Mentör: Yusuf Nas

Öğrenci: Naci Balcı

OWASP Top Ten

1-Broken Access Control (A01:2021):

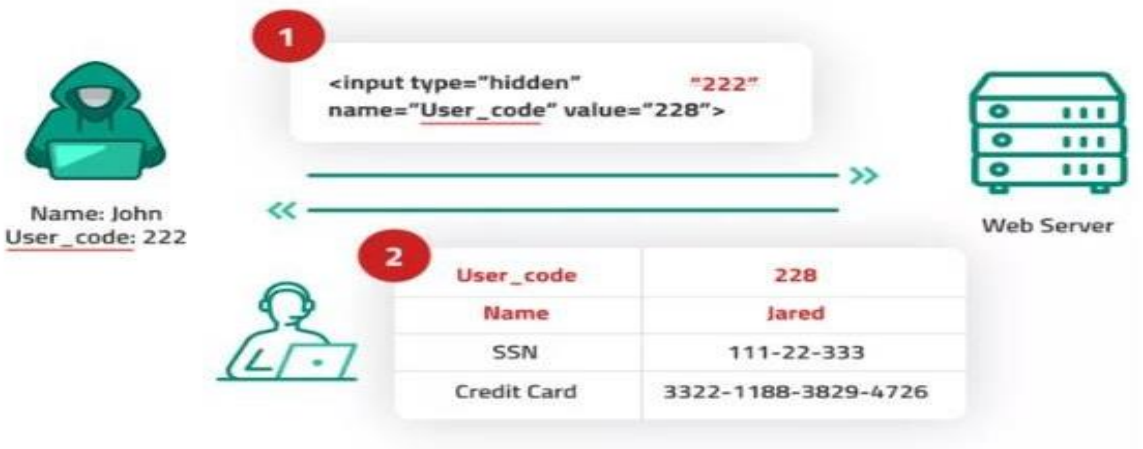
Yetkilendirme mekanizmalarının düzgün bir şekilde uygulanmamasından veya yanlış yapılandırılmasından kaynaklanır. Bu zafiyet, yetkisiz kullanıcıların veya düşük seviyeli yetkilere sahip kullanıcıların, belirli kaynaklara veya işlemlere erişim sağlayabilmesine yol açabilir.

Neden Kaynaklanır

- ***Unutulmuş Erişim Kontrolleri: Bazı geliştiriciler, hassas işlemler veya sayfalar için gerekli erişim kontrol mekanizmalarını uygulamayı ihmal edebilir. Örneğin, sadece yöneticilere açık olması gereken bir sayfa, kontrolsüz bir şekilde tüm kullanıcılara açık bırakılabilir.***
- ***Eksik Kimlik Doğrulama: Bazı işlevlerin, kullanıcıların kimliklerini doğrulamasını gerektirmemesi veya kimlik doğrulamanın eksik yapılması, yetkisiz kişilerin bu işlevlere erişmesine neden olabilir.***
- ***2. Yanlış Yapılandırılmış Erişim Kontrolleri***
- ***Rol Tabanlı Erişim Kontrolünün Yanlış Yapılandırılması: Kullanıcı rollerine yanlış izinlerin atanması, kullanıcıların kendi yetkileri dışında işlemler gerçekleştirebilmesine neden olabilir. Örneğin, bir "öğrenci" rolüne sahip bir kullanıcıya, "yönetici" rolüne ait yetkilerin yanlışlıkla verilmesi bu tür bir zafiyete sebep olabilir.***

Nasıl Önlenir?

