Debugging Network Application

Tomasz Urban

30 stycznia 2024

1 Task 1

1.1 Clone repository

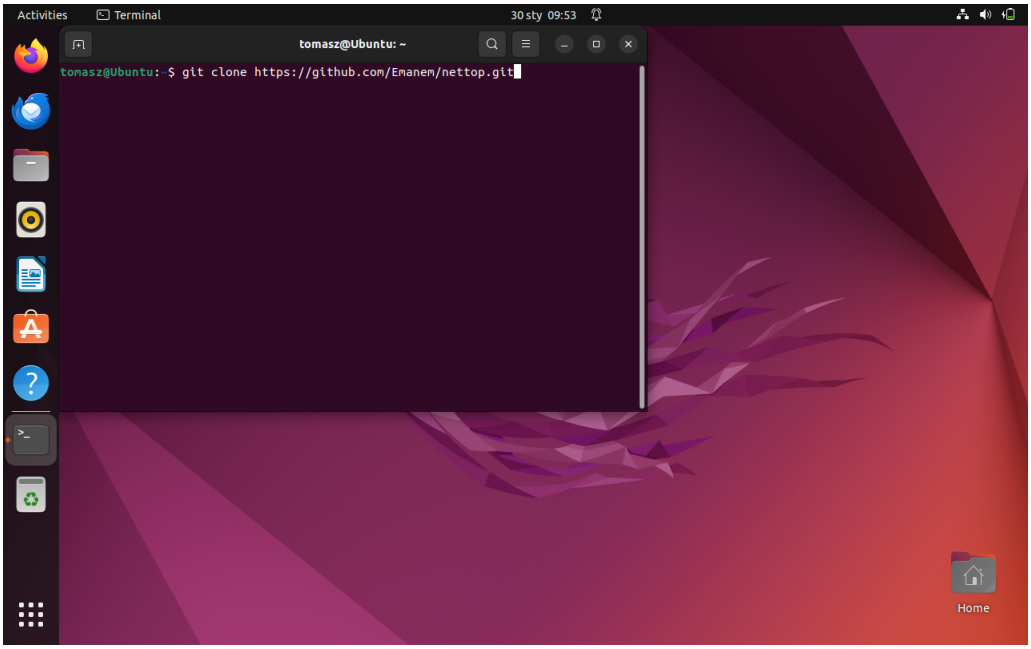

Figure 1: Clone repositroy with nettop source code

1.2 Install needed dependencies

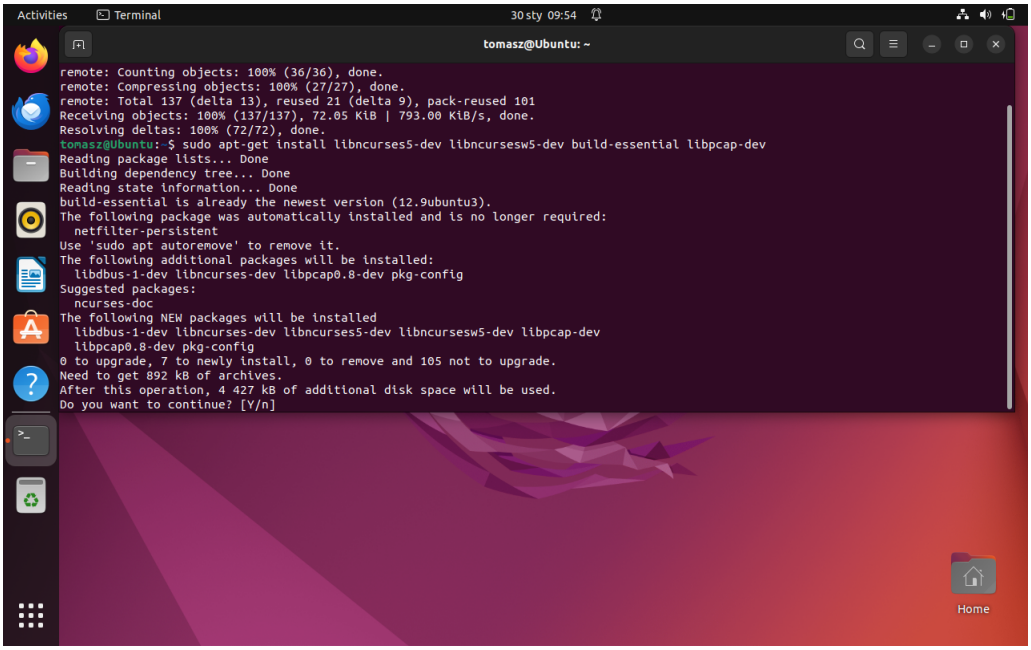

Figure 2: Install needed dependencies

1.3 Run nettop

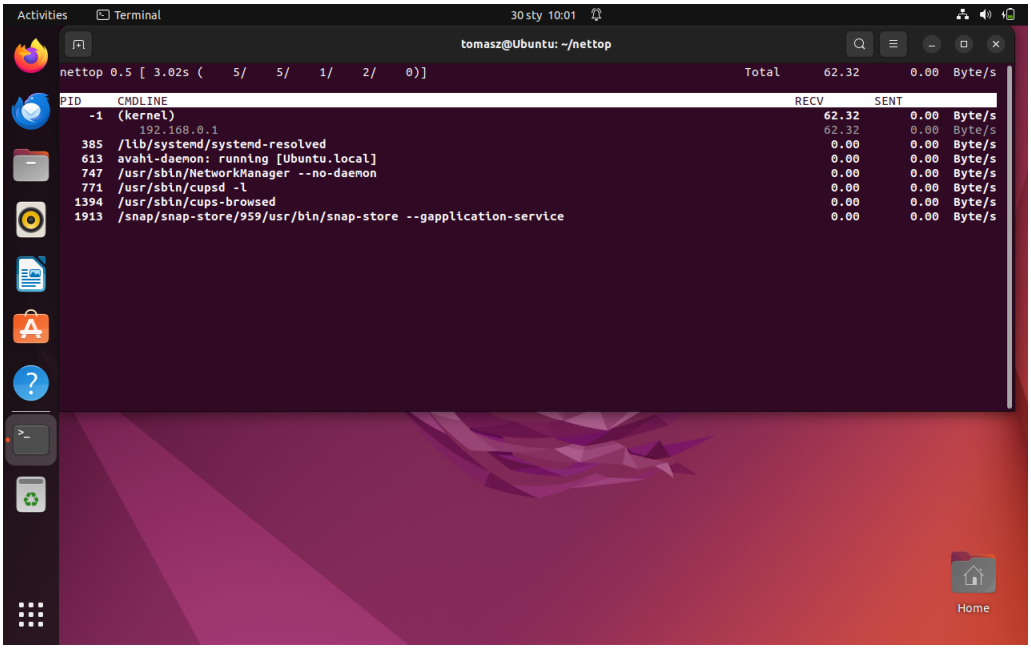

Figure 3: Run nettop

2 Task 2

To generate table network interface enp0s8 has been dumped (with global network address). Table with output is in file most_network_traffic.csv.

Output has been generated with use of jttop command. Command is provided on figure 4.

