پروژه عملی http درس برنامه نویسی وب دکتر الوانی

آرمان حاتمی ۹۷۳۰۰۰۸

سوال اول :

طبق صحبت های گفته شده در کلاس میدانیم میشود پورت های مختلف یک سرویس را از طریق اینترفیس های مختلفی باز کنیم. مثلا اگر پورتی را بر روی سیستم خود و اینترفیس wifi باز کنیم همه افراد میتوانند با ip به ما وصل شوند و محدودیتی ندارند ولی اگر پورتی را بر روی loopback داخلی یعنی 127.0.0.1 باز کنیم فقط خودمان آن را داریم و برای بقیه قابل دسترسی نیست.

سوال دوم :

این الگوریتم از کلمات Transport Layer Security ساخته شده است که به معنای ایمن سازی داده های تبادل شده بین مبدا و مقصد است. هدف این پروتکل حفظ حریم خصوصی در ارتباطات بین client و server است و به دلیل الگوریتم پیچیده و بهتر ان جایگزین پروتکل قبلی یعنی ssl شده است که از رمزنگاری ضعیف تری استفاده میکرد.
هدف tls این است که داده های تبادل شده توسط هیچ کس شنود نشود و اگر شنود شد غیر قابل فهم و غیر قابل استفاده باشند برای این کار از ۳ بخش استفاده میکند :

بخش اول : رمز نگاری کردن داده ها
بخش دوم : اهراز هویت طرفیت تبادل
بخش سوم : چک کردن دستکاری نشدن داده های ارسال شده

به این ترتیب با اعمال این ۳ بخش میتوانیم از انتقال امن داده ها اطمینان حاصل کنیم.

سوال سوم :

برای پیاده سازی این پروژه از ۲ روش مختلف رفتیم :
روش اول همان روشی است که در کلاس توسط اقای فاطمی اجرا شد و با استفاده از یک image که از داکر هاب دانلود کردیم توانستیم یک jwt authentication پیاده سازی کنیم با استفاده از فایل docker compose آن را همراه با image مربوط به وبسایت httpbin.org این دو را ادغام کردیم و توانستیم به درستی پروژه را پیاده سازی کنیم.

روش دوم به این صورت پیاده سازی شد که با استفاده از python و کتابخانه requests یک برنامه reverse proxy را پیاده سازی کردیم و سپس خودمان به آن با استفاده از کتابخانه pyjwt به آن reverse proxy اضافه کنیم که تصویر این قسمت از کد در

زیر قابل مشاهده است :

```
url = 'https://{}{}'.format(hostname, self.path)
req_header = self.parse_headers()
to_str = str(self.headers)
for line in to_str.split("\n"):
    auth = line[len("Authorization: Bearer ") : ]
    break

print(auth)
SECRET_KEY = "hatami"
token = auth
ALGORITHM = "HS256"
ACCESS_TOKEN_EXPIRE_MINUTES = 30
try:
    payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
except:
    print("invalid authentication")
    return
print(payload, flush=True)
valid = True
current_time = time.time()
if payload["iat"] + ACCESS_TOKEN_EXPIRE_MINUTES * 60 < current_time:
    valid = False
if payload["name"] not in valid_users:
    valid = False
```

این jwt authentication به این صورت عمل میکند که اگر نام کاربر ارسال شده جز کاربران valid و از زمان ساخت jwt کمتر از ۳۰ دقیقه طول کشیده باشد کاربر را به httpbin متصل میکند و در غیر این صورت error برمیگردانیم.

برای ساخت token ها از jwt.io و برای ارسال درخواست get از نرم افزار postman استفاده کردیم.

ممنون که توجه کردید.
در صورت هر گونه مشکل لطفا به ایدی تلگرام بنده پیام بدهید (@armanhtm)