OWASP TOP 10

OWASP (Open Web Application Security Project), açık kaynaklı bir topluluk tarafından geliştirilen, web uygulamalarının güvenliğini artırmayı hedefleyen bir proje topluluğudur. OWASP Top 10, web uygulamalarındaki en yaygın güvenlik açıklarını belirleyen ve bu konudaki en iyi uygulamaları öneren bir listedir.

Broken Access Control: Bu açık, kullanıcıların yetkileri dışında işlemler yapabilmesini sağlar. Örneğin, bir kullanıcı bir yöneticinin yetkilerine sahip bir sayfayı görüntüleyebilir veya değiştirebilir. Uygulamalarda yapılan yanlış veya yetersiz erişim kontrolü ayarları sonucu ortaya çıkar.

Türleri:
Vertical Privilege Escalation: Bir kullanıcı, yetkileri dışında bir seviyede erişim sağlar.
Horizontal Privilege Escalation: Bir kullanıcı, kendi yetkileri içinde olmayan diğer kullanıcıların verilerine erişir.

Önlemler:
Her bir istek için uygun yetkilendirme kontrolleri eklenmeli
Erişim kontrolleri düzgün yapılandırılmalı
Token-based authentication kullanılmalı
İşlemlerin logları tutulmalı

Cryptographic Failures: Bu kategori, verilerin korunmasında kullanılan kriptografik yöntemlerin yetersizliği veya yanlış kullanımı ile ilgili sorunları kapsar. Örneğin, verilerin şifrelenmemesi veya zayıf şifreleme algoritmalarının kullanılması, uygulamalarda kullanılan şifreleme algoritmalarının yanlış seçilmesi veya uygulanması, anahtar yönetimi hataları veya rastgele sayı üretimindeki eksiklikler nedeniyle oluşabilir.

Önlemler:
Şifreleme anahtarlarını güvenli bir şekilde yönetilmeli ve periyodik olarak değiştirilmeli
Endüstri standartlarına uygun, güçlü şifreleme algoritmaları (AES, RSA) kullanılmalı
Anahtar yönetimi hatalarından kaçınılmalı

Injection: Bu açık, kullanıcıdan alınan verilerin doğrudan bir veritabanına, komut satırına veya başka bir sistem bileşenine gönderilmesiyle ortaya çıkar. Uygulama tarafından yürütülen veritabanı sorgularına, komutlara veya diğer işlevlere veri girişlerinde yapılan hatalı denetlemeler veya filtrelemeler nedeniyle zararlı kod enjekte edilebilir.

Türleri:
SQL Injection: Kullanıcının girdiği verilerin SQL sorgularına dahil edilmesiyle ortaya çıkar ve veri tabanına yetkisiz erişim sağlar.
Command Injection: Kullanıcının girdiği verilerin sistem komutlarına dahil edilmesiyle çalışır ve komutların kötüye kullanılmasına yol açar.

Önlemler:
SQL sorgularında parametreli sorguları kullanarak veri girişini sorgulardan ayırılmalı
Nesne-ilişkisel eşleme araçları kullanarak veritabanı işlemleri soyutlanmalı
Kullanıcıdan alınan veriler doğrulanmalı ve temizlenmeli
WAF Kullanımı

Insecure Design: Güvenlik açıklarının tasarım aşamasında ele alınmaması durumudur. Güvenlik önlemleri uygulama tasarımı aşamasında düşünülmemiş olabilir. Güvensiz tasarım,

uygulamanın özellikle kimlik doğrulama, yetkilendirme, veri gizliliği ve bütünlüğü gibi önemli güvenlik konularında hatalar içermesiyle ortaya çıkabilir.

Önlemler:
Tasarım ekibi güvenlik rehberleri ve eğitimlerle desteklenmeli
Tasarım aşamasında güvenlik testleri ve risk değerlendirmeleri yapılmalı
Güvenli tasarım ilkeleri uygulanmalı

Security Misconfiguration: Sistem veya uygulama bileşenlerinin yanlış yapılandırılmasıyla oluşan açıkları ifade eder. Örneğin, varsayılan şifrelerin değiştirilmemesi veya gereksiz hizmetlerin açık bırakılması.

