Broken Access Control

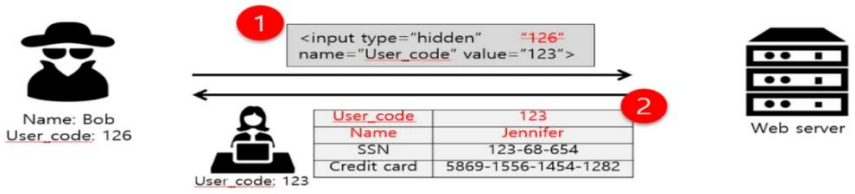
-Broken Access Control, web uygulamalarında kullanıcıların yetkisiz bir şekilde kaynaklara erişmelerine olanak tanıyan bir güvenlik açığıdır. Bu, kullanıcıların normalde erişim izni olmayan kaynaklara erişebildiği durumları ifade eder.

Neden Kaynaklanır?

Bu zafiyet, yanlış yapılandırılmış erişim kontrolü ve yetki doğrulama eksikliklerinden kaynaklanır. Uygulama geliştiricilerin erişim kontrol mekanizmalarını doğru bir şekilde uygulamaması, bu tür zafiyetlerin oluşmasına yol açar.

Broken Access Control Türleri

1. Dikey Erişim Kontrolü (Vertical Access Control): Kullanıcıların belirli bir rol veya pozisyonlarına bağlı olarak belirli kaynaklara erişimini kontrol eder. Örneğin, bir yönetici belirli işlevlere erişebilirken, normal bir kullanıcı bu işlevlere erişemez.
2. Yatay Erişim Kontrolü (Horizontal Access Control): Kullanıcıların kendi rolleri dışındaki kullanıcıların kaynaklarına erişimini kontrol eder. Bu, bir kullanıcının başka bir kullanıcının verilerine izinsiz erişmesini engeller.
3. Bağlamına Bağlı Erişim Kontrolü (Context-Dependent Access Control): Kullanıcının erişim izinlerini çeşitli bağlamlara (zaman, konum, durum) bağlı olarak değiştirebilir. Örneğin, belirli bir saat aralığında veya belirli bir konumda erişime izin vermek.



Broken Access Control açığından korunmak için alınması gereken önlemler:

- Güçlü kimlik doğrulama ve oturum yönetimi uygulamak.
- Yapılan işlemlerin logları tutulmalıdır.
- Erişim kontrollerini düzenli testlerle ve güncellemelerle iyileştirmek.
- Güvenlik güncellemelerini takip etmek ve yüklemek.

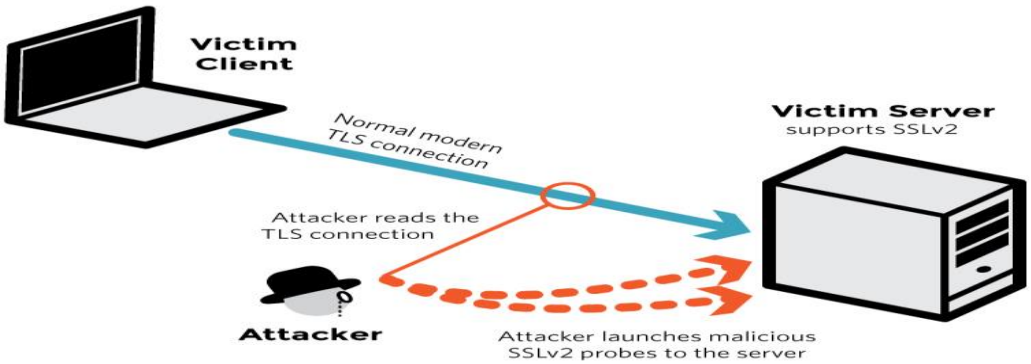
Cryptographic Failures

-Cryptographic Failures, web uygulamalarında verilerin gizliliği, bütünlüğü veya doğruluğunu sağlamak için kullanılan kriptografik yöntemlerin yetersiz ya da hatalı uygulanmasından kaynaklanan güvenlik zafiyetlerini ifade eder. Bu zafiyetler, hassas verilerin korunamamasına yol açabilir.

