1.Broken Access Control (Kırık Erişim Kontrolü)

Zafiyet Nedir?

Uygulamanın erişim kontrolü yeterince iyi değilse kullanıcılar diğer kullanıcıların da verilerine erişebilir. Haliyle görülmemesi gereken bilgiler ele başkaları tarafından gözükebilir.

Neden Kaynaklanır?

Kimlik doğrulama ve yetkilendirme mekanizmalarının yanlış yapılandırılmasının bir sonucu olarak ortaya çıkar ve hassas verilere yetkisiz erişen kullanıcının verileri çalması gibi kritik sonuçları olur.

Türleri

IDOR: Bir kullanıcının başka bir kullanıcının verilerine doğrudan erişim sağlamasıdır. (URL'deki id parametresini değiştirerek)

Yatay Erişim: Aynı seviyedeki kullanıcıların birbirlerinin verilerine yetkisiz erişimidir. (iki farklı kullanıcının birbirlerinin mesajlarını okuyabilmesi)

Dikey Erişim: Düşük seviyeli bir kullanıcının, daha yüksek seviyeli verilere erişmesidir. (standart kullanıcının admin paneline erişebilmesi)

Nasıl Önlenir?

Güvenlik Önlemleri Alarak: Bir kullanıcının sadece yetki düzeyinde erişim sağlamasını garanti edecek şekilde uygulamada erişim kontrolleri koyarak önlenebilir.

Roller Atayarak: Her bir kullanıcıyı uygun rollere dağıtıp, yetki sistemini bu roller üzerinden yapmayla önlenebilir.

Uygulamaya Girilenleri Kontrol Ederek: Bir kullanıcının uygulama tarafından atanan veriyi değiştirip zararlı içerik oluşturmasının önüne geçmek için filtrelemeyle önlenebilir.

2.Cryptographic Failures (Şifreleme Hataları)

Zafiyet Nedir?

Şifreleme tekniğinin iyi olmaması yada hatalı kullanılmasıyla meydana gelir. Bu yüzden bilgilerin ele geçirilmesi, şifrelerin kırılmasıyla sonuçlanabilir.

Neden Kaynaklanır?

Şifreleme Algoritmasının Zayıflığı: Şifreleme Algoritması zayıfsa yada güncelliğini yitirmişse veriler güvende olmaz.

Şifreleme Anahtarı: Anahtar yeterince güçlü değilse veya çalınırsa veriler güvende olmaz.

Düzgün olmayan veri şifrelemesi: Hassas verilerin veritabanına doğrudan yazılmaktansa maskelenmesidir. Bunun iyi yönetilememesi sonucu meydana gelir.

Güçsüz şifreleme protokolü kullanılmasıyla çalınan veriler şifrelenmiş olsa dahi kolay bir şekilde şifresi kırılabilir.

Nasıl Önlenir?

Güçlü şifreleme algoritması kullanarak önlenebilir

Anahtarı koruyarak önlenebilir

Veri maskelemeyi doğru şekilde yaparak önlenebilir. Böylece veritabanından hassas veriler çalınsada şifreli bir biçimde korunacaktır.

Şifreleme standartları yüksek şifreleme yöntemleriyle veriler şifrelenip saklanmalıdır. Böylece saldırgan verileri ele geçirse dahi açıp okuyamayacaktır. (AES-256 yada RSA ile şifrelenmiş verilerin sahibi dışında açılması çok daha zordur)

3. Injection

Nedir?

Saldırganın zararlı komutları uygulamaya enjekte ettiği güvenlik açığıdır. Çok çeşitli yerlerden bu enjeksiyon işlemi yapılabilir böylece görülememesi gereken bilgiler görülebilirken getirilmemesi gereken verilerde istemciye getirilir.

Neden Kaynaklanır?

Veri doğrulama filtresinin eksikliğinden dolayı, kullanıcının girdiği veriler kontrol edilmeden işlenirse bir açık meydana gelme olasılığı doğar.

En Yaygın Türleri:

SQL Injection:

Saldırganın SQL Sorgusuna zararlı kod eklemesiyle meydana gelir.