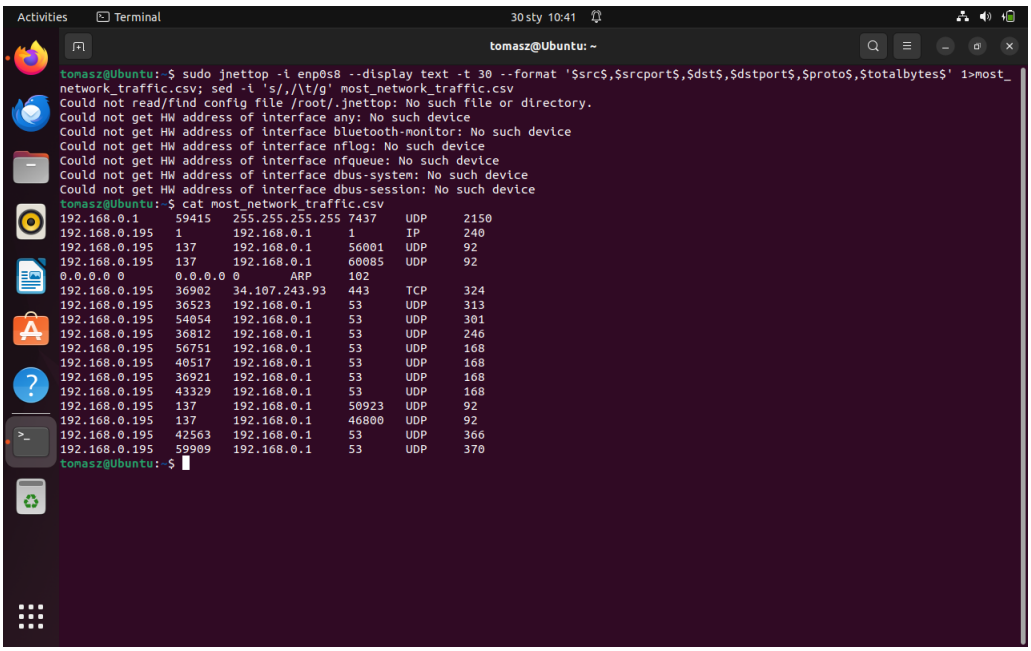

Figure 4: jttop output

2.1 First cron job

- job must be executed every 5 minutes on every hour,
- job include 10 seconds of gathering network traffic,
- traffic is recored to file
- number of bytes are captured
- script is provided in file task3_job1.sh

2.2 Second cron job

- at the beginning of hour in the output file of first job, a most networking consuming connection is found
- found value is written to syslog
- at the end file is clear
- script is provided in file task3_job2.sh

2.3 Crontab entries

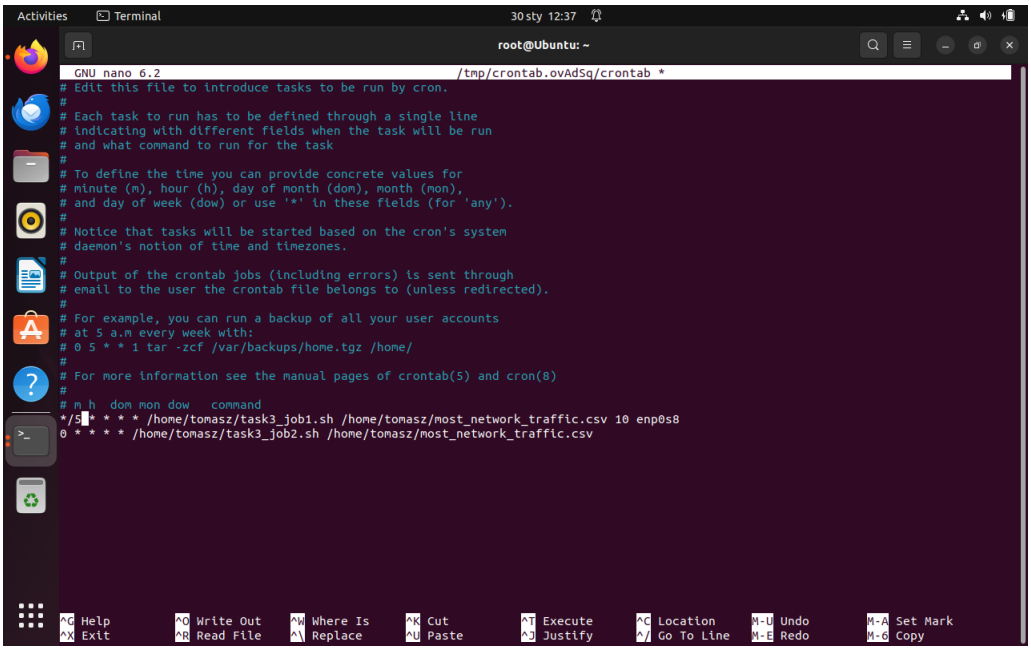

Figure 5: Crontab entries

2.4 Syslog

To check the functionality of cron jobs with sudo nano /var/log/syslog content of file has been examined to find out whether the proper line is inside. A new record at 6 PM has been found. Crontab is assumed to work correctly.

