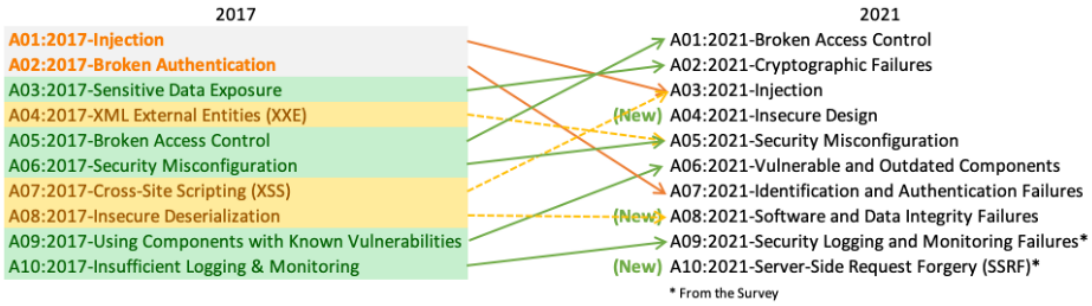
OWASP TOP 10 RAPORU

OWASP (Open Web Application Security Project) isimli açık kaynaklı bir proje topluluğudur. Türkçeye Açık Web Uygulaması Güvenlik Projesi olarak çevrilmektedir. Bu topluluğun amacı web uygulamalarında karşılaşılan güvenlik zafiyetlerini tespit etmek, bu zafiyetleri önlemek için çözümler üretmek ve güvenli yazılım geliştirme konusundaki farkındalığı artırmaktır. OWASP'ın en bilinen projelerinden biri olan OWASP Top 10, web uygulamalarındaki en yaygın güvenlik açıklarını listeleyen bir rehberdir. Bu rehberi oluşturmak için 200.000'den fazla kuruluştan ve sektör uzmanlarının anketlerinden veri toplar. Bu nedenle geniş bir uzmanlık havuzuna sahip olduğu söylenebilmektedir. Bu liste her 3 yılda bir güncellenmektedir. En son 2021 yılında güncel versiyonu yayımlanmıştır.



1) Broken Access Control (Bozuk Erişim Kontrolü)

Broken Access Control Zafiyeti Nedir?

Broken Access Control yani "Bozuk Erişim Kontrolü" zafiyeti sistem üzerinde yetersiz veya yanlış yetkilendirilmeler sonucunda oluşan bir zafiyettir. Bu duruma zayıf kimlik doğrulama, oturum yönetme kontrollerinin eksikliği, veritabanı kayıtlarına veya uygulamadaki dosyalara doğrudan erişebilme gibi hatalı yapılandırmalar neden olabilir.

Neden Kaynaklanır?

- Yanlış veya eksik yapılandırmalar
- Zayıf kimlik doğrulama
- Oturum yönetme kontrollerinin yetersizliği

Örnek Saldırı:

Aşağıda gösterilen örnek uygulamada hesap bilgilerine erişirken kullanılan bir SQL sorgusunda doğrulanmamış verilerin kullanıldığı gözlenmektedir.

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery( );
```

Saldırgan tarayıcıda 'acct' parametresini değiştirerek istediği hesap numarasını görebilmektedir. Eğer girdi doğrulama işlemi yapılmazsa saldırgan herhangi bir kullanıcının hesabına erişebilir.

