**Web Security 101**

**Dosya Yolu Geçiş Zafiyeti (File Path Traversal )**

Dosya yolu geçiş zafiyeti, saldırganın uygulamanın normalde erişemeyeceği dosyalara erişmesini sağlar. Genellikle dizin değiştirme karakterleri (../) kullanılarak üst dizinlere çıkılır ve hassas dosyalara ulaşılır.

Örneğin, bir web sitesi belirli bir dosyayı göstermek için URL'de bir parametre kullanıyorsa:

https://example.com/view?file=report.pdf

Bir saldırgan bu parametreyi değiştirerek şu şekilde sistem dosyalarına erişebilir:

https://example.com/view?file=../../../../etc/passwd

Bu sayede sistemdeki kullanıcı bilgilerini içeren dosya görüntülenebilir.

**Korunmasız Yönetici İşlevselliği (Unprotected Admin Functionality)**

Bir web uygulamasında yönetici (admin) işlevselliklerine yeterli erişim kontrolü uygulanmadığında, yetkisiz kişiler yönetim paneline veya kritik sistem ayarlarına erişebilir.

Örneğin, bir web sitesi yönetici girişini doğrulamadan doğrudan bir panel sunuyorsa:

https://example.com/admin

Bir saldırgan, herhangi bir kimlik doğrulaması yapmadan bu sayfaya erişebilir ve yönetici yetkilerine sahip olabilir. Eğer doğrudan belirli işlemleri gerçekleştiren URL’ler varsa, bunları da kullanabilir:

https://example.com/admin/deleteUser?user=1234

Bu durumda saldırgan, URL’yi kullanarak yetkisiz bir şekilde kullanıcıları silebilir veya sistemde değişiklik yapabilir.

**Öngörülemeyen URL ile Korumasız Yönetici İşlevselliği (Unprotected Admin Functionality via Unpredictable URL)**

Eğer bir web uygulamasındaki yönetici paneli veya kritik işlemler, özel bir kimlik doğrulama yerine sadece tahmin edilmesi zor bir URL’ye dayanıyorsa, saldırganlar bu URL’yi keşfederek yetkisiz erişim sağlayabilir.

Örneğin, yönetici paneline giriş yapmak için herhangi bir doğrulama olmadan sadece belirli bir URL’ye erişim yeterliyse:

https://example.com/secret-admin-panel-98765

Bir saldırgan bu URL’yi tahmin ederek ya da dizin tarama araçlarıyla (dirb, gobuster gibi) keşfederek giriş yapabilir ve yönetici yetkilerine sahip olabilir. Aynı şekilde, belirli işlemleri gerçekleştiren özel URL’ler (örneğin kullanıcı silme veya ayar değiştirme) doğrudan erişime açıksa, saldırganlar bunları kullanabilir. Javascript ve Html kodlarından BurpSuite Üzerinden Bulabilir.

**Parametre Tabanlı Erişim Kontrol Yöntemleri (Parameter-Based Access Control)**

Bazı web uygulamaları, kullanıcı yetkilendirmesini sadece URL parametreleri veya istemci tarafından kontrol edilebilen diğer veriler üzerinden yaptığı için saldırganlar bu değerleri değiştirerek yetkisiz erişim sağlayabilir.

Örneğin, bir kullanıcı profil sayfası şu şekilde çalışıyorsa:

https://example.com/profile?user=123

Bir saldırgan, user parametresini değiştirerek başkasının hesabına erişebilir:

https://example.com/profile?user=456

Eğer sistem sadece bu parametreye dayanarak yetkilendirme yapıyorsa, saldırgan başkasının bilgilerini görebilir veya değiştirebilir.

**Güvensiz Erişim Kontrol Mekanizmaları**

Bazı uygulamalar kullanıcının erişim haklarını veya rolünü oturum açma sırasında belirler ve ardından bu bilgileri kullanıcı tarafından kontrol edilebilen bir konumda depolar. Bu şunlar olabilir:

* **Gizli bir alan (Hidden Field)**
* **Bir çerez (Cookie)**
* **Önceden ayarlanmış bir sorgu parametresi (Query String Parameter)**

Uygulama, gönderilen değere göre erişim kontrol kararları verir. Örneğin:

<https://insecure-website.com/login/home.jsp?admin=true>  
<https://insecure-website.com/login/home.jsp?role=1>

Bu yaklaşım güvenli değildir çünkü bir kullanıcı değeri değiştirerek yönetici yetkileri kazanabilir ve yetkisiz erişim sağlayabilir.

**Kullanımı:**

Burp Suite Proxy ile isteği yakala.

Intercept is on durumundayken, GET veya POST parametrelerini değiştir (örneğin role=user → role=admin). (administrator de olabilir)

Forward diyerek isteği gönder ve yanıtı kontrol et.