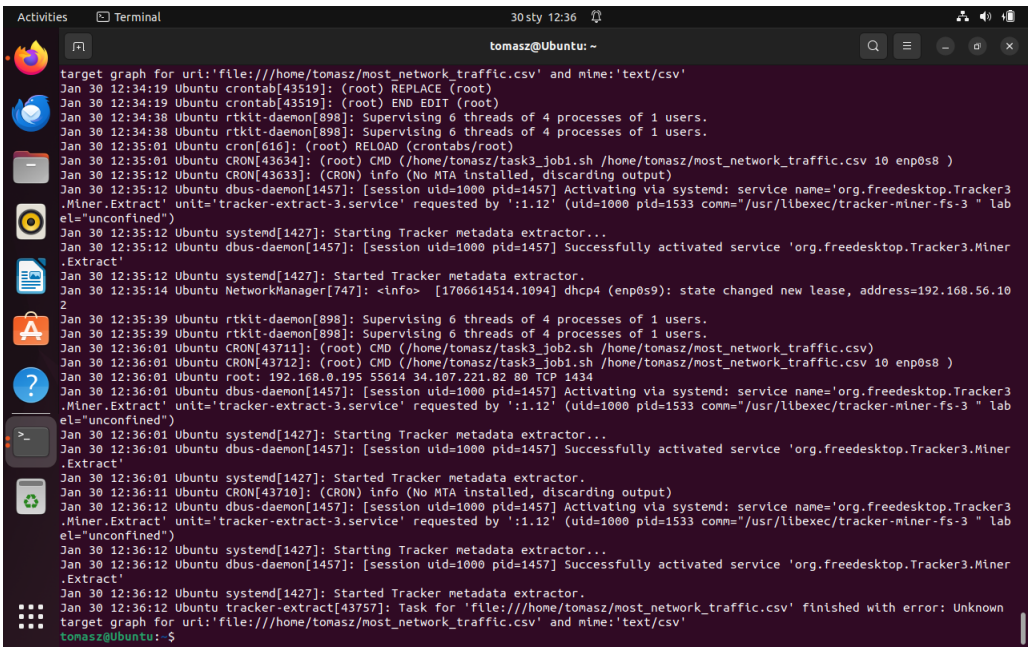

Figure 6: Syslog record

3 Task 4

3.1 Create and compress archive of home directory

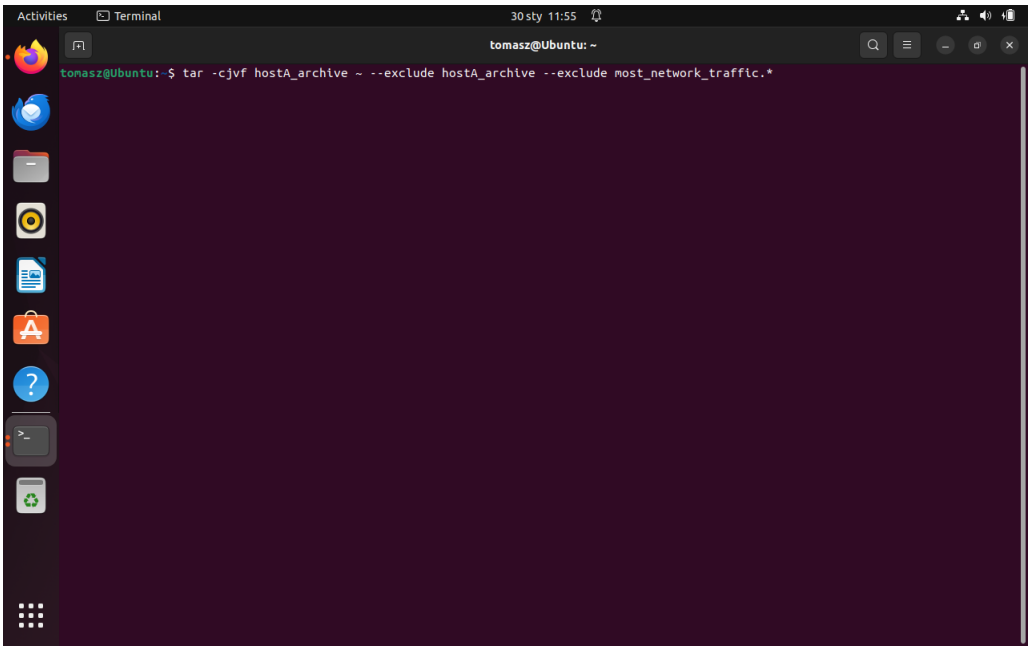

Figure 7: Syslog record

3.2 Send with nc

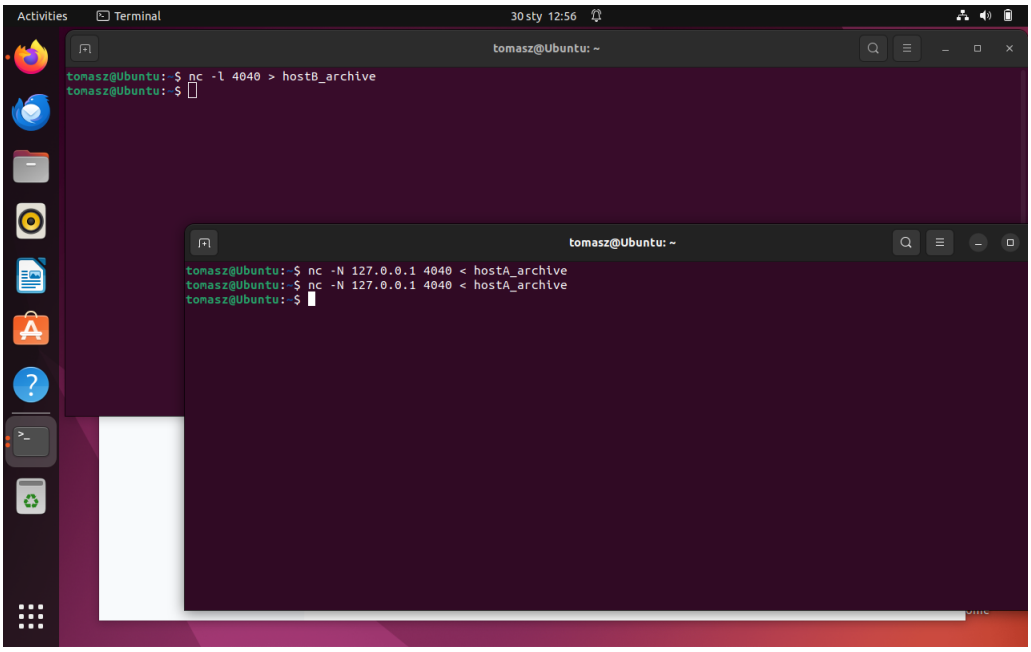

Figure 8: Send home archive via nc

On hostB received archive is saved as hostB_archive.

3.3 Content of archive

Both archives (hostB_archive and hostA_archive) are provided on a disk for evaluation.

