GÜVENLİ DOSYA TRANSFERİ İÇİN ŞİFRELEME TABANLI AĞ UYGULAMASI

PROJE ARA RAPRORU

27 NİSAN 2025

BURSA TEKNİK ÜNİVERSİTESİ
Yusuf Güney - 22360859041

1. GIRIŞ

Bu projede amacımız, ağ üzerinde güvenli şekilde dosya transferi gerçekleştiren, temel şifreleme, paket parçala/birleştir ve IP başlığı işleme gibi ağ seviyesi kavramları uygulayan bir sistem geliştirmektir. Ağ üzerinden gerçekleşen veri iletiminin güvenliğini arttırmak için dosya transferi aşamasında AES şifreleme kullanılmış, dosyalar parçalanarak aktarılmış ve IP katmanında temel düzeyde paket analizi yapılmıştır.

---

2. PROJE AMACI VE KAPSAMI

Projenin temel amacı, iki cihaz arasında (localhost üzerinden) çalışan, şifreli ve parçalanmış dosya transferi gerçekleştirebilen bir Python uygulaması geliştirmektir. Bu kapsamda geliştirilen modüller:

- AES şifreleme modülü
- Socket tabanlı dosya transfer sistemi
- Dosya parçalama ve birleştirme mekanizması
- IP paketi analiz modülü (Scapy kullanılarak)

---

3. KULLANILAN YÖNTEMLER VE TEKNİK DETAYLAR

3.1. Socket ile Dosya Transferi

Bu bölümde, sender.py ve receiver.py dosyaları kullanılarak temel bir TCP bağlantısı üzerinden dosya transferi gerçekleştirilmiştir.
Aşağıda sender tarafında dosya gönderme, receiver tarafında dosya alma ve kaydetme kod parçaları gösterilmektedir
Başarılı bir transfer sonrası “DOSYA GONDERILDI.” ve “DOSYA KAYDEDILDI.” mesajları alınmıştır.

```
32    # BAGLANTI KUR
33    s = socket.socket()
34    s.connect((HOST, PORT))
```

```
14    # SUNUCUYU BASLAT
15    s = socket.socket()
16    s.bind((HOST, PORT))
17    s.listen(1)
18
19    print("Bağlantı Bekleniyor...")
20    conn, addr = s.accept()
21    print("Bağlandı:", addr)
```

3.2. AES ile Şifreleme

Dosya, gönderilmeden önce AES algoritmasının EAX modu kullanılarak şifrelenmiştir.
Aşağıda şifreleme (**encrypt_and_digest**) ve çözümleme (**decrypt_and_verify**) işlemlerinin kod parçaları ile terminalde oluşan SHA-256 hash değerleri gösterilmiştir.
Bu sayede verinin gizliliği ve doğruluğu sağlanmıştır.

```
15    # AES SIFRELEYICIYI OLUSTUR
16    cipher = AES.new(key, AES.MODE_EAX)
```

```
36    # AES ILE COZ
37    cipher = AES.new(key, AES.MODE_EAX, nonce)
38    plaintext = cipher.decrypt_and_verify(ciphertext, tag)
```

3.3. Parçalama ve Birleştirme

Büyük verilerin ağ üzerinden güvenli ve hatasız aktarımı için dosya verisi 1024 baytlık parçalara bölünerek gönderilmiştir.
Alıcı taraf bu parçaları sırasıyla birleştirerek orijinal veriyi oluşturmuştur.
Aşağıda gönderim ve alım işlemleri sırasında kullanılan döngüler gösterilmiştir.

```
36    # VERIYI PARCA PARCA GONDER
37    for i in range(0, len(data), CHUNK_SIZE):
38        chunk = data[i:i+CHUNK_SIZE]
39        s.sendall(chunk)
```