(kullanıcı adı şifre yerine 'admin' OR '1'='1' yazılması sorguyu olduğu gibi true olarak gösterecektir. SQL sorgusundaki where şartındaki true bilgiye göre de çalışacağı için istenilen içerik admin şartı aranmaksızın getirilecektir)

Komut Injection:

Web uygulaması bir dosyayı sistem komutuyla okuyabilir bu sayede saldırgan zararlı içerikleri sisteme gömebilir.

(sistem komutuyla okunacak bir dosyaya '; rm-rf /' komutunu eklemesiyle root dosyayı sileceği için sistem çökecektir)

XML Enjeksiyonu

XML verileri veri iletimi yada depolanması açısından kullanılır fakat bu veriler güvenli bir şekilde işlenmezse içerisindeki zararlı kod uygulamanın işleyişine müdahale edebilir.

(örneğin kullanıcı girdilerini xml formatta kaydeden bir web uygulamasında,

```
<user> <name>John Doe</name>
        <role>admin</role>
       <password>password123</password>
       <!-- Zararlı içerik -->
        <externalEntity> <!ENTITY xxe SYSTEM "file:///etc/passwd"> </externalEntity>
 </user>
```

İçerikli bir XML gönderdiği vakit, saldırganın '/etc/passw' dosyasını okumasına izin verecektir.)

Önlemler:

Kullanıcıdan alınan XML içeriklerinde zararlı veri olmadığına emin olunacak bir biçimde algoritma kurgulanarak uygulamada işlenmelidir.

XML kütüphanelerindeki dış varlıkları(external entities) devre dış bırakmak koruma sağlayacaktır.

Kullanılan XML kütüphanelerinin güncel olduğundan, en son sürümünün bugları fixlenmiş olduğuna dikkat edilmelidir.

4. Insecure Design (Güvensiz Tasarım)

Zafiyet Nedir?

Web uygulamasını baştan zayıf bir şekilde dizayn ederken güvenlik risklerinin yeterince dikkate alınmayıp potansiyel açıklara kapı aralanması durumudur.

Neden Kaynaklanır?

Uygulamayı kurma aşamasında, mimaride gerekli güvenlik önlemleri alınmadığı taktirde meydana gelecek açık tipidir.

Kimlik doğrulama, yetkilendirme ve veri gizliliği gibi konularda açık sunacak bir zafiyet tipidir.

Sistemi tasarlarken şifreleri zayıf algoritmayla saklamak yada dümdüz metin olarak saklamak.

Kullanıcı girişleri eksik yada hiç doğrulama mekanizmasına tabii olmadan yetkisiz kişilerin sisteme erişmesine olanak tanımak.

Örneğin, sistemde ekstra güvenlik katmanı bulunmaması durumu.(iki faktörlü doğrulama)

Nasıl Önlenir?

Sistem mimarisi başında ihtiyaçlar listesine güvenlik gereksinimlerinin eklenmesiyle birlikte buna istinaden bir tasarım sürecinin yürütülmesiyle önlenebilir.

Gözden kaçan yada güncelliğini yitirmiş tasarım parçalarının tespiti için zafiyet testi yapılıp ilgili açıkların giderilmesiyle önlenebilir.

5. Security Misconfiguration (Güvenlik Yanlış Yapılandırması)

Zafiyet Nedir?

Uygulamanın yada sistemin güvenlik ayarlarının hatalı, eksik yada default haliyle bırakılmasından doğan bir zafiyettir. Bu sayede saldırganlar uygulama sunucusuna erişip veri sızıntısı yapabilir.

Neden Kaynaklanır?

Varsayılan ayarların kullanımına devam edilmesiyle meydana gelebilir (ilk kayıtta atanan şifrenin değiştirilmemesi)

Güvenlik açığı çıkmış bileşenlerin kullanımına devam edilmesiyle meydana gelebilir (güncelleme eksikliği)

Güvenlik ayarlarının yanlış yada eksik yapılandırılmasıyla meydana gelebilir (gereksiz hizmet veya portların açık tutulması)

Kullanılmayan özelliklerin açık tutulması saldırganlara saldırı yüzeyini genişletme imkanı sunar.

Nasıl Önlenir?

