Yavuzlar OWASP TOP10 Zaafiyetler Raporu

Yazar: Enes Ayhan Benli

A01:2021-Broken Access Control:

Broken Access Control(Bozuk Erişim Kontrolü) zaafiyetinde, erişim kontrolü normalde kullanıcıların yetkileri dışındaki alana müdahale edememelerini sağlarken bu zaafiyet sayesinde belirli yerlerden kullanıcının yetkisinin üzerinde veya dışında işlemler yapılabilir.

Nedeni erişim kontrolünün yanlış yapılandırılması veya eksik uygulanması gibi durumlardır.

5 türü vardır bunlar;

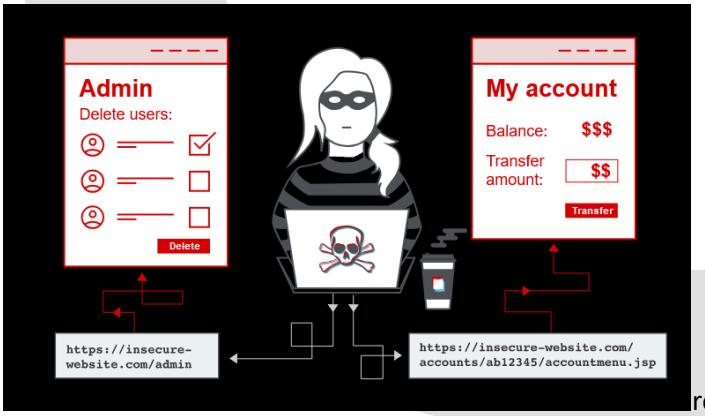
Insecure Direct Object References (IDOR) : nesne referanslarının güvensiz olması sonucu oluşan açık.

Violation of the Principle of Least Privilege : roller için gereken izinler olması gerekenden fazla olunca oluşan açık.

Access Control Checks Bypass : Sql injection veya Xss gibi güvenlik açıklarından yararlanarak erişim kontrollerinin atlatıldığı açık.

Session Management Flaws: Session tokenlerinin çalınıp yetkili kullanıcıların taklit edildiği açık.

Elevation of Privilege : Saldırganın alt düzey yetkilere sahip bir kullanıcıya sahipken üst seviyedeki kullanıcıların yetkilerini alıp yetkisini yükselttiği açık.



resim kaynağı:https://portswigger.net/web-security/access-control

Önlemek için yapılabilecekler, input filtresi, kod enjeksiyonu önleme, doğru yetki kontrolü ve yetkilendirmelerin doğru yapılması, rollerin gereğinden fazla yetki içermemesi gibi yöntemler uygulanabilir.

A02:2021-Cryptographic Failures:

Cryptographic Failures (Kriptografik Hatalar),hassas verilerin doğru şekilde şifrelenmemesi veya şifreleme algoritmalarının yanlış kullanılması sonucu oluşur.

Nedenleri uygulamalarda kullanılan şifreleme algoritmalarının yanlış seçilmesi, şifreleme algoritmalarının yanlış uygulanması, anahtar yönetimi hataları, rastgele sayı üretimindeki eksiklikler, eski şifreleme algoritmalarının kullanımı gibi durumlardır.

Önlemek için güçlü şifreleme algoritmaları kullanımı, düzenli olarak şifreleme kütüphanelerinin güncellenmesi, düzgün şifre yönetim prosedürleri kullanımı, https ve tls gibi güvenli iletişim protokolleri kullanımı gibi önlemler alınmalıdır

A03:2021-Injection:

Injection(Enjeksiyon), kullanıcı veri girişlerinin yeterince kontrol edilmemesi sonucu uygulamaya kötü niyetli komutların enjekte edilmesi sonucu oluşur.

Nedenleri kullanıcı girdilerinin doğrulanmaması, güvenlik kontrollerinin yetersiz olması ve tehlikeli girdilerin uygun şekilde işlenmemesidir.

Türleri:

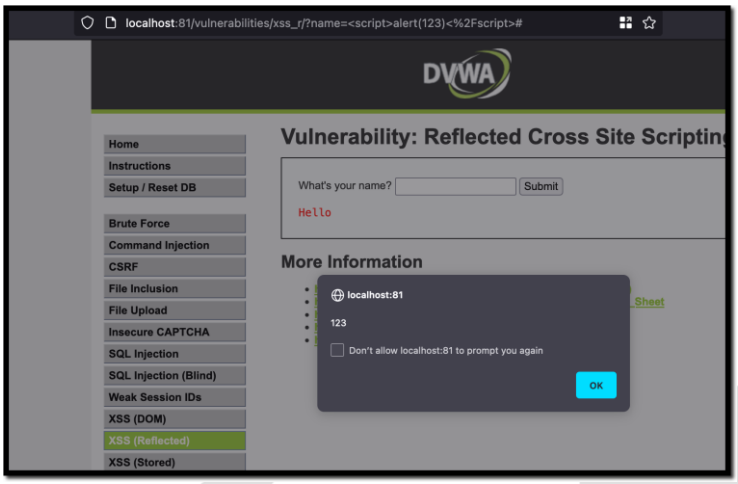
SQL/NoSQL Injection : Veritabanı komutları enjeksiyonu açığı.

