OWASP TOP 10

OWASP TOP 10 Nedir?

OWASP (Open Worldwide Application Security Project), web uygulamalarının güvenliğini sağlamaya çalışan, açık kaynaklara önem veren bir topluluktur. OWASP TOP 10, web uygulamalarında ortaya çıkan en kritik güvenlik risklerini listeler.

1. Broken Access Control

Web uygulamalarında her kullanıcının bir yetki alanı olmalıdır yani sıradan bir kullanıcı admin yetkilerine erişmemelidir buna erişim kontrolü denir. Erişim kontrolünde yapılan hatalar sonucunda hassas bilgilere yetkisiz erişim sağlanır, veri görüntüleme, değiştirme, silme gibi işlemler gerçekleştirilebilir.

1.1. Korunma Yöntemleri

- CORS (Cross-Origin Resource Sharing) kullanımını azaltın.
- Web sunucusu dizin listesini devre dışı bırakın.
- Erişim kontrol hatalarının log'larını tutun, tekrarlanan log'ları inceleyin.
- JWT token'ları kısa ömürlü olmalı, uzun ömürlü kullanılması gerekiyorsa OAuth standartı kullanın.

2. Cryptographic Failures

Web uygulamaları, kullanıcılarının hassas verilerini saklamak zorundadır. Bu verileri de şifreli şekilde tutması daha güvenlidir. Verilerin clean-text halde saklanması, şifrelemesinin zayıf tutulması önemli güvenlik sorunlarına yol açar.

2.1. Korunma Yöntemleri

- Zayıf şifreleme algoritmaları kullanmayın.
- Kullanımdan kaldırılmış hash fonksiyonları yerine salted hash fonksiyonları kullanın.
- Hassas verileri şifreleyin.

3. Injection

Kullanıcıdan alınan girdilerin filtrelenmediği, kontrol edilmediği durumlarda ortaya çıkar ve web uygulamalarına ciddi hasarlar verebilir. Zafiyeti tespit etmenin en iyi yolu uygulamanın kaynak kodunu okumaktır.

3.1. Korunma Yöntemleri

- Sunucu tarafı girdi doğrulama kullanın.
- Kullanıcıdan girdi alan her yeri test edin.

4. Insecure Design

Eksik veya etkisiz kontrol tasarımı olarak ifade edilen farklı zayıflıkları içerir. Güvensiz tasarım ile güvensiz uygulama arasında bir fark vardır. Güvenli tasarım içinde güvensiz uygulama barındırabilir. Güvensiz tasarım güvenli uygulama ile düzeltilemez.

4.1. Korunma Yöntemleri

- Güvenli bir yazılım geliştirme yaşam döngüsü oluşturun.
- Güvenli tasarım modelleri içeren kütüphaneler kullanın.
- Tehdit modellemesi kullanın.
- Tüm akışlar için birim ve entegrasyon testi uygulayın.

5. Security Misconfiguration

Uygulamaların işlevlerine uygun güvenlik konfigürasyonları yapılmadığında, varsayılan oturum açma kimlik bilgileri kullanıldığında, güncel olmayan ve/veya zafiyetli teknoloji kullanıldığında ortaya çıkar.

5.1. Korunma Yöntemleri

- Kullanılmayan özellikleri, framework'leri kaldırın.
- Kullanılan teknolojilerin güncelliğini kontrol edin.

6. Vulnerable and Outdated Components

Kullanılan bileşenler güncel değilse veya bilinen bir zafiyet içeriyorsa, güncellenen veya yamalanan bileşenlerin uygulamayla uyumluğu test edilmezse ortaya çıkar.

6.1. Korunma Yöntemleri

- Kullanılmayan bileşenleri, özellikleri, bağımlılıkları kaldırın.
- CVE ve NVD kaynaklarını düzenli inceleyin.
- Yalnızca güvenli bağlantılar üzerinden bileşenlerini güncelleyin.

7. Identification and Authentication Failures

Kullanıcıların kimliğinin doğrulanması ve oturumunun yönetilmesi kimlik doğrulama tabanlı saldırıları önlemek için önemlidir. Uygulama brute force gibi yöntemlere izin veriyorsa, varsayılan veya zayıf parola kullanımına izin veriyorsa, güvensiz şifre unuttum bölümlerini içeriyorsa, güvensiz çok faktörlü doğrulama süreçleri içeriyorsa bu zafiyet ortaya çıkar.

7.1. Korunma Yöntemleri

- Brute force gibi yöntemlerden korunmak için rate-limit kullanın.
- Zayıf parola kullanımını engelleyin.
- Kullanıcı adı, şifre gibi kısımların tespit edilmesine karşı tek tip hata mesajı kullanın.
- Başarısız oturum açma denemelerini log'layın.

8. Software and Data Integrity Failures

Yazılım ve veri bütünlüğü hataları, bütünlük kurallarına uymayan kod ve altyapı ile ilgilidir. Bir uygulamanın güvenilmeyen kaynaklardan eklenti, kütüphane veya modül indirmesiyle oluşur.

8.1. Korunma Yöntemleri

- Yazılım, eklenti indirilirken kaynakların güvenilirliğini kontrol edin.
- Yazılım destek zinciri güvenlik aracı kullanın.

9. Security Logging and Monitoring Failures

Uygulamalardaki aktif ihlallerin tespit edilmesine, artırılmasına ve bunlara yanıt verilmesine yardımcı olur. Log kaydı ve izleme olmadan ihlaller tespit edilemez.

9.1. Korunma Yöntemleri

- Sunucu tarafındaki girdi doğrulama hatalarının log dosyasına kaydedildiğinden emin olun.
- Log verileri şifreli olarak saklayın.
- Etkili bir izleme ve uyarı sistemi kullanın.

10. Server Side Request Forgery (SSRF)

SSRF zafiyeti, bir web uygulaması kullanıcı tarafından sağlanan URL'i doğrulamadan bir kaynağı getirdiğinde ortaya çıkar. Bulut hizmetlerinin karmaşıklığı nedeniyle SSRF daha ciddi bir sorun haline gelmektedir.

10.1. Korunma Yöntemleri

- Kullanıcıdan alınan girdi verilerini filtreleyin.
- HTTP yönlendirmelerini devre dışı bırakın.
- Black-list yerine white-list kullanın.