HTTP History sekmesinde değişikliklerin etkisini incele.

**Yatay Ayrıcalık Yükseltme (Horizontal Privilege Escalation)**

Kullanıcı, kendi hesabının dışında, aynı seviyede başka bir kullanıcının bilgilerine erişim sağlar. Burada kullanıcı kimliği tahmin edilmez, ancak kullanıcının yetkileri sınırlandırılmamıştır ve saldırgan, başka bir kullanıcının hesabına (aynı rol seviyesinde) erişebilir.

Bazı uygulamalarda, kullanıcı kimlikleri URL veya parametrelerde öngörülebilir şekilde gönderilir. Örneğin, kullanıcı ID'si 123 olan bir profil şu şekilde olabilir:

https://example.com/profile?id=123

Bir saldırgan, ID'yi değiştirerek başka kullanıcıların profillerine erişebilir:

https://example.com/profile?id=124

Bu tür öngörülebilir kimlikler, güvenlik açığı oluşturur çünkü saldırgan kimlik numaralarını tahmin ederek yetkisiz erişim sağlayabilir.

**Kullanımı:**

Blogda gezerken Carlosun profiline gir

<https://example.com/blogs?userId=32b42fdb-8c01-4888-aeb0-ad577643fe6c>

bu şekilde user id’i gösteriyor şimdi kendi profilimize girelim

<https://example.com/my-account?id=4407ca72-ced5-4ba8-a2b3-3cbcf69b969b>

buradaki id değerini Carlosun id değeri ile değiştiriyoruz. Bu sayede Carlos hesabına girmiş olduk.

**Parola Ifşasıyla İstek Parametresi Tarafından Kontrol Edilen Kullanıcı Kimliği**

Bu güvenlik açığı, uygulamanın kullanıcı kimliğini ve şifreyi kontrol etme şekliyle ilgilidir. Eğer uygulama, kullanıcı kimliği ve şifreyi URL parametreleri veya istek body’sinde gönderiyorsa, bu bilgiler kolayca ele geçirilebilir ve manipüle edilebilir.

**Zafiyetin Açıklaması:**

**Parola ifşası**: Uygulama, şifreyi açıkça URL parametrelerinde veya HTTP isteklerinde gönderiyorsa, bu şifre ifşa olur. Bu, saldırganın bu bilgiyi yakalayarak başka bir kullanıcı veya yönetici hesabına erişim sağlamasına neden olabilir.

**İstek Parametresi ile Kullanıcı Kimliği Kontrolü**: Kullanıcı kimliği, genellikle **id** parametresi gibi URL parametrelerinde kontrol edilir. Eğer **id** parametresi değiştirilerek yönetici kimliği girilirse, yönetici yetkilerine sahip kullanıcı bilgilerine ulaşılabilir.

<https://example.com/user_account?id=carlos>

Eğer id parametresi, başka bir kullanıcının kimliğiyle değiştirilirse (örneğin, id=administrator), yönetici bilgilerine ulaşılabilir

**Kullanımı:**

Kendi hesabımıza login olup o isteği burp ile repeater’a gönderiyoruz

my-account?id=wiener yazan yeri my-account?id=administrator olarak değiştiriyoruz

Response kısmında html koldarı içinden admin kullanıcı adı ve şifreyi buluyoruz ve giriş yapıyoruz

**Kimlik Doğrulama Açıkları (Authentication Vulnerabilities)  
Kimlik doğrulama açıkları**

saldırganların hassas verilere ve işlevlere erişmesini sağlayan açıklardır. Bu zafiyetler, kimlik doğrulama süreçlerindeki zayıflıklardan kaynaklanabilir ve genellikle şu şekilde istismar edilir:

* Zayıf şifreler
* Brute-force (kaba kuvvet) saldırıları
* Kullanıcı adı numaralandırma
* Kimlik doğrulama token'larının zayıf yönetimi  
  Bu tür açıklar, yetkisiz erişim sağlanmasına ve daha fazla güvenlik açığının ortaya çıkmasına yol açabilir.

**Kimlik Doğrulama ve Yetkilendirme Farkı (Authentication vs. Authorization)**  
Kimlik doğrulama, bir kullanıcının gerçekten iddia ettiği kişi olup olmadığını doğrulama işlemidir. Yetkilendirme ise bir kullanıcının bir işlemi yapmaya yetkili olup olmadığını kontrol eder.

**Brute-force (Kaba Kuvvet) Saldırıları (Brute-force Attacks)**  
Brute-force saldırısı, bir saldırganın kullanıcı kimlik bilgilerini tahmin etmek için deneme-yanılma yöntemini kullanmasıdır. Bu saldırılar genellikle kullanıcı adı ve parola listeleri ile otomatikleştirilir. Saldırganlar, bu süreci daha verimli hale getirmek için mantık kullanabilir veya halka açık bilgilere dayanarak tahminlerde bulunabilir.  
Bu tür saldırılar, şifre korumasız veya zayıf şifreler içeren sistemlerde ciddi güvenlik tehditlerine yol açar.

