MIT World Peace University

Wireless Devices and Mobile Security
Third Year B. Tech, Semester 5

Program to Send OTP to Mobile Phone using Python and Twilio API

Lab Assignment 6

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

November 26, 2023

Contents

1 Aim

To write a program to send OTP to mobile phone using Python and Twilio API.

2 Objectives

1. To learn how to use Twilio API to send SMS.
2. To learn how to use Python to send SMS.
3. To learn how to use Python to generate OTP.

3 Theory

3.1 Twilio API

1. **Overview:** The Twilio API is a cloud communications platform that allows developers to integrate messaging, voice, and video capabilities into their applications. It provides a set of RESTful APIs for building communication solutions.
2. **Key Features:**
   - Sending and receiving SMS and MMS messages.
   - Making and receiving voice calls.
   - Video conferencing capabilities.
   - Integration with various programming languages.
3. **Use Cases:**
   - Implementing two-factor authentication (2FA).
   - Building notification systems.
   - Creating interactive voice response (IVR) systems.

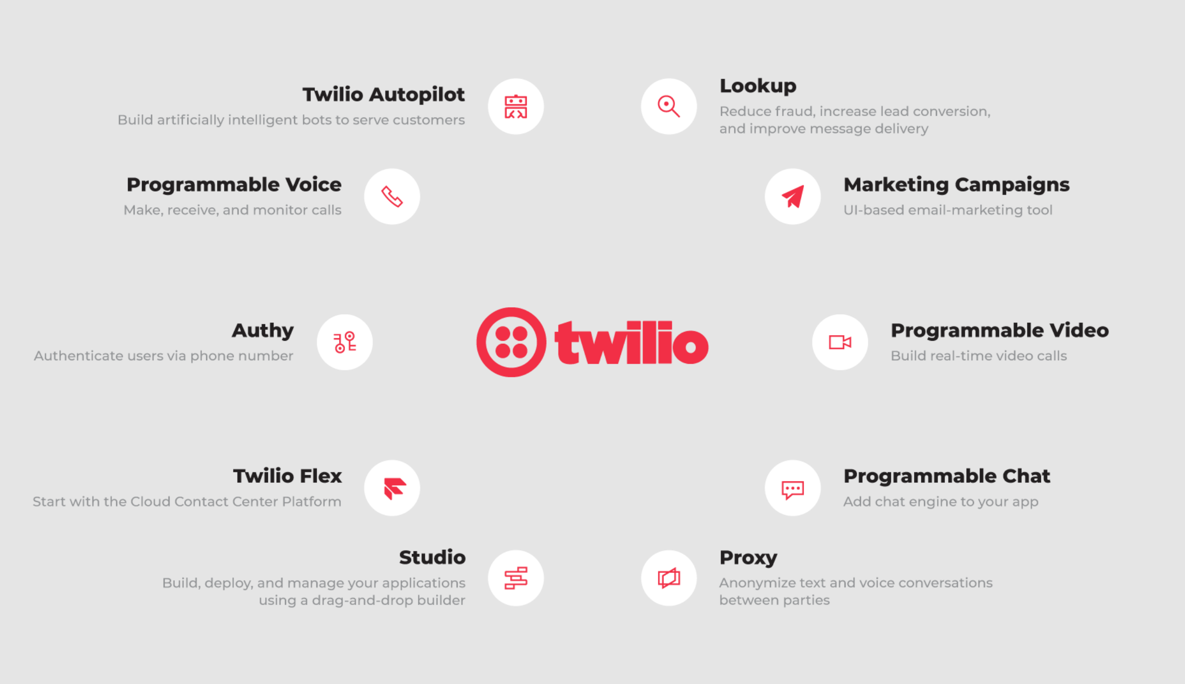

Twilio Autopilot
Build artificially intelligent bots to serve customers
Lookup
Reduce fraud, increase lead conversion, and improve message delivery
Programmable Voice
Make, receive, and monitor calls
Marketing Campaigns
UI-based email-marketing tool
Authy
Authenticate users via phone number
twilio
Programmable Video
Build real-time video calls
Twilio Flex
Start with the Cloud Contact Center Platform
Programmable Chat
Add chat engine to your app
Studio
Build, deploy, and manage your applications using a drag-and-drop builder
Proxy
Anonymize text and voice conversations between parties

Figure 1: Twilio Features

3.2 Pricing of Twilio API

1. **Billing Model:** Twilio charges based on usage, with costs associated with each message, call, or other communication type.

