Un exemple de panne amusante de tcpdump

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 août 2010

https://www.bortzmeyer.org/tcpdump-filtrage.html

Voici un cas rigolo qui m'est arrivé hier. Soit une machine Debian/Linux où les programmes de capture de paquets (comme tcpdump) tout marchaient autrefois. Tout à coup, plus aucun filtre BPF ne trouve un seul paquet...

tcpdump sans argument donne le résultat attendu :

```
% sudo tcpdump -i eth1 -n
12:36:40.723290 IP 193.243.207.47.31619 > 192.93.0.129.53: 40691 [1au] MX? doublev.fr. (39)
12:36:40.723440 IP 192.93.0.129.53 > 193.243.207.47.31619: 40691- 0/2/3 (112)
12:36:40.723480 IP 212.96.9.225.2123 > 192.93.0.129.53: 46420+ MX? fema.fr. (25)
12:36:40.723634 IP 192.93.0.129.53 > 212.96.9.225.2123: 46420- 0/2/0 (81)
...
```

Mais aucun filtre BPF ne trouve ne serait-ce qu'un seul paquet. Dès que je mets comme argument `port 53`, ou `udp` ou même `ip`, je n'ai plus aucun paquet. Par contre, les filtres `not ip` ou bien `not port NN` (pour n'importe quelle valeur de NN) me montrent le trafic... Le problème affecte tous les programmes utilisant la libpcap, pas seulement tcpdump.

La machine a deux interfaces dont une seule a ce problème (sur l'autre, même puce donc même "*driver*", tcpdump marche bien). L'interface à problème n'est pas utilisée par la machine elle-même, elle reçoit du trafic uniquement par "*port mirroring*".

Si j'enregistre les paquets avec l'option `-w` et que je tente de relire avec un filtre, zéro réponse :

```
% tcpdump -n -r /tmp/tcpdump-eth1.pcap | wc -l
reading from file /tmp/tcpdump-eth1.pcap, link-type EN10MB (Ethernet)
5722

% tcpdump -n -r /tmp/tcpdump-eth1.pcap ip | wc -l
reading from file /tmp/tcpdump-eth1.pcap, link-type EN10MB (Ethernet)
0
```

Utiliser l'option `-O` (ne pas optimiser le code BPF) ne change rien. Vider les caches <http://www.linuxinsight.com/proc_sys_vm_drop_caches.html> ou même redémarrer la machine, au cas où un rayon cosmique aie corrompu la mémoire ne change rien non plus.

Arrivé là, voyez si vous avez trouvé la solution...

Mon collègue Antonio Kin-Foo a trouvé. La configuration réseau avait été refaite et le commutateur Ethernet Foundry trouvait drôle de désormais "*taguer*" les paquets. C'est ce que montrait tcpdump si on utilise l'option `-e` (afficher la couche 2). Au lieu de « ethertype IPv4 (0x0800) » pour des paquets IPv4, on voyait « ethertype 802.1Q (0x8100) ». Lorsque les paquets sont ainsi "*tagués*", tcpdump sait se débrouiller dans certains cas (il formatait correctement les paquets) mais pas dans d'autres (définition du filtre). D'autre part, si l'encapsulation/décapsulation VLAN est gérée par le matériel de la carte ou par le système d'exploitation, parfois tcpdump n'a plus de problèmes puisqu'il ne voit plus le "*tag*" 802.1Q.

Le bon moyen est de rajouter pour réparer le filtre est d'ajouter `vlan and`. Par exemple, pour les paquets DNS :

```
tcpdump -n -i eth1 vlan and port 53
```

J'ai dû aussi modifier le code de l'outil que j'utilisais pour analyser les paquets. Quelque chose du genre (en C) :

```
        size_layer2 = SIZE_ETHERNET;
        ethernet = (struct sniff_ethernet *) (packet);
        ethertype = ntohs(ethernet->ether_type);
        if (ethertype == VLAN_ETHERTYPE) {  /* NEW: if the packets are
tagged, move four bytes forward */
            packet += 4;
            ethernet = (struct sniff_ethernet *) (packet);
            ethertype = ntohs(ethernet->ether_type);
        }
        if (ethertype == IPv6_ETHERTYPE) {
            ip_version = 6;
        } else if (ethertype == IPv4_ETHERTYPE) {
            ip_version = 4;
        } else {                /* Ignore other Ethernet types */
            goto next_packet;
        }
```