گزارش پروژه فاز دوم درس امنیت سیستم های کامپیوتری

سارا براتی 97521144

محدثه جعفری 97521198

محمد امین میرزاکوچکی 97522256

در این پروژه یک پیام رسان نظیر به نظیر غیر متمرکز که پیام ها را رمز گذاری میکند، تولید شده است.

برای برقراری ارتباط نظیر به نظیر از پروتکل UDP استفاده کرده ایم و برای رمزگذاری از الگوریتم AES استفاده کرده ایم.

UDP چیست؟

در شبکه های کامپیوتری، پروتکل دیتاگرام کاربر یکی از اعضای اصلی مجموعه پروتکل اینترنت است. با UDP، برنامه های کامپیوتری می توانند پیام هایی را که در این مورد به عنوان دیتاگرام نامیده می شود، به میزبان های دیگر در یک شبکه پروتکل اینترنت ارسال کنند.

این یک پروتکل بدون اتصال است و قابل اعتماد نیست. در سایر برنامه های چت مانند Facebook و WhatsApp استفاده نمی شود. سرعت ارتباط نسبتاً سریع از پروتکل TCP است. این در گشت و گذار ویدیویی و بازی آنلاین استفاده می شود.

AES چیست؟

AES (استاندارد رمزگذاری پیشرفته) یک رمز بلوک متقارن است که توسط NIST استاندارد شده است. اندازه بلوک داده ثابت 16 بایت دارد. کلیدهای آن می تواند 128، 192 یا 256 بیت باشد.

AES بسیار سریع و امن است و در واقع استاندارد رمزگذاری متقارن است.

```
1   import socket
2   import threading
3   import os
4   from Crypto.Cipher import AES
```

ابتدا پکیج های مورد نیاز را ایمپورت میکنیم.

پکیج socket را برای برقراری ارتباط UDP استفاده کرده ایم.

پکیج threading را برای ارسال و دریافت همزمان پیام ها استفاده کرده ایم.

لینک مستند پکیج AES

```
6   s = socket.socket(socket.AF_INET , socket.SOCK_DGRAM )
7   s.bind(("172.17.9.138",50001))
8   nm = input("ENTER YOUR NAME : ")
9   print("\nType 'quit' to exit.")
10
11  ip,port = input("Enter IP address and Port number: ").split()
```

برقراری ارتباط سوکت از آیپی آدرس و پورت خود به پورت و آیپی آدرس مطلوب. علت استفاده از پورت نامبر های 50000 و 50001 این است که محدوده پورت نامبرهای پروتکل UDP بین 4096 تا 65535 هست.

```
13  def send():
14      while True:
15          ms = input(">> ")
16
17          if ms == "quit":
18              os._exit(1)
19
20          key = b'Sixteen byte key'
21          data=bytes(ms,'ascii')
22          cipher = AES.new(key, AES.MODE_EAX)
23
24          ciphertext, tag = cipher.encrypt_and_digest(data)
25
26          file_out = open("encrypted.bin", "wb")
27
28          [ file_out.write(x) for x in (cipher.nonce, tag , ciphertext) ]
29          file_out.close()
30          s.sendto(ciphertext, (ip,int(port)))
31
```

در این تابع که برای ارسال پیام به طرف مقابل طراحی شده است در ابتدا یک ورودی از کاربر گرفته میشود و توسط یک کلید که از قبل بین دو طرف قرارداده شده است و پیامی که به داده های بایتی تبدیل شده است رمزگذاری صورت میگیرد و پیام به صورت رمز شده از طریق کانکشن UDP به طرف

مقابل ارسال میشود. یک تگ و یک nonce متخص رمزگذاری برای تایید اصالت پیام ها استفاده میشود.

```
33  def rec():
34      while True:
35          msg = s.recvfrom(1024)
36          file_in = open("encrypted.bin", "rb")
37          nonce, tag, ciphertext = [ file_in.read(x) for x in (16, 16, -1) ]
38          key = b'Sixteen byte key'
39          cipher = AES.new(key, AES.MODE_EAX, nonce)
40          data = cipher.decrypt_and_verify(msg[0], tag)
41          print("\t\t\t\t >> " +  data.decode()  )
42          print(">> ")
```

تابع rec هم مانند تابع send هست با این تفاوت که پیام را دریافت کرده و بجای رمزگذاری، رمزگشایی میکند سپس اصالت پیام را میسنجد و در صورت معتبر بودن به کاربر نمایش میدهد.

```
44    x1 = threading.Thread( target = send )
45    x2 = threading.Thread( target = rec )
46
47    x1.start()
48    x2.start()
```

این دو thread هم برای این است که همزمان بشود هم پیام را فرستاد و هم پیام را دریافت کرد.

هر دو طرف با استفاده از multithreading ارتباط برقرار می کنند

در پایتون، ما یک ماژول threading داریم، با کمک می‌توانیم رشته‌های زیادی را برای اجرای برنامه خود اضافه کنیم. ما همچنین می توانیم توابع مختلف را برای اجرا به رشته های مختلف ارسال کنیم.

بیایید با یک کمک بفهمیم بدن انسان را یک فرآیند می‌دانیم و اجزای ما رشته‌های مختلفی از آن هستند. به طور همزمان وظایف خود را انجام می دهند. کبد، قلب، ریه ها، مغز اندام های درگیر در این فرآیند هستند و هر فرآیند به طور همزمان در حال انجام است.

برای اینکه این پروژه رو تنها به صورت لوکال پیاده سازی کردیم سخت بودن پیاده سازی غیرلوکال بود. برای پیاده سازی غیر لوکال نیاز به پورت فورواردینگ وجود داشت که هم هزینه بر و هم زمان بر بود.

برای اینکه بخوایم پورت فورواردینگ انجام بدیم باید پورت نت داخلی خودمون رو (مثلن پورت نت داخلی نت خانگی مثل زیتل) به سوی Universal Network ارسال کنیم.

همچنین باید یک Public IP اجاره کنیم که این IP رو به پورتمون تخصیص بدیم. این اجاره IP در سال حداقل 5 یا 6 میلیون تومن هزینه دارد.

همچنین حجم کدی که برای پیاده سازی غیرلوکال داریم خیلی زیاد هستش و عملن در یک ترم قابل پیاده سازی نبودن.