Neden Kaynaklanır?

Bu zafiyetin başlıca nedenleri arasında:

- Zayıf veya eski kriptografik algoritmaların kullanılması.
- SSL/TLS protokollerinin yanlış yapılandırılması.
- Verilerin şifrelenmeden iletilmesi ya da saklanması örnek olarak verilebilir.



Cryptographic Failures açığından korunmak için alınması gereken önlemler:

- Güçlü ve güncel kriptografik algoritmalar kullanmak.
- Doğru anahtar yönetimi politikalarını uygulamak ve anahtarları güvenli yerlerde saklamak.
- SSL/TLS yapılandırmasını doğru yapmak ve geçerli, güvenilir sertifikalar kullanmak.
- Hassas verileri her zaman şifreli bir şekilde iletmek ve saklamak.

Injection

Injection, kullanıcıdan gelen verilerin bir komut veya sorguya dahil edilerek, kötü niyetli kodların çalıştırılmasına yol açan güvenlik açığıdır. Bu tür zafiyetler, bir saldırganın uygulamanın veri tabanına, dosya sistemine ya da diğer geri uç sistemlerine yetkisiz erişim sağlamasına olanak tanır.

Neden Kaynaklanır?

Injection zafiyeti genellikle aşağıdaki durumlardan kaynaklanır:

- Kullanıcıdan alınan girdilerin yeterince doğrulanmaması ve temizlenmemesi.
- Dinamik sorguların veya komutların doğrudan kullanıcı girdileriyle oluşturulması.
- Parametreli sorgular ya da hazırlanmış ifadelerin kullanılmaması.
- Eski ya da güvenlik açıklarına sahip kütüphanelerin veya çerçevelerin kullanılması.

Injection Türleri :

1. SQL Injection: Kullanıcı tarafından sağlanan veriler, SQL sorgularına dahil edilerek veri tabanına yetkisiz erişim sağlanmasına neden olur. Örneğin, bir kullanıcının form alanına zararlı SQL komutları eklemesi.
2. Command Injection: Kullanıcıdan alınan veriler, sistem komutlarına dahil edilerek sunucuda yetkisiz komutlar çalıştırılmasına neden olur. Bu durum, sunucu dosya sistemine ya da diğer kritik sistem bileşenlerine erişim sağlar.
3. Cross-Site Scripting (XSS): Kullanıcıdan alınan veriler, web sayfasına enjekte edilerek, diğer kullanıcılara kötü amaçlı kodlar gönderilebilir. Bu tür saldırılar, genellikle oturum çalma ya da zararlı içerik sunma amacı taşır.
4. LDAP Injection: Kullanıcıdan gelen girdiler, LDAP sorgularına dahil edilerek dizin hizmetlerine yetkisiz erişim sağlanabilir.

Injection açığından korunmak için alınması gereken önlemler:

- Parametreli sorgular veya hazırlanmış ifadeler kullanarak kullanıcı girdilerini sorgularda doğrudan kullanmamak.
- Girdi doğrulama ve temizleme mekanizmalarını uygulamak.
- WAF kullanmak.
- Kullanıcı girdilerinde sadece güvenli veri türlerini kabul etmek.