```
https://example.com/app/accountInfo?acct=notmyacct
```

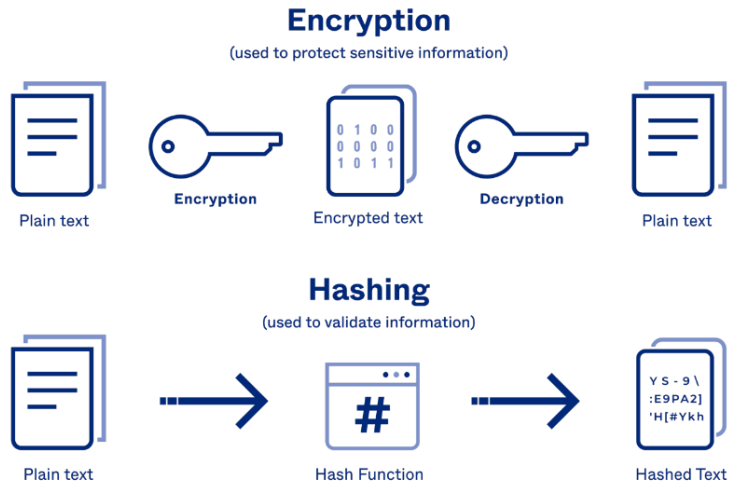
Nasıl Önlenir?

- Yapılandırılmalar yapılırken en az yetki ilkesi izlenmelidir. Rol tabanlı erişim kontrolü yöntemi uygulanabilir.
- Girdi doğrulama ve temizleme işlemleri yapılmalıdır.
- Çok faktörlü kimlik doğrulama uygulanmalıdır.
- Log kayıtları tutulmalı ve düzenli olarak denetlenmelidir.

2) Cryptographic Failures (Kriptografik Hatalar)

Cryptographic Failures Zafiyeti Nedir?

Cryptographic Failures, kriptografi ile korunan verilerin yanlış şifrelenmesi veya şifresinin çözülebilmesi durumunda meydana gelen bir zafiyettir. Bu tür zafiyetler genellikle hatalı şifreleme algoritmalarının seçilmesi ya da yanlış anahtar seçimi nedeniyle ortaya çıkar. Bu durum, saldırganların şifrelenmiş verilere erişim sağlamasına yol açabilir.



Neden Kaynaklanır?

- Zayıf veya yanlış şifreleme algoritmalarının kullanılması.
- Kısa şifreleme anahtarları kullanılması.
- Güvensiz parola depolama yöntemleri.
- Eski veya güvensiz şifreleme protokollerinin kullanımı.
- Zayıf SSL/TLS protokollerinin kullanımı.

Örnek Saldırı:

Zayıf şifreleme tekniklerinin kullanılması, hassas bilgilerin, (kredi kartı numaraları vb.) saldırganların eline geçmesine neden olabilir. Örneğin, zayıf SSL/TLS protokolleri kullanıldığında, saldırganlar internet trafiğini dinleyerek man-in-the-middle saldırıları gerçekleştirebilir. Ayrıca, şifreleme algoritmalarının yetersiz olması, saldırganların verilerin şifrelerini kırmasına yol açabilir.

Nasıl Önlenir?

- Doğru ve güçlü şifreleme algoritmaları kullanılmalı.
- Anahtarlar düzenli olarak yenilenmeli.
- Güçlü random sayı üretim teknikleri kullanılmalı.
- Güncel ve doğrulanmış algoritmalar kullanılmalı.
- Hassas verilerin hem depolama esnasında hem de iletim sırasında şifrelenmesi sağlanmalı.
- Düzenli güvenlik testleri ve zafiyet testleri yapılmalı.

3) Injection (Enjeksiyon)

Injection (Enjeksiyon) Zafiyeti Nedir?

Web uygulamalarında görülen bir zafiyet türüdür. Saldırganın uygulamadaki veri girişlerine zararlı kod yükleyerek sistemi manipüle etmesinden kaynaklanır. Saldırgan bu yöntemle uygulamanın veritabanını, sunucularını ve hassas bilgilerini ele geçirebilir.

Neden Kaynaklanır?

- Girişlerin doğru şekilde doğrulanmaması ve filtrelenmemesi.
- Kullanıcıdan alınan verilerin doğrudan sorgulara veya komutlara dahil edilmesi.
- Yetersiz veri doğrulama.

Örnek Saldırı:

- **SQL Injection:** Saldırganlar, veritabanı sorgularına zararlı SQL komutları ekleyerek veri çalabilir veya veritabanını manipüle edebilirler.