Command Injection : Komut enjeksiyonu açığı.

Server Side Template Injection : Yerel Şablon sözdizimi kullanılarak şablona kötü amaçlı yük eklenen açık

Header Injection: Http isteklerini modifiye ederek oluşturulan açık.

Content Injection (XSS, HTML Injection, CSS Injection) : Websiteye içerik yüklenmesi yoluyla oluşan açık.



örnek bir XSS açığı, resim kaynağı: https://pentest-tools.com/blog/xss-attacks-practical-scenarios

Önlemek için ise, kullanıcı girdisine asla güvenmemek, güvenli kelime filtresi oluşturmak, girdiyi işlemek filtrelemek, çıktıyı göndermeden önce kodlamak, çıktıda sınırlamalar yapmak, Web Application Firewall(WAF) kullanmak gibi önlemler alınması gerekir.

A04:2021-Insecure Design:

Insecure Design (Güvensiz Tasarım), Genellikle güvenlik düşünülmeden tasarlanan uygulamalarda, tasarımdaki hatalar ve eksiklikler sebebiyle oluşan zaafiyettir. Örneğin kullanıcı girdi sınırlarının eksik olması buffer overflow gibi sorunlara yol açabilir.

Nedenleri Güvenlik Kontrollerinin İhmali, Güvenlik Odaklı Olmayan Tasarım, Eski Güvenlik Pratiklerinin Kullanımı, Tehdit Modelleme ve Risk Değerlendirmesi Eksikliği gibi durumlardır.

Önlemek için ise güvenli tasarım ilkelerinin uygulanması, güvenlik göz önüne alınarak yapılan yazılım geliştirme, açıkların düzeltilmesi, modern güçlü ve iyi güvenlik pratiklerinin kullanılması, Kritik kimlik doğrulama, erişim kontrolü, iş mantığı ve anahtar akışları için tehdit modellemesinin kullanılması gerekir.

A05:2021-Security Misconfiguration:

Security Misconfiguration(Güvenlik Yanlış Yapılandırılması), sistemlerin uygulamaların yapılandırılmalarında yapılan miskonfigürasyonlardan oluşur.

Nedenleri varsayılan parolalar, kullanılmayan hizmetler, uygunsuz dosya izinleri gibi durumlardır.

Önlemek için güncel güvenlik sistemlerinin kullanılması ve güncellemelerin kaçırılmaması, erişimler için yalnızca gerektiği kadar yetkilendirmesi, sistemlerin detaylıca incelenip doğru şekilde konfigüre edilmesi gerekir.

A06:2021-Vulnerable and Outdated Components:

Vulnerable and Outdated Components(Zaaf ve Eskimiş Bileşenler), sistemde kullanılan bileşenlerin eskimiş ve zaafiyete sahip oldukları güvenlik zaafiyetidir.

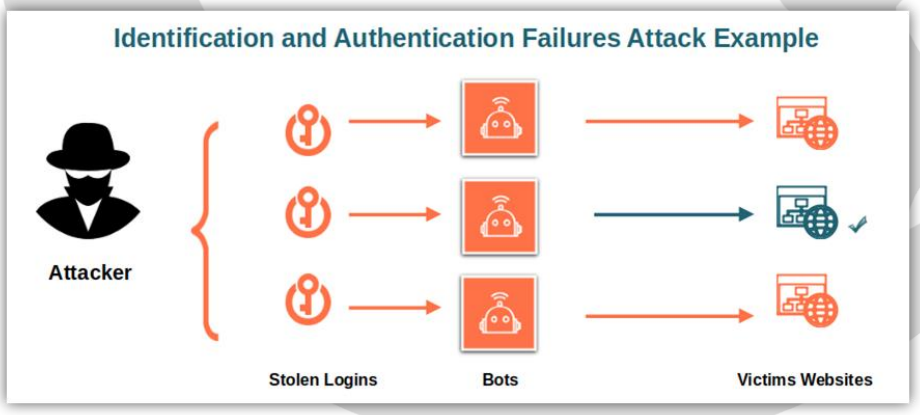
Nedenleri bileşenlerin kontrol edilmemesi, üçüncü parti bileşenlerin güncellenmemesi gibi durumlardır.

Önlemek için; düzenli olarak güvenlik açıkları taraması, bileşenlerin detaylı kontrolü, eski bileşenlerin yerine güncel bileşenlerin kullanımı, kütüphanelerin uyumluluğunun test edilmesi, konfigürasyonların kontrol edilmesi, güvenlik bültenlerinin takip edilmesi gerekir.

