Broken Access Control

Kullanıcıların erişmemesi gereken kaynaklara eriştiği durumlarda veya yapamaması gereken eylemleri yapabildiklerinde ortaya çıkan zafiyete denir.

Yaygın Saldırı Yöntemleri

1 – Parameter Tampering

URL parametreleri veya form inputları manipüle edilerek, karşıya başka kullanıcıların id'lerini göndermeyi deneyebiliriz. Zafiyetli sistemler bu inputları kabul edecek ve istenmeyen sonuçlar doğuracak.

2 – Horizontal Privilege Escalation

Session yönetimindeki zafiyetler veya yetkilendirme kontrollerindeki hatalar kullanıcının, başka bir kullanıcının session id'sini kullanarak onların adına işlemler yapmasına sebep olabilir.

3 – Vertical Privilege Escalation

Sadece yetkili kişilerin erişebildiği fonksiyonlara çeşitli bypass yöntemleriyle erişmenin sonucu olarak doğar. Böylelikle yetkisiz bir kullanıcıdan yetkili bir kullanıcıya terfi eder.

4 – Data Exposure Through Public URLs

Mantıksal hatalar veya konfigürasyon hataları sebebiyle public URL'lerde hassas verilerin gözüküyor olması. Gerekli müdahaleler yapılmadığı takdirde ciddi veri sızıntılarına sebep olacaktır.

5 – API Abuse

WEB API'ları doğru şekilde korunmazsa saldırganlar için ana hedef olabilir. API uç noktaları manipüle edilerek yetkisiz erişim elde edilebilir.

Nasıl Önlenir?

Kullanıcı tarafından girilen bütün değerlerin kontrol edilmesi ve temizlenmesi.

PoLP prensibi uygulanarak yetkiye ihtiyacı olmayan kullanıcılara gereksiz yetkilerin verilmemesi.

SQL Injection ve XSS saldırılarının önüne geçmek için önlemler alınması.

Session yönetiminin güvenli bir şekilde yapılması. HTTP-Only ve secure flaglerin kullanılması, token bazlı kimlik doğrulama kullanılması.

CI/CD süreçlerinin SAST, DAST yazılımlarıyla güçlendirilip yazılım güvenliğinin arttırılması.

Cryptographic Failures

Zayıf şifreleme, bir uygulamanın ve verilerinin güvenliğini doğrudan etkiler. Güvenlik eksikliği saldırganların dolandırıcılık ve kimlik hırsızlığı yapmak için verileri çalmasına ve değiştirmesine izin

verebilir. Cryptographic Failure; man-in-the-middle saldırılarına, veritabanının ele geçirilmesine ve öngörülemeyen sonuçların doğmasına sebep olabilir.

Neden Kaynaklanır?

Aktarımdaki verinin clear text olarak yollanması.

Verilerin clear text olarak saklanması.

Yetersiz/Eski sürüm şifreleme yöntemlerinin kullanılması.

Sistemlerde varsayılan anahtarlama şifrelerinin kullanılması.

Anahtar yönetimininin düzgün yapılmaması.

Uygulamaların güvenli bağlantılar üzerinden, onaylı sertifikalar ile iletişim kurmaması.

Nasıl Önlenir?

Şifreleme Anahtarları

Şifreleme anahtarlarının byte dizileri halinde saklanması. Text şifrelerin sadece kullanılacağı zamanlarda decrypt edilmesi ve encryptli bir şekilde tutulması. Güçlü bir şifreleme tekniği ve algoritması kullanılması.

Güvenli Kodlama

Geliştiricilerin yazılım geliştirme sürecinde uygulamanın güvenliğini sağlayacak şekilde ilerlemesi. Sağlam şifreleme yöntemlerinin uygulamanın çeşitli yerlerinde kullanılıyor olması şifreleme hatalarının azaltılmasına yardımcı olur.

Sızma Testleri

Şifreleme stratejilerinin hatasız bir şekilde ilerlediğinden emin olmak için devamlı sızma testleri yapılması. Algoritmadaki zayıflıkların tespit edilip bunların geliştirme süreçlerinde ilerleme kaydedilmesi.