- **Cross-Site Scripting (XSS):** Kullanıcı oturumlarının veya cookielerin çalınması gibi sonuçlara yol açabilir.
- **Command Injection:** Saldırganlar, komutlara kötü amaçlı komutlar ekleyerek sunucu kontrolünü ele geçirebilir.

Nasıl Önlenir?

- Tüm kullanıcı girişlerinin doğru ve güvenli bir şekilde doğrulanması ve temizlenmesi gerekir.
- Güvenli kodlama standartları izlenmeli
- WAF (Web Application Firewall) yani web uygulama güvenlik duvarı ile zararlı girişler engellenmeli.
- Güvenilir olmayan client davranışlarını tespit eden saldırı tespit sistemleri uygulanmalı.
- Sadece güvenli ve izin verilen girdiler kabul edilmeli.

4) Insecure Design (Güvensiz Tasarım)

Insecure Design (Güvensiz Tasarım) Zafiyeti Nedir?

Güvensiz Tasarım, bir web uygulamasının tasarımındaki hatalar veya eksiklikler nedeniyle ortaya çıkan bir güvenlik açığıdır. Bu tür hatalar, özellikle kimlik doğrulama, yetkilendirme, veri gizliliği ve bütünlüğü gibi kritik güvenlik konularında zafiyetlere yol açabilir. Güvensiz tasarımlar, saldırganların bu açıkları kullanarak uygulamalara zarar vermesine veya hassas verilere erişmesine neden olabilir.

Neden Kaynaklanır?

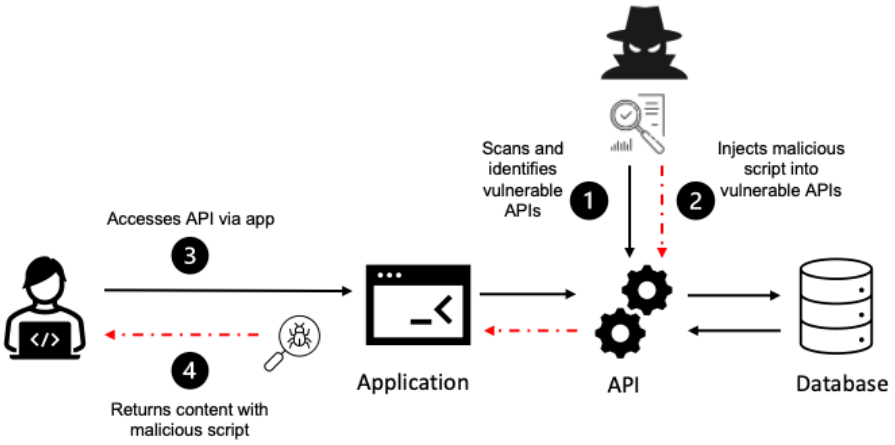
Tasarım aşamasında potansiyel tehditlerin yeterince öngörülmemesi ve değerlendirilmemesi de bu tür zafiyetlerin temel nedenleri arasındadır. Örneğin, aşırı detaylı hata mesajları veren bir uygulama, saldırganlara uygulamanın iç yapısı hakkında ipuçları verebilir ve bu bilgiler, SQL enjeksiyonu gibi diğer saldırılarda kullanılabilir.

Örnek Saldırı:

Örneğin saldırganlar hata mesajlarından elde ettikleri bilgilerle saldırıları (SQL enjeksiyonu gibi) başlatabilirler. Bu tür açıklar, hem bireyler hem de kurumlar için maddi ve itibar kaybına neden olabilir.

Nasıl Önlenir?

- Uygulama geliştirme sürecinde güvenli tasarım ilkeleri ve desenleri kullanılmalı.
- Yazılımın güvenlik yamalarının düzenli olarak güncellenmesi sağlanmalı.
- Sistem güvenliği için en iyi yazılımlar kullanılmalı ve düzenli olarak güncellenmelidir.



5) Security Misconfiguration (Güvenlik Yanlış Yapılandırmaları)

Security Misconfiguration (Güvenlik Yanlış Yapılandırmaları) Zafiyeti Nedir?

