



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Институт № 3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Лабораторная работа № 6
по дисциплине «WEB технологии»
на тему «Почтовый сервер»

Выполнил
студент группы МЗО-125БВ-24
Егоров А.В.

Приняли
ассистент каф. 304 Борисов А.И.

Москва
2025

Содержание:

| | |
|--|----|
| Постановка задачи..... | 3 |
| Код программы..... | 4 |
| Выполнение работы..... | 9 |
| Анализ работы программы в Wireshark..... | 12 |
| Вывод по работе:..... | 15 |
| Ссылки..... | 15 |

Постановка задачи:

1. Используя почтовые протоколы, отправить письмо на почтовый ящик одnogруппника. В почтовом сообщении написать: тема сообщения – название лабораторной работы, в тексте сообщения написать ФИО и группу отправителя, ФИО и группу получателя и время отправки сообщения. Установить флажок «Важное» для данного письма.

2. Используя протокол, получить письмо, зайти на свой ящик. Получить информацию (и привести в отчете) о количестве писем на ящике. Привести в отчете заголовки некоторых писем (не более 5 шт.). Получить письмо, присланное одnogруппником из почтового ящика. Привести его в отчете.

3. Получить письмо с вложенным файлом, получить вложенный файл.

4. Провести анализ передаваемых команд и данных с помощью программы Wireshark.

Код программы:

```
import imaplib, smtplib, mimetypes, time, os
from datetime import datetime
from tqdm import tqdm
import quopri
import base64

import email
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.header import decode_header
from email.message import EmailMessage

# Добавление почты и пароля для приложений
EMAIL = "eg89150319497@gmail.com"
PASSWD = "wfnm ywuo fzeo brwf"
RECIPIENT_EMAIL = "sawarinecodesu@gmail.com"
RECIPIENT_PASSWD = "jvtw fmya onec drey"

# IMAP сервер gmail
IMAP_SERVER = "imap.gmail.com"
IMAP_PORT = 993

# SMTP сервер gmail
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587

# Настройки изображения для отправки
IMAGE_PATH = "image.png"
IMAGE_SIZE = os.path.getsize(IMAGE_PATH)

# Объем загрузки файла за 1 итерацию цикла
CHUNK_SIZE = 1024 * 1024 # 1 MB

# Декодирование сообщения
def decode_header_value(header_value):
    decoded_parts = decode_header(header_value) # Разбиение заголовков на
    части
    result = ''
    for part, encoding in decoded_parts:
        if isinstance(part, bytes): # Если строка зашифрована
            result += part.decode(encoding or 'utf-8', errors='ignore') #
    Декодирование строки
        else:
            result += part
    return result

# Декодирование содержания письма (если оно закодировано)
def decode_body_part(body_part):
    content_transfer_encoding = body_part.get('Content-Transfer-Encoding')
# Получаем заголовок
    if content_transfer_encoding == 'quoted-printable': # Если кодировка
        'quoted-printable'
```

```

        return quopri.decodestring(body_part.get_payload()).decode('utf-8',
'ignore')
    elif content_transfer_encoding == 'base64': # Или 'base64'
        return base64.b64decode(body_part.get_payload()).decode('utf-8',
'ignore')
    else:
        return body_part.get_payload() # В противном случае просто
возвращаем

# Сохранение файла
def save_attachment(part):
    filename = part.get_filename() # Получаем имя файла из сообщения
    if filename: # Если файл есть
        folder = 'attachments'
        if not os.path.exists(folder): # Если нет директории то создаём её
            os.makedirs(folder)

        filepath = os.path.join(folder, filename) # Заходим в директорию

        # Сохраняем файл
        with open(filepath, 'wb') as f: # Сохраняем файл
            f.write(part.get_payload(decode=True))

        print(f"Файл сохранён: {filepath}")

#####
# SMTP SEND MESSAGE #
#####

msg = EmailMessage() # Создаём экземпляр сообщения

# Передаём заголовки сообщению
msg["From"] = EMAIL
msg["To"] = RECIPIENT_EMAIL
msg["Subject"] = "Задание №6. Почтовый сервер"
msg.set_content(f"Отправил - Егоров Александр Владиславович М30-125БВ-24,
Принял - Егоров Александр Владиславович М30-125БВ-24. Время отправки -
{datetime.now()}").

mime_type, _ = mimetypes.guess_type(IMAGE_PATH) # Присваиваем расширение
файлу
if isinstance(mime_type, str):
    maintype, subtype = mime_type.split("/") # Разделяем расширение файла '/'
для отправки
else:
    print("Error")
    exit(1)

file_data = bytearray() # Массив с данными файла

# Открываем файл в бинарном режиме и читаем его по частям
with open(IMAGE_PATH, 'rb') as file:
    # Создаём прогресс-бар
    with tqdm(total=IMAGE_SIZE, unit='B', unit_scale=True, desc="Загрузка
файла") as pbar:
        while True:
            # Читаем файл по частям

