OWASP (Open Web Application Security Project)

Web uygulamalarında ortaya çıkan yaygın güvenlik açıklarını belirlemek ve bu açıkların önüne geçilmesini sağlamak için web uygulamarında en fazla ortaya çıkan zaafiyetleri listeleyen açık kaynak kodlu bir projedir.
Güvenli web uygulamaları geliştirmek isteyen geliştiriciler için OWASP TOP 10 önemli bir konudur.

1-) Broken Access Control (Yetkisiz Erişim)

Bir web uygulamasının kullanıcılarının sadece belirli kaynaklara (veriler, sayfalar, özellikler vb.) belirli izin erişimine sahip olmalarını sağlayan bir mekanizmadır.

Zaafiyet:
Kullanıcıların sadece izin verilen kaynaklar ve verilere erişmeleri gerekirken, yetkisiz kaynaklara da erişim sağlaması durumunda ortaya çıkmaktadır.
Örneğin, bir admin kullanıcısının yönetim paneline girmeleri,
Kendi hesapları ve verileri dışında başka kullanıcılarında verilerine ve hesaplarına erişim sağlamaları.

Neden Kaynaklanıyor:
Uygulamalarda yapılan yanlış ve yetersiz erişim kontrolü ayarları,
Sistem ve uygulamalarda yanlış yapılandırma örneğin, tüm kullanıcıların belirli kaynaklar üzerinde yetki sahibi olması gibi.

Nasıl Önlenir:
Erişim kontrolleri düzgün ve güvenli yapılandırılımalıdır,
Uygulama geliştirilme sürecindeyken güvenlik testleri yapılmalı,

2-) Cryptographic Failures (Şifreleme Hataları)

Şifreleme hataları olarak geçmektedir.
Zaafiyet:
Kriptografi kullanılarak saklanan hassas verilerin, şifrelerin vb. yanlış şifrelemeler ve hatalı algoritma kullanımı nedeniyle saldırganların şifreleri çözüp hassas verilere erişim sağladığı bir zaafiyet türüdür.
Neden Kaynaklanıyor:
Zayıf ve eski şifreleme algoritmaları kullanımı, kriptografi anahtarların kötü şifrelenmesi ve korunması ,Kriptografik standartların ve en iyi uygulamaların ihmal edilmesi.

Nasıl Önlenir:
Doğru şifreleme algoritmaları kullanılmalı, Güncel algoritmalar kullanılmalı,
Kriptografik sistemler sürekli test edilmeli ve denetlenmeli

3-) Injection(Enjeksiyon)

Kullanıcıların web uygulamasındaki input yerlerine zararlı kod enjekte etmesi olayıdır.

Zaafiyet:

Bir saldırganın uygulamadaki veri giriş yerlerine zaralı kod göndererek veri tabanı işeyişini bozmasını, sunucu işleyişini bozmasını, kontrolü ele geçirmesini,

Hassas verileri çalmasına kadar gidebilecek bir güvenlik açığıdır.

Neden Kaynaklanıyor:

Kullanıcıdan gelen verilerin düzgün bir şekilde doğrulanmaması ve filtrelenmemesi sonucunda ortaya çıkmaktadır.