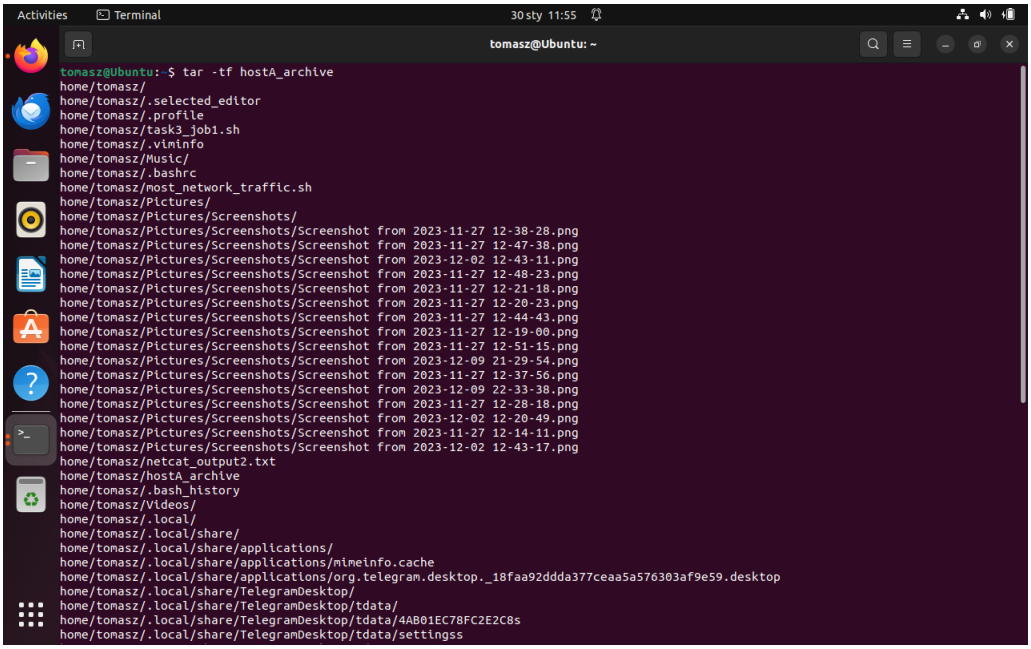

Figure 9: Content of archives

4 Task 5

4.1 Install nmap

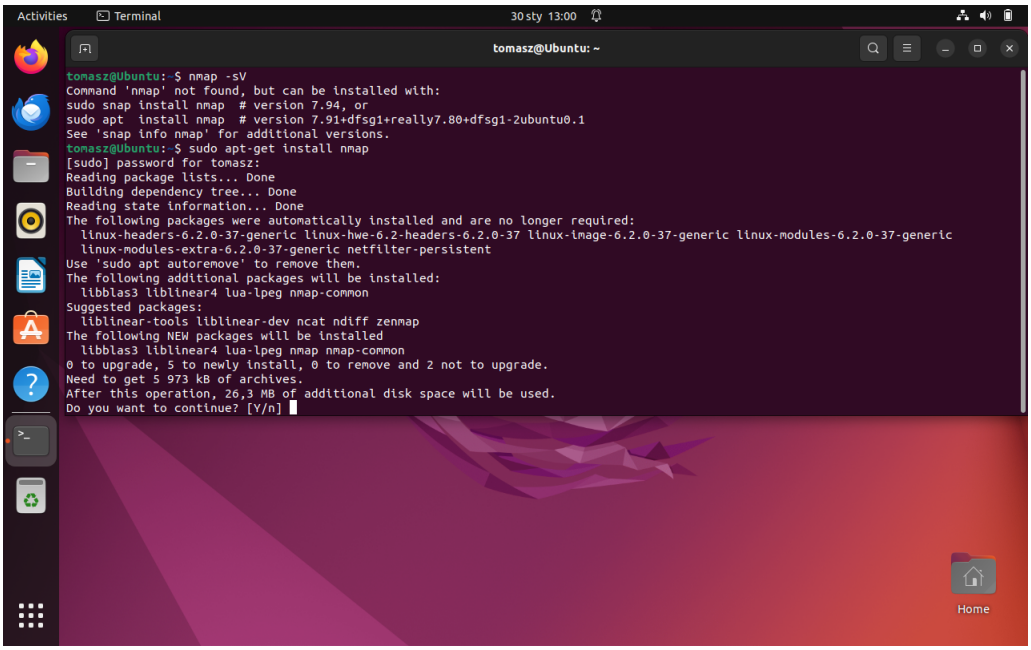

Figure 10: Search for OS and services

4.2 Nmap Security scanning, search for services an daemons

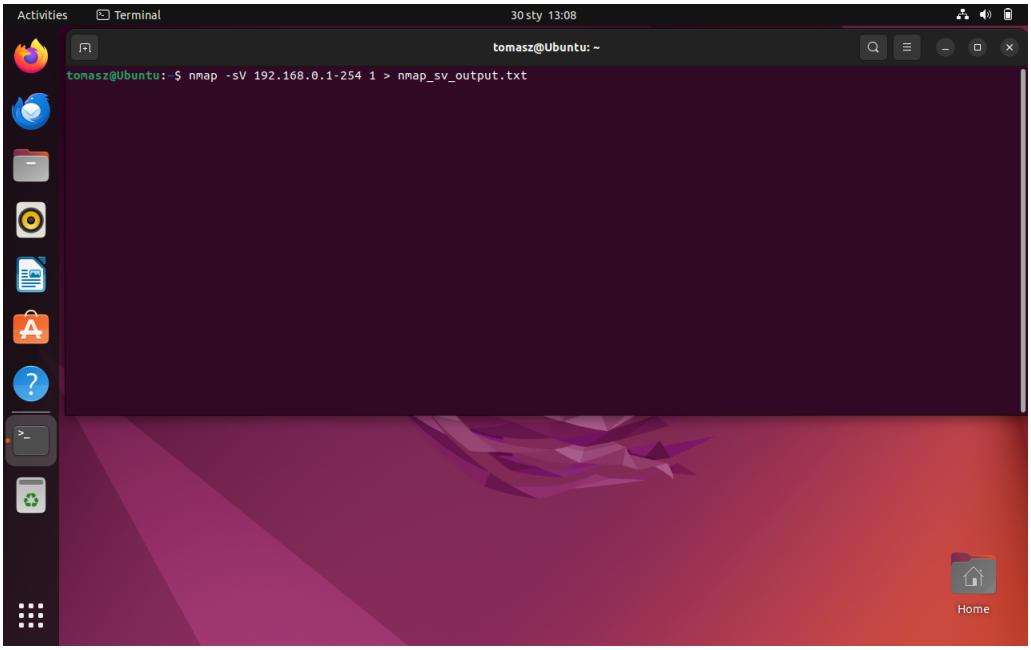

Figure 11: Search for services and daemons

File with search results is provided on the shared disk (nmap_sv_output.txt).

4.2.1 Search results

On gateway (192.168.0.1) there are following services available (all on tcp sockets):

- ssh server on port 22, input is filtered
- dns server on port 53, input is open
- http server on port 80
- netbios-ssn on port 139
- microsoft-ds on port 445
- upnp on port 1900

On host (192.168.0.195) all ports are closed. There are no available services.

4.3 Nmap Security scanning, search for OS and service detection

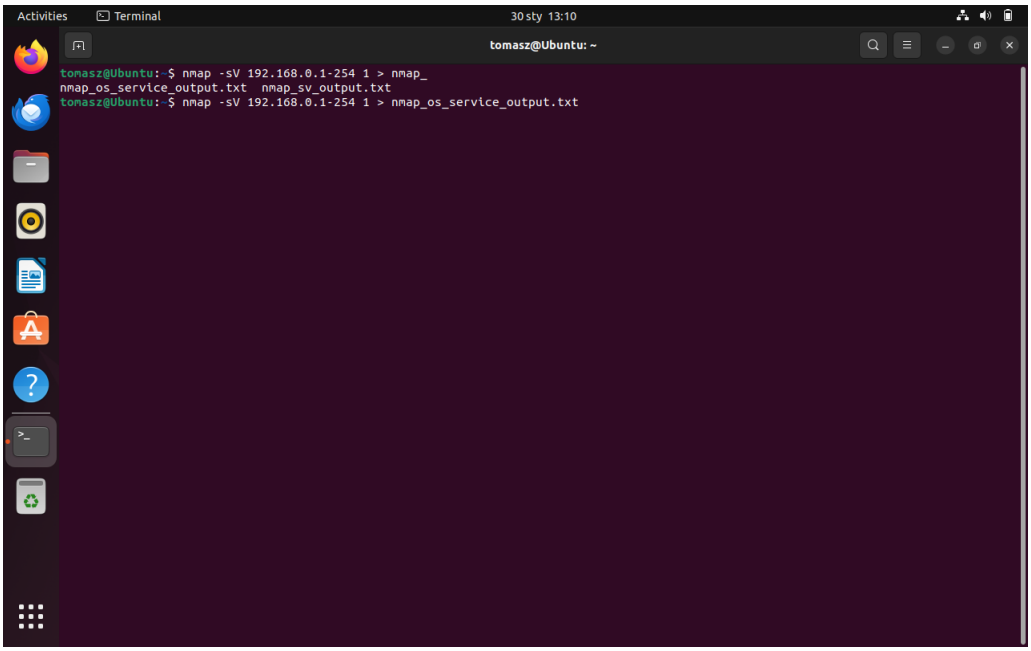

Figure 12: Search for OS and services

File with search results is provided on the shared disk (nmap_os_service_output.txt).