```

```

        chunk = file.read(CHUNK_SIZE)
        if not chunk:
            break # Выход из цикла, если файл прочитан полностью
        file_data.extend(chunk) # Добавляем данные в массив
        pbar.update(len(chunk)) # Обновляем прогресс-бар
        time.sleep(0.1) # Здесь можно сделать задержку для видимости
процесса

print("Файл успешно загружен.")
print("Отправка файла.")

msg.add_attachment(file_data, maintype=maintype, subtype=subtype,
filename=os.path.basename(IMAGE_PATH)) # Добавляем файл в письмо

with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server: # Подключаемся к smtp
серверу
    server.starttls() # Устанавливаем TLS соединение
    server.login(EMAIL, PASSWD) # Логинимся
    server.send_message(msg) # Отправляем письмо

print("Файл отправлен.")

print('-' * 50)

#####
# IMAP RECIVE MESSAGE #
#####

client = imaplib.IMAP4_SSL(IMAP_SERVER, IMAP_PORT) # Создание подключения к
gmail imap серверу
client.login(RECIPIENT_EMAIL, RECIPIENT_PASSWD) # Логин

status_select, messages_count = client.select("INBOX") # Выбор ящика

if isinstance(messages_count[0], bytes): # Проверка на None
    print(f"Status: {status_select}\nMessages:
{messages_count[0].decode()}") # Вывод статуса и количества сообщений
else:
    exit(1)

_, messages = client.search(None, 'ALL') # Подгрузка сообщений

message_ids = messages[0].split()[-5:][::-1] # Выбор 5 сообщений + реверс

print('-' * 50)

# Получение и обработка писем
for msg_id in message_ids:
    status, msg_data = client.fetch(msg_id, '(RFC822)') # Получение писем
    if status != 'OK':
        print(f'Не удалось получить письмо {msg_id.decode()}')
        continue

    raw_email = msg_data[0][1] # Получение сырых данных #type:ignore
    message = email.message_from_bytes(raw_email) # Преобразование байтов в
объект 'email.message.Message' #type:ignore

```

```

# Извлечение заголовков
from_ = decode_header_value(message.get('From'))
subject = decode_header_value(message.get('Subject'))
date = message.get('Date')

# Вывод заголовков
print(f"From: {from_}")
print(f"Subject: {subject}")
print(f>Date: {date}")
print('-' * 50)

# Поиск писем от отправителя
status_search, sender_messages = client.search(None, f'FROM "{EMAIL}"')

# Получаем список ID писем от этого отправителя
sender_message_ids = sender_messages[0].split()

print(f"Письмо от {EMAIL}:")

# Получение письма
if sender_message_ids:
    latest_msg_id = sender_message_ids[-1] # Самое последнее письмо
    status, msg_data = client.fetch(latest_msg_id, '(RFC822)') # Получаем
письмо

    if status != 'OK':
        print(f'Не удалось получить письмо {latest_msg_id.decode()}')
    else:
        raw_email = msg_data[0][1] # Получение сырых данных #type:ignore
        message = email.message_from_bytes(raw_email) # Преобразование
байтов в объект 'email.message.Message' #type:ignore

        # Получение заголовков
        from_ = decode_header_value(message.get('From'))
        subject = decode_header_value(message.get('Subject'))
        date = message.get('Date')

        # Вывод заголовков
        print(f"From: {from_}")
        print(f"Subject: {subject}")
        print(f>Date: {date}")
        print("-" * 50)

        # Обрабатываем тело письма
        if message.is_multipart():
            # Если письмо многокомпонентное (с несколькими частями)
            for part in message.walk():
                content_type = part.get_content_type()
                content_disposition = str(part.get("Content-Disposition"))

                # Если это текстовое содержимое
                if "attachment" not in content_disposition:
                    if content_type == "text/plain":
                        body = decode_body_part(part)
                        print("Текстовое содержимое (plain):")
                        print(body)
                    elif content_type == "text/html":

