Sicherheitsrisiko in der libpcap
Wann sind tcpdump, snort und wireshark blind?

Dr. Alexander Schinner

Abstract

Die libpcap ist die Basis vieler verschiedener Werkzeuge zur Überwachung, Diagnose und zum Schutz von Netzwerken. Ein Fehler in der Implementierung der Analyse von VLANs (802.1q) führt dazu, dass diese Programme wichtigen Netzwerkverkehr nicht erhalten.

1 libpcap

Die libpcap ist eine Bibliothek, die das Mitschneiden des vollständigen Netzwerkverkehrs ermöglicht. Sie stellt dabei eine komfortable Verbindung (Abb.1) zwischen den Netzwerkkartentreibern und den Anwenderprogrammen her. Zusätzlich bietet sie eine Pufferung sowie eine Filterung, die sogenannten Berkeley Packet Filter (BPF)

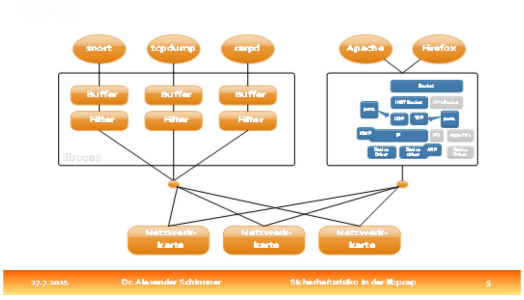

Abb. 1: Grundarchitektur libpcap

2 Berkeley Packet Filter

Die Filterung von Netzwerkpaketen durch eine einfach zu formulierende Sprache hat sicherlich zur weiten Verbreitung der libpcap geführt. Anstatt umständlich die Analyse der einzelnen Paketheader per Hand durchzuführen, kann der Netzwerkadministrator sich über den einfachen Befehl

```
root@host:~# tcpdump -n -l host 1.2.3.4
```

alle Pakete anzeigen lassen, die für den Rechner mit der IP-Adresse 1.2.3.4 bestimmt sind bzw. von ihm kommen. Bedenkt man, dass die Header eines Paketes unterschiedliche Längen haben können, bzw. dass IP-Adressen auch in anderen, als dem IP-Protokoll vorkommen können, wird klar, dass diese Filter nicht über einfache Zugriffe auf feste Adressen realisiert werden können.

Stattdessen wird diese Filterung durch die BPF Pseudo Maschine realisiert. Dies ist eine virtuelle CPU, die mit einem sehr beschränkten Befehlssatz mit kleinen Programmen die Analyse der Netzwerkpakete durchführt. Über die Option „`-d`“ von tcpdump kann man sich z.B. das Programm für den Filter „`host 1.2.3.4`“ anzeigen lassen:

```
tcpdump -d host 1.2.3.4
(000) ldh      [12]
(001) jeq      #0x800       jt 2    jf 6
usw…
```

Die ersten beiden Zeilen laden zwei Byte beginnend beim Offset 12 des Ethernetpaketes und vergleichen sie mit dem Wert 0x800. Dies ist nichts anderes, als die Prüfung, ob ein IP-Paket vorliegt. Die weitere Filterung erfolgt dann analog.

3 VLAN

Die Verbreitung von virtuellen LANs (802.1q) wächst im Rechenzentrumsumfeld immer mehr. Einerseits kann man damit Kabel und Switchports sparen, andererseits sind sie beim Einsatz von Virtualisierungstechniken fast unumgänglich geworden. Realisiert werden VLANs durch einen zusätzlichen 4 Byte großen Header, der auf den Ethernetheader folgt. Dieser enthält die VLAN-ID und die Protokollnummer des folgenden Headers. Prüfen kann man diese Header über den Berkeley Packet Filter

```
vlan [vlanid].
```

4 VLAN-Bug

Analysiert man Netzwerkverkehr, der Pakete mit und ohne VLAN transportiert, fällt auf, dass die Filterung auf einen bestimmten Port Pakete mit VLAN-IDs ignoriert. Verwirrender wird es dadurch, dass sich die Ausgaben

```
tcpdump -d vlan 32 or port 6000
```

und

```
tcpdump -d port 6000 ov vlan 32
```

deutlich voneinander unterscheiden, obwohl sie von der formalen Logik her identisch sein sollten. Der erste Aufruf zeigt alle Pakete im VLAN 32 und alle Pakete, die in irgendeinem VLAN sind und Port 6000 haben. Der zweite Aufruf hingegen gibt alle Pakete mit Port 600 aus, die in keinem VLAN sind und alle Pakete für VLAN 32. Eine Analyse der BPF zeigt:

```
port 6000                          vlan 32 or port 6000
   …                                  …
ldxb 4*([14]&0xf)                  ldxb 4*([14]&0xf)
ldh  [x + 14]                      ldh  [x + 18]
jeq  #0x1770  jt 22 jf 20   ⟺      jeq  #0x1770  jt 22 jf 20
ldh  [x + 16]                      ldh  [x + 20]
jeq  #0x1770  jt 22 jf 23   ⟺      jeq  #0x1770  jt 22 jf 23
   …                                  …
```

Über `ldxb 4*([14]&0xf)` wird die Größe des TCP-Header berechnet und in `x` gespeichert. Somit ist der Sourceport ohne VLAN-ID bei `[x+14]`, mit aber bei `[x+18]`. Dies bedeutet, dass, sobald der Parser das Keyword `vlan` sieht, grundsätzlich angenommen wird, dass der Ethernetheader um 4 Byte größer ist; die Offsets für IP und TCP-Header werden entsprechend berechnet. Dieses führt dazu, dass Analysesysteme nur noch unvollständigen oder falschen Netzwerkverkehr geliefert bekommen.

Literaturverzeichnis

[1] *TCP/IP*, W. Richard Stevens

[2] *The BSD Packet Filter: A New Architecture for User-level Packet Capture*; Steven McCanne & Van Jacobson, Usenix 93