4.3.1 Search results

On gateway (192.168.0.1) there are following services available (all on tcp sockets):

- ssh server on port 22, input is filtered
- dns server on port 53, input is open
- http server on port 80
- netbios-ssn on port 139
- microsoft-ds on port 445
- upnp on port 1900

On host (192.168.0.195) all ports are closed. There are no available services.

5 Task 6

5.1 NMAP script categories

5.1.1 auth

Deal with authentication credentials on the target system

5.1.2 broadcast

Typically these scripts discover hosts not listed on the command line by broadcasting on local network.

5.1.3 brute

Brute scripts use brute force attack to guess authentication credentials of a remote server.

5.1.4 default

Scripts which are the default set and are run when using the $-sC$ or $-A$. Several factors influence decision about choosing a script:

- speed- default scan must finish quickly
- usefulness- default script must produce useful information
- verbosity
- reliability
- intrusiveness- default scripts are almost always in safe cateory
- privacy

5.1.5 discovery

Discover more about the network by quering public registries.

5.1.6 dos

Scripts in these category may cause a denial of service. Sometimes used to test vulnerability to a denial of service method.

5.1.7 exploit

Scripts used to actively exploit some vulerability.

5.1.8 external

These scripts may send data to third-party database or other network resource.

5.1.9 fuzzer

This category contains scripts which are designed to send server software unexpected or randomized fields in each packet. While this technique can useful for finding undiscovered bugs and vulnerabilities in software, it is both a slow process and bandwidth intensive.

5.1.10 intrusive

Scripts that cannot be categorized to safe category.

5.1.11 malware

These scripts check, whether the target platform contains malware or backdoors.

5.1.12 safe

Scripts not designed to crash services, use large amounts of network bandwidt or of other resource, exploit security holes.

5.1.13 version

These scripts detect version features. Cannot be selected explicitly.

5.1.14 vuln

Check for special types of vulerabilities.