```

```

        body = decode_body_part(part)
        print("Текстовое содержимое (HTML):")
        print(body)
    else:
        # Если это вложение, сохраняем его
        save_attachment(part)
else:
    # Если письмо состоит из одной части
    body = decode_body_part(message)
    print("Текстовое содержимое:")
    print(body)

    print("-" * 50)
else:
    print(f"Не найдено писем от {EMAIL}")

# Закрытие соединения
client.logout()

```


Скриншоты работы:

9

| 15.8M/15.8M [00:01<00:00, 9.73MB/s]

нял - Егоров Александр Владиславович М30-125БВ-24. Время отправки - 2025-05-13 14:30:35.123995.

Письмо на ящике отправителя:

Задание №6. Почтовый сервер ➤



eg89150319497@gmail.com

кому: sawarinecodesu ▼

Отправил - Егоров Александр Владиславович МЗО-125БВ-24, Принял - Егоров Александр Владиславович МЗО-125БВ-24. Время отправки - 2025-05-13 14:30:35.123995.

Один прикрепленный файл • Просканировано системой Gmail ⓘ



← Ответить

→ Переслать



Письмо на ящике получателя:

Задание №6. Почтовый сервер Входящие x



eg89150319497@gmail.com

кому: мне ▼

Отправил - Егоров Александр Владиславович МЗО-125БВ-24, Принял - Егоров Александр Владиславович МЗО-125БВ-24. Время отправки - 2025-05-13 14:30:35.123995.

Один прикрепленный файл • Просканировано системой Gmail ⓘ



← Ответить

→ Переслать



Анализ работы программы в Wireshark:

SMTP:

Установка соединения:

| | | | | | | | |
|-----|-------------|-------------|-------------|---------|------|------------------------------------|------------------------|
| 583 | 15.24319... | 192.168.... | 173.194.... | TCP | 74 | 47828 → 587 | [SYN] Seq=0 Win=64240 |
| 586 | 15.26248... | 173.194.... | 192.168.... | TCP | 74 | 587 → 47828 | [SYN, ACK] Seq=0 Ack=1 |
| 587 | 15.26250... | 192.168.... | 173.194.... | TCP | 66 | 47828 → 587 | [ACK] Seq=1 Ack=1 Win= |
| 591 | 15.32606... | 173.194.... | 192.168.... | SMTP | 141 | S: 220 smtp.gmail.com ESMTP 2adb30 | |
| 592 | 15.32608... | 192.168.... | 173.194.... | TCP | 66 | 47828 → 587 | [ACK] Seq=1 Ack=76 Win |
| 593 | 15.32688... | 192.168.... | 173.194.... | SMTP | 87 | C: ehlo [192.168.1.32] | |
| 594 | 15.34598... | 173.194.... | 192.168.... | TCP | 66 | 587 → 47828 | [ACK] Seq=76 Ack=22 W |
| 595 | 15.34792... | 173.194.... | 192.168.... | SMTP | 235 | S: 250-smtp.gmail.com at your serv | |
| 596 | 15.34840... | 192.168.... | 173.194.... | SMTP | 76 | C: STARTTLS | |
| 597 | 15.36896... | 173.194.... | 192.168.... | SMTP | 96 | S: 220 2.0.0 Ready to start TLS | |
| 598 | 15.37305... | 192.168.... | 173.194.... | TCP | 1466 | 47828 → 587 | [ACK] Seq=32 Ack=275 W |
| 599 | 15.37307... | 192.168.... | 173.194.... | TLSv1.3 | 187 | Client Hello (SNI=smtp.gmail.com) | |
| 600 | 15.39250... | 173.194.... | 192.168.... | TCP | 66 | 587 → 47828 | [ACK] Seq=275 Ack=1553 |
| 601 | 15.39250... | 173.194.... | 192.168.... | TCP | 78 | 587 → 47828 | [ACK] Seq=275 Ack=32 W |
| 602 | 15.39293... | 173.194.... | 192.168.... | TLSv1.3 | 1466 | Server Hello, Change Cipher Spec | |
| 603 | 15.39349... | 173.194.... | 192.168.... | TCP | 1466 | 587 → 47828 | [PSH, ACK] Seq=1675 Ac |

```
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=4024308649 TSecr=0 WS=128
ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=795827677 TSecr=4024308649 WS=256
Seq=1 Ack=1 Win=64256 Len=0 TSval=4024308668 TSecr=795827677
com ESMTP 2adb3069b0e04-54fc644fae6sm1835067e87.16 - gsmtp
Seq=1 Ack=76 Win=64256 Len=0 TSval=4024308731 TSecr=795827739
.32]
Seq=76 Ack=22 Win=268800 Len=0 TSval=795827759 TSecr=4024308732
com at your service, [109.72.228.71] | SIZE 35882577 | 8BITIME | STARTTLS | ENHANCEDSTATUSCODES | PIPELINING | CHUNKING | SMTPUTF8
to start TLS
Seq=32 Ack=275 Win=64128 Len=1400 TSval=4024308778 TSecr=795827782 [TCP PDU reassembled in 599]
smtp.gmail.com)
Seq=275 Ack=1553 Win=267520 Len=0 TSval=795827806 TSecr=4024308778
Seq=275 Ack=32 Win=268800 Len=0 TSval=795827806 TSecr=4024308754 SLE=1432 SRE=1553
ge Cipher Spec
ACK] Seq=1675 Ack=1553 Win=267520 Len=1400 TSval=795827807 TSecr=4024308778 [TCP PDU reassembled in 604]
ns, Certificate, Certificate Verify, Finished
Seq=1553 Ack=4168 Win=60288 Len=0 TSval=4024308799 TSecr=795827807
```