- ***Minimize Edilmiş Erişim: Kullanıcıların yalnızca ihtiyaç duydukları en az erişimi almasını sağlayın. Örneğin, bir kullanıcı yalnızca kendi verilerine veya görevlerine erişebilmelidir, başkalarınınkine değil.***
- ***Rol Tabanlı Erişim Kontrolü (RBAC): Kullanıcıları, rollerine göre kategorize edin ve bu rollere özel erişim kuralları belirleyin. Her rol, sadece gerekli yetkilere sahip olmalıdır.***
- ***Yetki Zinciri: Üst düzey bir kullanıcı bile, yalnızca yetkilendirildiği görevleri yapabilmelidir. Bir rolün yetkileri, başka bir rolde bulunan yetkilerden devralınmamalıdır.***



2-Cryptographic Failures (A02:2021):

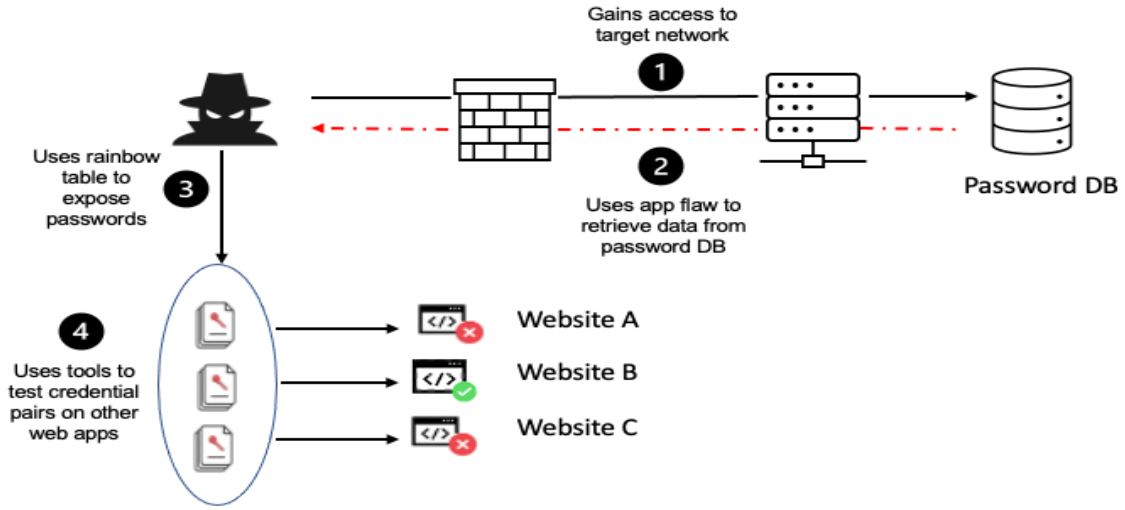
Veri güvenliğini sağlamak amacıyla kullanılan kriptografik yöntemlerin yanlış uygulanmasından veya uygun olmayan kriptografik mekanizmaların kullanılmasından kaynaklanan zafiyetleri tanımlar. Bu zafiyetler, hassas bilgilerin yetkisiz kişiler tarafından ele geçirilmesine, değiştirilmesine veya başka şekillerde kötüye kullanılmasına yol açabilir.

Neden Kaynaklanır

- ***Eski veya Zayıf Şifreleme Algoritmaları: Artık güvenli kabul edilmeyen, kırılmaya açık eski şifreleme algoritmalarının (örneğin, DES, RC4) kullanılması, verilerin ele geçirilmesi riskini artırır. Modern algoritmalar (örneğin, AES) bu tür saldırılara karşı daha dayanıklıdır ve kullanılmalıdır.***
- ***Yanlış Kriptografik Primitifler: Şifreleme, imza veya anahtar üretimi gibi işlemler için yanlış kriptografik fonksiyonların veya araçların kullanılması, güvenliği tehlikeye atabilir. Örneğin, MD5 veya SHA-1 gibi hash fonksiyonları artık güvenli kabul edilmemektedir.***
- ***Zayıf Anahtar Üretimi: Kriptografik anahtarların rastgeleliğinin ve yeterince güçlü olmasının sağlanmaması, anahtarların tahmin edilmesini veya kırılmasını kolaylaştırabilir.***

Nasıl Önlenir?

