Készítette: Mikolics Réka Szilvia (KD5XYP)

Sniffer

(TCP, UDP, ICMP)

A program célja:

Olyan csomagok információinak kinyerése, amelyek TCP, UDP vagy ICMP protokollt használnak.

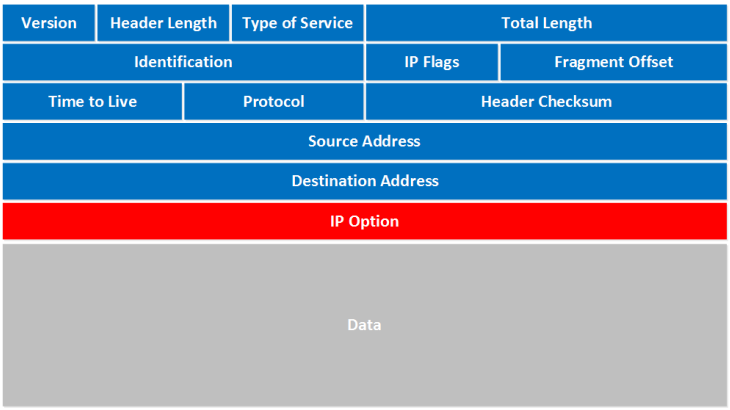
Bevezetés:
Az Ethernet szint:
Sajnos az Ethernetnek megvan a saját címzési módszere, mivel a létrehozók biztosítani akarták, hogy semelyik két gépnek se legyen ugyanaz az Ethernet címe. Azt is el akarták érni, hogy a felhasználónak ne kelljen a címek hozzárendelésével foglalkozni, ezért minden Ethernet vezérlő gyárilag beégetett címmel rendelkezik.
Minden Ethernet csomagnak egy 14 oktetes fejléce van, amely a forrás- és a célgép címét, valamint egy típuskódot tartalmaz. A hálózaton lévő gépek csak az olyan csomagokat figyelik, amelyek célmezőjében a saját Ethernet címüket találjak.

```
<<-------------------------32 bit----------------------------->>

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Célgép Ethernet címe (első 32 bit)                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ethernet cél (utolsó 16 bit) | Ethernet forrás (első 16 bit)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Forrásgép Ethernet címe (utolsó 32 bit)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Típuskód                  |                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IP fejléc, TCP fejléc, majd a tényleges adatok                |
|                                                               |
...
| adatok vége                                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ethernet ellenőrzőösszeg                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ha az Ethernet fejlécet E-vel, az ellenőrzőösszeget pedig C-vel jelöljük, akkor az eredeti állományunk így néz ki:
EIT....C EIT....C EIT....C EIT....C EIT....C EIT....C EIT....C
A csomagok megérkezésekor persze a fenti fejlécek mindegyikét leszedi a megfelelő protokoll. Az Ethernet interfész az Ethernet fejlécet és az Ethernet ellenőrzőösszeget szedi le. Ezekután ellenőrzi a típuskódot. Mivel az az IP-re mutat, ezért a datagrammot átadja az IP-nek, amely a Protokoll mező tartalmát ellenőrzi. Itt azt találja, hogy TCP, ezért a

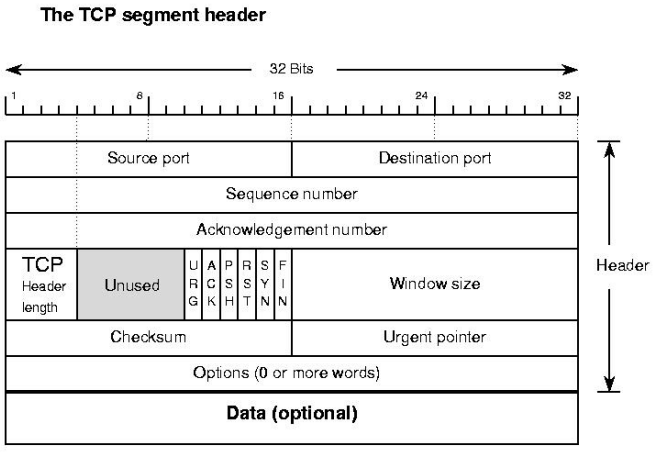
datagrammot a TCP-nek adja át. A TCP a Sorszám mező tartalma és egyéb információk alapján állítja össze az eredeti állományt.