На данных скриншотах можно наблюдать процесс подключения программы к SMTP серверу smtp.gmail.com и установку TLS соединения (STARTTLS).

Передача заголовков:

```
Frame 649: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits) on interface wlp5s0, id 0
Ethernet II, Src: Intel_98:a7:7e (48:45:20:98:a7:7e), Dst: TpLinkTechno_b5:f9:36 (c4:71:54:b5:f9:36)
Internet Protocol Version 4, Src: 192.168.1.32, Dst: 173.194.220.109
Transmission Control Protocol, Src Port: 47828, Dst Port: 587, Seq: 1771, Ack: 5000, Len: 73
Transport Layer Security
Simple Mail Transfer Protocol
  Command Line: mail from:<eg89150319497@gmail.com> size=21680454\r\n
    Command: mail
    Request parameter: from:<eg89150319497@gmail.com> size=21680454

Frame 655: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface wlp5s0, id 0
Ethernet II, Src: Intel_98:a7:7e (48:45:20:98:a7:7e), Dst: TpLinkTechno_b5:f9:36 (c4:71:54:b5:f9:36)
Internet Protocol Version 4, Src: 192.168.1.32, Dst: 173.194.220.109
Transmission Control Protocol, Src Port: 47828, Dst Port: 587, Seq: 1844, Ack: 5085, Len: 58
Transport Layer Security
Simple Mail Transfer Protocol
  Command Line: rcpt to:<sawarinecodesu@gmail.com>\r\n
    Command: rcpt
    Request parameter: to:<sawarinecodesu@gmail.com>
```