- ***Modern Algoritmalar: AES (Advanced Encryption Standard) gibi güçlü ve modern şifreleme algoritmalarını tercih edin. DES, RC4 gibi eski ve zayıf algoritmalardan kaçının.***
- ***Düzenli Güncelleme: Kriptografik kütüphanelerin ve araçların düzenli olarak güncellenmesi, bilinen zafiyetlere karşı korunma sağlar.***
- ***Anahtarları Güvenli Bir Şekilde Saklayın: Anahtarlar, güvenli bir şekilde saklanmalı ve erişim kontrolü altında olmalıdır. Anahtarları asla kaynak kodda veya açıkta bırakmayın.***
- ***Anahtar Rotasyonu: Anahtarlar düzenli aralıklarla yenilenmeli ve eski anahtarlar güvenli bir şekilde imha edilmelidir.***



3-Injection (A03:2021):

Saldırganların bir uygulamanın veri girdisini kötüye kullanarak, arka plandaki komutları veya sorguları manipüle etmesine olanak tanıyan kritik bir güvenlik açığıdır. Bu tür saldırılar, bir uygulamanın veri girişlerini doğru şekilde işlemediği durumlarda ortaya çıkar ve birçok farklı şekilde gerçekleştirilebilir.

Türleri:

1.SQL Injection (SQLi)
2.Cross-Site Scripting (XSS)
3.Command Injection
4.Css Injection
5.LDAP Injection
6.XML External Entity Injection (XXE)

7.NoSQL Injection
8.XPath Injection

Neden Kaynaklanır

- ***Kullanıcıdan alınan verilerin doğrudan SQL sorgularında, komut satırlarında veya başka bir yerde kullanılmadan önce doğrulanmaması.***
- ***Veri temizleme (sanitization) veya parametre bağlama (parameterized queries) işlemlerinin yapılmaması.***
- ***Kullanıcı girdisinin, uygulamanın beklediği formatta olup olmadığının kontrol edilmemesi.***
- ***SQL sorgularını veya komutları oluştururken, string birleştirme (string concatenation) kullanılması.***
- ***Kullanıcının girdiği verilerin, komut veya sorgu olarak yorumlanmasına izin veren zayıf kodlama alışkanlıkları.***

Nasıl Önlenir?

- ***SQL Injection'ı önlemek için en etkili yöntemlerden biri, parametrik sorgular (prepared statements) kullanmaktır. Bu yöntem, kullanıcı girdilerini veri olarak kabul eder ve bu verilerin SQL komutu olarak çalıştırılmasını engeller.***
- ***Tüm kullanıcı girdileri, beklenen formatta olup olmadığı açısından doğrulanmalı ve zararlı olabilecek karakterler temizlenmelidir.***
- ***Düzenli olarak güvenlik testleri (penetration testing) ve kod incelemeleri yaparak, injection zafiyetlerini tespit edin ve düzeltin.***
- ***OWASP ZAP gibi araçlar, injection zafiyetlerini tespit etmek için kullanılabilir.***

4-Insecure Design (A04:2021):

Güvenlik açısından zayıf mimari ve tasarım kararlarının verilmesi.