Következő az IP :

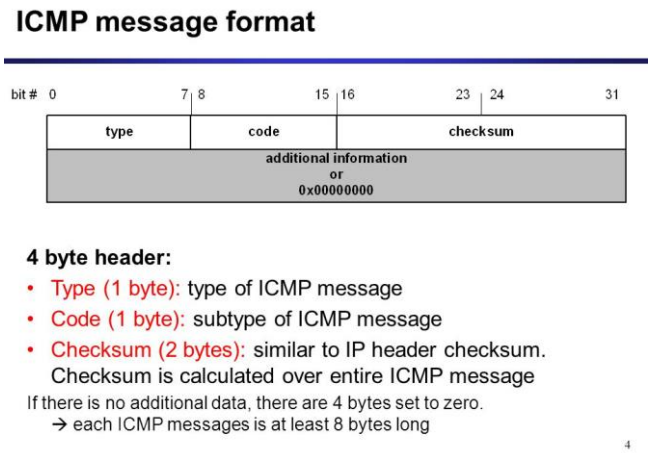


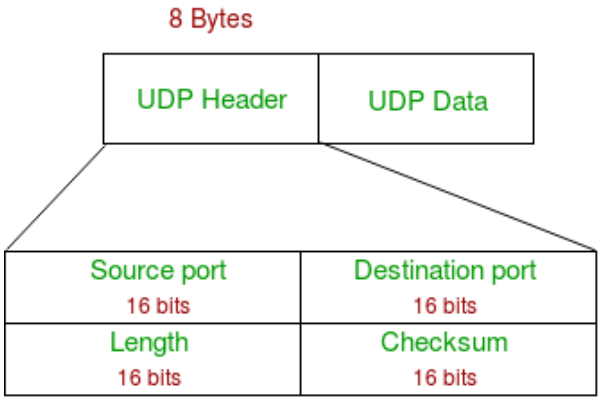
Ezt követően a protokoll száma alapján el kell döntenünk, hogy a kiolvasott csomag milyen protokollt használt.

Decimal Protocol Number	Keyword	Protocol
1	ICMP	Internet Control Message Protocol (ICMP)
6	TCP	Transmission Control Protocol (TCP)
17	UDP	User Datagram Protocol (UDP
50	ESP	Encapsulating Security Payload (ESP)
51	AH	Authentication Header (AH)

A különböző protokollok különböző felépítéssel rendelkeznek. Ezek alapján tudjuk kiolvasni őket.







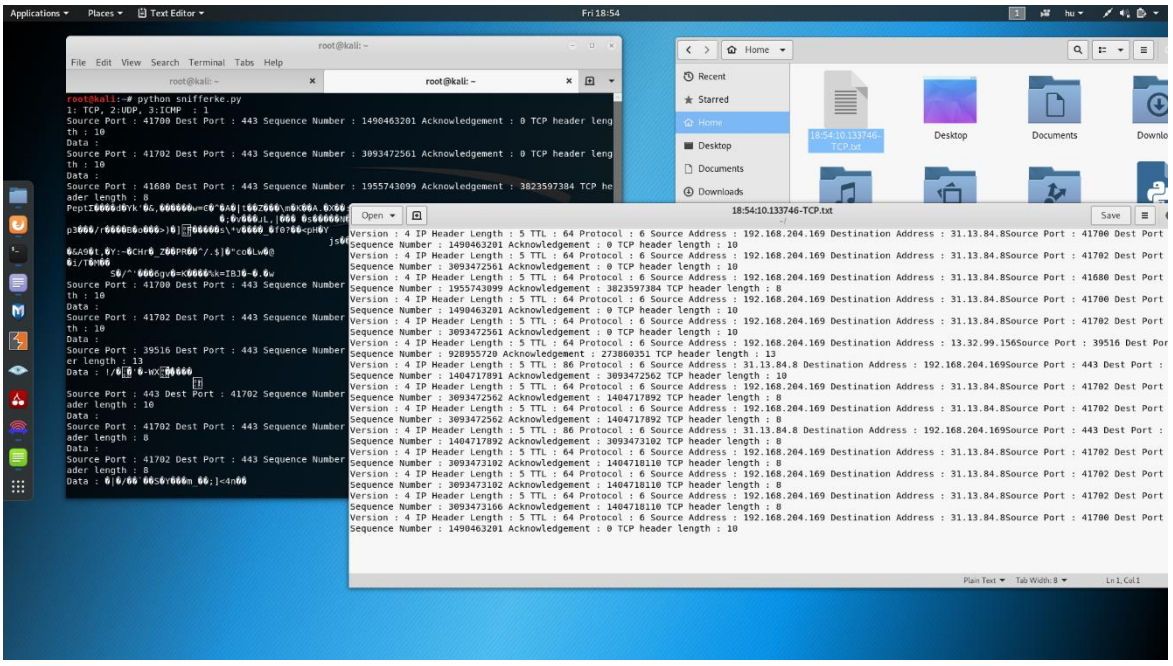
Ezeket a python socket.socket(*self*, *family*, *type*, *protocol*) kérjük le:

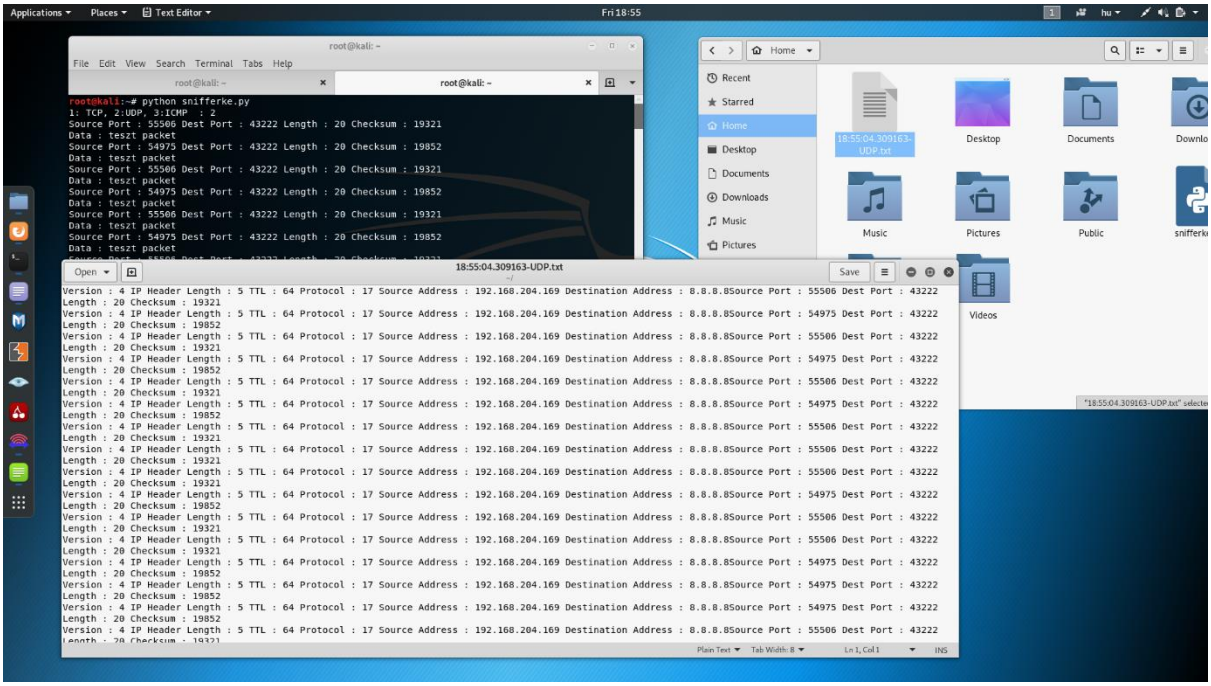
socket.socket(socket.AF_PACKET , socket.SOCK_RAW , socket.ntohs(0x0003))

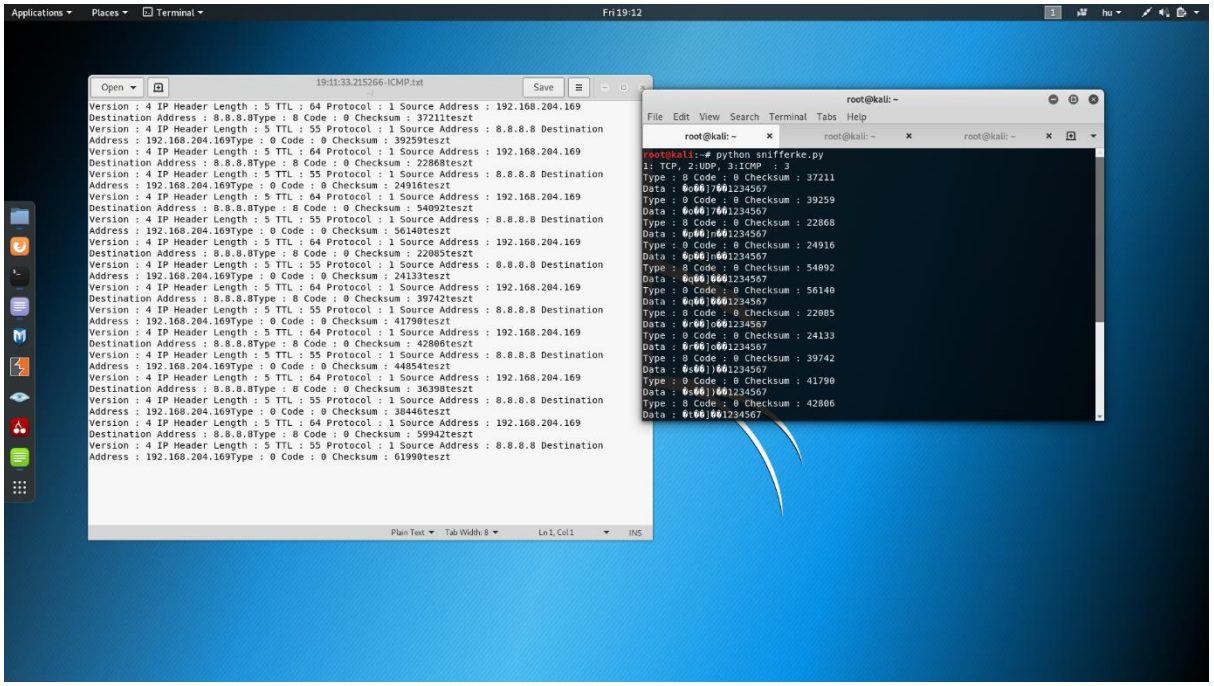
Így minden kimenő és bejövő forgamat figyelni tudunk (*socket.ntohs(0x0003)*) és az ethernet fejlécet is megkapjuk (*socket.AF_PACKET*).

A program:
Áll egy switcherből, eldönthetjük, hogy milyen protokollt akarunk leghallgatni és egy filewriterből, a kapott adatokat kiírja egy fájlba.

