BİL 520 Proje
Seyyid Hikmet Çelik
221111033

Proje kapsamında python dilinde Flask ve kriptografi kütüphaneleri kullanarak **Secure File Transfer Application** geliştirdim.

Uygulama, yerel ağ üzerinde çift taraflı güvenli dosya paylaşmaya yarıyor. Hem başlangıçta scripte path vererek hem de daha sonra site üzerinden dosya sürükleyerek ya da seçerek hem host hem client tarafından dosya paylaşabiliyorsunuz. İndirilecek dosyaları listeleyip hem host hem client tarafında indirebiliyorsunuz.

Uygulamada python ve generic kütüphaneler kullanıldığı için os independent, yine, bir cihazda host ettikten sonra tarayıcısı olan herhangi bir cihazdan da girilebildiği ve dosya gönderilip alınabildiği için client tarafında da os independent.

Program ve Özellikleri
Program aslında zipte görmüş olduğunuz **proje.py** python kodundan ibaret fakat uzun bir kod ve işlevselliği yüksek, güvenlik önlemleri sağlam ve güvenlik önlemleri alınırken çoğu yerde rastgelelikler de göz önünde bulunduruluyor, rastgele şifreleme yapılıyor.
proje-obf.py ise python kodunun https://github.com/Blank-c/BlankOBF adresindeki kod karıştırma(obfüskasyon) yöntemi kullanılmış hali ve işlevi proje.py ile birebir aynı, istediğinizi kullanabilirsiniz.

Uygulamadaki Güvenlik Önlemlerini Özet Geçmek Gerekirse;

- Python kodunun çalınması ve nasıl güvenlik önlemleri alındığı görülmesi ihtimaline karşı **obfüske edilmiş python kodu(**proje-obf.py**)**
- **Flask** uygulamasının kendi secret keyi, **random number** ile generate edildi, söylendiğine göre **CSRF(Cross-Site Request Forgery)** atağını **önlüyor**. (https://stackoverflow.com/questions/47687307/how-do-you-solve-the-error-keyerror-a-secret-key-is-required-to-use-csrf-whe)
- Siteye girebilecek IP'ler **whitelist** ile baştan belirleniyor, firewall gibi.
- Sitede **SSL** (doğal olarak **RSA**) sertifikası kullanılıyor. (**asimetrik şifreleme**)
- Siteye giriş şifresi **SHA-256** özet fonksiyonuyla kontrol ediliyor, bu sırada hem girilen hem de kontrol edilen şifre **rastgele salt**lanıyor. (tuzlanıyor.)
- Gönderilecek dosyalar **CBC modunda** (EBC dosya hakkında bilgi verdiği ve CBC daha güvenli olduğu için) **AES-128**bit ile şifreleniyor ve clientta çözülüp indiriliyor. Her seferinde rastgele AES keyi üretiliyor. (**simetrik şifreleme**)
- Şifreli gönderilen dosyaların doğruluğu şifreleri çözüldükten sonra orijinal dosyaların **MD5** özetiyle otomatik olarak karşılaştırılıp **dosya bütünlüğü doğrulanmış** oluyor. Bütünlüğü doğrulanmamış dosya indirilemiyor.
- Şifre girerken script injectionlari (**XSS** gibi) engellemek için password **HTML escape** ediliyor.

```
password = escape(request.form["password"])
```

(https://stackoverflow.com/questions/2334863/does-html-encoding-prevent-xss-security-exploits)

- Şifre ya da diğer veriler de bir veritabanından çekilmediği için **SQL Injection** saldırısı da yapılamaz.

Uygulamayı Çalıştırma ve Kullanım

proje.py kodu aşağıdaki şekilde terminalde çalıştırılabilir:

```
python3 proje.py -fp "." -ips "192.168.1.30, 192.168.1.32" -pw
"Admincim+&1hadiYiNeiyisin"
```