Security Misconfiguration, yani Güvenlik Yanlış Yapılandırmaları, bir uygulamanın ya da sistemin yanlış veya yetersiz yapılandırılması sonucu ortaya çıkan bir güvenlik açığıdır. Bu durum, sistemin güvenliğini sağlamak için gerekli olan işlemlerin tam olarak uygulanmaması veya hiç uygulanmamasından kaynaklanır. Örneğin, bir web sunucusunun yanlış yapılandırılması sonucu, saldırganlar bu sunucuya sızabilir ve hassas verilere ulaşabilirler.

Neden Kaynaklanır?

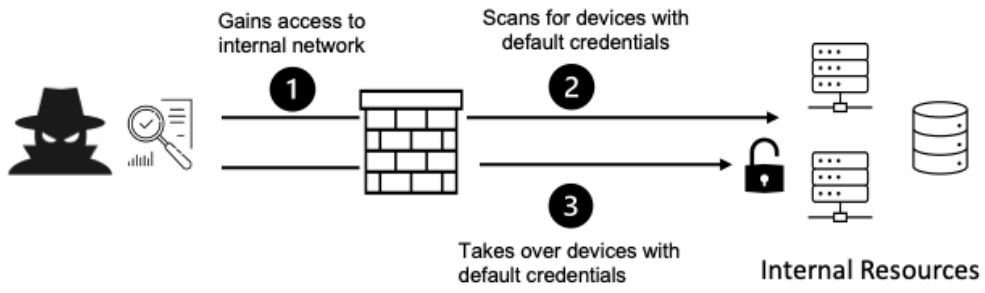
Güvenlik yanlış yapılandırmaları, genellikle sistemlerin varsayılan ayarlarla bırakılması, kullanılmayan sayfaların açık kalması, gereksiz hizmetlerin çalıştırılması veya güvenlik güncellemelerinin ihmal edilmesi gibi sebeplerden kaynaklanır.

Örnek Saldırı:

Varsayılan dosya izinlerinin değiştirilmediği bir sistemde, saldırganlar bu açıkları kullanarak sistemin kontrolünü ele geçirebilir veya hassas verilere erişebilirler. Yanlış yapılandırmalar, ayrıca güvenlik yamalarının ve güncellemelerinin ihmal edilmesi sonucu oluşan zafiyetlere de yol açar.

Nasıl Önlenir?

- Sistemleri ve uygulamaları kurarken varsayılan yapılandırmaları değiştirmek gereklidir.
- Yazılımın güvenlik düzeyini düzenli olarak kontrol edilmeli, güvenlik yamaları ve güncellemeleri zamanında uygulanmalıdır.
- Sistemler olabildiğince sade tutulmalıdır.



6) Vulnerable and Outdated Components (Zayıf ve Güncellenmemiş Bileşenler)

Vulnerable and Outdated Components (Zayıf ve Güncellenmemiş Bileşenler) Nedir?

Bir uygulamanın, güncellenmemiş veya bilinen güvenlik açıklarına sahip üçüncü taraf sistemler kullanması sonucu oluşan bir güvenlik açığıdır. Modern uygulamalar genellikle açık kaynak kütüphaneler, veritabanı yönetim sistemleri ve sunucu yazılımları gibi üçüncü taraf bileşenler kullanır. Eğer bu bileşenler güncel değilse veya bilinen güvenlik açıklarına sahipse, tüm sistemin güvenliği riske girebilir.

Neden Kaynaklanır?

Bu zafiyet, genellikle yazılımın düzenli olarak güncellenmemesi, eski veya artık desteklenmeyen sistemlerin kullanımı ve güncel güvenlik yamalarının uygulanmaması nedeniyle ortaya çıkar. Saldırganlar, bu tür zayıf bileşenleri tespit ederek, kötü amaçlı yazılım yaymak veya kimlik avı saldırıları başlatmak gibi kötü niyetli faaliyetlerde bulunabilirler. Bu nedenle, bileşenlerin güncellenmemesi, özellikle web uygulamaları için ciddi güvenlik riskleri oluşturur.

