Summary of the tool

The net_recon.py script/tool allows a user to passively or actively detect hosts on their network. This

tool takes in two arguments.

A network interface name (e.g. “enp0s3”) and an indicator for active or

passive mode. The network interface should be specified using “-i” or “--iface”.

A user who launches the tool with the “-p” or “—passive” argument will launch the tool in passive

mode, where the script will monitor ARP traffic on the given interface and use this traffic to detect the IP

address and MAC address pairings. IP and MAC addresses collected by the script are printed out to

the terminal while the script is running. The passive scan will continue until the user stops the script

using ctrl+c.

A user who launches the tool with the “-a” or “—active” argument will launch the tool in active

mode. In active mode, the tool will perform a ping sweep. The tool should ping every address in the

network and detect if a reply was received to determine if that address is active in the network.

You may use the scapy, sys, and os libraries/modules only. You must demonstrate that you can

write the code with these restrictions in place.

Part 1 – Set-up

Create a script called “net_recon.py”.

Add a main function that contains code to read the arguments passed in by the user from the

command line when the script was launched.

Add a function help (), which will print out the usage of the tool. If the user launches the script

without the necessary arguments, this function should be called and should print out the arguments

which a user can include when running the script.

Part 2 – Passive Recon

Add a function passive_scan() which is called if the user includes the “-p” or “—passive” augments

when launching the tool.

This function should use the Scapy sniff() function to listen for traffic at the interface provided by the

user. Include any function(s) necessary to handle traffic picked up by the sniff() function.

ARP traffic with an opcode of 2 or “is-at” should be parsed and the source IP address and source

MAC address stored. If an IP address has already been stored but a different MAC address is seen

then the script should also store this additional MAC address.

The user should see a list of hosts appear in the terminal while the script is running.

Part 3 – Active Recon

Add a function active_recon() which is called if the user includes the “-a” or “—active” arguments

when launching the tool.

This function should fetch the IP address for the given network interface. It should then send an

ICMP request to every host in the same network (you may assume it is a /24 network). The tool

should detect if an ICMP reply was returned. When the script has finished sending ICMP requests it

should output a list of addresses for which a reply was received.

Part 4 – Improved Display

Create an improved display for the user. This display should match the following:

The items on the top row should be dynamic. The network interface should show the interface

passed in by the user, the mode should show the current operating mode, and the total number of

detected hosts should be displayed to the right. The MAC and IP addresses observed while the

program is running should be displayed as shown above. If a MAC address is not known for a given IP

address, or vice versa, a “?” should be displayed in place of the missing address.

```
Interface: eth0                 Mode: Passive                   Found 3 hosts
-------------------------------------------------------------------------------
MAC                             IP
-------------------------------------------------------------------------------
11:22:33:44:55:66               192.168.1.2
de:ad:be:e7:ba:11               192.168.1.10
ea:77:a7:aa:ca:fe               192.168.1.3
```

Part 5 – Host Activity

The functionality of this part should only be active during “Passive” mode. Add a column to the table

in the display that shows the “Host Activity”, in terms of the total number of packets observed for a

given host. You will need to modify your code to capture this information. Displayed hosts should be

ordered by the number of packets seen. The output should resemble the following:

```
Interface: eth0                 Mode: Passive                   Found 3 hosts
-------------------------------------------------------------------------------
MAC                             IP                              Host Activity
-------------------------------------------------------------------------------
de:ad:be:e7:ba:11               192.168.1.10                    42
ea:77:a7:aa:ca:fe               192.168.1.3                     14
11:22:33:44:55:66               192.168.1.2                     5
```

Part 6 – Clean Readable Code (10%)

You code should be commented to clearly explain what each part of your code does.