

```

#----- Scapy CPL -----
# Envoyer des paquets avec payload
sendp(Ether()/"Hello World")

a = Ether()
data = "100110"
sendp(a/data)

# Paramètres de trames Ethernet
Ether(src = "11:22:33:44:55:66", dst = "ff:ff:ff:ff:ff:ff", type = 0x0801)

# Sniffer des paquets (ctrl+c pour arrêter)
a = sniff(filter = "ether src 000000000000") # avec adresse de source 00:00:00:00:00:00
a = sniff(filter = "ether dst ffffffff") # avec adresse de destination
ff:ff:ff:ff:ff:ff
a = sniff(filter = "ether proto 0x0801") # avec type de trame 0x0801
# En travaillant directement sur les datas (l'adresse source est du 6 au 12ème octet dans
une trame Ethernet):
a = sniff(filter = "ether[6:4] == 0x34363bd3 and ether[10:2] == 0xea14") # avec adresse
de source 34:36:3b:d3:ea:14

# Affichage du tableau des trames reçues
a.show()
# Affichage d'une des trames reçues
a[1].show()

#----- Exemples avec python -----
# N'affiche que les paquets avec une certaine adresse de destination (si pas filtré
avant)
for p in a:
    if p.dst == "ff:ff:ff:ff:ff:ff":
        p.show()
        print "-----"

# N'affiche que les paquets contenant une certaine séquence dans le payload
if p.load.__contains__('100001'): # Pourrait être un flag

# N'affiche que les paquets dont le payload commence par une certaine la séquence
if p.load[1:4] == "1101" : # Pourrait être une adresse de destination

# Lancer une fonction pour chaque paquet sniffé, ici affiche les paquets au fur et à
mesure qu'ils arrivent
def customAction(packet):
    packet.show()
    print "-----"

sniff(filter = "ether src 000000000000 and ether dst = ffffffff", prn = customAction)

```