

Департамент образования города Москвы

Государственное автономное образовательное учреждение высшего
образования города Москва «Московский Городской Педагогический
Университет»

Институт цифрового образования
Департамент информатики, управления и технологий

ЛАБОРАТОРНАЯ РАБОТА №5
по дисциплине «Распределенные системы»
Направление подготовки 38.03.05 – «бизнес-информатика»
Профиль подготовки «Аналитика данных и эффективное управление» (очная
форма обучения)

Выполнил:
Студент группы АДЭУ-221
Черенков Иван Романович

Проверил:
Босенко Т.М., доцент

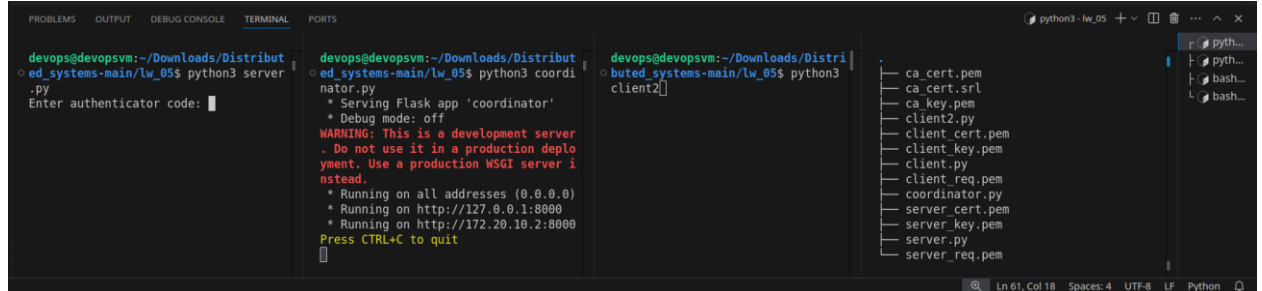
Москва
2024

Вариант 20. Сделать двойную аутентификацию файла server.py

Go Run Terminal Help

```

1  from flask import Flask, request, g
2  import ssl
3  import binascii
4  import pyotp
5  from cryptography.hazmat.primitives import hashes
6  from cryptography.hazmat.primitives.asymmetric import padding
7  from cryptography.x509 import load_pem_x509_certificate
8  from cryptography.hazmat.backends import default_backend
9  from cryptography.fernet import Fernet
10
11 app = Flask(__name__)
12
13 @app.before_request
14 def verify_client_cert():
15     cert = request.get_json()['certificate']
16     # Verify certificate against CA
17     if not verify_certificate(cert):
18         return "Invalid certificate", 401
19
20 @app.route('/api/data', methods=['POST'])
21 def get_data():
22     data = request.get_json()['data']
23     # Decrypt data
24     decrypted_data = decrypt_data(data)
25     # Process decrypted data
26     return {'result': 'ok'}
27
28 def verify_certificate(cert_pem):
29     # Load certificate
30     certificate = load_pem_x509_certificate(cert_pem.encode(), default_backend())
31     # Verify certificate against CA
32     try:
33         certificate.public_key().verify(
34             certificate.signature,
35             certificate.tbs_certificate_bytes,
36             padding.PKCS1v15(),
37             hashes.SHA256()
38         )
39         return True
40     except:
41         return False
42
43 def decrypt_data(encrypted_data):
44     # Load encryption key
45     key = open('encryption_key.txt', 'rb').read()
46     # Decrypt data
47     cipher = Fernet(key)
48     return cipher.decrypt(encrypted_data.encode())
49
50 if __name__ == '__main__':
51     totp = pyotp.TOTP("mcudistributedsys")
52     for x in range(3):
53         result = totp.verify(input("Enter authenticator code: "))
54         if result:
55             print("Authentication successful.")
56             context = ssl.create_default_context(ssl.Purpose.CLIENT_AUTH)
57             context.load_cert_chain('server_cert.pem', 'server_key.pem')
58             context.verify_mode = ssl.CERT_REQUIRED
59             context.load_verify_locations('ca_cert.pem')
60             app.run(host='0.0.0.0', port=5000, ssl_context=context)
61             break
62     else:
63         print(f"Invalid. Try again. Attempts: {3-x-1}")
```



```
devops@devopsvm:~/Downloads/Distributed_system$ s-main/lw_05$ python3 server.py
Enter authenticator code: 185545
Authentication successful.
* Serving Flask app 'server'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on https://127.0.0.1:5000
* Running on https://172.20.10.2:5000
Press CTRL+C to quit

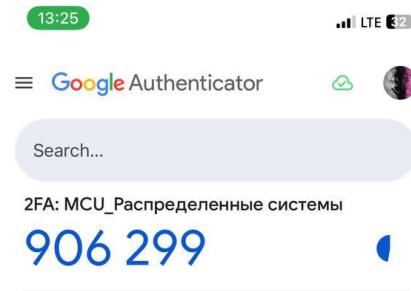
devops@devopsvm:~/Downloads/Distributed_system$ s-main/lw_05$ python3 coordinator.py
* Serving Flask app 'coordinator'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8000
* Running on http://172.20.10.2:8000
Press CTRL+C to quit

devops@devopsvm:~/Downloads/Distributed_system$ s-main/lw_05$ python3 client2.py

devops@devopsvm:~/Downloads/Distributed_systems-main$ in/lw_05$ tree
.
├── ca_cert.pem
├── ca_cert.srl
├── ca_key.pem
├── client2.py
├── client_cert.pem
├── client_key.pem
├── client.py
├── client_req.pem
├── coordinator.py
├── py.py
├── server_cert.pem
├── server_key.pem
├── server.py
└── server_req.pem

1 directory, 14 files
devops@devopsvm:~/Downloads/Distributed_systems-main$ in/lw_05$
```

Как это выглядит на телефоне:



Дерево проекта:

```
devops@devopsvm:~/Downloads/Distributed_systems-main$ in/lw_05$ tree
.
├── ca_cert.pem
├── ca_cert.srl
├── ca_key.pem
├── client2.py
├── client_cert.pem
├── client_key.pem
├── client.py
├── client_req.pem
├── coordinator.py
├── py.py
├── server_cert.pem
├── server_key.pem
├── server.py
└── server_req.pem
```