A teszteléshez Pakcet Sendert használtam (https://packetsender.com/) amellyel UDP és TCP protokollos csomagokat lehet küldözgetni Linuxon is. Az ICMP protokoll teszteléséhez a ping 8.8.8.8-at használtam, mivel a Linux pingelése ICMP protokollt használ.
Képek a futásról:




Applications
Places
Text Editor
Fri 18:55
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali:~# python snifferke.py
1: TCP, 2:UDP, 3:ICMP : 2
Source Port : 55506 Dest Port : 43222 Length : 20 Checksum : 19321
Data : teszt packet
Source Port : 54975 Dest Port : 43222 Length : 20 Checksum : 19852
Data : teszt packet
Source Port : 55506 Dest Port : 43222 Length : 20 Checksum : 19321
Data : teszt packet
Source Port : 54975 Dest Port : 43222 Length : 20 Checksum : 19852
Data : teszt packet
Source Port : 55506 Dest Port : 43222 Length : 20 Checksum : 19321
Data : teszt packet
Source Port : 54975 Dest Port : 43222 Length : 20 Checksum : 19852
Data : teszt packet
Home
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Public
18:55:04.309163-UDP.txt
Open
Save
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 17 Source Address : 192.168.204.169 Destination Address : 8.8.8.8Source Port : 55506 Dest Port : 43222
Length : 20 Checksum : 19321