5.2 Perform CVE detection using NMAP

5.2.1 Installing cve scripts

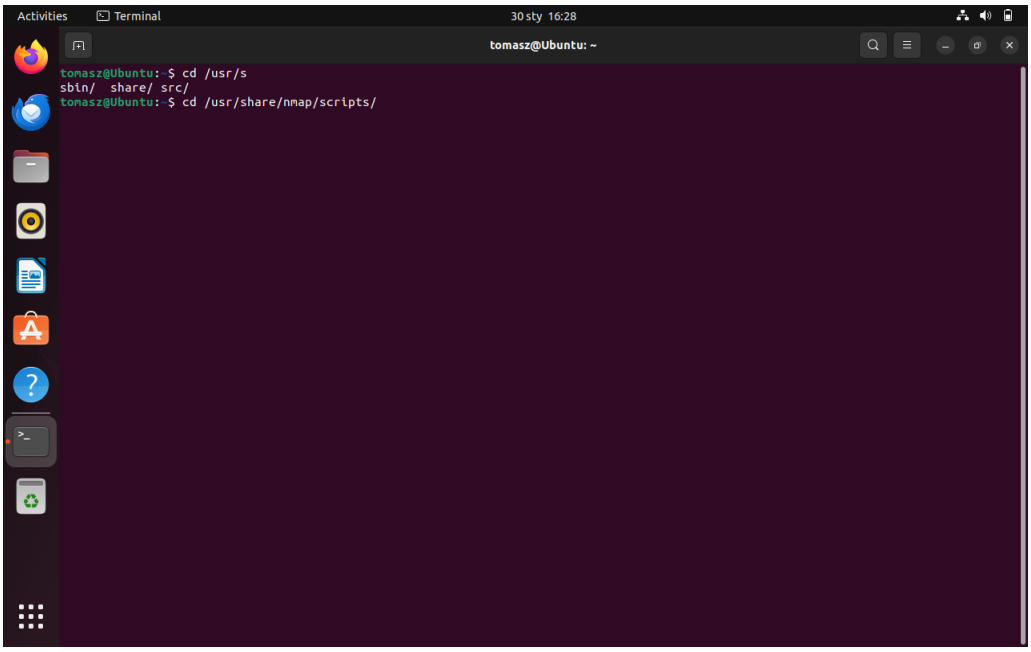

Figure 13: Change directory

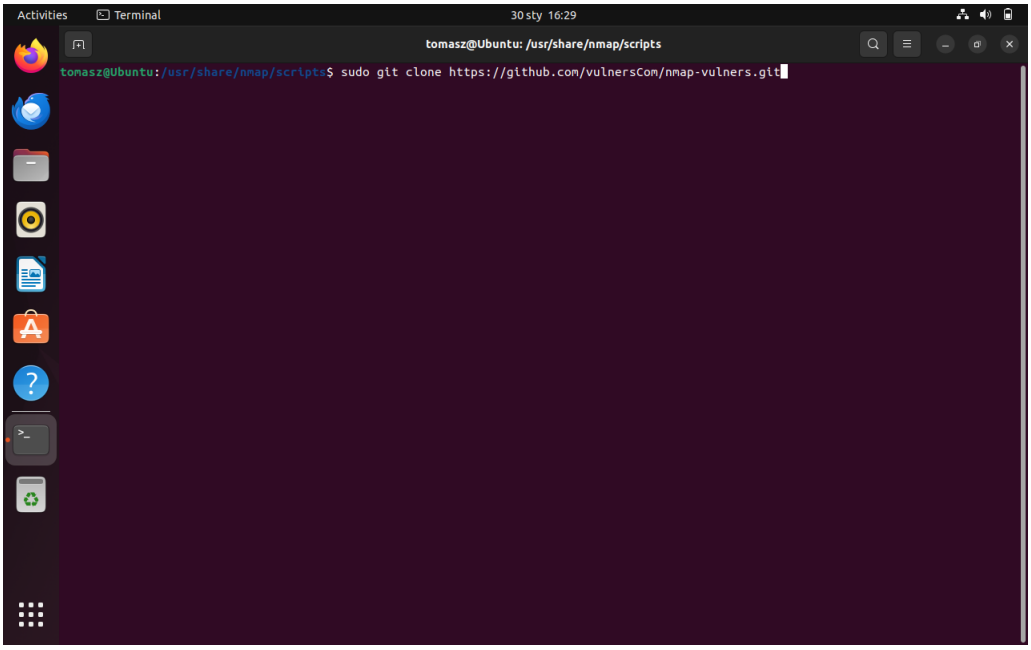

Figure 14: Cloning cve script

5.2.2 CVE searching

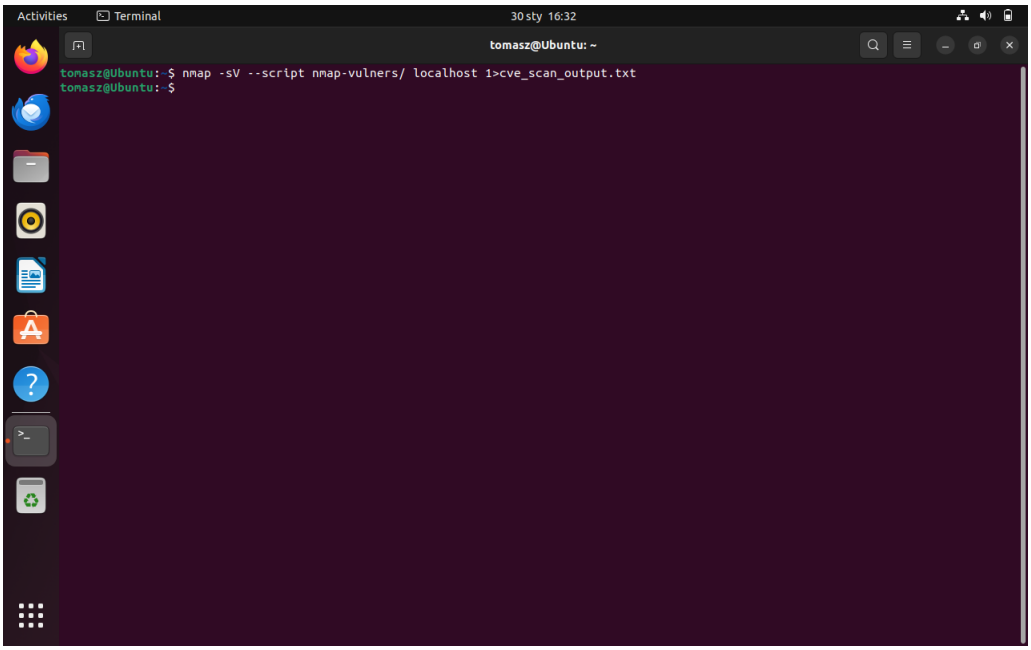

Figure 15: Searching for CVE vulnerabilities

Output of searching is provided in file cve_scan_output.txt.

5.3 Perform malware and backdoors detection scanning

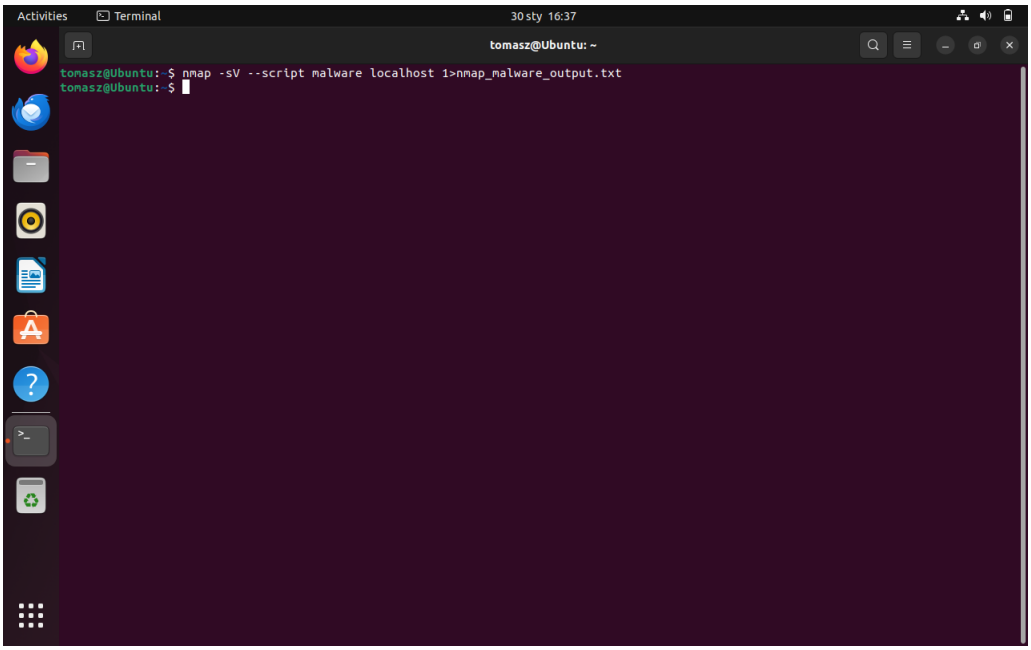

Figure 16: Searching for CVE vulnerabilities

Output of searching is provided in file nmap_malware_output.txt.