Фрагмент передаваемых данных:

```

> Frame 910: 1466 bytes on wire (11728 bits), 1466 bytes captured (11728 bits) on interface
> Ethernet II, Src: Intel_98:a7:7e (48:45:20:98:a7:7e), Dst: TpLinkTechno_b5:f9:36 (c4:71:1
> Internet Protocol Version 4, Src: 192.168.1.32, Dst: 173.194.220.109
> Transmission Control Protocol, Src Port: 47828, Dst Port: 587, Seq: 231530, Ack: 5255, L
> [13 Reassembled TCP Segments (16406 bytes): #885(922), #886(1400), #893(1400), #894(1400
> Transport Layer Security
> Simple Mail Transfer Protocol
  > Line-based text data (211 lines)
    bZmBk2ATdq0CdOMPPQSTY9tMzw+Bilcggdl0km4IYzsGsz+Rk7yLvPJ+/ZTKozwRpwmwM\r\n
    wSgReqaMzRGdIqkp4KL05KhlcJ2jAClQkawEgmUyPSgj2gxJzogpF8NoqUL0yPTZhmriJ08M5p1x\r\n
    Pvd8EnXae7DWNJcnoMBpm/saL6Za+/dT014PJwPyvm0cCjud0j4NHFCPKWjsnqmJoSuXucIGHkg1\r\n
    Ob8pHi99ddJLVaXvo9Q0DGBjs+FG/eTH2k/WJ8j0htopI1xnYFcU81GZHFRGHSwaR098mmN6kpgg\r\n
    z3veGCyGJ4mKG+0hft0wdZ5TGyRyb7SerhWensb1eJx02w1AFes1tVNr64Zr0fkJGRr7ZIXTFHv5\r\n
    YnJl4z9vkgQr/YEnsi8/lmF6+junAuJQ683pZzC6qieQ5rqmTimGvWnTpBixJvt0Ku4vvdZ8JX\r\n
    04CLqVnYmYzyZ0xUzmuTjJ6Qef5xjPXTCD09GLWHSK7bxzJTNUQ+81qh9JQZSa0Bj6jHWBv0tN4o\r\n
    LVoYrVyBcJa+e3Ak40l+CeMrsoaggCwi7I78n2Il+7W0UnNbzeKpRvA0L/Hs578ABq7sBRZy25uV\r\n
    u3pPTfg//39PA+6uaWhaaQoYS+56e911684AHXRLTUxQugeUxt0eJa3LWRsloKtioutm2t0sTXW9\r\n
    Qya9xKgZobLc9qU7eP8H/4nf/7XXcdZZ+8gnFiWVVCVMsvJZptYVteb+S8GrqgFN47nn6/fytbse\r\n
    40iRY9x1+D5u/8pXePY11/Mzv/EL7N+/LZRWmLFDRmGMygFtgDipowBiCEiktSa7vA6f0CWplnfH\r\n
    Ikafbu4fc0TNzfHFj9xEZQdcds3lqGZJtD350E991tN5+c+/kt/4/d/iF3/637P//LPJq6ekt+Ys\r\n
    MS0WHznKhZ/0ET7wyY9y4lTL6//xd7ji6U9mdhjIo5aU9eRzLzWjNw+o6fdF9W/Y5N53abYl+RS1\r\n
    BkaQCjhH91AnXR7Zbn6aJrWNmoCCD0pMx6qHOMiKlyZwVyafyaTcJ3qnb9N9Zc5cPmMoAkBC13IY\r\n
    0SiijpADJLRE26DPm0f9Py9l84fuov6Pb8H87mt4YNduZpYz1huChqw0RHBZczRmdh04l9f97k/x\r\n
    7773Z3nDH/8LP/m6nxZB+ho0y2Rv0koVeJqC2UG/jnU6CFQR1Hez6lLvdmZlSni6mJCCUMSc8vA\r\n
    uD0oiP9Lk3XBZurnBZwwa56yD2Q0xfgmYq4ayKSFqpNeCuGGNv+JtaCMT8ZG1Jj8vodfisyKaI6\r\n
    42QBuoKJCDGAd8q0Uv055KYBI6pDQnL1+85yP/6o7cwN+PYdc5eXvzSb+GCPWexZdsmlaX+PrD\r\n
    9/Nbv/Jr/PzC7/Gk51700VajosZHXSGbu0w5z+IdV1zBLR/6Ejf90818/H++kc1bLKf8mPBiAv2\r\n
    78MRuPnj/8wnPtCg0ioXX3wOr/n07+Csc/aQxick7QuZy6kkuhRtDckGCR3RkEI0IQEa0ypSPUvY\r\n
    POTzn/ky73/PB3noaw8zv7km5sDJpVMctiMezC1+s6G1GZ+XGCL4AdtUUSscWSl0Up0aTrL7LGe\r\n
    GXdq2BFOnTjKrgv0lx0i1qdB1kJKmj+ZWpe6J6BPhN6gVzZ1B/vaIUWZuXdw6d7omdWUHKJPwF5\r\n
