Лабораторна робота №4

з дисципліни «мережеві технології»

«Конфігурування мережевих інтерфейсів. Діагностичні утиліти TCP/IP»

Виконав: студент групи ІП-73мп
Олександр Ковальчук

Контрольні запитання:

1. Як можна визначити номер порту, який використовує мережевий застосунок?

   Визначити pid мережевого застосунку (наприклад, за допомогою утиліти `ps`). Після цього відфільтрувати вивід `sockstat -ls` по отриманому process id: `sockstat -ls | grep 710`

2. Які параметри фізичного, канального і мережевого рівнів моделі OSI можна змінити за допомогою утиліти ifconfig?

   Практично будь-які. Наприклад, IP-адресу, маску мережі, MAC-адресу, та ін. Детальніше написано у `man 8 ifconfig`

3. Що означає promiscuous mode мережевого інтерфейсу?

   Promiscuous mode або promisc mode - так званий «нерозбірливий» режим, в якому мережева плата дозволяє приймати всі пакети незалежно від того, кому вони адресовані.

   У нормальному стані на Ethernet-інтерфейсі використовується фільтрація пакетів канального рівня і якщо MAC-адреса в заголовку призначення прийнятого пакета не збігається з MAC-адресою поточного мережевого інтерфейсу і не є широкомовною, то пакет відкидається. У «нерозбірливому» режимі фільтрація на мережевому інтерфейсі відключається і все пакети, включаючи не призначені поточного вузла, пропускаються в систему.

4. Яким чином у FreeBSD виконується перетворення DNS-імені в IP-адресу?

   За допомогою бібліотечної функції gethostbyname (3). Типова поведінка цієї функції виглядає так: 1) проглядається файл `/etc/hosts`, в якому перераховано, які імена відповідають яким адресам; 2) якщо пошук не дав результатів, за допомогою `resolver (3)` здійснюються запити до серверів DNS. На

порядок цих дій можна впливати за допомогою файлу `/etc/nsswitch.conf`.

5. Яким чином утиліта traceroute визначає маршрут просування пакета?

   Посилає пакети з max ttl, що поступово збільшується, і аналізує ICMP повідомлення про помилки.

6. Які функції виконує утиліта ping?

   Надсилає ICMP ECHO_REQUEST пакети на мережеві вузли. Це використовується для базової діагностики доступності вузлів мережі.

7. Рівні стеку протоколів TCP / IP і їх функції.

   Прикладний — на цьому рівні працюють прикладні програми

   Транспортний – вирішує проблему доставки повідомлень

   Міжмережевої взаємодії — забезпечує взаємодію різних мереж. Головне завдання — маршрутизація пакетів даних між різнотипними комп'ютерними мережами.

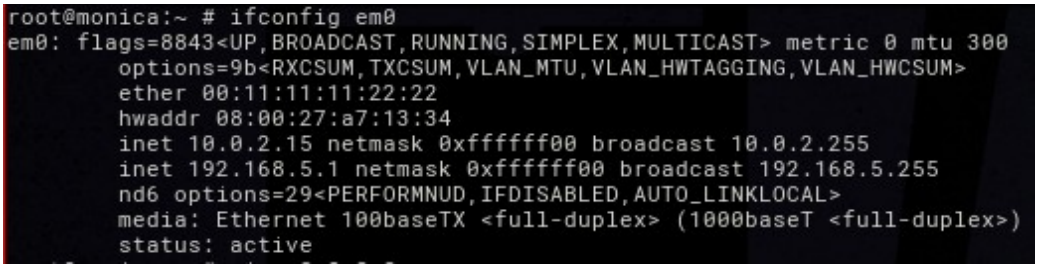
   Мережевих інтерфейсів — описує спосіб кодування даних для передачі пакета даних на фізичному рівні, тобто безпосередньо по дротах.

8. З якою метою може використовуватися утиліта tcpdump?

   Перехоплення мережевого трафіку. Це може виконуватися як із діагностичною метою, так і зі злочинною метою (наприклад, незаконно отримати дані, що передаються по мережі)

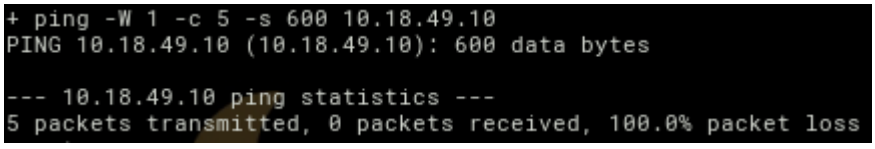
Налаштування мережевого інтерфейсу

```
#!/bin/sh
set -e

# All manipulations are done with em0 interface

# 1. Add additional IP address
# ============================
ifconfig em0 add 192.168.5.1

# 2. Change MAC address and working mode
# ======================================
ifconfig em0 lladdr 00:11:11:11:22:22
ifconfig em0 media 100baseTX
ifconfig em0 mediaopt full-duplex

# 3. Change MTU
# =============
ifconfig em0 mtu 300
```