Insecure Design

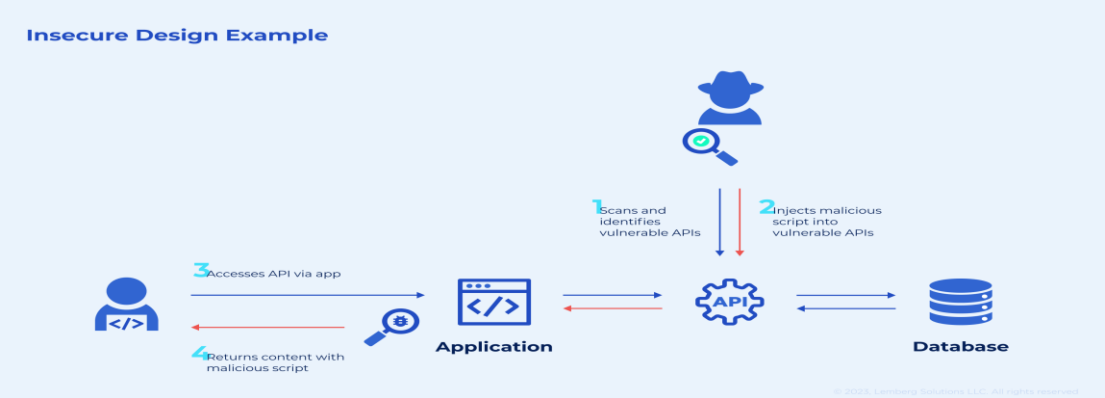
Insecure Design, bir uygulamanın veya sistemin tasarım aşamasında yeterli güvenlik önlemlerinin alınmaması nedeniyle oluşan güvenlik açıklarını ifade eder. Bu zafiyet, güvenlik

gereksinimlerinin dikkate alınmadığı ya da yetersiz şekilde ele alındığı durumlarda ortaya çıkar.

Neden Kaynaklanır?

Insecure Design zafiyeti genellikle aşağıdaki nedenlerden kaynaklanır:

- Güvenlik gereksinimlerinin tasarım aşamasında yeterince belirlenmemesi.
- Tehdit modelleme ve risk değerlendirme süreçlerinin ihmal edilmesi.
- Güvenlik testlerinin tasarım sürecine entegre edilmemesi.
- Yetersiz güvenlik kontrollerinin uygulanması ya da eksik bırakılması.
- Karmaşık ve zayıf mimariler oluşturulması.



Insecure Design açığından korunmak için alınması gereken önlemler:

- Güvenlik gereksinimlerini tasarım sürecinin erken aşamalarında belirlemek ve bunları düzenli olarak gözden geçirmek.
- Tehdit modelleme ve risk değelerlendirme yapmak
- Düzenli güvenlik testleri yapmak ve test sonuçlarına göre tasarımı iyileştirmek.
- Güvenli yazılım geliştirme yaşam döngüsü (SDLC) süreçlerini takip etmek.

Security Misconfiguration

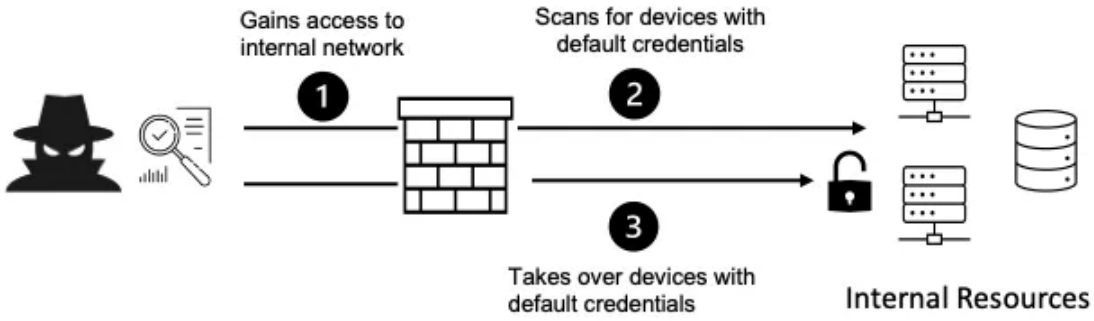
Security Misconfiguration, bir sistem, uygulama ya da ağın güvenlik ayarlarının hatalı ya da yetersiz yapılandırılmasından kaynaklanan bir güvenlik açığıdır. Bu zafiyet, saldırganların

sistemlere yetkisiz erişim sağlamasına, hassas verilere ulaşmasına ya da kötü amaçlı faaliyetler yürütmesine olanak tanır.