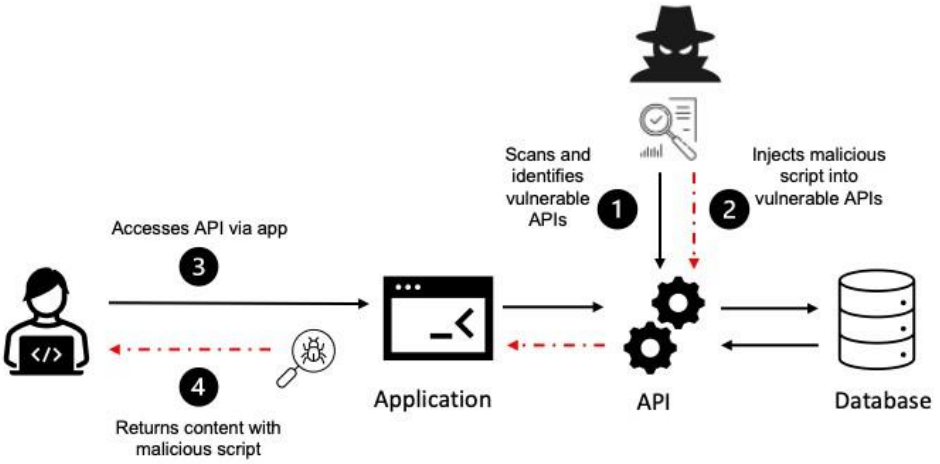
Neden Kaynaklanır

- ***Tasarım aşamasında güvenliğin öncelikli bir konu olarak ele alınmaması. Geliştiriciler ve tasarımcılar, güvenliği sonradan eklenebilecek bir özellik olarak görebilirler, bu da temel güvenlik zafiyetlerine yol açar.***
- ***Geliştirici ve tasarım ekibinin güvenlik konusunda yeterli bilgiye sahip olmaması.***
- ***Projenin başlangıç aşamasında, güvenlik gereksinimlerinin net bir şekilde tanımlanmaması. Güvenlik gereksinimleri, uygulamanın korunması gereken varlıklarını ve bu varlıkları korumak için alınması gereken önlemleri tanımlar.***

Nasıl Önlenir?

- ***Varsayılan Olarak Güvenli: Sistem, varsayılan olarak güvenli olacak şekilde tasarlanmalı; bu, yalnızca açıkça izin verilmiş olan işlemlere izin verilmesi gerektiği anlamına gelir.***
- ***Fail-Secure: Bir hata meydana geldiğinde, sistemin güvenli bir duruma geçmesini sağlayın.***

Örneğin, kimlik doğrulama sisteminde hata oluşursa, erişimi engelleyin.



5-Security Misconfiguration (A05:2021):

Bir sistemin veya uygulamanın güvenlik ayarlarının yanlış yapılandırılması veya varsayılan ayarlarla bırakılması sonucunda ortaya çıkan bir güvenlik zafiyetidir. Bu zafiyet, saldırganların sisteme yetkisiz erişim sağlamasına, hassas verilere ulaşmasına veya sistemi kötüye kullanmasına olanak tanıyabilir.

Neden Kaynaklanır

- ***Birçok yazılım veya donanım varsayılan (default) ayarlarla gelir. Bu ayarlar genellikle güvenlik açısından zayıf olabilir ve değiştirilmediklerinde güvenlik zafiyetlerine yol açabilir.***
- ***Güvenlik ayarlarının yanlış yapılması veya gerekli ayarların eksik bırakılması. Örneğin, güvenlik duvarının düzgün yapılandırılmaması veya gereksiz servislerin açık bırakılması.***
- ***Uygulama geliştiricilerinin veya sistem yöneticilerinin, güvenlik ayarlarını doğru şekilde yapılandırmak için gerekli bilgiye veya zamana sahip olmaması.***

Nasıl Önlenir?

- ***Sistem veya uygulama kurulduktan sonra varsayılan ayarların güvenlik açısından gözden geçirilmesi ve gerekliyse değiştirilmesi. Bu, varsayılan kullanıcı adlarının, parolaların ve diğer ayarların değiştirilmesini içerir.***
- ***Kullanılmayan veya gereksiz bileşenleri, modülleri, servisleri devre dışı bırakın veya sistemden kaldırın. Bu, saldırı yüzeyini küçültür ve potansiyel güvenlik risklerini azaltır.***

6-Vulnerable and Outdated Components (A06:2021):

Yazılım uygulamalarının, sistemlerin veya altyapıların eski, güncellenmemiş veya güvenlik açıklarına sahip bileşenleri kullanmasından kaynaklanır. Bu bileşenler, saldırganlar tarafından bilinen güvenlik açıkları üzerinden hedef alınabilir ve sistemlerin güvenliğini tehlikeye atabilir.

Neden Kaynaklanır

- ***Yazılım geliştiricilerin veya sistem yöneticilerinin, kullanılan bileşenlerin güncellemelerini veya güvenlik yamalarını takip etmemesi.***
- ***Güncellemelerin uygulanmasının zaman alıcı veya karmaşık olduğu durumlarda, bu işlemlerin ertelenmesi veya tamamen göz ardı edilmesi.***
- ***Eski bileşenlerin güncellenmesi durumunda, uygulamanın veya sistemin uyumsuz hale***

gelmesinden korkulması. Bu yüzden, eski bileşenlerin kullanımı devam ettirilir.
- *Yeni sürümlerin eski kod veya yapı ile uyumlu olmaması.*

Nasıl Önlenir?