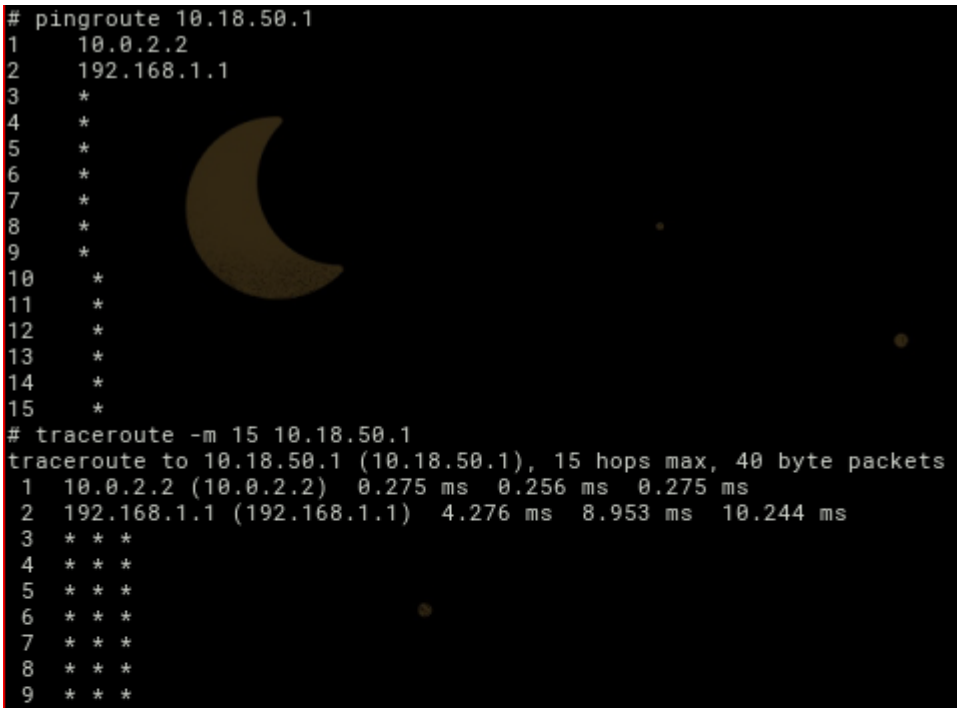
Діагностичні утиліти TCP/IP

```
# 4. Ping 10.18.49.10 with 5 packets and mtu 600
# ==============================================
ping -c 5 -s 600 10.18.49.10
```




```
# 5. Using ping and traceroute show the path to 10.18.50.1
# ========================================================

# Buggy function which emulates traceroute via ping command
pingroute()
{
  dest="$1"
```

```
  msg="(Destination (host|net) unreachable|Time to live exceeded)"
  for ttl in `seq 15`; do
    set +e
    ping_stats="$(ping -vvv -no -c 1 -m ${ttl} "${dest}")"
    is_fin=$?
    set -e
    if [ $is_fin -eq 0 ] ; then
        echo "${ttl}    ${dest}"
        break
    fi

    error_msg="$(echo "${ping_stats}" | egrep -i "${msg}")" || true
    case "$error_msg" in
      *"Time to live exceeded")
        addr=$(echo "${error_msg}" | awk '{print $4}' | sed -e 's/://g')
        echo "${ttl}    ${addr}"
        ;;
      *)
        echo "${ttl}    *    ${error_msg}"
        ;;
    esac
  done
}

# Traceroute using ping
pingroute 10.18.50.1


# Normal traceroute
traceroute 10.18.50.1
traceroute -I 10.18.50.1
```



```
# 6. Print statistics of TCP,UDP,ICMP,IP protocols
# ================================================

netstat -ni -p tcp -p udp -p icmp -p ip -h
```

```
# netstat -p tcp -p udp -p icmp -p ip -ni -h
Name    Mtu Network       Address              Ipkts Ierrs Idrop     Opkts Oerrs  Coll
em0     300 <Link#1>      00:11:11:11:22:22     9.0M     0     0      9.0M    59     0
em0       - 10.0.2.0/24   10.0.2.15              673     -     -      9.0M     -     -
em0       - 192.168.5.0/2 192.168.5.1           9.0M     -     -         0     -     -
em1    1.5K <Link#2>      08:00:27:89:82:97     1.8k     0     0       989     0     0
em1       - 10.18.51.0/24 10.18.51.101          1.7k     -     -       979     -     -
lo0     16K <Link#3>      lo0                    280     0     0       280     0     0
lo0       - ::1/128       ::1                    140     -     -       140     -     -
lo0       - fe80::%lo0/64 fe80::1%lo0              0     -     -         0     -     -
lo0       - 127.0.0.0/8   127.0.0.1              140     -     -       140     -     -
#
```

```
netstat -s -p udp
netstat -s -p tcp
netstat -s -p ip
netstat -s -p icmp
```

```
# netstat -s -p udp
udp:
        81 datagrams received
        0 with incomplete header
        0 with bad data length field
        0 with bad checksum
        0 with no checksum
        3 dropped due to no socket
        55 broadcast/multicast datagrams undelivered
        0 dropped due to full socket buffers
        0 not for hashed pcb
        23 delivered
        23 datagrams output
        0 times multicast source filter matched
```

```
# 7. Find IP address by name
# ==========================

# nslookup and dig utilities are not included in base system anymore
# however there are replacements: host and drill

host samba.org
drill samba.org
```

```
root@monica:~ # host samba.org
samba.org has address 144.76.82.156
samba.org has IPv6 address 2a01:4f8:192:486::443:2
samba.org mail is handled by 5 ns1.samba.org.
samba.org mail is handled by 7 smtp.samba.org.
samba.org mail is handled by 9 ns1.samba.org.
root@monica:~ # drill samba.org
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 46092
;; flags: qr rd ra ; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; samba.org.   IN      A

;; ANSWER SECTION:
samba.org.      6764    IN      A       144.76.82.156

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:

;; Query time: 9 msec
;; SERVER: 192.168.1.1
;; WHEN: Sun Nov  5 02:02:43 2017
;; MSG SIZE  rcvd: 43
root@monica:~ #
```

