E:\sniff.py

```
from scapy.all import *
sniff(filter="ip", prn=lambda x:x.sprintf("{IP:%IP.src% -> %IP.dst%\n}"))
import dpkt, pcap
pc = pcap.pcap()     # construct pcap object
pc.setfilter('icmp') # filter out unwanted packets
for timestamp, packet in pc:
    print dpkt.ethernet.Ethernet(packet)
import pcap

p = pcap.pcapObject()
dev = pcap.lookupdev()
p.open_live(dev, 1600, 0, 100)
#p.setnonblock(1)
try:
    for pktlen, data, timestamp in p:
        print "[%s] Got data: %s" % (time.strftime('%H:%M',
                                                   time.localtime(timestamp)),
                                     data)
except KeyboardInterrupt:
    print '%s' % sys.exc_type
    print 'shutting down'
    print ('%d packets received, %d packets dropped'
           ' %d packets dropped by interface') % p.stats()
import socket
s = socket.socket(socket.AF_INET,socket.SOCK_RAW,socket.IPPROTO_IP)
s.bind(("YOUR_INTERFACE_IP",0))
s.setsockopt(socket.IPPROTO_IP,socket.IP_HDRINCL,1)
s.ioctl(socket.SIO_RCVALL,socket.RCVALL_ON)
while True:
   data = s.recvfrom(10000)
   print data
# -*- coding: utf-8 -*-

# pip install scapy

"""
[{'name': 'Intel(R) 82574L Gigabit Network Connection',
  'win_index': '4',
  'description': 'Ethernet0',
  'guid': '{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}',
  'mac': '00:0C:29:5C:EE:6D',
  'netid': 'Ethernet0'}]
"""

from pprint import pprint
from scapy.arch.windows import get_windows_if_list
from scapy.all import *


# disable verbose mode
conf.verb = 0


def parse_packet(packet):
```

```
55      """sniff callback function.
56      """
57      if packet and packet.haslayer('UDP'):
58          udp = packet.getlayer('UDP')
59          udp.show()
60
61
62  def udp_sniffer():
63      """start a sniffer.
64      """
65      interfaces = get_windows_if_list()
66      pprint(interfaces)
67
68      print('\n[*] start udp sniffer')
69      sniff(
70          filter="udp port 53",
71          iface=r'Intel(R) 82574L Gigabit Network Connection', prn=parse_packet
72      )
73
74
75  if __name__ == '__main__':
76      udp_sniffer()
```