**Kullanıcı Adı Numaralandırma (Username Enumeration)**  
Kullanıcı adı numaralandırma, bir saldırganın, web sitesinin davranışındaki değişiklikleri gözlemleyerek geçerli bir kullanıcı adını belirlemesidir. Bu genellikle giriş sayfasında, doğru bir kullanıcı adı girildiğinde ancak yanlış parola girildiğinde meydana gelir.  
Ayrıca kayıt sayfasında, zaten alınmış bir kullanıcı adı girildiğinde de benzer bir durum yaşanabilir. Bu, saldırganların geçerli kullanıcı adlarını hızlıca listelemelerini sağlar, böylece brute-force saldırılarını daha verimli hale getirir.

**Kullanımı:**

**Burp ile bruteforce saldırsı :**

Hedef uygulamanın login sayfasına git ve geçersiz bir kullanıcı adı ve şifre girerek bir istek gönder. (testkullanıcı testşifre)

2. Burp Intruder’a Geçiş

Burp Suite'te Proxy > HTTP History sekmesine git. Gönderdiğin giriş isteğini bul ve bu isteği Burp Intruder'a gönder.

3. Username Parametresini Hazırlayın

Burp Intruder’da username parametresinin etrafındaki § sembollerini göreceksin (örneğin: username=§invalid-username§). Bu parametreyi seç payload pozisyonu olarak ayarla. Şifre geçici olarak testşifre kalsın

4. Payload Türünü Ayarla

Intruder'da Sniper attack seçeneğinin aktif olduğundan emin ol. Payloads paneline git ve Simple list payload türünü seç.

5. Kullanıcı Adı Listesini Yapıştır

Kullanıcı adı adaylarını içeren listeyi payload bölümüne yapıştır.

6. Saldırıya Başla

Start attack butonuna basarak saldırıyı başlat.

7. Sonuçları İncele

Saldırı tamamlandıktan sonra, Length sütununu kontrol et ve bir girişin diğerlerinden daha uzun olduğunu fark et. Bu, geçerli bir kullanıcı adı olduğunu gösterebilir. Invalid username mesajıyla karşılaşılan diğer cevaplardan farklı olarak, Incorrect password mesajını içeren yanıtı bul. Geçerli kullanıcı adını not al.

8. Şifreyi Brute-Force Et

Saldırıyı sonlandır ve username parametresini doğru kullanıcı adıyla değiştir. Ardından password parametresi için yeni bir payload pozisyonu ekle. Şimdi username=identified-user&password=§invalid-password§ şeklinde olmalı.

9. Şifre Listesini Yapıştır

Şifre adayları listesini payload bölümüne yapıştır ve tekrar Start attack butonuna bas.

10. Şifreyi Doğrula

Saldırı bitince, Status sütununu kontrol et. 200 durum kodu alan istekler başarısız girişleri, 302 durum kodu alan ise başarılı girişi gösterir. 302 yanıtını alan şifreyi not al. Giriş yap.

**2FA simple bypass (basit 2fa atlatma)**

Zayıf 2FA Uygulaması: E-posta veya SMS doğrulama kodlarını ele geçirme. Şifre sıfırlama sayfasında, hedefin e-posta adresini girip doğrulama kodunu almak.

TOTP Zaman Hatası: Zaman tabanlı 2FA (TOTP) kullanılıyorsa, zaman senkronizasyonu hatasından faydalanarak doğrulama kodunu tahmin etme.

Sosyal Mühendislik: Kullanıcıyı manipüle ederek, 2FA kodunu almak. Örneğin, telefon numarasını veya e-posta adresini ele geçirip, kullanıcıyı kandırarak kodu almak.

Çerez Ele Geçirme: XSS saldırısı ile kullanıcı oturum çerezlerini ele geçirip, oturum açmak.

SIM Swap: Hedefin telefon numarasını ele geçirerek, SIM kartını takas ettirip, SMS doğrulama kodlarını almak**.**

**Port Swigger çözümü:**

Kendi hesabına eriş linki not et /my-account

Hedef hesaba giriş yap doğrulama gelince url’e /my-account yaz giriş yap

Bu zafiyet, sistemin 2FA doğrulama mekanizmasının, kullanıcı kimlik doğrulama aşamasından sonra URL manipülasyonu ile bypass edilmesine olanak tanımasından kaynaklanır. Yani, sistem sadece şifreyi kontrol etmekte, ancak 2FA doğrulamasını geçerli kullanıcının hesabıyla bağdaştırmada eksik kalmaktadır.