Örnek Saldırı:

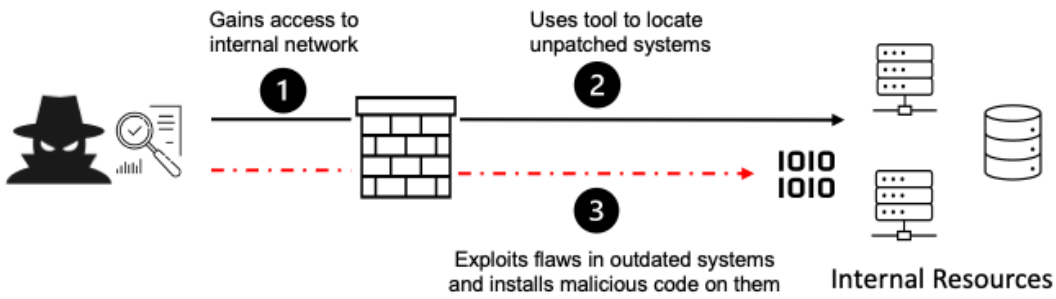
Bir web sitesinin, güvenlik açığı bulunan bir kütüphane kullandığını düşünelim. Bu durumda, saldırganlar bu güvenlik açığından faydalanarak siteye sızabilir, hassas verilere erişebilir veya siteyi kontrol altına alabilirler.

Nasıl Önlenir?

- Tüm yazılım bileşenlerinin güncel tutulması gerekmekte.
- Tarama için güncellenmiş sağlam bir zafiyet veritabanı kullanılmalı.

7) Identification and Authentication Failures (Kimlik Doğrulama Hataları)

Identification and Authentication Failures (Kimlik Doğrulama Hataları) Nedir?



Bir kullanıcının kimliğinin doğrulanması sırasında yaşanan hatalar sonucunda oluşan güvenlik açıklarıdır. Bu tür zafiyet, saldırganların sahte kimlik bilgileri ile sisteme erişim sağlamasına veya kullanıcı kimlik bilgilerini çalarak oturum açmasına yol açabilir. Güçlü kimlik doğrulama

yöntemlerinin kullanılmaması veya kimlik doğrulama süreçlerinin doğru bir şekilde yönetilmemesi bu tür zafiyetlerin sebeplerindendir.

Neden Kaynaklanır?

- Kısa, basit veya tahmin edilebilir parolalar kullanılması bu tür zafiyetlerini artırır.
- Sadece kullanıcı adı ve parola ile kimlik doğrulama yapılması, sistemleri savunmasız bırakır.
- Kullanıcı erişim izinlerinin doğru yönetilmemesi, yetkisiz kullanıcıların sisteme erişimine yol açabilir.

Örnek Saldırı:

Basit parolalar kullanıldığında ve giriş denemelerine herhangi bir sınırlama getirilmediğinde, saldırganlar kaba kuvvet saldırıları ile sisteme sızabilirler.

Nasıl Önlenir?

- **Çok Faktörlü Kimlik Doğrulama** kullanımı, kullanıcıların kimlik doğrulama sürecinde ek bir güvenlik katmanı sağlar.
- Uzun, karmaşık ve tahmin edilmesi zor parolalar oluşturulmalıdır. Parolaların düzenli olarak değiştirilmelidir.
- Zaman kısıtlamalı oturum kimlikleri oluşturulmalı. Varsayılan kimlik bilgileri değiştirilmelidir.
- Başarısız giriş denemeleri izlenmeli ve bu tür girişimlere yönelik sınırlamalar getirilmelidir.
- Kullanıcıların güvenilir parola yöneticileri kullanması teşvik edilmelidir.

8) Software and Data Integrity Failures (Yazılım ve Veri Bütünlüğü Hataları)

Software and Data Integrity Failures (Yazılım ve Veri Bütünlüğü Hataları) Nedir?

