OWASP TOP 10

Bugün bahsedeceğim konu OWASP(Open Web Application Security Project) TOP 10.

OWASP TOP 10, 2003 yılında ortaya çıkmıştır. Bizlere web uygulamalarına yönelik en kritik güvenlik riskleri hakkında bilgiler verir. Bu bilgiyi kullanmak, kurumlarımızda güvenli kod üretimini bizlere sağlamakta yardımcı olur.

1-)Broken Access Control: OWASP 2017' de 5. Sıradayken 2021'de paylaşılan OWASP listesinde 1. Sırada yer almaktadır. İstismara uğrayan en popüler açıkların başında yer alır.

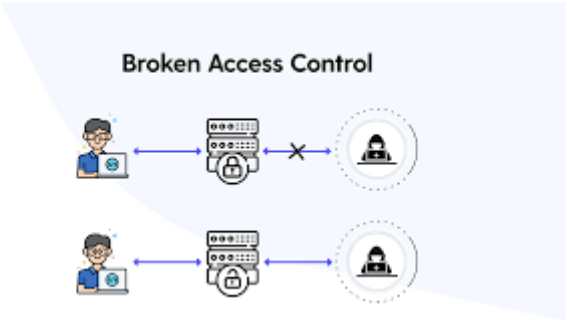
Uygulamalarda yapılan yanlış veya hatalı erişim kontrolü sonucu, yetkisi olmayan kullanıcıların uygulama içerisinde yer alan kaynaklara izinsiz erişmesine veya bunları değiştirmesine neden olabilir.

3 tür Access Control açığımız vardır. Bunlar:

• *Dikey Erişim Kontrolü(Vertical Access Control):* Diğer kullanıcı türleri için mevcut olmayan hassas işlevlere erişimi kısıtlayan(ne kadar erişebileceğini düzenler) mekanizmalardır.

• *Yatay Erişim Kontrolü(Horizontal Access Control):* Kaynaklara erişimi, bu kaynaklara erişmelerine özel olarak izin verilen kullanıcılarla kısıtlayan mekanizmalardır. (Kullanıcıların birbirlerinin kaynaklarına yetkisiz erişimi engellenir.)

• *Bağlama Bağlı Erişim Kontrolleri:* Uygulamanın durumuna veya kullanıcının onunla etkileşimine bağlı olarak işlevlere ve kaynaklara erişimi kısıtlar. (Bir kullanıcı çalıştığı projeye göre erişim alır.)



<u>NASIL ÖNLENİR?</u>

☺ Bir kaynağın herkese açık olması gerekmedikçe, varsayılan kullanıcılar için erişimi kısıtlayın.(reddedin)

☺ Erişim kontrolünü mümkün olduğunca kısıtlayın.

☺ Çalıştıklarından emin olmak için erişim kontrollerini baştan sona test edin, bu açık ciddi bir açıktır. (Düzenli olarak bu testleri devam ettirin.)

2-)Cryptographic Failures: OWASP 2017'de 3.sırada yer alırken 2021'de 2.sıraya yükselmiştir.

Hassas verilerin şifrelenme işlemi sırasında ortaya çıkan eksik veya yanlış uygulanmasıdır. Uygulama güvenliği için çok önemli bir risk faktörüdür. Hassas verilerin ortaya çıkması veya sistemin tehlikeye atılması gibi sonuçlara yol açabilir.

NASIL ÖNLENİR?

- Doğru şifreleme algoritmaları kullanılmalıdır.
- Kriptografik anahtarlar düzenli olarak yenilenmelidir.
- Saklanan veriler sınıflandırılmalı (böylelikle gereksiz verilerin saklanması önlenir.)
- Gerekli hassas veriler şifrelenerek(simetrik, asimetrik vb.) saklanmalı

3-)INJECTION: Web uygulamalarında en sık görülen açıklardan birisidir. OWASP 2017'de 1.sıradayken kurumların bu konu üzerinde aldıkları önlemler sonrası 2021'de 3.sıraya gerilemiştir.

Uygulamalarda kullanılan veri girişleri yoluyla; kötü niyetli kullanıcıların veri tabanı sorguları, komutları veya diğer işlevleri veri girişlerinde yapılan hatalı denemeler sonucunda zararlı kod enjekte etmesine yol açar. Veri tabanına erişen kötü niyetli kullanıcılar kişisel bilgilere ve daha birçok hassas verilere erişebilir.

SQL, NoSQL: Veri tabanı dillerindendir. Bu diller ile sorgular yapılarak hassas verilere ulaşılabilir.

OS komutları: İşletim sistemi dilinde sorgular yapılarak istenmeyen verilere ulaşılabilir.

LDAP: Dizin yönetiminde bulunan açıklardan yararlanılarak zafiyet ortaya çıkabilir.