3.4. SHA-256 ile Hash Kontrolü

Gönderilen ve alınan verilerin bütünlüğünü doğrulamak için SHA-256 algoritması kullanılarak özet değerler (hash) hesaplanmıştır.
Terminalde elde edilen gönderici ve alıcı hash değerleri karşılaştırılmış ve eşleştiği görülmüştür.
Bu, aktarım sırasında veri kaybı veya bozulması olmadığını kanıtlamaktadır.

```
29    hash_sender = hashlib.sha256(plaintext).hexdigest()
30    print("Gönderilen Dosya HASH:", hash_sender)
```

```
47    hash_receiver = hashlib.sha256(plaintext).hexdigest()
48    print("Alınan Dosya HASH:", hash_receiver)
```

```
PS C:\Users\Yusuf\Desktop\Bilgisayar_Aglari_Projesi\Proje Ara Raporu> python .\sender.py
Gönderilen Dosya HASH: 69f7d11c346fc673666e1242aed66499478ab49e17ec1fa0a3c3566ac5ac8766
Dosya Gönderildi.
```

```
PS C:\Users\Yusuf\Desktop\Bilgisayar_Aglari_Projesi\Proje Ara Raporu> python .\receiver.py
Bağlantı Bekleniyor...
Bağlandı: ('127.0.0.1', 53043)
Dosya Kapatıldı: received.txt
Alınan Dosya HASH: 69f7d11c346fc673666e1242aed66499478ab49e17ec1fa0a3c3566ac5ac8766
```

3.5. Scapy ile IP Paketi İncelemesi

Scapy kütüphanesi kullanılarak oluşturulan örnek bir IP paketi üzerinden başlık bilgileri (version, ttl, flags gibi) analiz edilmiştir.
Aşağıda IP paketinin oluşturulması ve packet.show() çıktısının terminal görüntüsü sunulmuştur. Böylece ağ katmanı seviyesinde veri paketlerinin yapısı gözlemlenmiştir.

```python
from scapy.all import IP

packet = IP(dst="8.8.8.8", ttl=64, flags="DF")

packet.show()

```

```
PS C:\Users\Yusuf\Desktop\Bilgisayar_Aglari_Projesi\Proje Ara Raporu> python .\spacy_test.py
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = ip
  chksum    = None
  src       = 172.16.0.33
  dst       = 8.8.8.8
  \options   \
```

---

4. GERÇEKLEŞTİRİLEN ADIMLAR VE GELİŞİM

- Socket tabanlı dosya transferi başarıyla gerçekleştirilmiştir.
- AES şifreleme ve çözümleme işlevleri çalışır duruma getirilmiştir.
- Dosya parçalama ve birleştirme işlemleri sorunsuz tamamlanmıştır.
- SHA-256 hash kontrolü yapılmıştır.
- IP paketi Scapy üzerinden analiz edilmiş, başlık içeriği doğrulanmıştır.

Bu aşamada projede geliştirilen kod dosyaları da tamamlanmış ve proje klasörüne eklenmiştir. Eklenen kodlar:

- sender.py

- receiver.py
- scapy_test.py

Bu dosyalar, projenin mevcut durumunu ve ilerlemesini belgelemek amacıyla rapora dahil edilmiştir.

Projeyi Test Etme Yöntemi:

- Terminal 1'de receiver.py dosyasını çalıştırarak bağlantı beklenir.
- Terminal 2'de sender.py çalıştırılarak dosya gönderimi başlatılır.
- Gönderilen dosyanın hash değeri ile alınan dosyanın hash değeri karşılaştırılır.
- Scapy testi için scapy_test.py çalıştırılarak IP başlığı analizi yapılır.

---

5. PROJE KALANINDA YAPILACAK ADIMLAR

- Ping komutu ile ağ performansı testi yapılacak.
- Wireshark kullanılarak ağ trafiğindeki şifreli paketler gözlemlenecek.
- Kodların temizlenmesi ve yorumların eklenmesi sağlanacak.
- Final raporu hazırlanacak ve proje tamamlanacaktır.

---

6. GITHUB PROJE DEPOSU

Projeye ait kodlara aşağıdaki GitHub adresinden ulaşılabilirsiniz:

- https://github.com/Yusuf-Guney/Bilgisayar-Aglari-Proje

---

7. KAYNAKLAR

- **https://docs.python.org/3/library/socket.html**
- **https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html**
- **https://scapy.readthedocs.io/en/latest/introduction.html**
- **https://www.geeksforgeeks.org/sha-in-python/**