    0q/a1KC0YUZptg8yX77lvt73d+/hts/zeXnXMTvvu03u0q5lz0/aT0qLu9xScGyW7ew+fpLWP3i\r\n
    XYTLmbWrZH6sDHZutk9L69bWmHJvwNQF0u1ji9YGjSLEqEXGTC4xknLC2YrQTHJPp5J4c0z9uWgC\r\n
    GGTnu9D9rK6Gn76GxtgeztCZPDFmjWbGFMNxnOqldT3Fro42GpxSBCEzv+PHjpG1xmSF0R6LK53\r\n
    soZqXVK/I20IpT+asMYSYivpvib3cI7u7GRrSYWTP0cSIGwtESNH9C0d5iNmMwnUzvXJZrmYiH\r\n
    2DHG9Ca16X5j07Zds1XeqZJMXg8dVS1ar3EbGYVI3LGI3c99MvXuzZASo7sfZekrD5JHI+q5W\r\n
    Uc0UU0Lri15xlJvWcAszJctJAXqpUjWyPmpg26WfUs0UoYdl53D7tazSz/Mwmtfwxve9U5e96a3\r\n
    8l9+6F9x3q7NL1Iwivlnog8KsaS2aTHThCga2u7VaVu5XnVdy34ZS6KydSQyMQaB0qPLcY51Ph\r\n
    yiImrLM07RgfPNpaKmeJvswFkryTsfX+Z5z09FwpT8v8Q3XGX21LR88QfUmk8WgV194H5h2seJb\r\n
    Pv+Vr/Glu+7i3oceglwzt2WBCy84h+d895Xs2LqFZC0j5ZYbb/g0Bz/3Dbbt3Mb0ndt50rXnsH/3\r\n
    boaDmrg8IsfI4rGTHAkrfPhue/mr277KloUBz3/aU3nuM65jMG5ofSCMxtSDIc64YqBWA7VMkPv\r\n
    EhZVgRPMONHYmMpBSKImxQ4r0R7bbu4B0SDpXDMtlagqVnMA9ZY0FqAy9aQk8eXuQnOyVykhdy\r\n
    TRRng6ld86IjV2s01b3ZqYUc4QUsJhmJtd4Kt3f4NPfeBmvvV38KTzzubieUTxbQFeTkQig54\r\n
    ODPkVLNnueDsFQUQvhnjjCW0nkgimFAM4vIOu2LiSsVksH0r720SWY5CJHuVqQHnWmSpBlzwjg\r\n
    JEIXdaXS84tNy6CqGK+0cMZBFNOJdqIh8Sh40e4+atf50uP3Ms3TjzK4tISpxZPsTwe0+ZIMlJX\r\n
    ZG1Amf6kNg1stLqisjXz51/Ec/bv5RnXPZ2ZGc0b/vzPGS2t8vq3vZ7RYMCP/vt4Sd/8keY37SD\r\n
    pZ0ie1OqIqUg/oWsBc6Tp4Frk4TFnSnAp4UY0P+eHmw3BTLr8VJZere5E+10zYem17cOgESenM\r\n
    HafXAJLwnadP030rRMxglcaoyJZ5zd+9/b383q//Ijd+7Bae+vSrMJAj1VWE41F0eMlLGjMNHb\r\n
    lds/8M9Uq4Z6525S2xK1pDjrNNF7mvEYpWS9rKqKztwIFCAD0eObFpSirgfkfHKEHMU/S+GzQsL\r\n
    PPvp13Pe2Qf42Mdu5PDJR6mHA7Ifs80MSGdXvOuNbyYPLa/6gVdhBo4v3nQb737n33LXh2/k+U95\r\n
    Nj/0X1/Hjm3zNCghmmJor0BN73k/73vru/jRSy/GzsySvJgxKwmsPY0uT/WA1vQy6EFkx6NImiP\r\n
    jgatDE2I7LvwLBaeeglf+dqXueBpV20GFWlpjKuHEoLgXLRV0SsGZKtp2gaTM2bg8CcWyT4xHMz\r\n
    k8e0RRXb6Sa1Jsfa2I8ZDIYkFGnUoIa16HdaQPZBUj/rAWhF0x7jjGLsA8GLZgInRLLVeAECJHmX\r\n
    u3XJR9FoV+Ud1sXoHQuowHvfn89dVeGcBFxUzUkCG0L9CMW0X4yWpebr1hWZNRVDr3GS9FtVtFG0\r\n
    SCXuQMLZQpDwsKKF0CVIJuusHmqLlyAEaufEJVN0ctGKbl0rxbgY51QBqBgL4V10IyEewkFVYZw\r\n
    kmnHDWQxaCmlaNaq2wPPBpwjGSOK8Ev1dN+M0SIBDbAMqRFBjKvVgIDWuEqi4sgLopuh/0WIoZZ4W\r\n
    z44tZ82MUWLqC7nFVhV1Xcv5LxTAitH4IAY1rbXs/+U7+uBRZKwrmpoQ0DnhnJJJQ4wSFlaCaR\r\n
    1n8lgSNVSS900WJLQnA2Gt82qIyYgbPoSEMUElIikobgsrsnJ/le1/5wt7wN29gdnmRudk5wi9A\r\n
    F6vEn6LkmTxlhFwmhE6J4TbmVGBzitB6rDZS0/gGjC7fy6NVRiu7Bt6tLBEz05KKqouuPJV9V6D\r\n
    6/0krTkLrJuz2aS/WQJVLGDEi07qlamsE3Jt6MysRoy7krxswajniGLMGjdUOVFXQ1aNpm38xGyb\r\n