Varsayılan güvenlik ayarlarının gözden geçirilmesi ve atanan username/password bilgilerinin değiştirilmesi

Kullanılan yazılım/donanım bileşenine yama geldikçe vakit kaybetmeden güncellemesinin gerçekleştirilmesi

Zafiyet taraması yapılması

Gereksiz hizmetlerin ve portların devre dışı bırakılmasıyla önlenebilir

6. Vulnerable and Outdated Components (Savunmasız ve Güncel Olmayan Bileşenler)

Zafiyet Nedir?

Savunmasız ve güncel olmayan bileşenler, yazılımda kullanılan ve artık desteklenmeyen, eski veya bilinen güvenlik açıklarına sahip bileşenlerdir. Bu tür zafiyetler, bu bileşenlerin uygulama içerisindeki yetkisi arttıkça daha büyük bir risk haline gelir.

Neden Kaynaklanır?

Güncellenmemiş bileşenlerin kullanılması.

Güvenlik zafiyetleri içeren bileşenlerin mevcut durumda kullanılması.

Artık geliştirici tarafından desteklenmeyen bileşenlerin kullanılması.

Yanlış yapılandırılmış bileşenlerin kullanılması.

Türleri Nelerdir?

Outdated Components (Güncellenmemiş Bileşenler): Eski versiyonların kullanılmasından kaynaklanır.

Vulnerable Components (Zafiyet İçeren Bileşenler): Zafiyet barındıran veya sonradan bu duruma düşen bileşenler.

Unsupported Components (Desteklenmeyen Bileşenler): Geliştirici tarafından artık desteklenmeyen bileşenler.

Misconfigured Components (Yanlış Yapılandırılmış Bileşenler): Yanlış yapılandırmadan doğan zafiyetler.

Nasıl Önlenir?

Bileşenlerin güncellemelerini düzenli olarak takip etmek.

Kullanılmayan bileşenleri silmek.

Desteklenmeyen bileşenlerin yerine yeni teknolojiler kullanmak.

Bileşenleri resmi kaynaklardan temin etmek.

Zafiyet taraması yaptırmak.

7. Identification and Authentication Failures (Tanımlama ve Kimlik Doğrulama Hataları)

Zafiyet Nedir?

Oturum ve kimlik bilgisi yönetiminin düzgün yapılmadığı durumlarda ortaya çıkan zafiyetlerdir. Saldırganlar, kullanıcıları yanlış veya eksik denetimlerden dolayı taklit etmeye çalışır ve o kullanıcıymış gibi doğrularlar.

Neden Kaynaklanır?

Zayıf parolaların kullanılması.

Parolaların güçlü bir hash algoritması ile saklanmaması.

Yeterli deneme hakkı sınırlamasının olmaması.

Çok faktörlü kimlik doğrulamanın kullanılmaması.

Türleri Nelerdir?

Zayıf Parola Koruması: Kolayca tahmin edilebilecek veya brute force saldırılarına maruz kalabilecek parolaların kullanılması.

Parola Saklanma Şekli: Parolaların düz metin olarak veya güvensiz bir şekilde saklanması.

Oturum Süresi: Kullanıcı oturumlarının çok uzun süreli olması.

Yetki Kontrol Zayıflığı: Doğrulama sonrasında, yetkilerin doğru kontrol edilmemesi.

Parola Geri Alma Süreci: Güvensiz parola geri alma yöntemleri.

Nasıl Önlenir?

Güçlü parolalar seçilmeli.

Çok faktörlü doğrulama kullanılmalı.

Kullanıcının bilgileri şifrelenerek saklanmalı.

Kullanıcıya belirli bir deneme hakkı verilmeli.

Yanlış giriş yapıldığında hangi bilginin yanlış olduğu kullanıcıya söylenmemeli.

Captcha kullanımı.

Oturum süresinin çok uzun tutulmaması.

Zafiyet taraması yaptırılmalı.

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

Zafiyet Nedir?

Yazılım ve veri bütünlüğü hataları, kullanılan altyapı ve kodların bütünlük ihlallerine karşı koruma sağlamadığında ortaya çıkar. Güvenilmeyen kaynaklardan alınan kodlar ve otomasyon süreçlerinde yeterli güvenlik önlemlerinin olmaması bu tür zafiyetlerin oluşmasına neden olabilir.