A07:2021-Identification and Authentication Failures:

Identification and Authentication Failures(Tanımlama ve Kimlik Doğrulama Hataları), kullanıcının kimliğini doğrulama aşamasındaki zaafiyetlerdir, Kimlik doğrulama ve oturum yönetimi, kimlik doğrulama ile ilgili saldırılara karşı koruma sağlamak için kritik öneme sahiptir.

Nedenleri; bruteforce'a izin veren sistemler, varsayılan parolalar, plain text halde tutulan parola verileri, eksik veya etkisiz çok faktörlü doğrulama, oturum tanımlayıcısının tekrar kullanılması gibi durumlardır



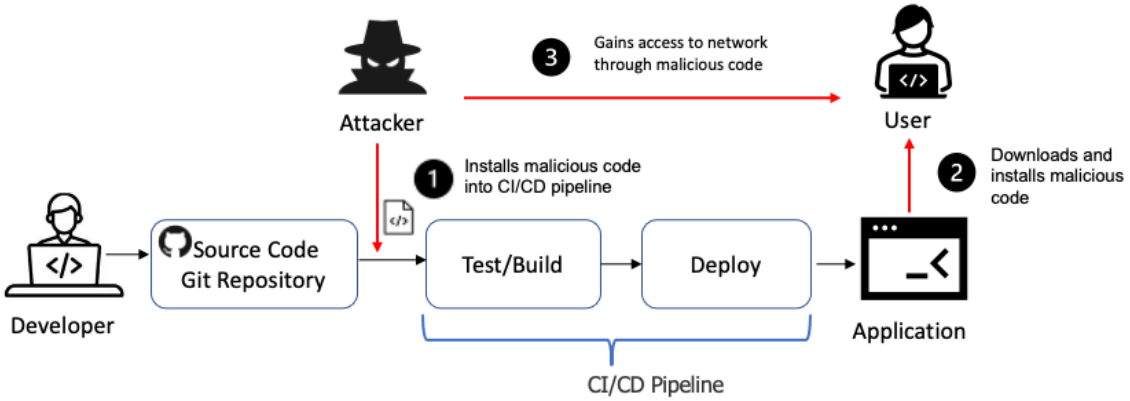
örnek bir zaafiyet resmi, kaynak: https://cheapsslsecurity.com/blog/what-are-the-owasp-top-10-vulnerabilities-and-how-to-mitigate-them/

Önlemek için; doğru ve güvenli çok faktörlü doğrulama sistemlerinin kullanılması, varsayılan şifrelerin kullanılmaması, parolaların en kötü 10.000 parola listesine karşı test edilmesi, parola politikalarının güncellenmesi, başarısız girişlerin rate limitle cevaplandırılması, güvenli oturum kimliği kullanılması gibi önlemler alınmalıdır.

A08:2021-Software and Data Integrity Failures:

Software and Data Integrity Failures(Yazılım ve Veri Bütünlüğü Hataları) zaafiyeti, bütünlük ihlallerine karşı koruma sağlamayan kod ve altyapı ile ilgilidir. Yazılımın ve verilerin bütünlüğünün korunmaması, saldırganların sisteme yetkisiz erişim sağlamasına ve verileri manipüle etmesine olanak tanır.

Nedenleri; Güvensiz Yazılım Güncellemeleri, Zayıf Kod Bütünlüğü Kontrolleri, Güvensiz Bileşenler, Yanlış Konfigürasyonlar, Eksik Veri Doğrulama gibi durumlardır



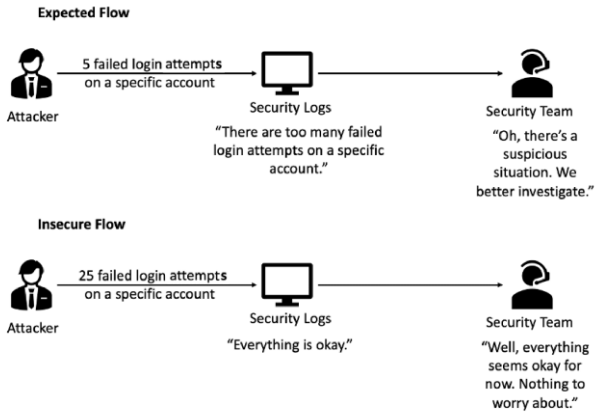
zaafiyetle ilgili örnek görsel, kaynak: https://my.f5.com/manage/s/article/K50295355

Önlemek için; dijital imzalar veya benzer mekanizmaların kullanılması, npm veya Maven gibi kütüphanelerin ve bağımlılıkların güvenilir depoları kullanıldığından emin olunması, yazılıma kötü amaçlı kod eklenmemesi için kod ve yapılandırma değişiklikleri için bir inceleme süreci oluşturulması, bütünlük kontrolü için güncel sistemlerin kullanılması gibi önlemler alınmalıdır.