```

Заккрытие соединения:

| | | | | | |
|-------|-------------|------------|------------|------|----------------------------|
| 24595 | 24.42743... | 192.168... | 173.194... | SMTP | 94 C: QUIT |
| 24597 | 24.44649... | 173.194... | 192.168... | TCP | 66 587 → 47828 [ACK] Seq= |
| 24598 | 24.44649... | 173.194... | 192.168... | SMTP | 167 S: 221 2.0.0 closing c |
| 24599 | 24.44669... | 173.194... | 192.168... | TCP | 66 587 → 47828 [FIN, ACK] |
| 24600 | 24.44679... | 192.168... | 173.194... | TCP | 66 47828 → 587 [FIN, ACK] |
| 24606 | 24.46531... | 173.194... | 192.168... | TCP | 66 587 → 47828 [ACK] Seq= |

IMAP:

Установка соединения:

| | | | | | |
|-------|-------------|----------------|-------------|---------|------|
| 24605 | 24.45346... | 192.168.1.32 | 209.85.2... | TCP | 74 |
| 24607 | 24.47332... | 209.85.233.109 | 192.168.... | TCP | 74 |
| 24608 | 24.47335... | 192.168.1.32 | 209.85.2... | TCP | 66 |
| 24609 | 24.47413... | 192.168.1.32 | 209.85.2... | TCP | 1466 |
| 24610 | 24.47417... | 192.168.1.32 | 209.85.2... | TLSv1.3 | 187 |
| 24611 | 24.49623... | 209.85.233.109 | 192.168.... | TCP | 66 |
| 24612 | 24.49623... | 209.85.233.109 | 192.168.... | TLSv1.3 | 2866 |
| 24613 | 24.49623... | 209.85.233.109 | 192.168.... | TLSv1.3 | 1158 |
| 24614 | 24.49627... | 192.168.1.32 | 209.85.2... | TCP | 66 |
| 24615 | 24.49630... | 192.168.1.32 | 209.85.2... | TCP | 66 |
| 24616 | 24.49783... | 192.168.1.32 | 209.85.2... | TLSv1.3 | 146 |
| 24617 | 24.51844... | 209.85.233.109 | 192.168.... | IMAP | 701 |

```
35114 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1382680381 TSecr=0 WS=128
993 → 35114 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval=520111236 TSecr=1382680381
35114 → 993 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1382680401 TSecr=520111236
35114 → 993 [ACK] Seq=1 Ack=1 Win=64256 Len=1400 TSval=1382680401 TSecr=520111236 [TCP PDU reassembled]
Client Hello (SNI=imap.gmail.com)
993 → 35114 [ACK] Seq=1 Ack=1522 Win=267520 Len=0 TSval=520111257 TSecr=1382680401
Server Hello, Change Cipher Spec
Encrypted Extensions, Certificate, Certificate Verify, Finished
35114 → 993 [ACK] Seq=1522 Ack=2801 Win=61568 Len=0 TSval=1382680424 TSecr=520111257
35114 → 993 [ACK] Seq=1522 Ack=3893 Win=60544 Len=0 TSval=1382680424 TSecr=520111257
Change Cipher Spec, Finished
Response: * OK Gimap ready for requests from 109.72.228.71 i10mb14553038ltk
```

На данных скриншотах wireshark происходит установка TCP соединения с TLS шифрованием.

