NGUYỄN QUANG HUY - DHKL16A1HN

MSV: 22174600113

Bài thực hành 2:

```python
import hashlib
# Mật khẩu lưu trữ dưới dạng mã băm SHA-256
stored_password = hashlib.sha256(b"mypassword").hexdigest()

# Yêu cầu người dùng nhập mật khẩu
password = input("Nhập mật khẩu: ")
hashed_password = hashlib.sha256(password.encode()).hexdigest()

if hashed_password == stored_password:
    print("Xác thực thành công!")
else:
    print("Xác thực thất bại!")
```

```
Xác thực thất bại!
```

```python
import pyotp
import time
# Tạo khóa bí mật và mã OTP
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)

print("Mã OTP của bạn là:", totp.now())

# Yêu cầu nhập mã OTP
otp_input = input("Nhập mã OTP: ")

if totp.verify(otp_input):
    print("Xác thực thành công!")
else:
    print("Xác thực thất bại!")
```

```
Mã OTP của bạn là: 298288
Xác thực thành công!
```

```python
import pyotp
import time
# Bước 1: Xác thực bằng mật khẩu
stored_password = hashlib.sha256(b"mypassword").hexdigest()
# Mật khẩu lưu trữ dưới dạng mã băm SHA-256
password = input("Nhập mật khẩu: ")
hashed_password = hashlib.sha256(password.encode()).hexdigest()

if hashed_password == stored_password:
    print("Xác thực mật khẩu thành công! Chuyển sang bước xác thực bằng mã OTP.")
else:
    print("Xác thực mật khẩu thất bại!")
    exit()  # Thoát chương trình nếu sai mật khẩu

# Bước 2: Xác thực bằng mã OTP nếu mật khẩu đúng
# Tạo khóa bí mật và mã OTP
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)

# In mã OTP (trong thực tế sẽ được gửi qua SMS hoặc Email)
print("Mã OTP của bạn là:", totp.now())

# Yêu cầu người dùng nhập mã OTP
otp_input = input("Nhập mã OTP: ")

if totp.verify(otp_input):
    print("Xác thực hai yếu tố thành công!")
else:
    print("Xác thực bước 2, mã OTP thất bại!")
```

```
Xác thực mật khẩu thành công! Chuyển sang bước xác thực bằng mã OTP.
Mã OTP của bạn là: 672994
Xác thực bước 2, mã OTP thất bại!
```

1. Tại sao xác thực hai yếu tố (2FA) lại an toàn hơn so với xác thực chỉ bằng

mật khẩu?- Vì nó bổ sung một lớp bảo mật thứ hai, giảm rủi ro ngay cả khi mật

khẩu bị lộ.- Mật khẩu (Yếu tố thứ nhất): dễ bị rò rỉ qua các hình thức tấn công như

phishing, keylogger, hoặc bị đoán brute-force.- OTP (Yếu tố thứ hai– thường là thứ bạn có, ví dụ ứng dụng trên điện

thoại): là mã ngắn, thay đổi liên tục (ví dụ mỗi 30 giây), nên ngay cả khi

hacker có được mật khẩu, họ vẫn cần điện thoại hoặc thiết bị sinh mã

OTP đểđăngnhập.

=> Vì vậy, 2FA giúp ngăn chặn truy cập trái phép kể cả khi mật khẩu đã bị đánh

cắp.

2. Có thể cải tiến thêm tính năng bảo mật nào cho chương trình này không?

có

Tính năng bảo mật

Môtả

Giới hạn số lần nhập sai OTP

Hạn chế thời gian hiệu lực của OTP

Sử dụng HTTPS khi truyền OTP (nếu web)

Lưu khóa secret một cách an toàn

Thêmmãhóathôngtin đầu vào/đầura

Kết hợp thêm các yếu tố xác thực khác

Ngăn brute-force (ví dụ: khóa tài khoản sau 3

lần nhập sai).

MãOTPmặcđịnhhếthạnsau30s,cần đảm

bảo chương trình kiểm tra đúng thời điểm.

Tránh bị đánh cắp OTPqua mạng.

Không in ra màn hình hoặc lưuở dạng mã

hóa.

Nếu triển khai thực tế, cần tránh rò rỉ dữ liệu

người dùng.

Ví dụ: nhận diện thiết bị, vị trí địa lý, hoặc

email xác nhận.

3. Dựa trên kết quả thực hành, Anh/Chị rút ra được bài học gì về tính bảo mật

của mật khẩu và mãOTP?

Bài học chính: Mật khẩu không đủ an toàn nếu dùng một mình. OTP giúp tăng

cường bảo mật.

- Mật khẩu dễbị rò rỉ, đoán được, hoặc lặp lại giữa các tài khoản.- OTP sinh ra tức thời, chỉ có hiệu lực trong thời gian rất ngắn, và chỉ sử

dụng mộtlần.- Kết hợp cả hai giúp bảo vệ tài khoản trước các mối đe dọa mạng ngày

càng tinh vi.- Nhưng OTP cũng không hoàn toàn tuyệt đối an toàn nếu secret bị lộ

hoặc thiết bị sinh OTP bị mất.

=> Bài học: Hệ thống an toàn nên dựa vào nhiều lớp bảo vệ và người dùng cần

có ý thức bảo mật như: không chia sẻ OTP, không dùng cùng mật khẩu cho

nhiều tài khoản, và bật 2FA bất cứ khi nào có thể.