



**Moje riešenie:**

Pcap súbor som prečítal a uložil si ako string s ktorým som ďalej pracoval

p = bytes\_hex(pkt\_data).decode("utf-8")

Čítal som informácie zatiaľ po dvoch znakoch v stringu a transformoval som si ich podľa potreby na hexadecimálne čísla, decimálne čísla (IP adresa), alebo som ich nechal v stringovom tvare.

Ďalej som si dal do súboru **ethertypes.txt**  tieto základné typy ethernetov a len čítaním prvých 4och znakov som ich porovnával so znakmi v stringu, či sa zhodujú. Nebolo mi zatiaľ potrebné si ukladať tieto informácie z packetu, keďže som výstup rovno zapisoval do výstupného súboru **hexramce.txt.**

0800 IPv4  
0806 ARP  
86dd IPv6

Ak som našiel daný ethertype tak rovnakým štýlom som preň hladal v inom súbore menom **protocols.txt** prislúchajúci protokol.

01 ICMP  
02 IGMP  
06 TCP  
09 IGRP  
11 UDP  
2f GRE  
32 ESP  
33 AH  
39 SKIP  
58 EIGRP  
59 OSPF  
73 L2TP

Ak to nebol Ethernet ale IEEE 802.3 tak tie som rozlišoval v súbore **saps.txt**

ffff IEEE 802.3 - RAW  
e0e0 IEEE 802.3 - IPX  
aaaa IEEE 802.3 - SNAP  
4242 IEEE 802.3 - STP

Celé som to aplikoval v Pythone (pycharme) a konzolu som používal len na pomocné výpisky (rýchly debugger).

Môj výstup:

ramec 2138  
dlzka ramca poskytnuta pcap API - 92 B  
dlzka ramca prenasaneho po mediu - 96 B  
Ethernet II  
Zdrojova MAC adresa: 00 11 2F 67 B7 5F   
Cielova MAC adresa: FF FF FF FF FF FF   
IPv4  
zdrojova IP adresa: 147.175.144.91  
cielova IP adresa: 147.175.145.255  
UDP  
ff ff ff ff ff ff 00 11 2f 67 b7 5f 08 00 45 00  
00 4e 15 a2 00 00 80 11 db 43 93 af 90 5b 93 af  
91 ff 00 89 00 89 00 3a c2 a0 8f 9a 01 10 00 01  
00 00 00 00 00 00 20 45 4f 46 41 45 4a 44 45 44  
46 45 46 45 42 44 44 44 45 43 41 43 41 43 41 43  
41 43 41 43 41 41 41 00 00 20 00 01