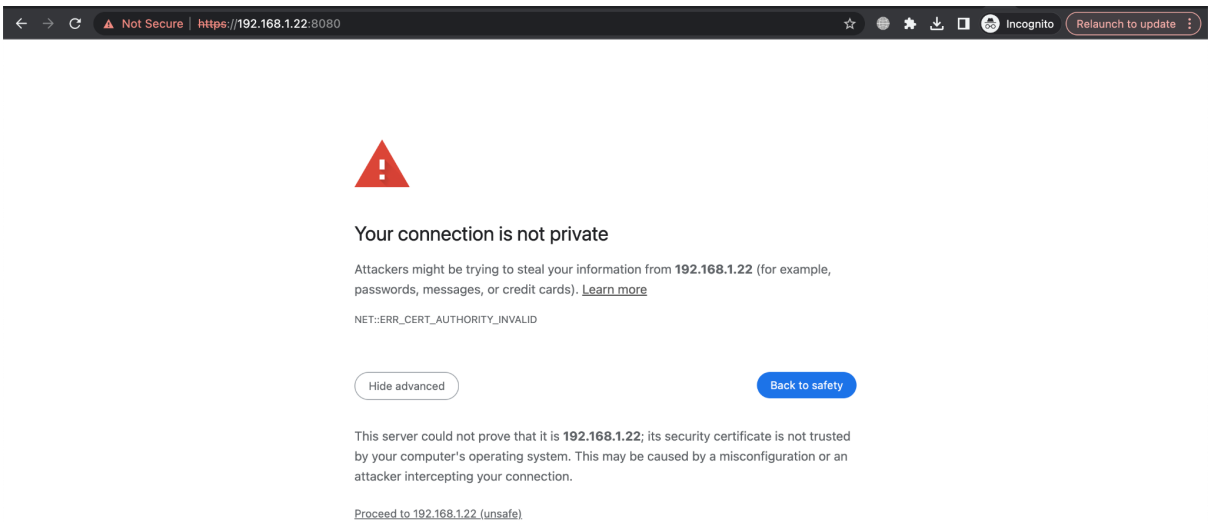
Uygulama bu şekilde host (benim ağ üzerindeki ip'si 192.168.1.22) cihazda 8080 portu üzerinde, https://192.168.1.22:8080 adresinde dosya gönderip almayı sağlayan basit bir web sitesi oluşturuyor.

Burada;

- **-fp** ya da **--folderpath** opsiyonu paylaşılacak dosyayı (bulunulan dizinde başka bir folder olmaması lazım, bunu kontrol olarak çok basit şekilde ekleyebilirdim ama amaç file paylaşmak zaten) (varsayılan path ".")
- **-ips** ya da **--iplist** opsiyonu tarayıcı üzerinde dosya alıp gönderebilecek ip beyaz listesini (varsayılan boş, yani sadece host)
- **-pw** ya da **--loginpassword** opsiyonu ise admin dışında farklı bir şifre kullanılmak istendiğinde giriş şifresini belirtme amacıyla kullanılıyor. (varsayılan admin)

- Eğer folderpath şu an bulunduğumuz dosyaysa kodun kendisini asla paylaşmamak için de kodun içinde kontrol var.
- Kodun çalıştığı host bilgisayarın ipsi listede olmasa bile host her zaman izinli. Burada 192.168.1.30 ve 192.168.1.32 ip adreslerine sahip telefon, tablet gibi cihazlara siteye giriş izni veriyoruz. Eğer -ips "all" opsiyonunu kullansaydık ağdaki tüm cihazlara izin vermiş olacaktık, default olarak sadece hosta izin var.
- Güçlü bir şifre kullanmak güvenlik açısından daha iyi, o yüzden değiştiriyoruz.

Programı terminalde çalıştırıp izin verilen cihazlardan ya da host bilgisayardan siteye (https://192.168.1.22:8080) girdiğinizde aşağıdaki uyarıyı alıyorsunuz:



Bunun sebebi kodda işi Flask'ın adhoc SSL özelliğini kullanmam(kendim key verip kullanmamdan farklı olarak uygulama her açıldığında rastgele):

```
app.run(host=get_ip(), port=8080, ssl_context="adhoc")
```

Bu şekilde yerel ağda olduğum ve **kendi imzalı** sertifikam olduğu için tarayıcının güvenmemesi çok normal ve burayı "Proceed to 192.168.1.22" deyip geçiyorum. Fakat bu sayede SSL (dolayısıyla RSA şifreleme) kullanarak (http yerine https kullanarak) **asimetrik şifreleme** ile dosya paylaşım sitemin güvenliğini artırmış oluyorum.