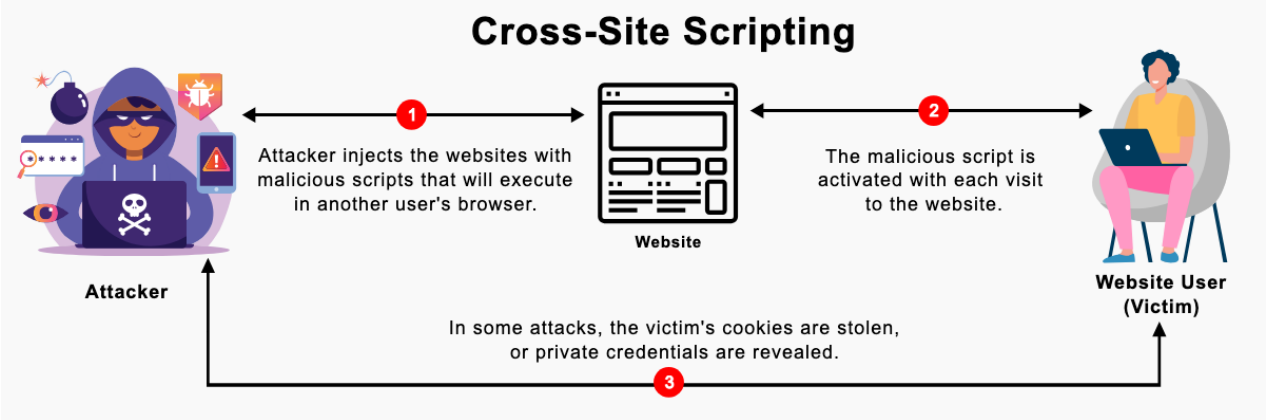
Örneğin, kullanıcıdan alınan bir verinin direkt olarak sunucuda çalıştırılması

Nasıl Önlenir:

Kullanıcıdan gelen verileri filtreleyip doğrulayarak zararlı kodları temizleyerek girdileri uygun duruma getirmeliyiz.

SQL enjeksiyonu önlemek için parametrik sorgular kullanmak,

Kodlama standartlarına uymak



4-) Insecure Design (Güvensiz Tasarım)

Zaafiyet:
Uygulama tasarımında yapılan hatalar ve eksiklikler nedeniyle ortaya çıkan bir açıktır.
Tasarım aşamasında uygulama güvenlik önlemlerinin yeterince göz önünde bulundurulmaması ve hatalı uygulanmasıdan dolayı oluşur.

Neden Kaynaklanıyor:
Uygulama tasarlanırken daha çok kullanıcı deneyimi ve işlevsellik göz önünde bulundurulur ve güvenlik göz ardı edilir.
Güvenlik gereksinimlerinin belirlenirken yetersiz ve eksik kalması
Hızlı bir şekilde geliştirme isteğinden dolayı güvenlik için gerekli kodlamaların yapılmaması
Nasıl Önlenir:
Uygulama tasarım aşamasının başından beri güvenlik gereksinimleri tespit edilmeli ve gerekli önlemler alınmalı,
Potansiyel tehdit ve risklerin tahmin edilmesi,
Tasarım aşamasında sürekli güvenlik testleri yapılmalı,
Kodlama ve güvenlik standartlarına uymak

5-) Security Misconfiguration (Güvenlik Yanlış Yapılandırması)

Zaafiyet:
Bir sunucu veya web uygulamasındaki tüm güvenlik kontrollerinin yapılmaması veya yanlış yapılandırılmasından dolayı oluşan bir güvenlik açığıdır.
Neden Kaynaklanıyor:
Varsayılan Parolaların değiştirilmemesi, kullanıcıların kolayca tahmin edebileceği parolalar anlamına gelir.
Sistem veya uygulama yapılandırmalarının düzenli olarak yapılmaması (denetleme ve güncelleme)
Kullanılmayan hizmetlerin ve gereksiz özelliklerin etkin olması saldırganların bu hizmetler üzerinden sisteme erişmesini kolaylaştırır.
Uygunsuz dosya izinleri yetkisiz erişime neden olabilir.
Nasıl Önlenir:
Sistem kurulurken varsayılan ayarlar (dosya izinleri, varsayılan parolalar, kullanıcı adları) değiştirmeli güvenli olmayan varsayılan ayarları güvenli hale getirmeliyiz.
Kullanılmayan hizmetleri ve uygulamaları devre dışı bırakarak sadece gerekli sistem bileşenlerini çalıştırmak.
Güvenlik yamalarını ve güncellemelerini takip etmeliyiz
Yazılımın güvenlik düzeyini denetleyip test etmeliyiz

6-) Vulnerable and Outdated Compenents (Zayıf ve Güncellenmemiş Bileşenler)

Bir sistemde kullanılan yazılım bileşenlerinin(açık kaynaklı kütüphaneler , frameworkler, modüller) eski, zayıf ve güvenlik açıklarına sahip olan sürümlerinin kullanılmasını ifade eder.