Injection

Kullanıcı girdilerinin kontrol edilmediği, temizlenmediği ve filtrelenmediği durumlarda yaygın olarak görülen zafiyet türüdür. Saldırganın programa kod enjekte ettiği, bilgisayara zararlı yazılım enjekte ettiği veya sunucuya zararlı sorgu enjekte ettiği durumlarda karşımıza çıkar.

Injection Türleri

Blind SQL Injection

Bir saldırganın veritabanı sunucusunun döndürdüğü bir hata sayfasını kullanarak SQL ifadeleri yardımıyla True ve False sorusu sormasına ve böylece veritabanının tam kontrolünü ele geçirmesine veya sistemde komutlar yürütmesine olanak tanır.

Blind XPath Injection

Bir XML belgesinin yapısını bilmeyen bir saldırganın belgenin yapısını belirlemeye çalışan yöntemler kullanmasına olanak tanır.

Buffer Overflow

Belleğin bazı bölümlerini üzerine yazarak bir uygulamanın akışını değiştirir.

LDAP Injection

Bu tür saldırılarda, saldırgan keyfi komutları yürütmek veya LDAP ağacının içeriğini değiştirmek için yerel bir proxy kullanarak LDAP ifadelerini değiştirebilir.

OS Commanding

HTTP istekleri aracılığıyla, sistemde komut çalıştırabilmemize yarayan zafiyetler. Bu şekilde saldırgan sistem üzerine zararlı yazılım yükleyerek kontrol sahasını genişletebilir.

SQL Injection

SQL sorgularında bulunabilecek zafiyetlerden yararlanarak veritabanından hassas verileri çıkarmaya, verileri güncellemeye ve verileri hatta yok etmeye kadar varabilen eylemler yapılabilmektedir.

Nasıl Önlenir?

Güvenli bir API kullanarak araya girmelerin minimuma indirilmesi.

Sunucu taraflı girdi doğrulamalarının yapılması ve özel karakterlere izin verilmemesi.

Parametrik sorgular kullanarak kullanıcı girdisinin güvenli bir şekilde SQL içinde kullanılması.

WAF kullanarak SQL saldırılarını minimuma indirme.

Kullanıcı girdilerini kontrol etmemize yardım eden kütüphanelerin kullanılması.

Insecure Design

Yazılım geliştirme aşamalarında veya mimari aşamalarda verilen yanlış kararların bir sonucu Insecure Design. Tasarım aşamasında güvenliğin sona bırakılması sonucu oluşan zafiyet yığılımı.

Neden Kaynaklanır?

SDLC süreçlerinde döngünün güvenliğine önem verilmemesi.

Standartlara uyulmadan yazılım geliştirme süreçlerinin gerçekleştirilmesi.

Kullanıcı yetkilendirme stratejilerinin eksik veya yanlış yapılandırılması.

Görev ayrılığı ve en az ayrıcalık gibi güvenlik ilkelerinin önemsenmemesi.

Nasıl Önlenir?

SSDLC sürecine önem verilerek tasarım aşamasından rapor aşamasına kadar bütün süreçlerin denetlenmesi.

Günümüz standartları karşılayacak ve hatta daha kalitelisini sağlayacak şekilde güvenlik standartlarına uyulması.

Yazılım geliştirme sürecinde DevSecOps üçlüsünün tercih edilmesi.

Security Misconfiguration

Herhangi bir uygulama veya sunucu üzerindeki güvenlik konfigürasyonlarının doğru şekilde yapılmaması, varsayılan ayarların/şifrelerin kullanılması veya standartlara uygun olarak gerçekleştirilmiyor olması durumu.

Neden Kaynaklanır?

Uygulama yığınının herhangi bir bölümünde uygun güvenlik sertleştirmesinin olmaması veya bulut hizmetlerinde uygunsuz şekilde yapılandırılmış izinler.

Gereksiz özellikler etkinleştirilmiş veya yüklenmiş olması.

Varsayılan hesaplar ve parolalarının hala aktif olması.

Debug mesajlarını, hata mesajlarını ve hatta kritik log mesajlarının kullanıcıya gösterilmesi.