Neden Kaynaklanır?

Güvenilmeyen veya doğrulanmamış kaynaklardan alınan kodların kullanılması.

Eski veya desteklenmeyen bileşenlerin kullanılması.

İşlem boyunca yeterli doğrulama yapılmaması.

Türleri Nelerdir?

Güvenli Olmayan Kod Kullanımı: Güvenilmeyen veya doğrulanmamış kaynaklardan alınan kodların kullanılması.

Eski veya Desteklenmeyen Bileşenler: Güvenlik güncellemeleri almayan yazılımlar.

Giriş Doğrulaması: İşlem hattı boyunca yeterli doğrulama yapılmaması.

Nasıl Önlenir?

Verilerin kaynağından emin olunmalı.

Bileşenlerin kısıtlı erişime sahip olması sağlanmalı.

CI/CD düzeninde uygun ayırma, erişim denetimi ve yapılandırma yapılmalı.

Kodlar test edilmeli ve güncellenmeli.

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

Zafiyet Nedir?

Güvenlik günlüğü ve izleme hataları, sistem veya uygulamanın güvenliğini sağlamak için gerekli olan günlüğe kaydetme ve izleme mekanizmalarının eksik veya yetersiz olmasından kaynaklanan zafiyetlerdir. Bu tür zafiyetler, saldırganların sistemdeki faaliyetlerini fark ettirmeden sürdürmesine olanak tanır.

Neden Kaynaklanır?

Kayıtların eksik veya hatalı tutulması.

Olayların izlenememesi veya yetersiz izlenmesi.

Toplanan verilerin analizinin eksik olması.

Türleri Nelerdir?

Yetersiz Günlükleme: Önemli olayların veya kullanıcı aktivitelerinin yeterince güncellenmemesi.

Yanlış Yapılandırma: Günlükleme ve izleme sistemlerinin hatalı yapılandırılması.

Günlüklerin Saklanma Süresi: Günlüklerin yeterince uzun bir süre boyunca saklanmaması.

İzleme Araçlarının Eksikliği: Güvenlik izleme araçlarının eksik veya yetersiz olması.

Yetersiz Anomali Tespiti: Sıradışı aktiviteleri tespit etmek için yeterli algoritmaların kullanılmaması.

Nasıl Önlenir?

Loglama ve izleme süreçleri düzgünce yürütülmeli.

IDS/IPS sistemleri kullanılmalı.

Loglar yedeklenmeli.

Gerçek zamanlı alarm oluşturabilen monitoring sistemleri kullanılmalı.

Logların bütünlüğü güvence altına alınmalı.

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

Sunucular da istemciler gibi kendi iç ağlarında pek çok servise istekte bulunur. Ve gelen cevapları işleyip kullanıcıya istediğini verir. Kullanıcının kendisi iç ağdaki servislere erişemeyeceği için sunucular bir aracı görevi üstelenirler.İşte sunucu taraflı istek sahteciliği olarak da bilinen SSRF, sunucuların yaptığı isteklerin manipüle edilmeye çalışıldığı bir zafiyet türüdür. Geçtiğimiz günlerde React içerisinde API istekleri için kullanılan Axios içerisinde de SSRF açığı çıkmıştı.

Neden Kaynaklanır?

Normalde erişilemeyen iç ağlarda yer alan diğer bileşenler, bunların Ip'leri ve çalıştırdıkları servislerin hangi portlarda çalıştıkları öğrenilebilir ve ardından saldırılabilir.

Türleri

Basic SSRF: Sunucu, istenen yanıtı saldırgana gösterir.

Blind SSRF: Saldırgana yanıt dönmez. Bu sebeple saldırganın güvenlik açığını doğrulamak için yollar bulması gerekir.

Nasıl Önlenir?

Uygulamalara erişmesi gereken DNS adları ve Ip'ler whitelist'e alınmalı.

Gerekirse blacklist de oluşturulmalı.

Kullanıcı girdileri mutlaka filtrelenmeli.

Kullanılmayan url şemaları devre dışı bırakılmalı.

İç ağdaki servislerde kimlik doğrulaması yapılmalı.

Sunucuların yolladığı yanıtlar kontrol edilmeli.