Передача данных по imap:

| | | | | | |
|-------|-------------|-------------|-------------|------|---|
| 24617 | 24.51844... | 209.85.2... | 192.168.... | IMAP | 701 Response: * OK Gimap ready for requests from 109.72.228.71 i10mb14553038ltk |
| 24618 | 24.51868... | 192.168.... | 209.85.2... | IMAP | 106 Request: INAD0 CAPABILITY |
| 24619 | 24.53903... | 209.85.2... | 192.168.... | IMAP | 297 Response: INAD0 OK Thats all she wrote! i10mb14553038ltk |
| 24620 | 24.53926... | 192.168.... | 209.85.2... | IMAP | 148 Request: INAD1 LOGIN sawarinecodesu@gmail.com "jvtw fmya onec drey" |
| 24639 | 25.48686... | 209.85.2... | 192.168.... | IMAP | 365 Response: INAD1 OK sawarinecodesu@gmail.com authenticated (Success) |
| 24640 | 25.48726... | 192.168.... | 209.85.2... | IMAP | 108 Request: INAD2 SELECT INBOX |
| 24651 | 26.16947... | 209.85.2... | 192.168.... | IMAP | 461 Response: INAD2 OK [READ-WRITE] INBOX selected. (Success) [THROTTLED] |
| 24652 | 26.16986... | 192.168.... | 209.85.2... | IMAP | 106 Request: INAD3 SEARCH ALL |
| 24659 | 26.78400... | 209.85.2... | 192.168.... | IMAP | 377 Response: INAD3 OK SEARCH completed (Success) [THROTTLED] |
| 24663 | 26.78438... | 192.168.... | 209.85.2... | IMAP | 114 Request: INAD4 FETCH 213 (RFC822) |
| 41582 | 30.49119... | 192.168.... | 209.85.2... | IMAP | 114 Request: INAD5 FETCH 212 (RFC822) |
| 57996 | 34.06275... | 192.168.... | 209.85.2... | IMAP | 114 Request: INAD6 FETCH 211 (RFC822) |
| 64925 | 37.82354... | 192.168.... | 209.85.2... | IMAP | 114 Request: INAD7 FETCH 210 (RFC822) |
| 64969 | 38.87163... | 192.168.... | 209.85.2... | IMAP | 114 Request: INAD8 FETCH 209 (RFC822) |
| 65022 | 40.09939... | 192.168.... | 209.85.2... | IMAP | 133 Request: INAD9 SEARCH FROM "eg89150319497@gmail.com" |
| 65027 | 41.32563... | 192.168.... | 209.85.2... | IMAP | 115 Request: INAD10 FETCH 213 (RFC822) |
| 72973 | 45.70441... | 192.168.... | 209.85.2... | IMAP | 103 Request: INAD11 LOGOUT |

Завершение соединения:

```
78 35114 → 993 [ACK] Seq=2199 Ack=86970695 Win=1916928 Len=0 TSval=1382704011 TSecr=520134690 SLE=
66 35114 → 993 [FIN, ACK] Seq=2199 Ack=86970695 Win=1916928 Len=0 TSval=1382704011 TSecr=520134690
66 993 → 35114 [ACK] Seq=86970695 Ack=2200 Win=267520 Len=0 TSval=520134866 TSecr=1382704011
```

| | | | | |
|-------|-------------|-------------|-------------|-----|
| 73001 | 48.08369... | 192.168.... | 209.85.2... | TCP |
| 73003 | 48.08404... | 192.168.... | 209.85.2... | TCP |
| 73007 | 48.10569... | 209.85.2... | 192.168.... | TCP |

Вывод по работе:

Работа наглядно продемонстрировала ключевые принципы функционирования почтовых протоколов, в частности, использование SMTP (Simple Mail Transfer Protocol) для отправки электронных писем и IMAP (Internet Message Access Protocol) для их получения и управления сообщениями на сервере. Кроме того, исследование подчеркнуло важность анализа сетевого трафика для глубокого понимания процесса обмена данными между клиентом и сервером. Такой анализ позволяет детально изучить этапы установления соединения, передачи информации и возможные уязвимости в системе, что является важным аспектом обеспечения безопасности и оптимизации работы почтовых сервисов.

Ссылки:

Все файлы можно посмотреть на github:

https://github.com/swrneko/mai_shit/tree/main/1lvl_2sem/web/laba_4