Zaafiyet:

Sistemde bulunana üçüncü taraf yazılım bileşenlerinin eski sürümleri veya güvenlik açığı bulunan sürümlerinin saldırganlar tarafından istismar edilmesidir.

Bu bileşenler bilinen zaafiyetleri barındırdıkları için ve bu zaafiyetler saldırganlar tarafından açıkça istismar edilebilir.

Neden Kaynaklanıyor:

Kullanılan bileşenlerin eski sürümlerinin kullanılması, güncellenmemesi
Güvenilir olmayan kaynaklardan indirilen bileşenlerin güvenilirliği yoktur zararlı ve zayıf olma ihtimali daha yüksektir.
Sistemde bulunan yazılım bileşenlerinin güvenlik durumunun takip edilmemesi

Nasıl Önlenir:

Sistemde ve uygulamalarda kullanılan bütün bileşenleri düzenli olarak güncellemek
Güvenlik yamaları yayınlandığında en kısa sürede uygulanmalı
Sistemde kullanılan bütün bileşenlerin sürümünü takip etmek

7-) Identification and Authentication Failures (Kimlik Doğrulama ve Kimlik Belirleme Hataları)

Zaafiyet:

Kimlik doğrulama bir kullanıcının iddia ettiği kimlik olduğunu doğrulaması sürecidir, kimlik belirleme ise bir kullanıcının belirleme sürecidir. İşte bu süreçelerde meydana gelen hatalar kullanıcıların kimliklerinin yanlış belirlenmesine ya da kimlik doğrulama mekanizmalarının atlatılmasına neden olmaktadır.

Bu tür bir açık bir saldırganın bir kullanıcının kimliğini çalmasına veya sahte bir kimlik kullanarak uygulamaya erişim sağlamasına neden olabilir.

Neden Kaynaklanıyor:

Kullanıcıların kolay ve kolay tahmin edilebilir parolalar kullanmasına izin veren veya güçlü parola gereksinimlerinin olmaması
Çok faktörlü kimlik doğrulama eksikliği olması sadece kullanıcı adı ve parola gereksinimi saldırganlar tarafından kolayca aşılabilir
Oturumların doğru bir şekilde yönetilmemesi örneğin, oturumun zaman aşımına uğramaması

Nasıl Önlenir:

Güçlü parola politikaları uygulanmalı, kullanıcıların güçlü parolalar seçmelerini zorunlu kılmalı

Çok faktörlü kimlik doğrulama günümüz önemli uygulamaların olmazsa olmazıdır, kullanıcı kimliğini doğrulamak iki veya daha fazla doğrulama faktörü kullanılabilir.

8-) Software and Data Integrity Failures (Yazılım ve Veri Bütünlüğü Hataları)

Zaafiyet:
Bir sistemdeki yazılım ve verilerin yetkisiz veya zararlı değişikliklere karşı korunmaması durumunda ortaya çıkan güvenlik açığıdır. Bu tür hatalar, saldırganın sisteme zarar vermesine verileri manipüle etmesine olanak tanıyabilir.

Neden Kaynaklanıyor:
Yazılım güncellemeleri, yamalar ve verilerin bütünlüğünü doğrulamak için yeterli güvenlik önlemlerinin alınmaması
Yazılım tedarik zincirine yapılan saldırılardan dolayı zararlı kodun yazılıma eklenmesi
Yazılım ve bileşenlerinin dijital olarak imzalanmaması, bu bileşenlerin bütünlüğünü doğrulama sürecini zorlaştırır.

