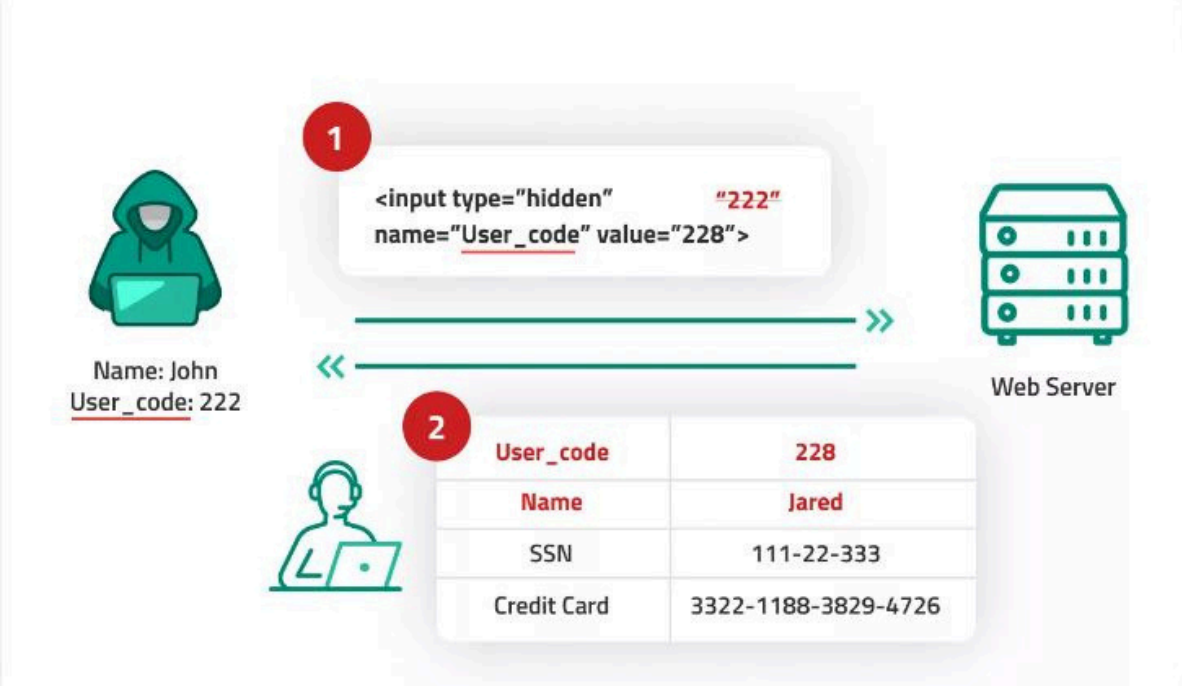
OWASP TOP 10

Owasp (Open Web Application Security Project), başta web uygulamaları olmak üzere yazılım güvenliklerii artırmak amacıyla 2003 yılında proje hayata geçirilmiştir. Owasp genel olarak gönüllülerden oluşan bir projedir. Dünya çapında yayılmış olan sektör gönüllüleri ile birlikte ortak bir çalışma yapılarak düzenli bir şekilde liste güncellenmektedir. Liste güncellenmesi, topluluk gönüllülerinin ankete teşvik edilerek verilerin Github reposunda toplanarak bir ortalama alınması ve bu ortalama ile de listenin güncellenmesi ve yenilenmesini sağlar. Bu sayede sektör çalışanları bilinçlenir ve daha güvenli bir web, yazılım ortamı geliştirmeye çalışırlar.

A01:2021 – Broken Access Control: Erişim doğrulamanın düzgün bir şekilde yapılmamasının yol açtığı bir zafiyettir.
Örnek Senaryo: Yetkim olmamasına rağmen başka bir kullanıcının hesabına giriş yapıyor olmam bu zafiyeti doğurur. Genellikle tahmin edilebilir veya kolaylıkla kırılabilir şifreler, ulaşılabilir oturum cookie'leri ve brute force saldırıları sonucu bu zafiyet sömürülebilir.
Broken Access Control Nedenleri: Zayıf oturum yönetimi, kullanıcıların yetkilerinin doğru konfigüre edilmemesi gibi nedenlerden ortaya çıkmaktadır.
Broken Access Control Önleme: Oturumları düzenli olarak yenilemek, veri doğrulama (input validation) yapmak.