6 Task 7

6.1 Capture traffic on ports 67 and 68

6.1.1 Run tcpdump to capture network on public interface enp0s8

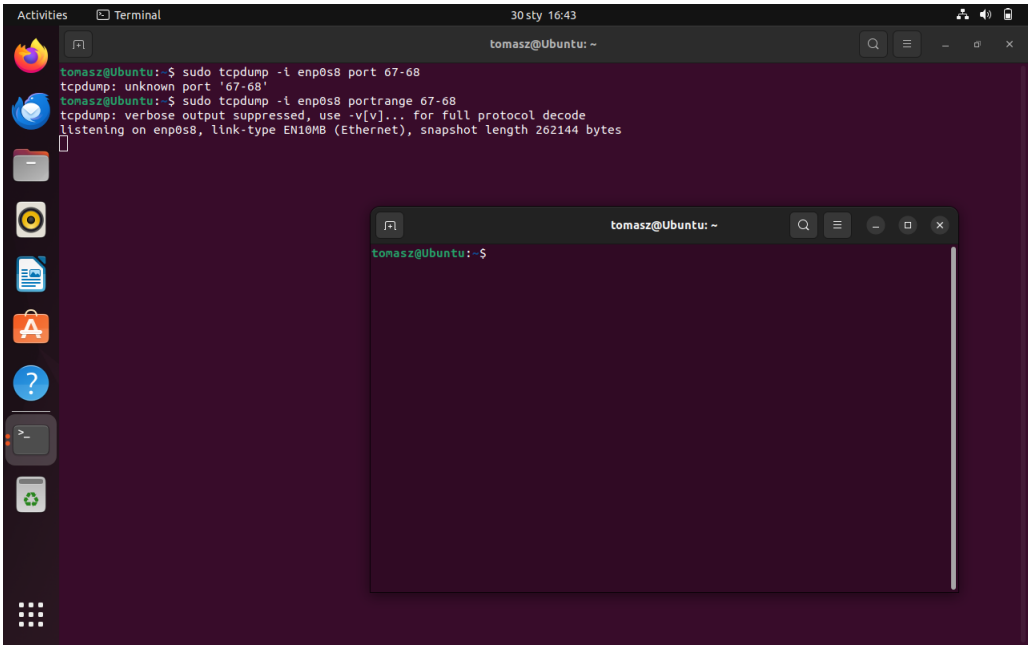

Figure 17: Run tcpdump to listen

6.1.2 Request IP address

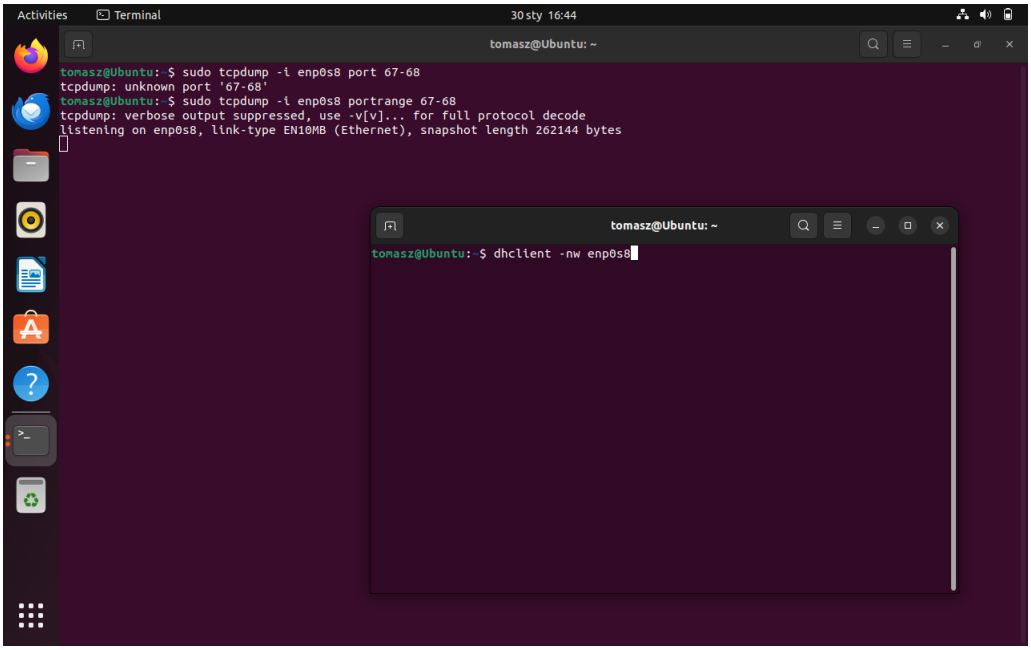

Figure 18: Request IP address

6.1.3 Answer of DHCP server

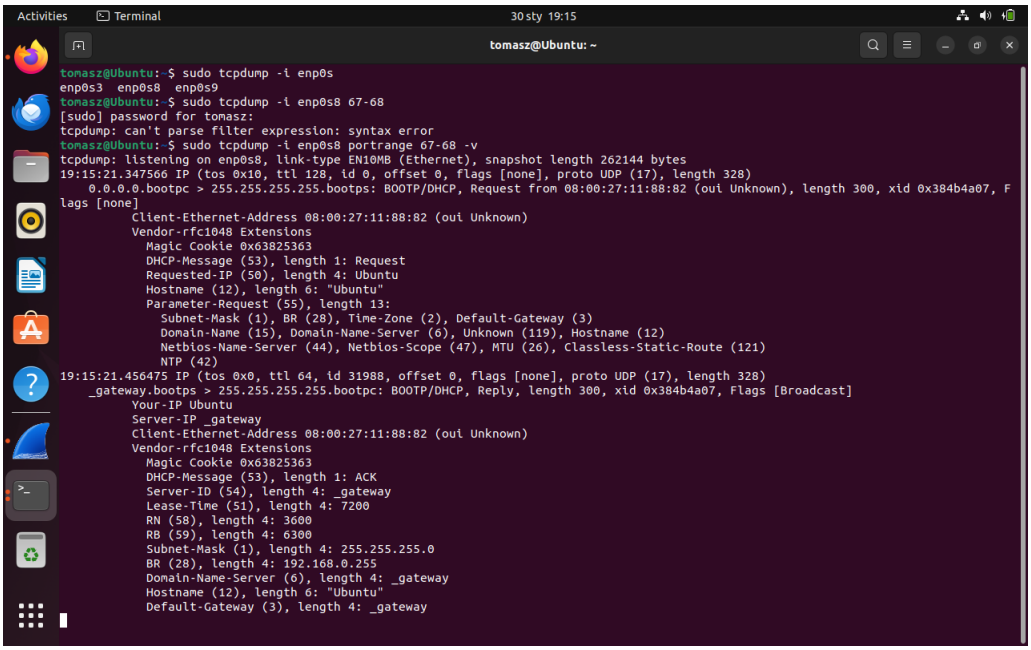

Figure 19: Answer of DHCP request

6.2 Evaluate content

- Client-Ethernet-Address: 08:00:27:11:88:82
- Requested-IP Option 50: Ubuntu
- Hostname Option 12: Ubuntu
- SRC IP Address for the initial Request Message: 0.0.0.0
- DST IP Address for the initial Request Message: 255.255.255.255
- SRC IP Addresses for the ACK Reply Message: 192.168.0.1 (_gateway)
- DST IP Addresses for the ACK Reply Message: 255.255.255.255

