МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

**Лабораторна робота № 6**

з дисципліни «Тестування та контроль якості (QA) вбудованих систем»

|  |
| --- |
| Виконала:  студентка групи ІА-92  Артеменко К.С |

Київ – 2022

**Мета лабараторної роботи:**

Ознайомитися з технологіями автоматизації тестування. Ознайомитися з основами використання мови програмування Python для автоматизації тестування. Написати функцію для підключення клієнта до сервера, використовуючи модуль subprocess.

**Порядок виконання завдання лабораторної роботи**

Налаштування мережного оточення

# Імпортуємо необхідний модуль, для виконання Linux команд із скрипта

import re

import subprocess

# Ip-адреса сервера

server\_ip = 'server'

def client():

# З підключенням до серверу допомагає модуль subprocess та клас subprocess.Popen.

pro = subprocess.Popen(

# -c – запуск клієнта та підключення до сервера, -i – інтервал між виведенням результату тестування,-u - підключення до протоколу UDP

['iperf', '-c', server\_ip, '-i', '1', '-u'],

# STDOUT (стандартний висновок)

stdout=subprocess.PIPE,

# STDERR (стандартний висновок помилок)

stderr=subprocess.PIPE,

)

# wait() повертає ідентифікатор процесу та код завершення, упакований у 16-бітове значення.

pro.wait()

# Повертая вивід та помилку.

return pro.stdout.read().decode('utf8'), pro.stderr.read().decode('utf8')

def parser():

# Парсер повертає список усіх інтервалів з iperf, щоб у подадьшому їх можна було проаналізувати.

keys = ['Interval', 'Transfer', 'Bandwidth']

parsed\_result = []

stdout, stderr = client()

parser\_result\_interval = re.compile(r'[0-9].[0-9]{4}-[0-9].[0-9]{4}')

parser\_result\_transfer = re.compile(r'[0-9]+\sKBytes')

parser\_result\_bandwidth = re.compile(r'[0-9].[0-9]{2}\sMbits/sec')

# Угрупованний список інтервалів

grouped\_result = list(

zip(

parser\_result\_interval.findall(stdout),

parser\_result\_transfer.findall(stdout),

parser\_result\_bandwidth.findall(stdout)

)

)

# Цикл з додаванням даних до parsed result

for res in grouped\_result:

parsed\_result.append({key: value for key, value in list(zip(keys, res))})

# список усіх інтервалів iperf

return parsed\_result, stderr

# Використовується для виконання деякого коду, лише якщо файл було запущено безпосередньо, а не імпортовано.

if \_\_name\_\_ == '\_\_main\_\_':

result, error = parser()

QA завдання:

1) Проаналізувати трафік для сесії на ваш вибір (Skype, SMTP, ftp та ін.)

2) Розробити тестову документацію.

Title: Аналіз трафіку для сесії UDP

Description: Переконайтеся, що клієнт підключаєтеся до iperf-серверу по протоколу UDP

Setup Description:  
PC1 ----ethernet---- PC2

PC1: 10.0.2.15

PC2: 10.0.2.8

Prerequisites:

1) Потрібно дізнатися правильні IP-адреси для потрібних інтерфейсах на PC1 і PC2

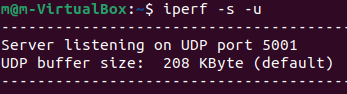
2)Для запуску python коду потрібно на PC2 завнтажити сам python

3) Основний код main.py повинен бути на головній сторінці

4) Встановлений і запущений wireshark на PC1

Steps:

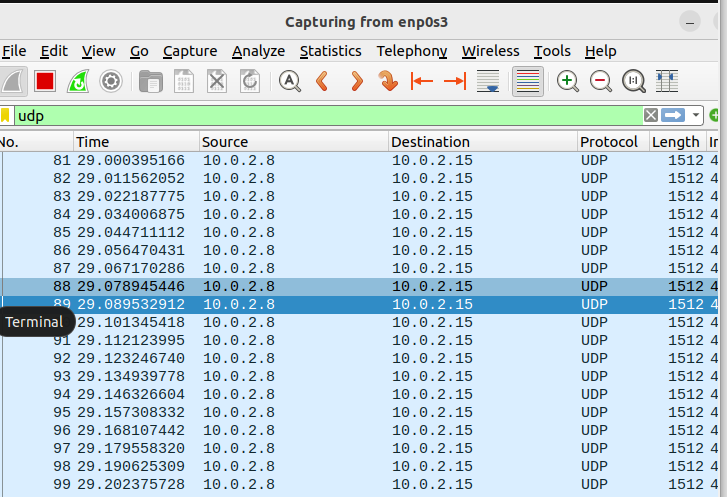
1. На PC1 запустити UDP iperf-сервер



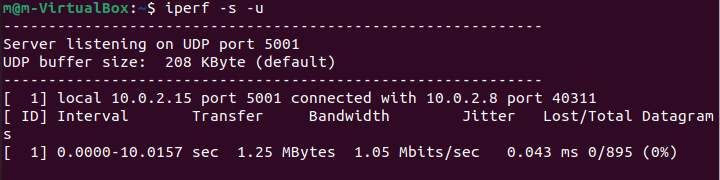
1. Запустити код (main.py) на PC2



1. Перевірити сигнал UDP протоколу на PC1 у Wireshark:



1. Перевірити результат ітервалу на сервері



Посилання на репозиторій: <https://github.com/kkotuha/LR5>

**Висновок:** Під час виконання даної лабораторної роботи я Ознайомилась з технологіями автоматизації тестування. Також я ознайомилася з основами використання мови програмування Python для автоматизації тестування, і написала функцію для підключення клієнта до сервера, використовуючи модуль subprocess.