XSS(Cross,Site Scripting): Siteler arası komut dosyası çalıştırma saldırısıdır.

NASIL ÖNLENİR?

☺ Parametrelerin doğrulanması

☺ Oluşabilecek kod çalıştırma zafiyetlerini kapatacak kodların eklenmesi

☺ WAF kullanımı

☺ Gerekli testlerin yapılması

4-)Insecure Design: OWASP 2021'de ilk defa çıkarak 4.sırada yerini aldı.

Web uygulama tasarımında ortaya çıkan hata veya eksiklerden kaynaklanan açıktır.

NASIL ÖNLENİR?

• Güvenli tasarım ilkeleri uygulanmalıdır.

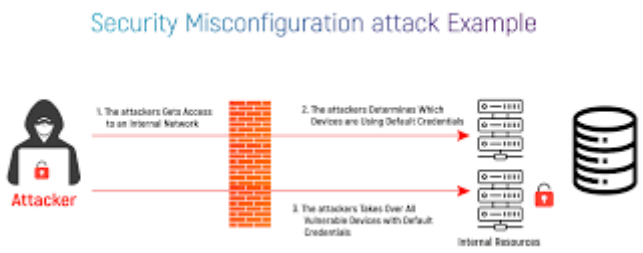
• Güvenlik odaklı bir yaklaşım sergilenmelidir.

• Güvenlik testleri düzenli olarak yapılmalıdır.

5-)Security Misconfiguration: OWASP 2017'de 6.sırada yer alırken 2021'de 5.sıraya yükselmiştir.

Bir uygulamanın veya sistemin yanlış yapılandırılması sonucu ortaya çıkan güvenlik açığıdır. Konfugürasyon dosyasındaki hatalı veya eksik işlemlerden kaynaklanır. Saldırganlar uygulama sunucusuna erişebilir ve hassas verileri ele geçirebilirler.

XML (XXE): Bir saldırganın uygulamanın XML verilerini işlemesine müdahale etmesine izin veren güvenlik açığıdır.



NASIL ÖNLENİR?

☺ En iyi uygulamaları takip etmek

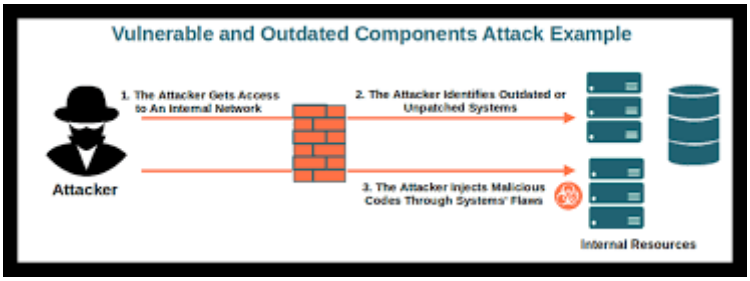
☺ Yazılımın güvenlik düzeylerini test ve kontrol etmek

☺ Sistem yapılandırmasını güncel tutmak

6-)Vulnerable and Outdated Components: OWASP 2017'de 9.sırada yer alırken 2021'de 6.sıraya yükselmiştir. Savunmasız(güvenlik açığı olan) ve Güncel Olmayan Bileşenler olarak tercüme edilebilir.

Bu durum işletim sistemlerini, web veya uygulama sunucularını, veri tabanı yönetim sistemlerini, API'leri etkiler. Saldırganın bunların yalnızca birinde bir açık bulması yeterli olur.

Bilinen tüm güvenlik açıklarına karşı koruma yapar fakat bilinmeyen açıklar sorun oluşturur ve her yıl 18.000 gibi bir miktarda güvenlik açığı ortaya çıkmaktadır.



NASIL ÖNLENİR?

• Güncelleme politikaları oluşturmak

• Güvenlik testlerini düzenli ve sürekli yapmak(proaktif)

7-)Identification and Authentication Failures: OWASP 2017’ 2.sırada 2021’de 7.sıraya kadar gerilemiştir. Bozuk kimlik doğrulama olarak dilimize tercüme edilir.

Bir kullanıcının kimliğinin doğrulanması veya yetkilendirilmesi sırasında yaşanan sorunlar nedeniyle ortaya çıkar. Bu açık, bir saldırganın bir kullanıcının kimlik bilgilerini çalmasına veya sahte bir kimlik kullanarak uygulamaya erişmesine izin verebilir.

NASIL ÖNLENİR?

☺ Güçlü kimlik doğrulama yöntemleri

☺ Kimlik doğrulama işlemlerini kayıt altına almak veya izlemek

☺ Kimlik bilgilerini arka tarafta şifrelemek

☺ Çok faktörlü kimlik doğrulama yöntemlerini kullanmak

8-)Software and Data Integrity Failures: OWASP 2021’de yeni çıkan bir güvenlik açığıdır. Yazılım ve Veri Bütünlüğü Arızaları olarak dilimize tercüme edilir.