- *Kullanılan tüm bileşenler, kütüphaneler ve sistemler için düzenli olarak güncellemeler ve yamalar kontrol edilmelidir. Kritik güvenlik güncellemeleri mümkün olan en kısa sürede uygulanmalıdır.*
- *Yazılım projelerinde kullanılan bağımlılıkların ve bileşenlerin sürümlerini yönetmek için bağımlılık izleme araçları kullanın (örneğin, npm audit, Dependabot). Bu araçlar, güncellenmesi gereken bileşenler konusunda uyarılar sağlar.*
- *Gerektiğinde, bağımlılıkları güncellemek için süreçler oluşturun.*

7-Identification and Authentication Failures (A07:2021):

Bir sistemin veya uygulamanın, kullanıcıların kimliğini doğru şekilde tanımlayamaması veya doğrulayamaması durumunda ortaya çıkan güvenlik zafiyetlerini ifade eder. Bu tür hatalar, yetkisiz kişilerin sisteme erişmesine, meşru kullanıcıların hesaplarına erişim sağlanmasına veya hassas verilere erişilmesine yol açabilir.

Neden Kaynaklanır

- *Kullanıcıların basit, tahmin edilmesi kolay veya kısa parolalar seçmesine izin verilmesi. Bu durum, brute-force (kaba kuvvet) saldırılarıyla parolaların kolayca kırılmasına yol açabilir.*
- *Parola politikalarının yetersizliği, örneğin parolaların düzenli olarak değiştirilmemesi veya karmaşık karakter gereksinimlerinin olmaması.*
- *Kullanıcıların kimlik doğrulama sırasında sadece bir faktöre (genellikle parola) dayanması. Bu, bir saldırganın yalnızca parolayı ele geçirerek hesaba erişebilmesine olanak tanır.*
- *MFA kullanımı, ek bir doğrulama katmanı ekleyerek güvenliği artırır, ancak bazı sistemlerde bu uygulanmaz.*

Nasıl Önlenir?

- *Parolaların minimum uzunlukta, karmaşık karakterler içermesi ve belirli aralıklarla değiştirilmesi gibi güçlü parola politikaları oluşturun.*
- *Kullanıcıları aynı parolayı birden fazla sistemde kullanmamaları konusunda bilgilendirin ve zorunlu kılın.*
- *Parolaların en az 8 karakter uzunluğunda, harf, sayı ve özel karakterler içermesi gerektiğini zorunlu hale getirin.*
- *Kullanıcıların hesaplarına erişmeden önce birden fazla doğrulama faktörü kullanmasını sağlayın. Örneğin, bir parola ve SMS ile gönderilen bir doğrulama kodu.*

8-Software and Data Integrity Failures (A08:2021):

Yazılım ve veri bütünlüğünü korumak için kullanılan mekanizmaların eksiklikleri veya hataları sonucu ortaya çıkan güvenlik açıklarını ifade eder. Bu tür hatalar, yazılım ve verilerin yetkisiz değişikliklere, manipülasyonlara veya bozulmalara karşı korunamamasına neden olabilir.

Neden Kaynaklanır

- *Yazılım ve veri bütünlüğü için gerekli olan güvenlik kontrollerinin (örneğin, hash doğrulama, imza doğrulama) eksikliği.*
- *Yazılım güncellemelerinin veya veri değişikliklerinin güvenli bir şekilde doğrulanmaması.*
- *Yazılım güncellemeleri veya yamalarının doğruluğunun ve bütünlüğünün kontrol edilmemesi.*

- ***Güncellemelerin veya yamaların doğruluğunu ve yetkili kaynaklardan geldiğini doğrulamadan uygulanması.***

Nasıl Önlenir?

