Pcap

Dataformaat

Een pcap-file bestaat uit een "gobal-header" en achtereen volgens een "packet-header" en de "packet-data". De "global header" specificeert o.a. de versie en heeft een magic-nuber om het file-type te herkennen. Zie ook: https://wiki.wireshark.org/Development/LibpcapFileFormat

hva_pcap.py

`regel 13-16` We gebruiken een `named-tuple` om gemakelijk de headers te specifieren. Het formaat beschrijft de grootte van de velden en met `calcsize` wordt de lengte van headers in bytes bepaald.

open_offline

De routine `open_offline` lees een pcap-file en levert steeds een nieuwe header (hdr) en packet (pkt) op. Deze routine is a generator door het gebruik van `yield` .

`regel 19` Het betreft een binary file, dus we open de file read-only(`r`) in binary mode(`b`). Het `with` statement zorgt er voor dat als de suite afgelopen is de file netjes gesloten wordt.

`regel 20-21` We lezen de bytes van de "global-header", is niet het verwachte aantal gelezen, dan gooien we een exceptie.

`regel 22-24` We decoderen de bytes en dat levert een lijst van waarden. De `*_glbHdr` zorgt er van dat deze lijst als argumenten aan `GblHdr` wordt doorgeven en vervolgens wordt het `named-tuple` initialiseerd. We controleren of we deze pcap-file kunnen lezen door de `assert` met magic en versie.

`regel 27-29` We proberen de volgende "packet-header" te lezen. Indien geen bytes gelezen kunnen worden, zijn we aan het einde van de file en stopt de generator door de `return` . Als we niet voldoende kunnen lezen gooien we een exceptie.

`regel 30-32` Net als bij de "global-header" decodeeren we de bytes en creeren een named-tuple `PktHdr` . Echter we vinden het handig om het tijdstip van het packet ook als een `datetime` te hebben. We creeren een `datetime` timestamp met behulp van `ts_sec` en `ts_usec` . Deze tijd is in UTC.

`regel 33-38` De lengte van het packet is vermeld in de header (`incl_len`) We moeten `incl_len` bytes lezen, dit zou niet in een keer kunnen lukken, dus een while loop todat we genoeg gelezen hebben.

`regel 39` De header en het packet zijn gelezen en kunnen worden terug gegeven. Bij de volgende aanroep wordt vergegaan met regel `26`

`regel 43-46` Met `sys.argv` kunnen we de externe argumenenten, die gebruikt zijn om het programma aan te roepen benaderen. Als er een argument is wordt deze als `pcap-filename` gebruikt anders gebruiken de standaard naam. Om te testen of een en ander werkt tonen we header van elk packet af.