Bir yazılım veya veri sisteminin beklenmeyen şekilde değiştirilmesi sonucu meydana gelir. Saldırgan yazılım veya veri sistemini kötü amaçlı yazılım yükleyerek verileri çalabilir, bozabilir veya sistemi kontrol edebilir.

NASIL ÖNLENİR?

• Yazılımın veya verilerin istenen kaynaktan olduğunu ve değişmeden geldiğini doğrulamak amacıyla hash, dijital imza gibi mekanizmalar kullanmalıdır.

• Kod ve yapılandırma değişiklikleri için inceleme süreci oluşturmak

• Yazılım güncellemelerini düzenli olarak yüklemek

• Güçlü erişim kontrolü uygulamak

• Veri yedekleme işlemleri

• Güvenli yazılım geliştirme yöntemleri uygulamak

9-)Security Logging and Monitoring Failures: OWASP 2017'de 10.sıradayken 2021'de 9.sıraya yükselmiştir.

Güvenlik olaylarının izlenmesi ve kaydedilmesi sırasında oluşan yetersiz veya hatalı yapılandırma sonucu açığa çıkar. Kötü amaçlı aktivitelerin tespit edilememesi eya güvenlik olaylarına yanıt verilememesi gibi sonuçlara neden olur.

NASIL ÖNLENİR?

☺ Güvenlik olaylarının izlenmesi ve kaydedilmesi için uygun araçlar kullanmak

☺ Günlük olayların düzenli olarak incelenmesi

☺ Uyarı ve alarm sistemleri kullanmak

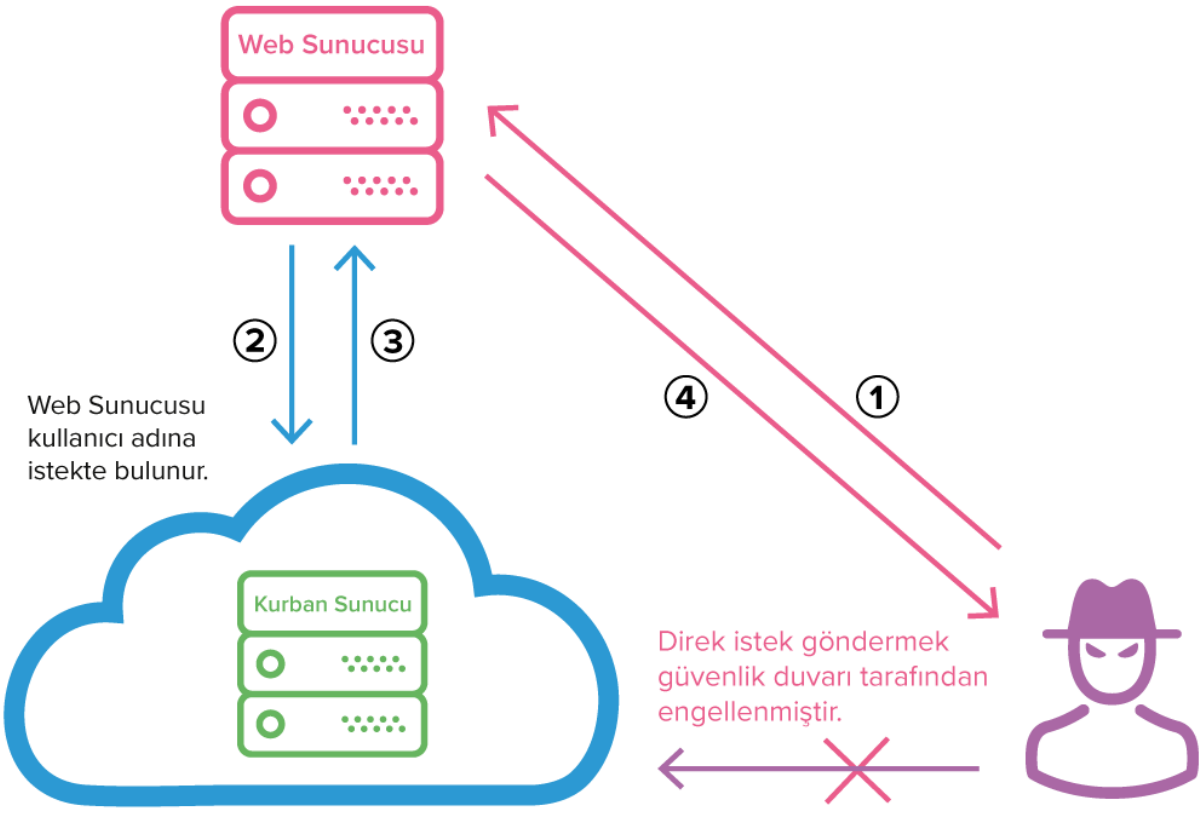
10-)Server-Side Request Forgery: Diğer adıyla **SSRF** (Sunucu Taraflı İstek Sahtekârlığı). 2021'de listemize katılan ve OWASP TOP 10' un sonunda yer alan bir web güvenlik zafiyetidir.