6.3 Task 8

6.3.1 Save tcpdump to file of .pcap format

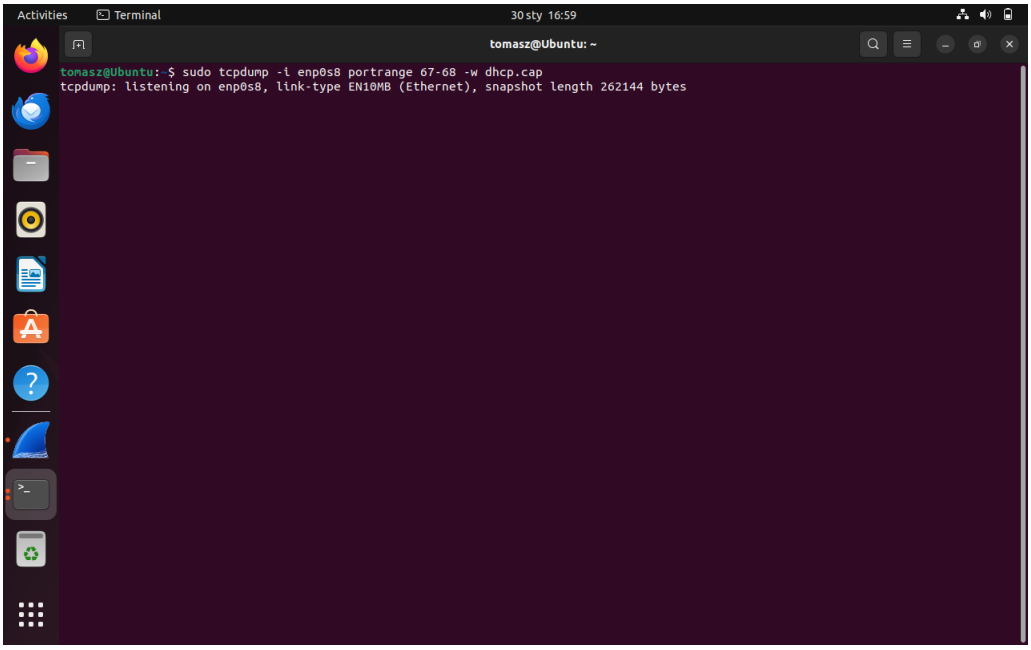

Figure 20: Answer of DHCP request

6.3.2 Wireshark output

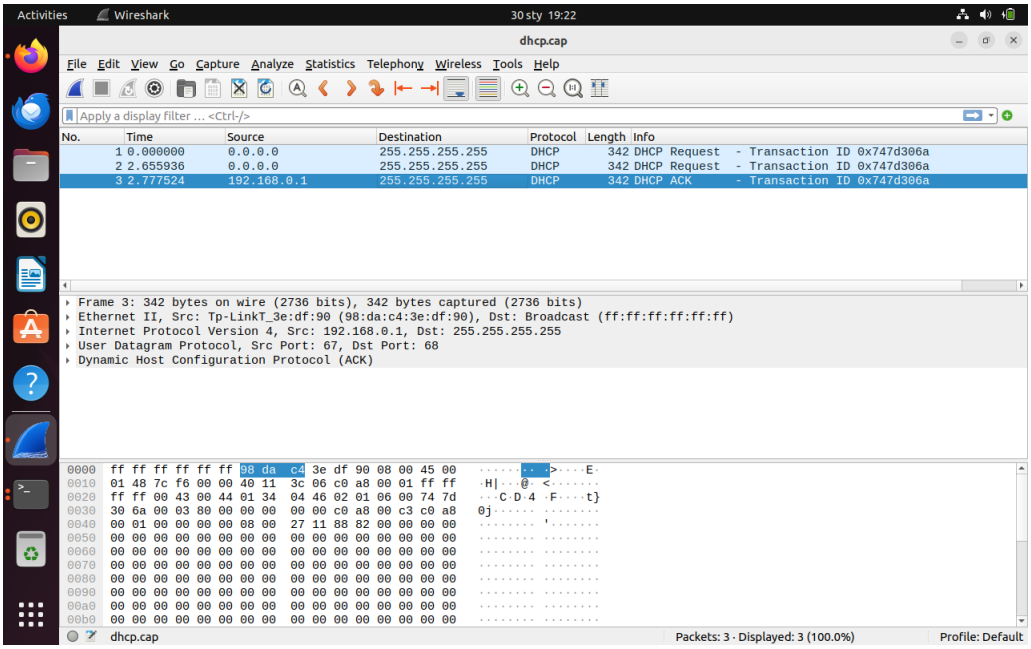

Figure 21: Wireshark output

6.4 Evaluate content

- Client-Ethernet-Address: 08:00:27:11:88:82
- Hostname Option 12: Ubuntu

- SRC IP Address for the initial Request Message: 0.0.0.0
- DST IP Address for the initial Request Message: 255.255.255.255
- SRC IP Addresses for the ACK Reply Message: 192.168.0.1
- DST IP Addresses for the ACK Reply Message: 255.255.255.255

7 Task 9

Task has been made on VM machine and on host machine via host-only adapter.

7.1 Install iperf

On both computers iperf utility has been installed.

```
tomasz@tomasz-Aspire-A515-54G:~$ sudo apt install iperf
[sudo] password for tomasz:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libqt5help5 libqt5sql5 libqt5sql5-sqlite libqt5xml5 libsdl-ttf2.0-0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  iperf
0 upgraded, 1 newly installed, 0 to remove and 19 not upgraded.
Need to get 121 kB of archives.
After this operation, 315 kB of additional disk space will be used.
Get:1 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 iperf amd64 2.1.5+dfsg1-1 [121 kB]
Fetched 121 kB in 1s (105 kB/s)
Selecting previously unselected package iperf.
(Reading database ... 276473 files and directories currently installed.)
Preparing to unpack .../iperf_2.1.5+dfsg1-1_amd64.deb ...
Unpacking iperf (2.1.5+dfsg1-1) ...
Setting up iperf (2.1.5+dfsg1-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
tomasz@tomasz-Aspire-A515-54G:~$
```

Figure 22: Iperf installation

7.2 Measure bandwith

On host computer iperf has been turned on in server mode. VM has played the role of client. Communication has been established on port 2020. Measurement has been made twice. One time for tcp connection, secondly for udp connection.

```
tomasz@tomasz-Aspire-A515-54G: ~
tomasz@tomasz-Aspire-A515-54G:~$ iperf -s -p 2020 -o iperf_output.txt
Output from stdout and stderr will be redirected to file iperf_output.txt
^Ctomasz@tomasz-Aspire-A515-54G:~$ iperf -s -u -p 2020 -o iperf_output.txt
Output from stdout and stderr will be redirected to file iperf_output.txt
```

Figure 23: Measure bandwith

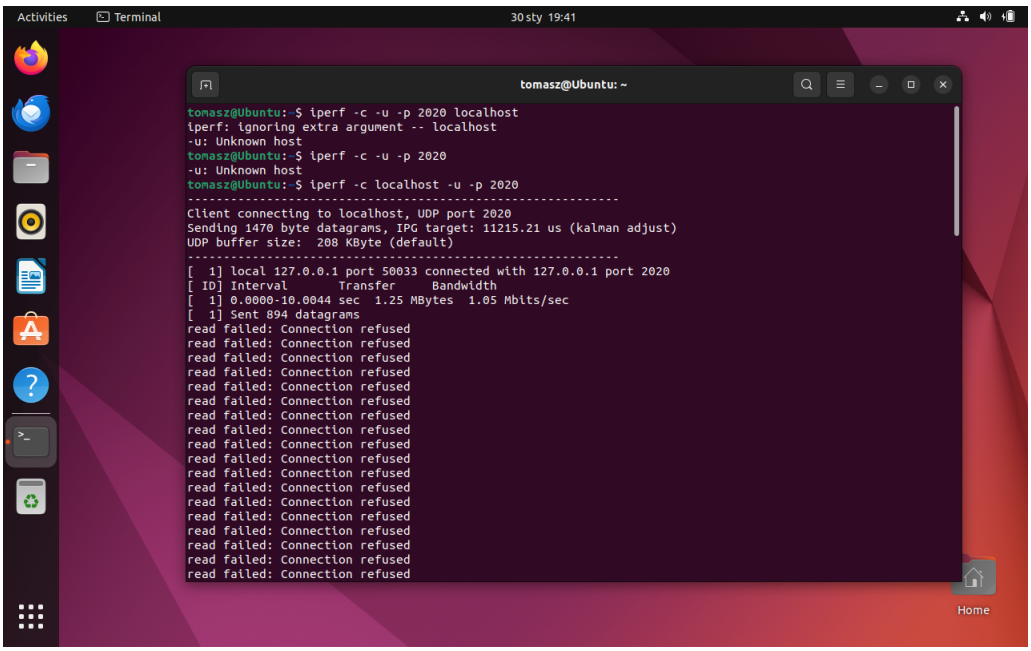

Figure 24: Iperf for client and udp protocol

Command for server:

- tcp: iperf -s -p 2020 -o iperf_output.txt
- udp: iperf -s -u -p 2020 -o iperf_output_udp.txt

Commands for client:

- tcp: iperf -c -p 2020 localhost
- udp: iperf -c -u -p 2020

Ouput of iperf has been provided on disk:

- iperf_output.txt for tcp

Measured values:

- tcp: $8.14 \frac{Gbits}{sec}$
- udp: $1.05 \frac{Mbits}{sec}$