Nasıl Önlenir:
Yazılım ve verilerin değiştirilmediğini doğrulamak için dijital imzalar kullanılmalı
Yazılım güncellemeleri düzenli olarak yüklenmeli
Güçlü erişim kontrolleri uygulanmalı
Güvenli ve standartlara uygun yazılım geliştirme yöntemleri kullanılmalı

9-) Security Logging and Monitoring Failures (Güvenlik Günlüğü ve İzleme Hataları)

Monitor ve loglama yapılmadan ihlaller tespit edilemez. Örneğin, evinizin veya arabanızın kapısı her açıldığı ve kapandığında size bilgi gelse, bu durumda hırsızları durdurma ve yakalama şansınız çok yüksektir.

Zaafiyet:

Web uygulamalarında güvenlik olaylarının yeterince izlenmemesinden veya kaydedililmemesinden dolayı kaynaklanan bir güvenlik açığıdır. Bu tür hatalar potansiyel saldırıların tespit edilmesini ve saldırılara hızlıca yanıt verilmesini zorlaştırabilir.

Logları kaydetmek, izlemek tehdit analizleri yapmayı kolaylaştırır ve bu da gelecekteki saldırıları önlemek için kritik öneme sahiptir.

Neden Kaynaklanıyor:

Eksik log tutma örneğin, başarısız giriş denemeleri, veri erişimi veya yetkisiz işlem denemeleri gibi olaylar loglanmadığında bu olaylar tespit edilemez.

Log'ların yeterince izlenmemesi veya analiz edilmemesi olası saldırıların tespit edilememesine neden olabilir.

Log'ların yeterince korunmaması durumunda saldırganlar izlerini silebilir.

Olası saldırılar gerçekleştiğinde uyarı mekanizmalarının bulunmaması

Nasıl Önlenir:

Güvenlik olaylarının izlenmesi ve kaydedilmesi için uygun mekanizma ve yazılımlar kullanmak

Log kayıtlarının sürekli olarak incelenmesi

Gerekli uyarı mekanizmaları kullanmak

10-) Server-Side Request Forgery SSRF (Sunucu Taraflı İstek Sahteciliği)

Zaafiyet:

Bir saldırganın bir uygulamanın sunucusunu kullanarak sunucuya veya sunucunun erişebileceği diğer sistemlere yetkisiz istekler göndermesine yol açan bir güvenlik açığıdır.

Bu tür saldırılar, sunucunun arka uç sistemlerle, veri tabanlarıyla, iç hizmetlerle, veya üçüncü taraf API'lerle etkileşime girdiği durumlarda meydana gelebilir.

Neden Kaynaklanıyor:

Kullanıcı girdilerinin yeterince doğrulanmaması kullanıcıya IP adresi URL veya diğer kaynaklar üzerinden erişim imkanı verildiğinde saldırgan bu input yerlerini manipüle ederek sunucu isteklerini kendi amaçları doğrultusunda yönlendirebilir.

Uygulamalar iç ağdaki kaynakların güvenli olduğunu varsayabilir. Ama bir saldırgan bu zaafiyeti kullanıp suncu üzerinden iç ağlara erişim sağlayabilir.

Uygulamalar üçüncü taraf servisleri veya API'leri çağırırken güvenilir olduklarından emin olamalıdır. Saldırganlar bu servisleri manipüle ederek SSRF saldırılarını gerçekleştirebilir.

Nasıl Önlenir:

Kullanıcıdan gelen girdiler titizlikle doğrulanıp filtrelenmeli

Güvenli URL listesi kullanılabilir sunucunun erişebileceği IP adresi ve URL leri sınırlandırarak sadece bu listedeki kaynaklara istek yapılmasına izin verilebilir.

Güvenlik duvarı kullanılarak girdileri filtreleyip olası saldırı durumu önlenebilir.