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 17 Source Address : 192.168.204.169 Destination Address : 8.8.8.8Source Port : 54975 Dest Port : 43222
Length : 20 Checksum : 19852
Plain Text
Tab Width: 8
Ln 1, Col 1
INS


Applications
Places
Terminal
Fri 19:12
19:11:33.215266-ICMP.txt
Open
Save
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 1 Source Address : 192.168.204.169
Destination Address : 8.8.8.8Type : 8 Code : 0 Checksum : 37211teszt
Version : 4 IP Header Length : 5 TTL : 55 Protocol : 1 Source Address : 8.8.8.8 Destination
Address : 192.168.204.169Type : 0 Code : 0 Checksum : 39259teszt
root@kali: ~
File Edit View Search Terminal Tabs Help
root@kali:~# python snifferke.py
1: TCP, 2:UDP, 3:ICMP : 3
Type : 8 Code : 0 Checksum : 37211
Type : 0 Code : 0 Checksum : 39259
Type : 8 Code : 0 Checksum : 22868
Type : 0 Code : 0 Checksum : 24916
Type : 8 Code : 0 Checksum : 54092
Type : 0 Code : 0 Checksum : 56140
Type : 8 Code : 0 Checksum : 22085
Type : 0 Code : 0 Checksum : 24133
Type : 8 Code : 0 Checksum : 39742
Type : 0 Code : 0 Checksum : 41790
Type : 8 Code : 0 Checksum : 42806
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

Packet Sender használata: