OWASP (Open Web Application Security Project) Top 10 Zafiyetleri

Top 10 Zafiyet Listesi:

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

1. BROKEN ACCESS CONTROL

a. Zafiyet nedir?

Broken Access Control bir web uygulamasındaki güvenlik açıklarından biridir. Bu zafiyet sonucunda saldırgan normalde erişmemesi gereken kaynaklara erişebilir ve daha yüksek erişim iznine sahip olabilir.

b. Neden Kaynaklanır?

Erişim kontrolü kullanıcılara yalnızca belirli düzeyde, belirli kaynaklara erişime izin veren bir mekanizmadır.
Bu zafiyet erişim kontrolünün yanlış ya da eksik yapılandırılmasından kaynaklanır.

c. Zafiyetin Türleri

Yatay Erişim Kontrolü: Aynı yetkiye sahip kullanıcıların birbirlerinin verilerine veya işlemlerine izinsiz erişim sağlaması durumudur.

Dikey Erişim Kontrolü: Daha düşük yetkiye sahip bir kullanıcının, daha yüksek yetkili bir kullanıcı gibi davranarak erişim sağlaması durumudur.

d. Nasıl Önlenir?

- Hata mesajları en basit halde tutulmalı.
- Kullanıcının belirleyeceği parolalar belirli şartlar sunularak (uzunluk, karmaşıklık) oluşturulmaya izin verilmeli.
- Kullanıcılara minimum düzeyde yetki verilmeli.
- Olası anormal yetki erişimlerini fark edebilmek adına log kaytları tutulmalı.
- Düzenli güvenlik testleri yapılmalı.

2. CRYPTOGRAPHIC FAILURES

a. Zafiyet nedir?

Cryptographic Failures verilerin şifrelenmemesi ya da eski, daha önce çözülmüş şifreleme algoritmalarının kullanılmasından ortaya çıkan bir zafiyettir.

b. Neden Kaynaklanır?

Eski şifreleme algoritmaları kullanılmasından ya da verilerin düzgün şifrelenmemesinden kaynaklanır. Sunucu tarafında bilgiler kaydedilirken şifrelense bile, kullanıcıdan sunucuya aktarılırken veriler şifrelenmiyorsa da zafiyet ortaya çıkar.

c. Nasıl Önlenir?

- Http (Hyper Text Transfer Protocol), ftp (File Transfer Protocol) gibi güvensiz cleartext olarak çalışan protokoller yerine, şifreli bir şekilde transfer yapan https (Secure Hyper Text Transfer Protocol), ftps (File Transfer Protocol Secure) gibi protokoller kullanılmalıdır.
- Eski ve açığa çıkmış şifreleme algoritmaları yerine güçlü ve yeni algoritmalar kullanılmalıdır.
- Veriler şifreli bir şekilde kaydedilmelidir.

3. INJECTION

a. Zafiyet Nedir?

Injection zafiyeti, saldırganın jötü amaçlı verileri ya da komutları uygulamanın input alanlarına girerek çalıştırmasına olanak sağlayan bir güvenlik açığıdır.

b. Neden Kaynaklanır?

Injection zafiyeti, kullanıcının input verilerinin düzgün şekilde doğrulanmaması ve temizlenmemesiden oluşur. Uygulama, kullanıcıdan gelen inputu direkt bir komut veya sorgu içinde kullanırsa, saldırgan bu inputu manipüle ederek kötü amaçlı komutlar enjekte edebilir. Inputun güvenli olmayan şekilde işlenmesi, bu zafiyetin oluşmasını sağlar.

c. Zafiyetin Türleri

SQL Injection: SQL sorgularına zararlı komutlar enjekte edilerek veritabanına yetkisiz erişim sağlanabilir ve bu açık veritabanındaki bilgilerin değiştirilmesine, silinmesine ve çalınmasına olanak sağlar.

Command Injection: Özellikle bir uygulamanın kullanıcıdan input alınan kısımlarında bulunabilen bu açık, saldırganın Windows ya da Linux komutları çalıştırmasına olanak sağlayarak sisteme uzaktan erişim sağlama, dosya ekleme, silme, değiştirme gibi eylemlerde bulunabilmesine yol açar.

Cross-Site Scripting (XSS): Bir web uygulamasına zararlı JavaScript kodlarının enjekte edilmesi ile saldırganın diğer kullanıcıların tarayıcısında bu kodların çalışmasına olanak sağlar. Bunun sonucunda kullanıcı verileri çalınabilir veya oturumlarının ele geçirilebilir.

d. Nasıl Önlenir?

- Özel karakterlerin uygulama içerisinde kullanımı kısıtlanmalı ya da engellenmeli.
- Web uygulamalarına gelen tanımlanmamış robot trafiğinin fark edilmesi ve engellenmesi gibi çeşitli özelliklere olanak sağlayan WAF (Web Application Firewall) hizmeti kullanılmalı.

4. INSECURE DESİNG

a. Zafiyet Nedir?

Bir sistemin ya da uygulamanın tasarım aşamasında güvenlik önlemerine yeteri öenm verilmediğinde ortaya çıkan güvenlik açıklarıdır.

b. **Neden Kaynaklanır?**
   Tasarım aşamasında işlevsellik ve performansa öncelik verilmesi, güvenliğin 2. plana atılması bu zafiyete neden olabilir. SAFETY FIRST!

c. **Nasıl Önlenir?**
   - Güvenlik öncelik haline getirilmeli.
   - Kullanıcı yetkileri mümkün olduğunca kısıtlanmalı.
   - Güvenli yazılım mimarisi kullanılmalı.

5. SECURITY MISCONFIGURATION

a. **Zafiyet Nedir?**
   Security Misconfiguration zafiyeti, bir uygulamanın, ağın ya da sistemin güvenlik ayarlarının yanlış yapılandırılmasından kaynaklanan bir zafiyettir.

b. **Neden Kaynaklanır?**
   Gereksiz servislerin açık olması, varsayılan ayarların kullanılması, güvenlik protokollerinin yeterince sıkı olmaması gibi sebeplerden dolayı bu zafiyet ortaya çıkabilir.

c. **Nasıl Önlenir?**
   - Gereksiz servisler devre dışı bırakılmalı.
   - Varsayılan ayarlar değiştirilmeli, yerine özel ayarlar kullanılmalı.
   - Erişim kontrolleri yapılmalı.

6. VULNERABLE AND OUTDATED COMPONENTS

a. **Zafiyet Nedir?**
   Zayıf ve Güncel Olmayan Bileşenler anlamına gelen Vulnerable and Outdated Components zafiyeti, güncel olmayan bileşenlerin sistem güvenliğini tehlikeye atması ve güvenlik risklerini arttırmasıdır.

b. **Neden Kaynaklanır?**
   Bir uygulama veya sistemin eski veya güvenlik güncellemeleri yapılmamış bileşenler (kütüphaneler, yazılımlar) kullanmasıyla ortaya

çıkar. Bu tür bileşenler genellikle bilinen güvenlik açıklarına sahip olabilir ve bu açıklar, saldırganların sisteme yetkisiz erişim sağlamasına veya veri sızdırılmasına sebep olabilir.

c. **Nasıl Önlenir?**
   - Kullanılan bileşenlerin son sürüm olmasına ve güncellemelerin yapılmış olmasına dikkat edilmeli.
   - Bileşenler güvenilir kaynaklardan temin edilmeli.

7. IDENTIFICATION AND AUTHENTICATION FAILURES

a. **Zafiyet Nedir?**
   Kimlik Tanıma ve Kimlik Doğrulama Hataları anlamına gelen Identification and Authentication Failures zafiyeti, kullanıcıların kimliklerini doğrulamakta yaşanan sorunlar sebebiyle saldırganlara yetkisiz erişim ve saldırı yollarına ulaşım sağlar.

b. **Neden Kaynaklanır?**
   Zayıf parola politikaları, kötü şifreleme ugulamaları, güvenlik testlerinin eksikliği güvensiz kimlik doğrılama mekanizmaları bu zafiyete sebep olabilir.

c. **Nasıl Önlenir?**
   - Çok faktörlü kimlik doğrulama (MFA) kullanılmalı.
   - Güçlü şifreler kullanılmalı.
   - Oturum süreleri sınırlandırılmalı.

8. SOFTWARE AND DATA INTEGRITY FAILURES

a. **Zafiyet Nedir?**
   Yazılım ve Veri Bütünlüğü Hataları anlamına gelen Software and Data Integrity Failures zafiyeti, yazılım veya veri bütünlüğünün sağlanamaması durumunda ortaya çıkar ve sistemin güvenliğini tehlikeye atar.

b. **Neden Kaynaklanır?**
   Yazılım ve verilerin bütünlüğünü koruyan mekanizmaların yetersizliği veya hatalı çalışmasından kaynaklanır. Bu tür hatalar, verilerin veya

yazılımın yetkisiz kişiler tarafından değiştirilmesine veya bozulmasına neden olabilir.

c. **Nasıl Önlenir?**
   - Yazılım geliştirme sürecinde güvenli kodlama uygulamaları kullanılmalı.
   - Anormal faaliyetleri görebilmek ve önlem alabilmek adına log kayıtları tutulmalı.
   - Yazılım güncellemeleri güvenilir kaynaklardan alınmalı.
   - Dijital imza kullanılmalı.
   - Verilerin bütünlüğünden emin olarak yedekleme yapılmalı.

9. SECURITY LOGGING AND MONITORING FAILURES

a. **Zafiyet Nedir?**
   Güvenlik Günlüğü ve İzleme Hataları anlamına gelen Security Logging and Monitoring Failures zafiyeti, loglama hatalarından dolayı şüpheli davranışların fark edilmesini güçleştirir.

b. **Neden Kaynaklanır?**
   Log kayıtlarının yetersizliği, izleme araçlarının yetersizliği, kayıtların saklanmaması, kayıtların yeterince incelenmemesi nu zafieyete sebep olabilir.

c. **Nasıl Önlenir?**
   - Log kayıtları düzenli ve kapsamlı tutulmalı
   - Etkin izleme araçları kullanılmalı
   - Kayıtlar düzenli incelenmeli

10. SERVER-SIDE REQUEST FORGERY (SSRF)

a. **Zafiyet Nedir?**
   Sunucu Taraflı İstek Sahteciliği anlamına gelen SSRF zafiyeti, saldırganın bir sunucuyu kendisi adına istekte bulunmaya zorlamasına dayanan bir güvenlik açığıdır. SSRF, saldırgana bu güvenlik açığını barındıran sunucudan istek oluşturması veya buradan gelen istekleri

kontrol edebilmesi için web uygulamasındaki bir parametreyi değiştirmesine izin verir.

b. Neden kaynaklanır?

Kullanıcı girdilerinin doğrulanmaması, yanlış yapılandırılmış istekler, güvenlik kontrollerinin eksikliği bu zafiyete sebep olabilir.

c. Nasıl Önlenir?

- Kullanıcı girdileri doğrulanmalı
- Kaynaklara erişim sınırlandırılmalı
- Yalnızca belirli IP adreslerinden ve URL'lerden gelen isteklerin işlenmesine izin verilmeli.
- Firewall ve Proxy kullanılmalı.

Kaynaklar:

- https://docs.yavuzlar.org/web-guvenligi/broken-access-control
-
- https://www.siberguvenlik.web.tr/index.php/2021/04/15/command-injection-nedir/
-
- https://www.oracle.com/tr/security/cloud-security/what-is-waf/#:~:text=WAF%20hizmeti%2C%20web%20uygulamalarınıza%20gelen,olanak%20tanıyan%20çeşitli%20özellikler%20içerir.
-
- https://medium.com/@ebru_arslan/a04-insecure-design-636997095adb
-
- https://www.manageengine.com/vulnerability-management/misconfiguration/?network=g&device=c&keyword=&campaignid=11564688153&creative=478020053267&matchtype=&adposition=&placement=&adgroup=116703025350&targetid=dsa-471014726558&gad_source=1&gclid=Cj0KCQjwrKu2BhDkARIsAD7GBosCXv6BioBSDDw-tK5IJJKWYJA67PpF8cQZ-ne7RE369zipvxaa0cIaAniqEALw_wcB
-
- https://fordefence.com/owasp/#:~:text=A06%3A2021-Vulnerable%20and%20Outdated,ve%20yanlış%20yapılandırmasına%20neden%20olmaktadır.
-
- https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
-
- https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/
-
- https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/
-
- https://medium.com/@ahmetumitbayram/server-side-request-forgery-sunucu-taraflı-istek-sahteciliği-nedir-2d006d914799
-

- https://portswigger.net/web-security/ssrf