Sistemlerde logların varsayılan haliyle tutuluyor olması ve detaylı log analizinin yapılamıyor olması.

Uygulama sunucularında, uygulamarda çalışan servislerde geliştirme araçlarının deaktif edilmemesi.

Güncel yazılımların kullanılmaması veya güvenlik açığı bulunan yazılımların kullanılması.

Nasıl Önlenir?

Varsayılan şifre ve ayarların uygulama/sunucu kurulumda değiştirilmesi. Bunların ürün ortamına taşınmaması.

Sistemde bulunan gereksiz bütün servislerin kapatılması ve diğer servislerin konfigürasyonlarının doğru yapılması.

Güncel yazılımlar kullanılması ve kullanılan 3. parti yazılımların ayrıca SCA testlerinden geçirilmesi.

Vulnerable and Outdated Components

Web uygulamalarının bir sürü kütüphane, framework ve bileşenlerden oluşuyor olması ömrü dolmuş bileşenlerin kontrolünü zorlaştırır. Zafiyetli veya ömrü dolmuş bileşenlerin kullanılması bunu tetikler.

Neden Kaynaklanır?

Zafiyetli bileşen kullanımı: Uygulama üzerinde saldırganların kullanabileceği zafiyetler barındırır. Bu bileşenler farkında olunmadan yıllar boyunca kullanılıyor olabilir.

Ömrü dolmuş bileşen kullanımı: Artık geliştiricisi tarafından desteklenmeyen bileşenleri kapsar. Güvenlik güncellemelerini almayacak bileşenler geliştirilen yazılımın güvenliğini doğrudan etkiler.

Nasıl Önlenir?

Bileşen envanteri tutulması: Yazılım geliştirme sürecinde kullanılacak bileşenlerin bir envanterinin tutulması ve bunların versiyonlarının takip edilmesi.

Yama güncellemelerinin takip edilmesi: Geliştirisi tarafından yayımlanan güvenlik güncellemelerinin sıklıkla takip edilmesi ve bileşenlerin güncellenmesi için aksiyon alınması.

Güncel zafiyetlerin takibi: Günümüzde bulunan zafiyetlerin takip edilerek, etkilenebilecek bileşenlerin tespit edilmesi ve önlemler alınması.

Alternatif bileşen kullanımı: Ömrü dolmuş bileşenler yerine alternatif bileşenlerin aranması ve onların kullanılması.

Identification and Authentication Failures

Kullanıcı kimliğinin doğrulanması, session yönetimi ve kullanıcının yetkilenmesi büyük önem arz etmektedir. Bu aşamada yapılan hatalar, uygulamada yetki hiyerarşisini ortadan kaldıracak ve uygulamanın gizliliğini/veri bütünlüğünü tehlikeye atacaktır.

Neden Kaynaklanır?

Bruteforce veya buna benzer otomatik gerçekleşen saldırıların önlenmemesi.

Zayıf kullanıcı adı ve şifre kombinasyonlarının kullanılıyor olması.

“Şifremi unuttum” gibi süreçlerin zayıf doğrulama yöntemlerine dayanıyor olması.

Şifrelerin zayıf şifreleme yöntemleri kullanılarak veya clear text halinde tutuluyor olması.

MFA süreçlerinin eksik veya verimsiz gerçekleştiriliyor olması.

Session bilgilerinin URL’de açığa çıkarılıyor olması.

Kullanıcı deaktif olduğunda oturumun kapatılmıyor olması.

Nasıl Önlenir?

Mümkün olduğu müddetçe MFA süreçlerinin uygulanması.

Varsayılan şifre ve hassas bilgilerin ürün ortamında kullanılmaması.

Zayıf şifrelerin kullanılmasına müsade edilmemesi.

Kayıt olma, şifre kurtarma ve API yollarının güvenli işlemler aracılığıyla gerçekleştiğinin kontrolünün yapılması.

Sunucu tarafında güvenli bir oturum yönetimi sisteminin kullanılması. Session ID’lerin tahmin edilemeyecek veya enumeration yöntemleriyle bulunamayacak şekilde oluşturulması.