Neden Kaynaklanır?

Security Misconfiguration zafiyeti genellikle şu nedenlerden kaynaklanır:

- Varsayılan ayarların kullanılması.
- Gereksiz özelliklerin veya hizmetlerin etkinleştirilmesi.
- Güvenlik yamalarının veya güncellemelerin uygulanmaması.
- Yanlış izinler ve eksik yapılandırmalar.
- Hatalı hata ayıklama ve hata mesajları.



Security Misconfiguration açığından korunmak için alınması gereken önlemler:

- Varsayılan ayarları ve şifreleri değiştirerek uygulamanın güvenlik seviyesini artırmak.
- Gereksiz hizmetleri ve özellikleri devre dışı bırakmak ve sadece gerekli olanları etkin tutmak.
- Güvenlik yamalarını ve güncellemelerini düzenli olarak uygulamak.
- Doğru izin yapılandırmaları kullanarak dosya, klasör ve sistem kaynaklarına yetkisiz erişimi engellemek.
- Hata mesajlarını kullanıcıya gizlemek ve ayrıntılı hata bilgilerini yalnızca güvenlik kayıtlarına yansıtmak.

Vulnerable and Outdated Components

Vulnerable and Outdated Components, uygulama veya sistemde kullanılan yazılım bileşenlerinin (kütüphaneler, çerçeveler, modüller vb.) güvenlik açıklarına sahip olması veya bu bileşenlerin eski sürümlerinin kullanılması durumunda ortaya çıkan bir güvenlik açığıdır.

Bu zafiyet, saldırganların bilinen açıkları istismar etmesine ve sistemi tehlikeye atmasına olanak tanır.

Neden Kaynaklanır?

Vulnerable and Outdated Components zafiyeti genellikle aşağıdaki durumlardan kaynaklanır:

- Güncellemelerin ve güvenlik yamalarının zamanında uygulanmaması.
- Güvenlik açıklarına sahip bileşenlerin kullanılması.
- Üçüncü taraf kütüphanelerin veya modüllerin güvenliğinin yeterince denetlenmemesi.
- Eski veya desteklenmeyen yazılımların kullanılması.
- Bağımlılıkların yönetilmemesi veya gözden kaçması.

Vulnerable and Outdated Components açığından korunmak için alınması gereken önlemler:

- Güncellemeleri ve güvenlik yamalarını düzenli olarak uygulamak ve kullanılan tüm bileşenlerin en son sürümlerini kullanmak.
- Üçüncü taraf kütüphaneler ve modüllerin güvenlik durumunu düzenli olarak gözden geçirmek.
- Yazılım bileşenlerinin güvenlik açıklarını takip etmek ve riskli bileşenleri güncellemek veya değiştirmek.
- Eski ve desteklenmeyen yazılımları mümkün olan en kısa sürede sistemden kaldırmak veya güncel sürümleriyle değiştirmek.
- Bağımlılık yönetim araçları kullanarak, kullanılan bileşenlerin güvenlik durumunu otomatik olarak izlemek ve uyarıları dikkate almak.

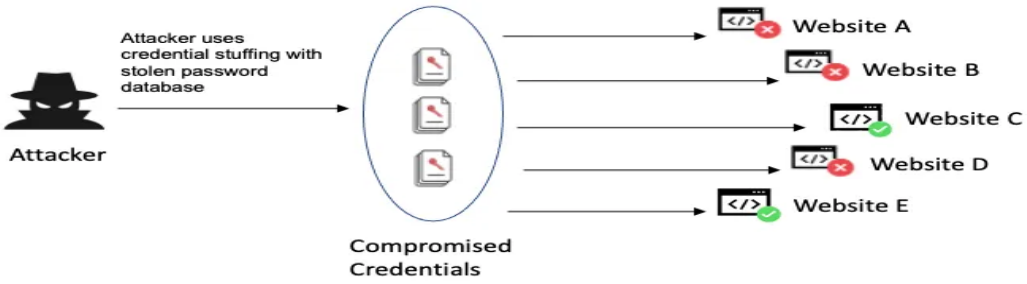
Identification and Authentication Failures

Identification and Authentication Failures, bir kullanıcının kimliğinin doğru bir şekilde tanımlanamaması veya doğrulanamaması durumunda ortaya çıkan güvenlik açıklarını ifade eder. Bu zafiyet, kimlik doğrulama süreçlerinin zayıf olması nedeniyle yetkisiz kişilerin sisteme erişim sağlamasına olanak tanır.