2. **Factors Affecting Pricing:**

   - Message type (SMS, MMS).
   - Destination country for calls and messages.
   - Type of phone number used (local, toll-free).
   - Volume of usage.

3. **Pricing Details:** Twilio provides a detailed pricing page on their official website, allowing users to estimate costs based on their specific use case.

3.3 OTP Generation with Python

1. **Python Libraries:** Use libraries like ‘pyotp‘ or ‘onetimepass‘ to generate OTPs (One-Time Passwords) in Python.

2. **Time-based OTP (TOTP):** TOTP is a widely used algorithm for generating OTPs based on the current time.

3. **Implementation:** Sample Python code involves importing the library, creating an OTP object, and generating OTPs based on the provided key.

4. **Security Considerations:** Ensure the secure storage of secret keys and follow best practices for OTP implementation.

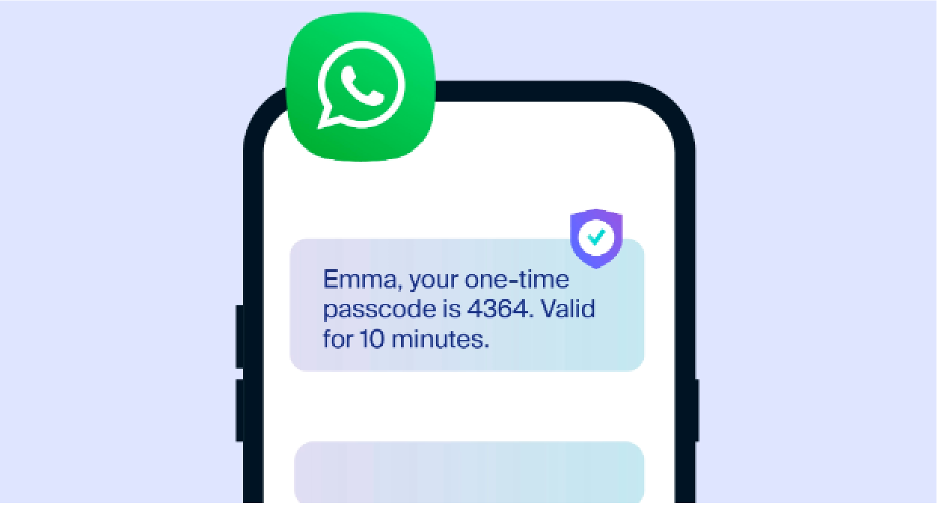


Figure 2: OTP Example SMS

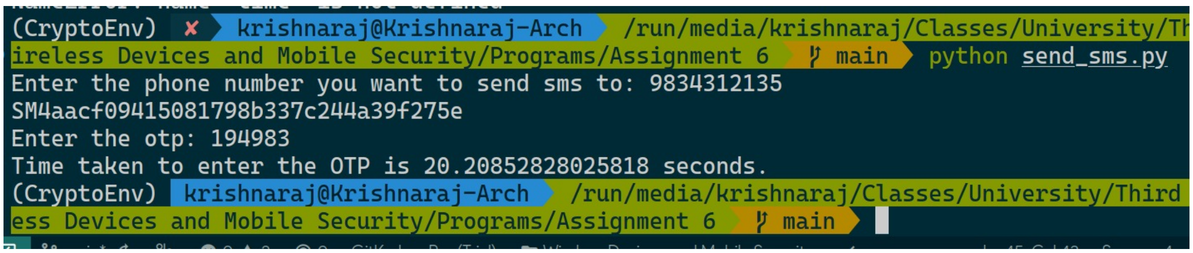
3.4 Working of OTPs for Enhanced Security

1. **Two-Factor Authentication (2FA):** OTPs are commonly used as a second factor to enhance security along with passwords.
2. **Dynamic Authentication Codes:** OTPs change dynamically at regular intervals, providing a time-sensitive layer of security.
3. **Use in Identity Verification:** OTPs are employed in identity verification processes, ensuring that the entity accessing the system has possession of the valid OTP.
4. **Avoiding Replay Attacks:** OTPs are designed to be used only once, mitigating the risk of replay attacks.