Налаштування фільтру tcpdump для перехоплення трафіку TCP syn, UDP з мережі 10.18.51.0/24 та ICMP в мережу 10.18.48.0/24.

```
# Dump traffic:
# INCOMING: TCP syn, UDP from network 10.18.51.0/24
# OUTGOING: ICMP to 10.18.48.0/24

src_net='10.18.51.0/24'
dst_net='10.18.48.0/24'
incoming_filter="((tcp[tcpflags] & tcp-syn != 0) or udp) and src net ${src_net}"
outgoing_filter="icmp and dst net ${dst_net}"
for interface in `ifconfig -l`; do
  tcpdump -i "$interface" "(${incoming_filter}) or (${outgoing_filter})" &
done
```

```
00:33:17.434153 IP 10.0.2.15 > 10.18.48.4: ICMP echo request, id 7171, seq 0, length 64
00:33:18.438033 IP 10.0.2.15 > 10.18.48.4: ICMP echo request, id 7171, seq 1, length 64
00:33:19.445915 IP 10.0.2.15 > 10.18.48.4: ICMP echo request, id 7171, seq 2, length 64
00:33:20.506926 IP 10.0.2.15 > 10.18.48.4: ICMP echo request, id 7171, seq 3, length 64
+ true
+ sleep .5
00:33:21.724030 IP 10.18.51.2.36267 > 10.18.51.101.metagram: UDP, length 5
00:33:32.717406 IP 10.18.51.2.54766 > 10.18.51.101.ssh: Flags [S], seq 2524166953, win 29200, options [mss 1460,sackOK,TS val 1910979240 ecr 0,nop,wscale 7], length 0
00:33:32.717540 IP 10.18.51.101.ssh > 10.18.51.2.54766: Flags [S.], seq 833892043, ack 2524166954, win 65535, options [mss 1460,nop,wscale 6,sackOK,TS val 1426559250 ecr 1910
979240], length 0
00:33:45.247837 IP 10.18.51.2.53792 > 10.18.51.101.cdc: Flags [S], seq 2399861363, win 29200, options [mss 1460,sackOK,TS val 1910991771 ecr 0,nop,wscale 7], length 0
00:35:06.741615 IP 10.18.51.2.55361 > 239.255.255.250.1900: UDP, length 170
00:35:07.742773 IP 10.18.51.2.55361 > 239.255.255.250.1900: UDP, length 170
00:35:08.743124 IP 10.18.51.2.55361 > 239.255.255.250.1900: UDP, length 170
00:35:09.743955 IP 10.18.51.2.55361 > 239.255.255.250.1900: UDP, length 170
```