Neden Kaynaklanır?

Identification and Authentication Failures zafiyeti genellikle aşağıdaki nedenlerden kaynaklanır:

- Zayıf parola politikaları ve parola yönetimi.
- Çok faktörlü kimlik doğrulamanın (MFA) eksikliği.
- Oturum yönetimi hataları (örneğin, oturum süresinin dolmaması).
- Hatalı kimlik doğrulama işlemleri ve kimlik doğrulama eksiklikleri.
- Varsayılan kimlik doğrulama bilgileri kullanılması.



Identification and Authentication Failures korunmak için alınması gereken önlemler:

- Güçlü kimlik doğrulama yöntemleri kullanmak
- Kimlik doğrulama işlemlerini izlemek
- Kimlik bilgilerini şifrelemek
- Düzenli olarak kimlik doğrulama politikalarını kontrol etmek

Software and Data Integrity Failures

Software and Data Integrity Failures, yazılım güncellemelerinin, kritik verilerin ve veri işleme süreçlerinin bütünlüğünün bozulduğu durumlarda ortaya çıkan güvenlik açıklarını ifade eder. Bu zafiyet, saldırganların yetkisiz değişiklikler yapmasına veya kötü amaçlı yazılımlar enjekte etmesine olanak tanır, bu da sistemin güvenilirliğini tehlikeye atar.

Neden Kaynaklanır?

Software and Data Integrity Failures zafiyeti genellikle şu nedenlerden kaynaklanır:

- Güvenilmeyen kaynaklardan gelen güncellemelerin veya yazılımların yüklenmesi.
- Kod bütünlüğü kontrollerinin olmaması veya yetersiz olması.
- Kötü amaçlı yazılımların veya zararlı kodların sisteme sızması.
- Veri işleme süreçlerinin yeterince korunmaması.
- Dijital imzaların eksikliği veya hatalı uygulanması.

Türleri

1. Güvensiz Yazılım Güncellemeleri: Kaynağı doğrulanmamış yazılım güncellemelerinin yüklenmesi, sistemin kötü amaçlı yazılımlar tarafından ele geçirilmesine neden olabilir.
2. Kod Bütünlüğü Kontrollerinin Eksikliği: Yazılım bileşenlerinin kod bütünlüğünün doğrulanmaması, saldırganların kötü amaçlı kod eklemesine olanak tanır.
3. Veri Manipülasyonu: Veri işleme süreçlerinde yeterli güvenlik önlemlerinin alınmaması, hassas verilerin saldırganlar tarafından değiştirilmesine veya bozulmasına yol açabilir.
4. Eksik Dijital İmzalar: Yazılım bileşenlerinin veya verilerin dijital imzalarının eksik veya hatalı olması, sistemin güvenlik açıklarına karşı savunmasız hale gelmesine neden olabilir.
5. Tedarik Zinciri Saldırıları: Yazılım tedarik zincirine yapılan saldırılar, güncellemelerin veya yeni bileşenlerin zararlı yazılımlar içererek sisteme girmesine yol açabilir.