İlerlediğinizde siteye erişim için şifre giriş ekranı geliyor, burada belirlediğiniz şifreyi (ya da default admin) girerek ilerliyorsunuz:

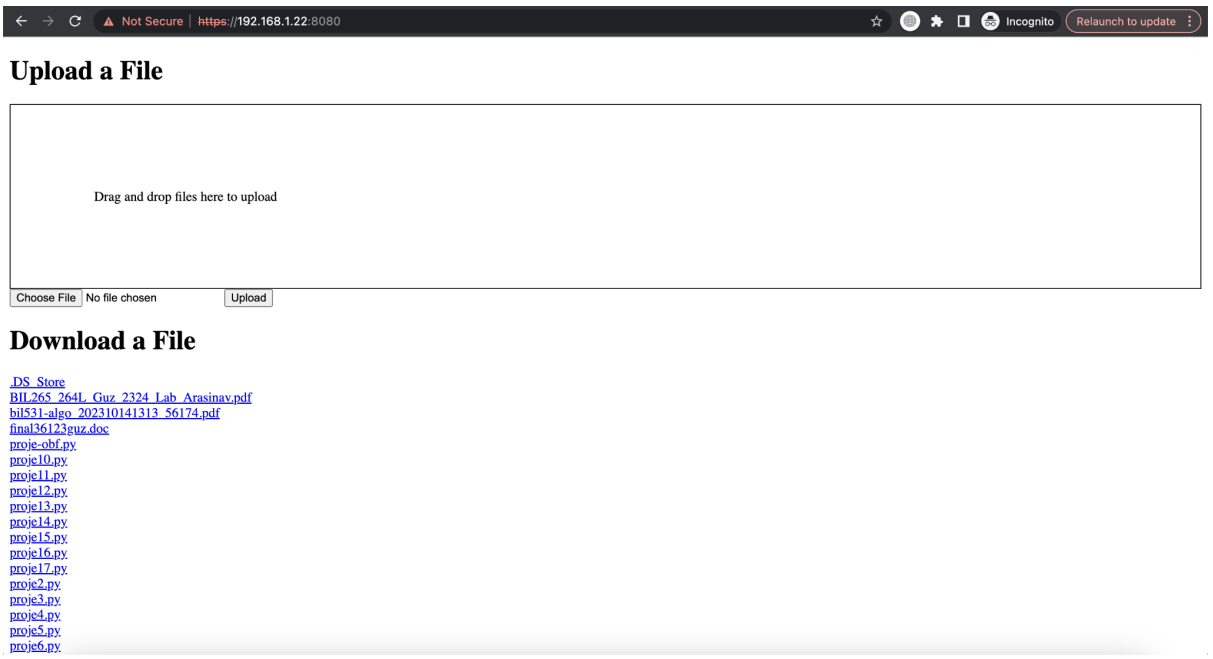
Kodda şifre **SHA256 özet fonksiyonu**yla kontrol ediliyor. generate_password_hash ve check_password_hash fonksiyonlarını kullanıyorum ve şifre aynı olsa bile bu fonksiyonlara her girildiğinde şifre tuzlanıyor (**salt**) ve yine tuzlanmış haline göre kontrol ediliyor:

```
hashedpw = generate_password_hash(password, method="sha256")
if check_password_hash(hashedpw, args.loginpassword):  # parola kontrol
   session["logged_in"] = True
```