Saldırgan, hedef sunucuda bir URL'ye istek gönderirken sahte kaynaklı bir IP adresi ve alan adı(host) sağlayarak sunucunun kendi iç ağına erişmesine izin veren bir web güvenlik açığıdır. Kısacası saldırgan kendisini hedef kullanıcının sunucusu gibi göstererek gelen isteği, kendi oluşturduğu kötü amaçlı isteğe yönlendirir. Bir uygulamanın doğrulama sürecini atlayarak veya bir URL'de bir sunucu adresi veya IP adresi gibi güvenliği kontrol etmeyen bir parametre kullanarak gerçekleştirebilir.

Nelerden kaynaklanabilir?

-Kullanıcı girdilerinin kontrol edilmemesi, doğrulanmaması

- Yanlış yapılandırma

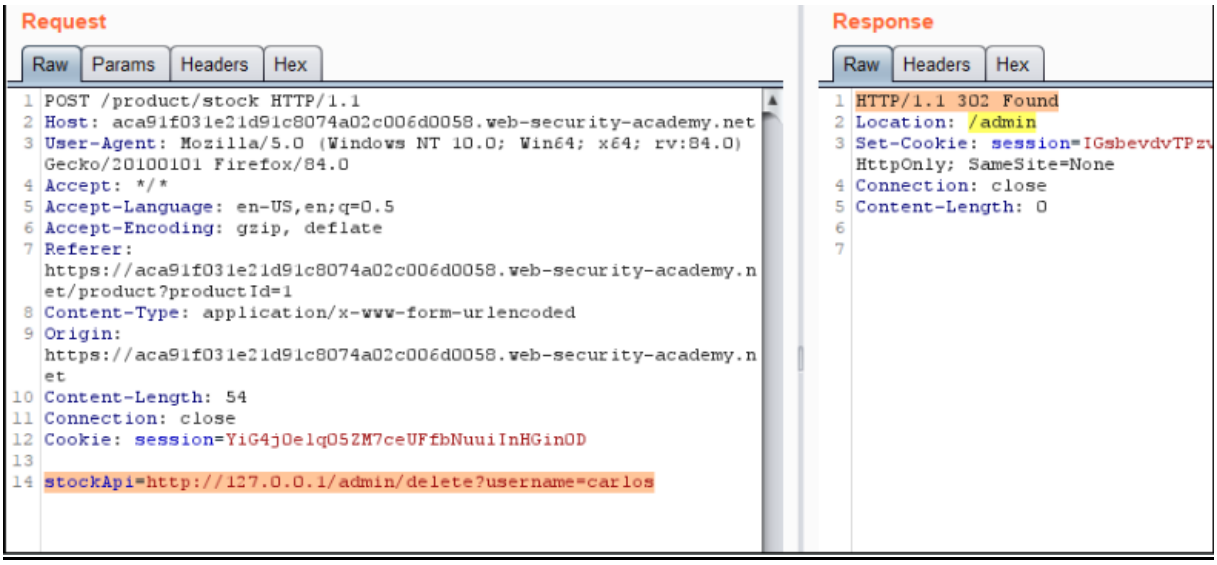


Hedef sunucu için ciddi bir güvenlik tehdittir çünkü saldırgan sunucuya belirli istekler göndererek hassa verileri çalabilir, sistemi ele geçirebilir veya devre dışı bırakabilir.

SSRF Türleri

◆ Blind (Kör) SSRF: Saldırgan sunucu isteklerinin sonucu göremese de bu istekleri arka tarafta manipüle edebilir. Genellikle veri tabanlarını hedef alır.

◆ Zincirleme SSRF: SSRF kullanılarak başka zafiyetlerinde ortaya çıkması durumudur. Reflected XSS, RCE(uzaktan komut çalıştırma), LFI(local file inclusion) gibi zafiyetler ortaya çıkabilir.



NASIL ÖNLENİR?

• Giriş doğrulaması yapılması

• Güvenlik duvarı kurulumu

• Güvenli URL işleme

• Sunucu ayarlarının kontrol edilmesi

• HTTP isteklerini sınırlama

• DNS çözümleme ve whitelist oluşturma

• Kullanılmayan URL şemalarını devre dışı bırakma

• İç ağda bulunan servislere erişim için kimlik doğrulama

• Sunucu taraflı yanıt kontrolü