Software and Data Integrity Failures açığından korunmak için alınması gereken önlemler:

- Güvenilir kaynaklardan gelen güncellemeleri ve yazılımları kullanmak ve bunların dijital imzalarını doğrulamak.
- Kod bütünlüğü kontrolleri uygulayarak, yazılım bileşenlerinin yetkisiz değişikliklere karşı korunmasını sağlamak.
- Veri işleme süreçlerinde güçlü güvenlik önlemleri uygulamak ve verilerin bütünlüğünü sürekli olarak izlemek.
- Dijital imzaları kullanarak yazılım ve veri bileşenlerinin bütünlüğünü korumak.
- Tedarik zinciri güvenliği stratejileri geliştirerek, yazılım tedarik zincirindeki olası zafiyetleri minimize etmek.

Security Logging and Monitoring Failures

Security Logging and Monitoring Failures, güvenlik olaylarının kaydedilmemesi, yetersiz kaydedilmesi veya izlenmemesi durumunda ortaya çıkan güvenlik açıklarını ifade eder. Bu zafiyet, saldırıların tespit edilememesine veya saldırılara zamanında müdahale edilememesine neden olur, bu da sistemin güvenlik risklerini artırır.

Neden Kaynaklanır?

Security Logging and Monitoring Failures zafiyeti genellikle şu nedenlerden kaynaklanır:

- Güvenlik olaylarının yeterince kaydedilmemesi veya hiç kaydedilmemesi.
- Kaydedilen olayların düzenli olarak izlenmemesi ve analiz edilmemesi.
- Güvenlik uyarı mekanizmalarının eksikliği veya yanlış yapılandırılması.
- Kaydedilen verilerin yeterli süre saklanmaması veya güvenli bir şekilde depolanmaması.
- Otomatik tehdit tespit sistemlerinin kullanılmaması veya yetersiz olması.

Security Logging and Monitoring açığından korunmak için alınması gereken önlemler

- Kapsamlı bir güvenlik kaydı politikası oluşturarak, kritik olayların düzenli olarak kaydedilmesini sağlamak.
- Güvenlik kayıtlarının güvenli bir şekilde saklanması ve belirli bir süre boyunca erişilebilir tutulması.
- Olay izleme ve analiz süreçlerini otomatikleştirmek, böylece potansiyel tehditlerin erken tespit edilmesini sağlamak.
- Güvenlik uyarı sistemlerini doğru bir şekilde yapılandırarak, önemli güvenlik olayları için zamanında uyarılar almak.

Server-Side Request Forgery (SSRF)

Server-Side Request Forgery (SSRF), saldırganın bir sunucuyu kötüye kullanarak sunucunun dışarıdaki veya iç ağlardaki başka bir kaynağa istek göndermesini sağladığı bir güvenlik zafiyetidir. Bu zafiyet, saldırganların hedef sunucu üzerinden gizli bilgileri ele geçirmesine, dahili sistemlere erişim sağlamasına veya diğer sistemlerdeki güvenlik açıklarını kötüye kullanmasına olanak tanır.

Neden Kaynaklanır?

SSRF zafiyeti genellikle şu nedenlerden kaynaklanır:

- Kullanıcı girdisinin doğru şekilde doğrulanmaması ve filtrelenmemesi.
- Sunucunun güvenlik duvarı kurallarının yetersiz olması.
- Sunucunun dışa ve içe dönük istekleri kontrol etmeden işleme alması.
- URL'lerin kullanıcı tarafından doğrudan kontrol edilebilmesi.
- Güvenlik politikalarının eksikliği veya yanlış uygulanması.

Server-Side Request Forgery açığından korunmak için alınması gereken önlemler:

- Giriş doğrulaması.
- Güvenlik duvarı kurulumu.
- Güvenli URL işleme.
- Sunucu ayarlarının kontrol edilmesi.

Referanslar

1. https://cybershieldcommunity.com/owasp-top-10-zafiyetleri-ve-alinmasi-gereken-onlemler/

2- https://owasp.org/www-project-top-ten/

3- https://medium.com/@aysekaya/owasp-top-10-zafiyetleri-veal%C4%B1nmas%C4%B1-gereken-%C3%B6nlemler-a1a38280148e

4- https://www.youtube.com/watch?v=hryt-rCLJUA