Bir yazılımın veya veri sisteminin beklenmedik şekilde değiştirilmesi ya da bozulması sonucu oluşan güvenlik açıklarıdır. Bu tür hatalar, yazılım veya veri üzerinde yetkisiz değişiklikler yapılmasına olanak tanır ve bu durum saldırganların sistemi ele geçirmesine, verileri bozmasına veya çalmasına yol açabilir. Örneğin, kötü amaçlı bir güncelleme, yetkisiz bir kullanıcı tarafından yapılan değişiklikler ya da güvenilmeyen kaynaklardan gelen yazılımlar, bu tür zafiyetlere neden olabilir.

Neden Kaynaklanır?

- Güncellemelerinin kaynağını doğrulamadan indirilmesi ve uygulanması, kötü amaçlı yazılımların sisteme sızmasına olanak sağlar.
- Veri ve yazılım üzerinde yetersiz erişim kontrolü, saldırganların sistem üzerinde değişiklik yapmasına olanak tanır.

Örnek Saldırı:

Bir yazılım güncelleme süreci sırasında, güncellemelerin bütünlüğü doğrulanmadan indirilmesi, saldırganların bu güncellemeleri manipüle ederek kötü amaçlı yazılım yüklemelerine imkân tanır. Özellikle otomatik güncellemeler, bu tür riskleri daha da artırır. Bu güncellemeler, daha önce güvenilen bir uygulama veya sistem üzerinden yayıldığında, kullanıcıların güvenini suistimal ederek geniş çapta etki yaratabilir.

Nasıl Önlenir?

- Yazılım ve veri güncellemeleri sırasında dijital imzalar veya benzer doğrulama mekanizmaları kullanarak, güncellemelerin beklenen kaynaktan olup olmadığı kontrol edilmelidir.
- Üçüncü taraf kütüphaneler için yalnızca güvenilir depoları kullanılmalı.
- Veri ve yazılım yedeklemelerini düzenli olarak oluşturulmalı ve bu yedeklemelerin güvenliğinin sağlanması gerekir.
- Yazılım ve veri erişiminin sıkı bir şekilde kontrol edilmesi gerekir. Yalnızca yetkili kişilerin erişim sağlamasına izin verilmelidir.

9) Security Logging and Monitoring Failures (Güvenlik Kayıt ve İzleme Hataları)

Security Logging and Monitoring Failures (Güvenlik Kayıt ve İzleme Hataları) Nedir?

Bir uygulamanın veya sistemin güvenlik olaylarını izleme ve kaydetme işlevlerinin yetersiz olması veya hatalı yapılandırılması nedeniyle ortaya çıkan güvenlik açıklarıdır. Bu tür zafiyetler, kötü amaçlı aktivitelerin tespit edilememesi, şüpheli hareketlerin fark edilememesi ve güvenlik olaylarına zamanında müdahale edilememesi gibi ciddi sonuçlara yol açabilir.

Neden Kaynaklanır?

- İzleme sistemlerinin yanlış yapılandırılması, uyarıların oluşturulmamasına veya önemli güvenlik olaylarının gözden kaçmasına neden olur.
- Kayıtların güvenliğinin sağlanmaması, saldırganların izlerini silmesine veya kayıtları manipüle etmesine olanak tanır.
- Güvenlik olaylarına anında yanıt verilmesi gereken durumlarda uyarı sistemlerinin olmaması, olayların geç fark edilmesine neden olabilir.

Örnek Senaryo:

Bir siber saldırı sırasında, başarısız oturum açma girişimlerinin ve sunucu hatalarının kaydedilmemesi veya bu kayıtların doğru analiz edilmemesi, saldırganların sisteme sızmasına ve zararlı aktivitelerini uzun süre fark edilmeden yürütmesine olanak tanır. Örneğin, bir saldırgan bir uygulamanın kimlik doğrulama sistemini aşarak erişim elde eder ve bu erişim uzun süre fark edilmezse, saldırgan sistem üzerinde daha fazla zararlı işlem gerçekleştirebilir.

