.217.214.101.80 > 172.18.0.2.49498: Flags [S.], seq 1854070124, ack 520371112, w
in 65535, options [mss 1420,sackOK,TS val 2843262775 ecr 3731059148,nop,wscale 8], length 0
        0x0000:  4500 003c 0000 4000 7e06 cd68 acd9 d665  E..<..@.~..h...e
        0x0010:  ac12 0002 0050 c15a 6e82 dd6c 1f04 3ba8  .....P.Zn..l..;.
        0x0020:  a012 ffff f79c 0000 0204 058c 0402 080a  ................
        0x0030:  a978 bf37 de63 71cc 0103 0308            .x.7.cq.....
20:09:23.428802 IP 172.18.0.2.49498 > 172.217.214.101.80: Flags [.], ack 1, win 1021, options [nop,nop
,TS val 3731059149 ecr 2843262775], length 0
        0x0000:  4500 0034 2d2c 4000 4006 de44 ac12 0002  E..4-,@.@..D....
        0x0010:  acd9 d665 c15a 0050 1f04 3ba8 6e82 dd6d  ...e.Z.P..;.n..m
        0x0020:  8010 03fd 2f7a 0000 0101 080a de63 71cd  ..../z.......cq.
        0x0030:  a978 bf37                                .x.7
20:09:23.428869 IP 172.18.0.2.49498 > 172.217.214.101.80: Flags [P.], seq 1:86, ack 1, win 1021, optio
ns [nop,nop,TS val 3731059149 ecr 2843262775], length 85: HTTP: GET / HTTP/1.1
        0x0000:  4500 0089 2d2d 4000 4006 ddee ac12 0002  E...--@.@.......
        0x0010:  acd9 d665 c15a 0050 1f04 3ba8 6e82 dd6d  ...e.Z.P..;.n..m
        0x0020:  8018 03fd 2fcf 0000 0101 080a de63 71cd  ..../........cq.
        0x0030:  a978 bf37 4745 5420 2f20 4854 5450 2f31  .x.7GET./.HTTP/1
        0x0040:  2e31 0d0a 486f 7374 3a20 6f70 656e 736f  .1..Host:.openso
        0x0050:  7572 6365 2e67 6f6f 676c 652e 636f 6d0d  urce.google.com.
        0x0060:  0a55 7365 722d 4167 656e 743a 2063 7572  .User-Agent:.cur
        0x0070:  6c2f 372e 3734 2e30 0d0a 4163 6365 7074  l/7.74.0..Accept
        0x0080:  3a20 2a2f 2a0d 0a0d 0a                   :.*/*....
20:09:23.429216 IP 172.217.214.101.80 > 172.18.0.2.49498: Flags [.], ack 86, win 1051, options [nop,no
p,TS val 2843262776 ecr 3731059149], length 0
        0x0000:  4500 0034 0000 4000 7e06 cd70 acd9 d665  E..4..@.~..p...e
        0x0010:  ac12 0002 0050 c15a 6e82 dd6d 1f04 3bfd  .....P.Zn..m..;.
        0x0020:  8010 041b 21d0 0000 0101 080a a978 bf38  ....!........x.8
        0x0030:  de63 71cd                                .cq.
20:09:23.431855 IP 172.217.214.101.80 > 172.18.0.2.49498: Flags [P.], seq 1:583, ack 86, win 1051, opt
ions [nop,nop,TS val 2843262779 ecr 3731059149], length 582: HTTP: HTTP/1.1 301 Moved Permanently
        0x0000:  4500 027a 0000 4000 7e06 cb2a acd9 d665  E..z..@.~..*...e
        0x0010:  ac12 0002 0050 c15a 6e82 dd6d 1f04 3bfd  .....P.Zn..m..;.
        0x0020:  8018 041b f86a 0000 0101 080a a978 bf3b  .....j.......x.;
        0x0030:  de63 71cd 4854 5450 2f31 2e31 2033 3031  .cq.HTTP/1.1.301
        0x0040:  204d 6f76 6564 2050 6572 6d61 6e65 6e74  .Moved.Permanent
        0x0050:  6c79 0d0a 582d 436f 6e74 656e 742d 5479  ly..X-Content-Ty
        0x0060:  7065 2d4f 7074 696f 6e73 3a20 6e6f 736e  pe-Options:.nosn
        0x0070:  6966 660d 0a43 726f 7373 2d4f 7269 6769  iff..Cross-Origi
        0x0080:  6e2d 5265 736f 7572 6365 2d50 6f6c 6963  n-Resource-Polic
```