4 Platform

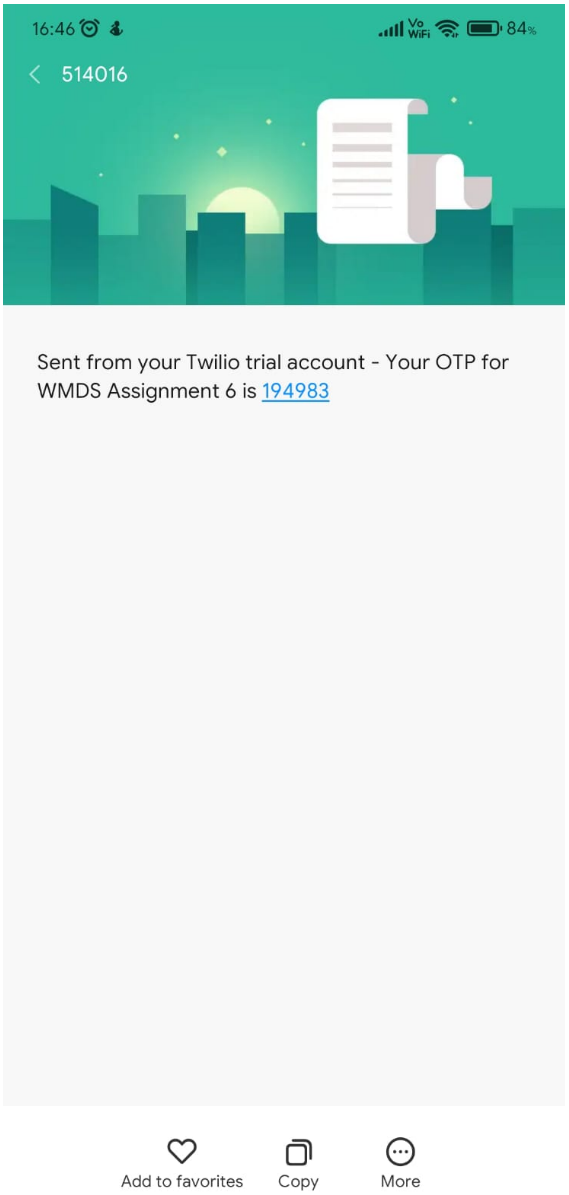
Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Python 3.10.1

5 Input and Output



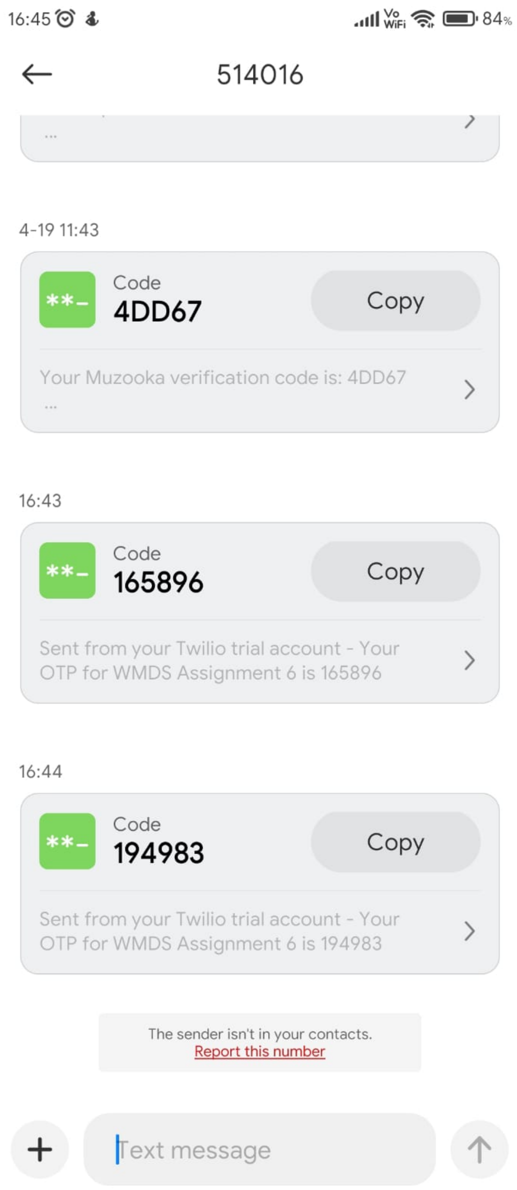
```
NameError: name 'time' is not defined
(CryptoEnv)  ✘  krishnaraj@Krishnaraj-Arch  /run/media/krishnaraj/Classes/University/Th
ireless Devices and Mobile Security/Programs/Assignment 6  ᚴ main  python send_sms.py
Enter the phone number you want to send sms to: 9834312135
SM4aacf09415081798b337c244a39f275e
Enter the otp: 194983
Time taken to enter the OTP is 20.20852828025818 seconds.
(CryptoEnv)  krishnaraj@Krishnaraj-Arch  /run/media/krishnaraj/Classes/University/Third
ess Devices and Mobile Security/Programs/Assignment 6  ᚴ main 
```

