Hướng dẫn kết nối với hệ thống truyền dữ liệu VGU Robocon 2019

21/04/2019

Võ Lê Tùng

Outline

- Tổng quan hệ thống
  - Secure Hypertext Transfer Protool (HTTPS)
  - JSON Web Token (JWT)
  - Websocket
- Phương thức hoạt động
  - Đăng nhập
  - Nhận dữ liệu
- Demo
- Một vài lưu ý

Tổng quan hệ thống

Secure Hyper Text Transfer Protocol (HTTPS)

- Xác thực kết nối giữa 2 bên
- Mã hoá thông tin trao đổi

JSON Web Token (JWT)

- Xác thực người dùng
- Hạn chế thời gian đăng nhập
- Hạn chế số lượng đăng nhập

Websocket

- Hỗ trợ kết nối thời gian thực
- Đảm bảo độ trễ thấp
- Được hỗ trợ bởi nhiều nền tảng

Phương thức hoạt động

Hệ thống mã hoá

Mỗi đội sẽ được BTC cung cấp 3 tập tin, sử dụng để xác thức và mã hoá thông tin được trao đổi giữa server và mỗi đội

- `cacert.pem` : Xác nhận danh tính
- `clientcert.pem` : Mã hoá thông tin
- `clientkey.pem` : Giải mã thông tin

Đăng nhập

- Đăng nhập bằng tài khoản được cung cấp
- Nhận mã xác thực khi đăng nhập thành công
    - Thời hạn sử dụng 10 phút

Nhận dữ liệu

- Thiết lập kết nối bằng mã xác thực
- Dữ liệu được gửi về ngay sau khi kết nối
- Xác nhận để được gửi tin nhắn tiếp theo

Demo

Ngôn ngữ và thư viện

- Ngôn ngữ lập trình Python3.
- Thư viện:
  - json (Đọc và dịch chuỗi json)
  - ssl (Cài đặt hệ thống bảo mật)
  - time (Xử lý dữ liệu thời gian)
  - urllib.request (Gửi yêu cầu HTTP)
  - pathlib (Tạo đường dẫn đến đúng tập tin)
  - websocket-client (Thiết lập kết nối Websocket)

```
pip install websocket-client
```

Khai báo các thư viện được sử dụng

```
import websocket
import json
import ssl
import time
import urllib.request
from pathlib import Path
```

Các hàm được sử dụng

Cài đặt và trả về thông tin cho giao thức mã hoá HTTPS

```
def makeSSLContext(ca, crt, key):
    sslCTX = ssl.create_default_context(
        purpose=ssl.Purpose.SERVER_AUTH,
        cafile=ca
    )

    sslCTX.load_cert_chain(crt, key)

    return sslCTX
```

Trả về chuỗi JSON chứa thông tin đăng nhập

```
def makeJSONCredentials(username, password):
    creds = {
        "username": username,
        "password": password
    }

    return json.dumps(creds).encode("utf-8")
```

Cài đặt và trả về thông tin của yêu cầu HTTPS

```
def makeRequestHeader(url, contentType, content):
    req = urllib.request.Request(url)

    req.add_header('Content-Type', contentType)
    req.add_header('Content-Length', len(content))

    return req
```

Gửi yêu cầu đăng nhập và trả về mã xác thực

```
def getToken(url, username, password,
             ca, crt, key):
    reqSSLContext = makeSSLContext(ca, crt, key)
    reqContent = makeJSONCredentials(username, password)
    req = makeRequestHeader(
        url,
        'application/json; charset=utf-8',
        reqContent
    )

    # Gửi yêu cầu và nhận kết quả trả về
    resp = urllib.request.urlopen(
        req, data=reqContent, context=reqSSLContext)

    # Đọc và trả về mã xác thực
    respBody = resp.read()
    respBodyJSON = json.loads(respBody.decode('utf-8'))

    return respBodyJSON["token"]
```

Đăng nhập và nhận dữ liệu

Cài đặt thông tin của giao thức mã hoá cho Websocket

```
CA_CRT = Path("cacert.pem")
CRT = Path("clientcert.pem")
KEY = Path("clientkey.pem")

sslopt = {
    'cert_reqs': ssl.PROTOCOL_SSLv23,
    'keyfile': KEY,
    'certfile': CRT,
    'ca_certs': CA_CRT,
}
```

Nhận mã xác thực và thêm mã xác thực vào thông tin yêu cầu Websocket

```
HOST = "vgurobocon2019.local"
PORT = 4433

url = 'https://%s:%s/subscribe' % (HOST, PORT)
token = getToken(url,
                 'user', 'password',
                 CA_CRT, CRT, KEY)

header = {
    'Authorization': 'Bearer %s' % (token)
}
```

Thiết lập kết nối Websocket và bắt đầu nhận dữ liệu

```
url = 'wss://%s:%s/data' % (HOST, PORT)
ws = websocket.create_connection(url,
                                 header=header,
                                 sslopt=sslopt)

while True:
    msg = ws.recv()
    packet = json.loads(msg.decode('utf-8'))
    print(packet)
    ws.send(json.dumps({'finished': True}).encode('utf-8'))
```

Một vài lưu ý

- Tối đa 2 kết nối cho mỗi tài khoản
- Mọi dữ liệu trao đổi đều được mã hoá dưới dạng base64
- Ngắt kết nối nếu hệ thống nhận nhiều hơn 30 tin nhắn mỗi giây từ mỗi đội
- Chú ý thiết lập kết nối lại với hệ thống nếu có lỗi xảy ra