Software and Data Integrity Failures

Kullanılan yazılım veya altyapıların proje ile uyumlu olmaması durumlarında ortaya çıkan zafiyet. Uygulamanın plugin, kütüphane veya modüllere dayanarak güvenilmez kaynaklardaki kodları, repoları veya CDN’leri kullanması sonucu oluşur. Güvenliksiz CI/CD süreçleri saldırganlar tarafından açık hedef haline gelir ve potansiyel yetkisiz erişime sebebiyet verir.

Neden Kaynaklanır?

CI/CD süreçlerinin güvenilir ortamlarda gerçekleştirilmesi.

Artık kullanılmayan cihazlarda hassas bilgilerin tutulması ve bunların silinmemesi.

Sistem içerisine entegre edilmiş zararlı yazılımlar.

Açık kaynak kodlu yazılımlarda doğrulanmamış yayımcıların projelerinin kullanılması.

Nasıl Önlenir?

Kullanılan yazılımların imzalarının kontrol edilmesi ve yazılımının sahibinin 3. parti kişiler olmaması.

Npm veya Maven gibi platformlarda güvenilir kütüphanelerin kullanıldığının teyit edilmesi.

İstemcilere imzalanmamış veya şifrelenmemiş verilerin gönderilmemesi ve bunların veri uyumluluk kontrollerinin yapılması.

Security Logging and Monitoring Failures

Doğrudan bir zafiyet oluşturmasa da loglama ve monitörlemenin zayıf olması saldırganların tespitinde ve müdehalesinde kritik hatalar olmasına sebep olur. Uzun zaman fark edilmeyen zararlılar ileriki süreçlerde büyük hasarlar ortaya çıkarabilir.

Neden Kaynaklanır?

Yetersiz loglama işlemleri.

Logların yedeklenmemesi.

Gereksiz logların tutularak ihtiyacımız olan verilere erişilememesi.

IDS ve IPS gibi yazılımların yetersiz kurulumu sebebiyle şüpheli aktivitelerin alert oluşturmaması veya gerçek zamanlı alertlerin gönderilmiyor olması.

Nasıl Önlenir?

Giriş ve başarısız giriş eylemlerinin loglandığından emin olun.

Sunucu hatalarına karşın tuttuğunuz logları başka bir kaynakta yedekleyin.

Oluşturulan logların gerekli bilgileri içerdiğine ve log formatlarına uyduğuna dikkat edin.

Monitörleme sistemlerinin çalıştığını test etmek için eylemler alın.

Logların yetkisiz kişiler tarafından erişilemediğinden emin olun.

Server-Side Request Forgery

SSRF, saldırganın sunucu taraflı saldırılar yapabilmesine olanak sağlar. Sunucunun sadece sunucuda çalışması gereken methodları çalıştırmasını sağlar ve bunları sömürür.

Neden Kaynaklanır?

Hedef uygulama, bir URL'den veri içe aktarma, verileri bir URL'de yayınlama veya başka bir şekilde kurcalanabilecek bir URL'den veri okuma işlevine sahip olabilir. Saldırgan, tamamen farklı bir URL sağlayarak veya URL'lerin yollarını değiştirerek bu işlevselliğe yapılan çağrıları değiştirir.

Sunucuya giden değiştirilmiş istekler, sunucunun sanki kendi içinde haberleşiyor gibi düşünmesine sebep olur ve hassas bilgilerin sızdırılmasını sağlar.

Nasıl Önlenir?

API’lere girilen URL’lerin kontrollerinin yapılması. Hatalı yazılan kodlarda saldırganlar domain taklidi yaparak sistemi bypasslayabilir.

DNS sunucusunda domainlerin whitelist süreçlerinin uygulanması.

Sunucudan istemciye ham responların gönderilmemesi. Gönderilecek response’larda sadece kullanıcının ihtiyacına yarayacak bilgilerin gönderilmesi.

URL şemalarının zorunlu tutulması. Sadece HTTPs kullanılan bir ortamda diğer URL şemalarının kullanılmaması.

Kullanıcıya asla güvenilmemesi. Sunucuya gelen girdilerin kontrol edilmesi ve bunların temizlenmesi.

Bütün servislerde gerekli kimlik doğrulama aşamalarının yapılması.