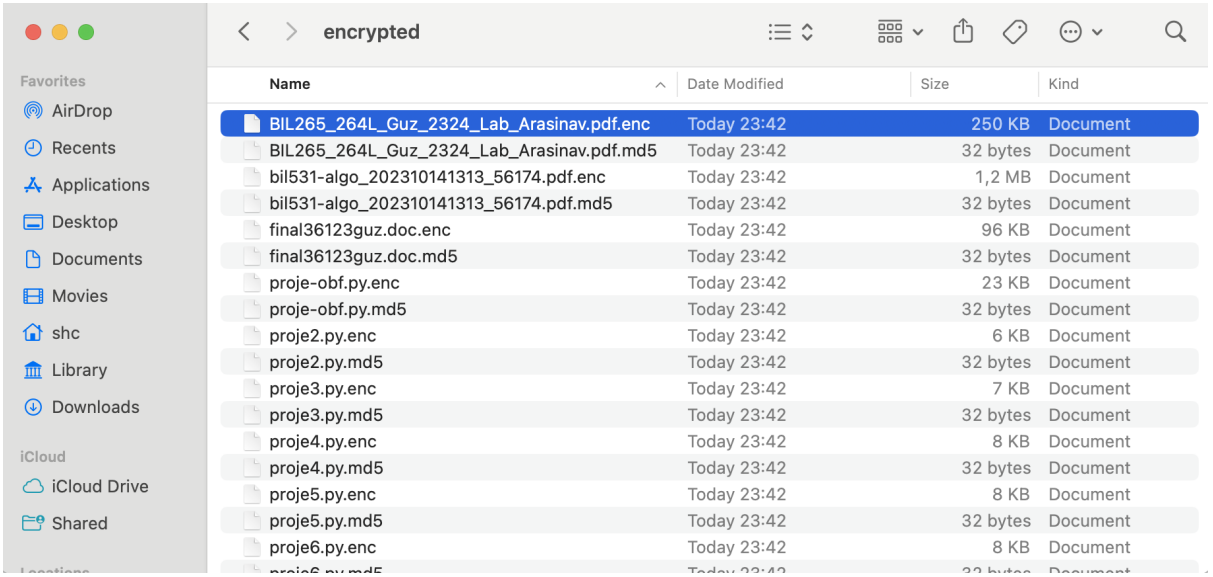
Eğer client ipnin siteye erişimi yoksa yukarıdaki giriş ekranı gelmiyor bile direkt "Access Denied" hatası veriyor site:

```
return (
              "Access Denied. Unauthorized IP Address!",
              403,
          )  # beyaz listede degilse yetkisiz erisim
```

Doğru şifreyi girerek ilerlediğinizde folder-path'ten paylaşılan dosyaları "Download a File" kısmında görebiliyor aynı zamanda yeni dosya paylaşmak istediğinizde "Upload a File" kısmından dosya seçerek ya da sürükleyip bırakarak yeni bir dosyayı hem client hem host tarafından yükleyebiliyorsunuz (Yükleme yüzdesi, hızı ve başarılı olup olmadığı da gösteriliyor)(Eklenen yeni dosyayı download kısmında görebilmeniz için tekrar sayfayı yenilemeniz gerekli):



Burada dosyalar her iki taraftan (client-host) gönderilirken folder-path dizininde oluşturulan encrypted dizininde şifreli halleri (AES-128bit CBC) ve orijinal dosyalarının md5leriyle birlikte tutuluyor:



Ve dosyalar indirilirken bu dosyalar tekrar decrypt edilip orijinal dosyaların md5leri ile karşılaştırılarak bütünlükleri doğrulanmış oluyor.

Eklenebilecek Özellikler
Dosyalar client tarafında şifreli olarak indirilip client tarafında bir python scripti ile çözülebilir, bu durumda çözücü python scripti ve AES anahtarını da cliente ulaştırmak gerekiyor, farklı portlardan paylaşılarak veri dağıtılabilir ama bu sefer kullanışlılık azalır.

Bakıldığında, hem proposalda söylediğim özellikleri gerçekleştirdim hem de daha fazla özellik eklemiş oldum. Kodlara da bakabilirsiniz, yeteri kadar yorumladım.
Proposalda bahsettiğimiz soruları yeniden yanıtlayacak olursak;
Yazdığım araç hangi problemleri çözdü?
Dosya gönderme, alma, ağ üzerinde paylaşma gibi işlemleri sağlayan birçok uygulama var fakat yine bu araçların birçoğunu public ağlar üzerinde kullandığınız zaman araya birinin girme ve dosyanıza erişme şansı var. Bu problemi çözmek için dosyaları şifreli bir şekilde göndermek ya da almak bir çözüm yolu olabilir, bu şekilde dosya hırsızlığı ve manipülasyonu önlenebilir.
Bu problem neden önemli? Ne gibi sorunlara yol açıyor?
Bu problem dosyalarınızın ağ üzerindeki güvenliği için önemli. Tersi durumda dosyalarınıza yetkisiz erişim sağlanabilir ya da manipüle edilebilir, bu durum istenmeyen sonuçlar doğuracaktır. Yetkisiz erişim hassas bilgileriniz için, dosya manipülasyonu ise alıcı tarafından alınan dosyanın güvenilirliği için tehlikeli.