Önlemler:
Konfigürasyon ayarları düzenli olarak gözden geçirilmeli ve güncellenmeli.
Güvenli konfigürasyon standartları belirlenmeli ve uygulanmalı

Vulnerable and Outdated Components: Kullanılan bileşenlerin veya kütüphanelerin eski ve güvenlik açıklarına sahip olmasıdır. Bu, bilinen güvenlik açıkları içeren yazılımların kullanılmasını içerir. Birçok modern uygulama, üçüncü taraf bileşenlerin kullanımını içerir. Bu bileşenler, genellikle web uygulama çerçeveleri, veritabanı yönetim sistemleri, açık kaynak kütüphaneler, sunucu yazılımı ve diğer araçlar gibi yazılım bileşenleri olabilir. Bu bileşenlerde güvenlik açıkları keşfedilmesi halinde, uygulama açığa karşı savunmasız hale gelebilir.

Önlemler:
Güvenlik açıklarına sahip olan bileşenleri kullanmaktan kaçınılmalı
Kullanılan yazılımların ve kütüphanelerin güncellemeleri düzenli olarak takip edilmeli
Güncelleme politikaları oluşturmak

Identification and Authentication Failures: Kullanıcı kimlik doğrulama ve yetkilendirme süreçlerindeki hataları kapsar. Örneğin, zayıf şifreler, eksik çok faktörlü kimlik doğrulama veya kimlik doğrulama hataları. Bu tür bir açık, bir saldırganın bir kullanıcının kimliğini çalmasına veya sahte bir kimlik kullanarak uygulamaya erişmesine izin verebilir.

Önlemler:
Çok faktörlü kimlik doğrulama kullanmak
Düzenli olarak oturum sürelerini ve oturum yenileme işlemlerini yönetmek
Güvenli kimlik doğrulama protokollerini uygulayın

Software and Data Integrity Failures: Yazılım ve verilerin bütünlüğünün korunmamasıyla ilgili sorunları ifade eder. Örneğin, yazılım güncellemelerinin veya verilerin doğruluğunun kontrol edilmemesi. Bu zafiyet, bir saldırganın yazılım veya veri sistemini hedef alarak sistemi istismar etmesine veya manipüle etmesine izin verebilir. Yazılım ve veri bütünlüğü hataları, bir yazılım güncellemesi sırasında, yazılımın kötü amaçlı bir saldırgan tarafından değiştirilmesi veya bir saldırganın bir veri depolama ortamına kötü amaçlı yazılım yerleştirmesi gibi birçok farklı şekilde ortaya çıkabilir. Bu tür saldırılar sonucunda, saldırgan verileri çalabilir, verileri bozabilir veya sistemi kontrol etmeye başlayabilir.

Önlemler:
Yazılım ve veri kaynakları güvenilir ve doğrulanmış kaynaklardan temin edilmeli
Kod ve yapılandırma değişiklikleri için bir inceleme süreci oluşturmak
Yazılım ve veri yedeklemeleri oluşturmak
Yazılım güncellemeleri ve veri bütünlüğü için dijital imzalar kullanın.
Yazılım güncellemelerini düzenli olarak yüklemek

Security Logging and Monitoring Failures: Güvenlik olaylarının yeterince kaydedilmemesi ve izlenmemesiyle ilgili eksikliklerdir. Güvenlik olaylarına dair yeterli loglama ve izleme olmaması. Bu tür bir açık, kötü amaçlı aktivitelerin tespit edilememesi veya güvenlik olaylarına yanıt verilememesi gibi sonuçlara neden olabilir.

Önlemler:
Otomatik izleme araçları kullanımı
Log doğrulama ve analizi
Kapsamlı loglama yapılması

Server-Side Request Forgery (SSRF): Bir saldırganın, sunucunun kendi kaynaklarına veya arka plandaki hizmetlere kötü niyetli istekler göndermesiyle ilgili bir açık türüdür. Bir SSRF saldırısı, bir uygulamanın doğrulama sürecini atlayarak veya bir URL'de bir sunucu adresi veya IP adresi gibi güvenliği kontrol etmeyen bir parametre kullanarak gerçekleştirilebilir. SSRF, hedef sunucu için ciddi bir güvenlik tehdidi oluşturur, çünkü saldırgan sunucuya istekler göndererek hassas verileri çalabilir, sunucunun kaynaklarını tüketebilir, sunucunun kontrolünü ele geçirebilir veya sunucunun çalışmasını bozabilir.

Önlemler:
Giriş doğrulaması
Yetkisiz Erişimi Engelleme
Güvenli URL işleme
Firewall ve Güvenlik Duvarları kullanımı
Sunucu ayarlarının kontrol edilmesi