Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра комп’ютерної інженерії та програмування

ЗВІТ

з лабораторної роботи № 8

з дисципліни «Сучасні технології безпечного програмування»

«**TIME-BASED ONE TIME PASSWORD**»

Виконав:

студент гр. КН-Н922б

Кулик Д.І.

Перевірив:

Бульба С. С.

Харків – 2022

Мета роботи: Дослідити і реалізувати механізм генерації одноразових паролів TOTP.

Індивідуальне завдання

Time-based One Time Password. Створити програму, що демонструє роботу розробленого алгоритму. Організувати взаємодію з мобільним додатком Google Authenticator.

Хід роботи

TOTP означає Time-based One-Time Passwords і є поширеною формою двофакторної автентифікації (2FA). Унікальні цифрові паролі генеруються за стандартизованим алгоритмом, який використовує поточний час як вхідні дані. Паролі на основі часу доступні в автономному режимі та забезпечують зручність і підвищену безпеку облікового запису, якщо використовувати їх як другий фактор.

Для реалізації двофакторної аутентифікації з використанням TOTP необхідно враховувати основну вимогу – пароль повинен створюватись на стороні користувача, а також постійно змінюватись.

Вирішення цього завдання може виглядати так:

Коли користувач включає двофакторну автентифікацію, відбувається таке

• Внутрішній сервер створює секретний ключ для цього користувача

• Потім сервер передає цей секретний ключ до телефонної програми користувача

• Телефонний додаток ініціалізує лічильник

• Телефонний додаток генерує одноразовий пароль, використовуючи цей секретний ключ та лічильник

• Телефонний додаток змінює лічильник через певний інтервал і відновлює одноразовий пароль, роблячи його динамічним.

Однак, у цій послідовності дій є кілька проблем. Перша з них полягає в тому, як програма буде генерувати одноразовий пароль. З цією проблемою справляється попередник методу TOTP – алгоритм HOTP.

HOTP перекладається як "Одноразовий пароль на основі HMAC". Цей алгоритм опубліковано інженерною групою Інтернету (IETF) як RFC4226. HOTP визначає алгоритм створення одноразового пароля із секретного ключа та лічильника.

Важливі фрагменти програми

Приведена нижче функція оновлює тимчасовий пароль у файлі (див. рис. 1)

```
def task():
    """Функція створює одноразовий пароль на стороні користувача,
    що записується в файл. Функція емулює пристрій клієнта"""
    log.debug('Generating TOTP...')
    totp = get_totp_token(config.SECRET)
    write_to_file(totp, config.FILENAME)
    log.debug(f'Wrote new TOTP [{totp}] to file.')


def set_background_update():
    """Функція створює одноразовий пароль на стороні користувача,
    що змінюється через певний проміжок часу (30 секунд)
    Через 30 секунд попередній код стає невалідним"""
    scheduler = BackgroundScheduler()
    job = scheduler.add_job(task, 'interval', seconds=config.INTERVAL)
    job.func()
    scheduler.start()
```

Рисунок 1 – Реалізоване оновлення TOTP

Приведені нижче функції генерують TOTP (див. рис. 2)

```
def get_hotp_token(secret, intervals_no):
    """Фукнція визначає алгоритм створення одноразового пароля
    із секретного ключа та лічильника на основі HMAC"""
    key = base64.b32decode(secret, True)
    msg = struct.pack(">Q", intervals_no)
    h = hmac.new(key, msg, hashlib.sha1).digest()
    o = o = h[19] & 15
    h = (struct.unpack(">I", h[o:o + 4])[0] & 0x7fffffff) % 1000000
    return h


def get_totp_token(secret):
    """Функція, що генерує одноразовий пароль із шести символів
    за допомогою алгоритма HOTP"""
    x = str(get_hotp_token(secret, intervals_no=int(time.time()) // INTERVAL))
    while len(x) != 6:
        x += '0'
    return x
```

Рисунок 2 – Генерація TOTP

```
if __name__ == '__main__':
    set_background_update()
    login()
    entered_code = enter_code()
    while not validate(entered_code):
        entered_code = input('Невірний код! Спробуйте ще раз:\n')
    print('Авторизація пройшла успішно!')
```

Рисунок 3 – Перевірка на TOTP

config.py | password.txt

```
FILENAME = 'password.txt'
INTERVAL = 30
SECRET = 'MNUGC2DBGBZQ====='


CREDENTIALS = {
    'username': 'demiurg',
    'password': '321'
}
```

Рисунок 4 – Файл config.py

Результати роботи програми

```
C:\Users\Daniil\PycharmProjects\stbp\Scripts\python.exe C:/Users/Daniil/PycharmProjects/stbp/LABS/kulyk08/main.py
Введіть нікнейм: demiurg
Введіть пароль: 321
Введіть код верифікації: 46263
Невірний код! Спробуйте ще раз:
fkweio
Невірний код! Спробуйте ще раз:
580365
Авторизація пройшла успішно!
```

Рисунок 5 – Результат виконання програми

password.txt

```
580365
```

Рисунок 6 – Файл password.txt

password.txt | file.log

```
2022-11-27 22:07:39 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:07:39 | DEBUG - Wrote new TOTP [310582] to file. (function = task)
2022-11-27 22:08:09 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:08:09 | DEBUG - Wrote new TOTP [489168] to file. (function = task)
2022-11-27 22:08:39 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:08:39 | DEBUG - Wrote new TOTP [247422] to file. (function = task)
2022-11-27 22:09:09 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:09:09 | DEBUG - Wrote new TOTP [582140] to file. (function = task)
2022-11-27 22:09:39 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:09:39 | DEBUG - Wrote new TOTP [922349] to file. (function = task)
2022-11-27 22:10:09 | DEBUG - Generating TOTP... (function = task)
2022-11-27 22:10:09 | DEBUG - Wrote new TOTP [580365] to file. (function = task)
```

Рисунок 7 – Файл log.file

Висновки: в результаті виконання лабораторної роботи було досліджено і реалізувано механізм генерації одноразових паролів TOTP.