Figure 3: Terminal Input and Output



Sent from your Twilio trial account - Your OTP for WMDS Assignment 6 is 194983

Figure 4: Message Received on Phone (+919834312135)



...

4-19 11:43

Code
4DD67

Your Muzooka verification code is: 4DD67
...

16:43

Code
165896

Sent from your Twilio trial account - Your OTP for WMDS Assignment 6 is 165896

16:44

Code
194983

Sent from your Twilio trial account - Your OTP for WMDS Assignment 6 is 194983

The sender isn't in your contacts.
Report this number

Figure 5: Other Messages Received on Phone (+919834312135)

6 Code

```
from twilio.rest import Client
import time
account_sid = 'AC2729594588fa8c7cd37d00283acdd58e'
auth_token = 'c88cb19255f4c77cce0baebc73c72df4'
client = Client(account_sid, auth_token)

def make_otp():
    import random
    otp = ""
    for i in range(6):
        otp += str(random.randint(0,9))
```

```python
    return otp


if __name__ == "__main__":
    otp = make_otp()

    body_string = "Your OTP for WMDS Assignment 6 is " + otp
    send_phone_number = '9834312135'
    try:

        send_phone_number = int(input("Enter the phone number you want to send sms to: "))
        # verify the validity of the phone number
        while len(str(send_phone_number)) != 10:
            print("Invalid phone number. Please try again.")
            send_phone_number = int(input("Enter the phone number you want to send sms to: "))
    except ValueError:
        print("Invalid phone number. Please try again.")
        send_phone_number = int(input("Enter the phone number you want to send sms to: "))

    message = client.messages.create(
    from_='+12165034403',
    body=body_string,
    to='+91'+str(send_phone_number)
    )
    print(message.sid)

    # start timer here.
    start_time = time.time()
    # wait for user to enter the otp
    user_otp = input("Enter the otp: ")
    # if it is correct, print the time taken to enter the otp.
    if user_otp == otp:
        end_time = time.time()
        time_taken = end_time - start_time
        print("Time taken to enter the OTP is " + str(time_taken) + " seconds.")

        # if the time taken is more than 60 seconds, stop the program.
        if time_taken > 60:
            print("Time taken to enter the OTP is more than 60 seconds. Please try again.")
            user_otp = input("Enter the otp: ")
    # if it is wrong, stop the program.
    if user_otp != otp:
        print("Wrong OTP. Please try again.")
        user_otp = input("Enter the otp: ")
```

Listing 1: Script to Send SMS via Twilio API

7 Conclusion

Thus, we have successfully used Twilio API to send OTP to a mobile phone using Python, and verified it on the script.

References

[1] Twilio Documentation.
https://www.twilio.com/docs

[2] Twilio Pricing.
https://www.twilio.com/pricing

[3] PyOTP Documentation.
https://github.com/pyauth/pyotp

[4] onetimepass Documentation.
https://github.com/tadeck/onetimepass

[5] NIST Digital Identity Guidelines.
https://www.nist.gov/publications/digital-identity-guidelines