- ***Yazılım güncellemeleri ve yamalarının, güvenilir kaynaklardan geldiğini doğrulamak için imzalama ve hash doğrulama yöntemleri kullanın.***
- ***Güncellemeleri ve yamaları, uygun şekilde test edin ve doğrulayın, ardından sistemde uygulayın.***
- ***Güncellemelerin bütünlüğünü sağlamak için PGP (Pretty Good Privacy) veya GPG (GNU Privacy Guard) gibi araçlarla imzalama yapın.***

9-Security Logging and Monitoring Failures (A09:2021):

Bir sistemin veya uygulamanın güvenlik olaylarını etkili bir şekilde kaydedememesi ve izlememesi durumunda ortaya çıkan güvenlik zafiyetleridir. Bu tür hatalar, güvenlik ihlallerinin tespit edilmesini, analiz edilmesini ve yanıtlanmasını zorlaştırır, bu da saldırganların gizlice sistemde kalmasına ve zararlı faaliyetlerde bulunmasına neden olabilir.

Neden Kaynaklanır

- ***Kritik güvenlik olaylarının veya sistem aktivitelerinin günlüğe kaydedilmemesi. Örneğin, oturum açma denemeleri, başarısız giriş girişimleri veya yetkilendirme hataları gibi olayların günlüğe kaydedilmemesi.***
- ***Günlükleme yapılandırmalarının yetersiz veya yanlış olması.***
- ***Güvenlik günlüğü kayıtlarının güvenli bir şekilde saklanmaması veya erişim kontrolü olmadan açık bir şekilde bırakılması.***
- ***Günlüklerin düzenli olarak yedeklenmemesi veya korunmaması.***

Nasıl Önlenir?

- ***Sistem ve uygulama düzeyinde kritik güvenlik olaylarını (örneğin, oturum açma denemeleri, veri erişimleri, başarısız girişler) günlüğe kaydetmek için kapsamlı bir günlükleme politikası oluşturun.***
- ***Olayların doğru ve detaylı bir şekilde günlüğe kaydedilmesini sağlayın.***
- ***Güvenlik günlüğü kayıtlarının güvenli bir şekilde saklanmasını ve yetkisiz erişimlere karşı korunmasını sağlayın.***
- ***Günlüklerin düzenli olarak yedeklenmesi ve uzun süreli saklanması için uygun prosedürler oluşturun.***

10-Server-Side Request Forgery (SSRF) (A10:2021):

Bir saldırganın, sunucunun kendi adıyla veya sunucu tarafında çalıştırılan başka bir servisle kötü niyetli istekler yapmasını sağlamasıyla ortaya çıkan bir güvenlik zafiyetidir. Bu tür bir saldırı, sunucunun iç ağdaki veya diğer hizmetlerdeki kaynaklara istek göndermesini sağlayarak, hassas verileri çalabilir, iç sistemlere erişebilir veya hizmetleri manipüle edebilir.

Neden Kaynaklanır

- ***Kullanıcının verdiği URL'lerin veya IP adreslerinin yeterince doğrulanmaması veya filtrelenmemesi.***
- ***Doğrulama ve filtrasyon eksiklikleri, saldırganların zararlı isteklere neden olabilecek verileri sunucuya göndermesine olanak tanır.***
- ***Sunucunun, iç ağda veya başka sistemlerdeki kaynaklara istek yapabilmesi, yanlış yapılandırılmış***

HTTP istekleri ve yönlendirmelerle sonuçlanabilir.

- ***Uygulama, URL veya IP adreslerini kullanarak iç kaynaklara istek gönderirken bu kaynakları yeterince kontrol etmemesi.***

Nasıl Önlenir?

- ***Kullanıcılardan alınan URL veya IP adreslerini sıkı bir şekilde doğrulayın ve filtreleyin. Geçerli URL formatlarını ve IP adreslerini kontrol edin.***
- ***İç ağ IP adreslerini veya URL'leri içeren kullanıcı girişlerini engelleyin.***
- ***İç ağ kaynaklarına yapılacak istekleri kontrol etmek için beyaz listeleme yöntemi kullanın. Yalnızca belirli ve güvenilir kaynaklara erişime izin verin.***
- ***İç kaynaklara yapılacak istekleri sınırlamak için beyaz listeleme politikaları oluşturun.***