Nasıl Önlenir?

- Güvenlik olaylarını izlemek ve kaydetmek için uygun araçlar seçilmeli ve bunların doğru yapılandırıldığından emin olunulmalıdır.
- Günlük kayıtlarını düzenli olarak inceleyerek şüpheli aktiviteleri tespit edilmelidir ve gerektiğinde hızlı müdahale edilmelidir.
- Güvenlik olaylarına karşı hızlı tepki vermek için etkili uyarı ve alarm sistemleri kullanın.
- Güvenlik politikalarını ve prosedürlerini düzenli olarak gözden geçirilmeli ve güncellemeler zamanında yapılmalıdır.

10) Server-Side Request Forgery (Sunucu Taraflı İstek Sahteciliği - SSRF)

Server-Side Request Forgery (Sunucu Taraflı İstek Sahteciliği - SSRF) Nedir?

Bir saldırganın hedef sunucuyu, sunucunun erişebileceği herhangi bir iç veya dış kaynağa istek göndermesi için kandırdığı bir web güvenlik açığıdır. Bu güvenlik açığında saldırgan, sunucunun kontrol ettiği bir parametreyi, genellikle URL veya IP adresini, manipüle ederek sunucuya sahte istekler gönderir. SSRF saldırıları, genellikle bir uygulamanın dahili ağlarına erişmek, hassas verilere ulaşmak, sistemlerin çalışmasını engellemek veya sunucuyu manipüle etmek için kullanılır.

Nasıl Çalışır?

SSRF, saldırganın hedef sunucunun URL'lere istek göndermesine neden olacak şekilde sunucuyu yanıltması ile çalışır. Bu saldırılar, sunucunun istemci tarafından sağlanan bir girdiye güvenmesi ve bu girdiyi doğrulamadan işleme alması sonucunda gerçekleşir. SSRF saldırıları sırasında, saldırgan, sunucuya yetkisiz istekler gönderebilir ve bu istekler sunucunun dahili sistemlerine, özel ağlarına veya başka harici kaynaklara yönlendirilir.

Örnek Senaryo:

Bir web uygulaması, kullanıcılardan aldığı bir URL ile resim indirip gösteren bir fonksiyon içeriyor olabilir. Saldırgan bu URL parametresine yerel ağı hedefleyen bir IP adresi veya sunucuya zarar verebilecek bir URL girerek sunucuyu bu isteği gerçekleştirmeye zorlayabilir. Bu sayede saldırgan, sunucuya zarar verebilir, ağ içi servislere erişebilir veya sunucudan veri sızdırabilir.

Neden Kaynaklanır?

- Kullanıcı tarafından sağlanan girdilerin doğrulanmaması, sunucunun potansiyel olarak tehlikeli kaynaklara istek göndermesine neden olur.
- SSRF, sunucunun normalde erişememesi gereken dahili ağlara ve güvenlik duvarlarının arkasındaki hizmetlere erişim sağlamak için kullanılabilir.
- Sunucunun, girişleri güvenilir bir şekilde doğrulayamaması veya temizleyememesi, SSRF saldırılarına yol açabilir.

Nasıl Önlenir?

- Tehlikeli girdilere karşı filtreler ve kısıtlamalar uygulanmalıdır.
- Sadece belirli güvenilir URL'lere veya IP adreslerine istek göndermeye izin verilmelidir.
- Sunucudan giden istekleri sınırlamak için güvenlik duvarlarının ve ağ ayarlarının yapılandırılması gerekmektedir.
- Sunucunun, clientlara detaylı hata mesajları veya gereksiz bilgi içeren yanıtlar göndermemesini sağlamak gerekmektedir.

REFERANSLAR

https://owasp.org/Top10/

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

https://www.reflectiz.com/blog/owasp-top-ten-2023/

https://certera.com/blog/mitigating-the-owasp-top-10-vulnerabilities/