A09:2021-Security Logging and Monitoring Failures:

Security Logging and Monitoring Failures(Güvenlik Günlüğü ve İzleme Hataları), uygulamaların veya sistemlerin güvenlik olaylarını kaydetme ve izleme süreçlerinde yetersizliklerinden kaynaklanan bir zaafiyettir. Önemli olayların kaçırılmasına ve saldırıların geç fark edilmesine yol açabilir.

Nedenleri; yetersiz veya eksik günlük kaydı(Örneğin giriş yapmak,hatalı giriş denemeleri gibi önemli olayların günlük kaydının tutulmaması), izleme ve uyarı mekanizmalarının olmaması, olayların yeterince analiz edilmemesi gibi durumlardır.



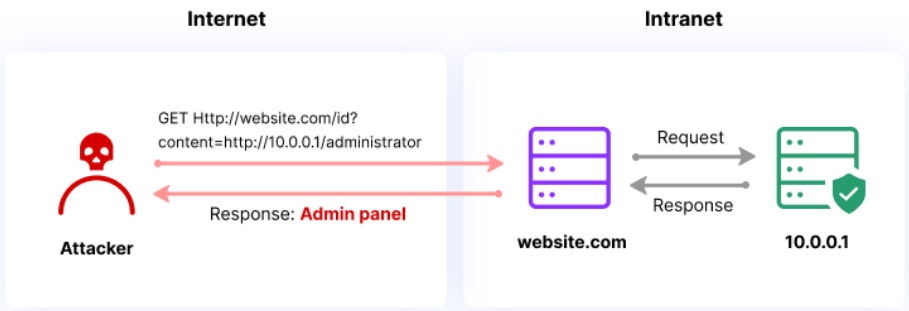
örnek görsel, kaynak: https://www.commencis.com/thoughts/top-10-web-application-vulnerabilities-an-owasp-overview/

Önlemek için; kapsamlı bir güvenlik günlüğü tutulması, tüm önemli olayların kaydedildiğinden ve bu günlüklerin düzenli olarak incelendiğinden emin olunması, gelişmiş izleme ve uyarı sistemlerinin kullanılması gibi önlemler alınması gerekir.

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

Server-Side Request Forgery(Sunucu Tarafı İstek Sahteciliği), sistemdeki bir urlye istek gönderirken sahte ip adresi ve alan adı sağlanarak normalde sadece içeriden erişilebilecek kısımlara dışarıdan erişilebilen zaafiyettir. Bir web uygulaması kullanıcı tarafından sağlanan URL'yi doğrulamadan uzak bir kaynağı getirdiğinde ortaya çıkar.

Nedenleri; kullanıcının sağladığı girdilerin yeterince doğrulanmaması, güvenli olmayan şekilde dış kaynaklardan veri alma süreçleri, sunucu yapılandırmasındaki hatalar gibi durumlardır.



örnek görsel, kaynak: https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/

Önlemek için sunucu tarafındaki isteklerin kaynağını ve hedefinin doğrulanması, dış kaynaklardan gelen istekleri güvenli bir şekilde işlenmesi, güvenlik duvarları ve ağ segmentasyonu kullanarak sunucunun yalnızca güvenli kaynaklara erişim sağladığından emin olunması ve ağ ve uygulama katmanında önlemler alınması gerekir.

Kaynakça

- https://owasp.org/www-project-top-ten/
- https://medium.com/@aysekaya/owasp-top-10-zafiyetleri-ve-al%C4%B1nmas%C4%B1-gereken-%C3%B6nlemler-a1a38280148e
- https://portswigger.net/web-security/access-control
- https://www.authgear.com/post/what-is-broken-access-control-vulnerability-and-how-to-prevent-it
- https://www.pentestpeople.com/blog-posts/owasp-top-ten-cryptographic-failures
- https://www.securityjourney.com/post/owasp-top-10-cryptographic-failures-explained
- https://portswigger.net/web-security/server-side-template-injection
- https://portswigger.net/web-security/host-header
- https://pentest-tools.com/blog/xss-attacks-practical-scenarios
- https://medium.com/@shivam_bathla/a04-2021-insecure-design-9e16449f29ef
- https://www.wallarm.com/what/a04-2021-the-insecure-design
- https://www.securityjourney.com/post/owasp-top-10-identification-and-authentication-failures
- https://cheapsslsecurity.com/blog/what-are-the-owasp-top-10-vulnerabilities-and-how-to-mitigate-them/
- https://my.f5.com/manage/s/article/K50295355
- https://www.commencis.com/thoughts/top-10-web-application-vulnerabilities-an-owasp-overview/
- https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/