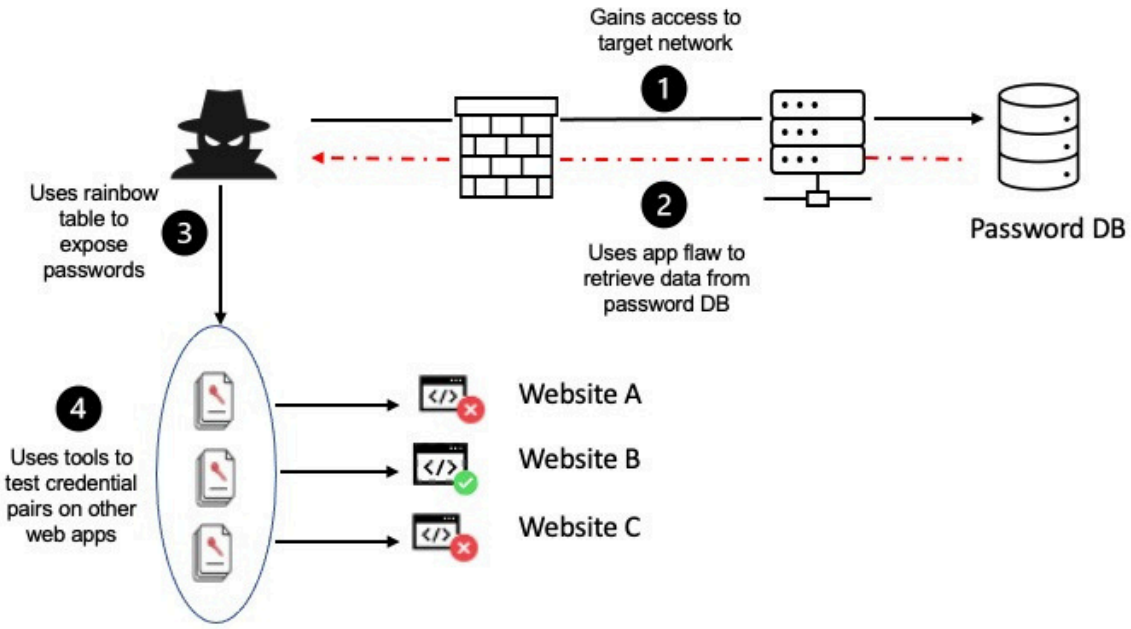


A02:2021 – Cryptographic Failures: Şifreler, kredi kartları ve kişisel veriler gibi bilgilerin şifrelenmeden veyahut güvensiz algoritmalar ile şifrelenerek transfer edilmesi ya da sunucu üzerinde saklanması sonucu ortaya çıkan bir zafiyettir.

Örnek Senaryo: Bir uygulama kredi kartı bilgilerini otomatik database encryption kullanarak kendi bünyesinde tutuyor fakat lazım olduğu zaman kullanıcıya decyrpt edilip veriliyor sql injection açığını kullanarak database içinden almak istediğimizde kredi kartı bilgisini düz metin olarak alabiliyoruz.

Cryptographic Failures Nedenleri: Zayıf şifreleme algoritmalarının kullanımı, anahtar yönetimi hataları gibi nedenlerden ortaya çıkmaktadır.

Cryptographic Failures Önleme: Güçlü şifreleme algoritmaları kullanmak, anahtarlı güvenli şekilde depolamak, şifreleme standartlarına uygun uygulamalar geliştirmek, kullanmak.



A03:2021 – Injection: Bu zafiyet genellikle kullanıcıdan alınması gereken verilerin düzgün yapılandırılmaması ve bu sebepten saldırganın uygulama içerisine kendi komutlarını enjekte etmesine, çalıştırmasına yol açan bir zafiyettir.

İnjectin Türleri;

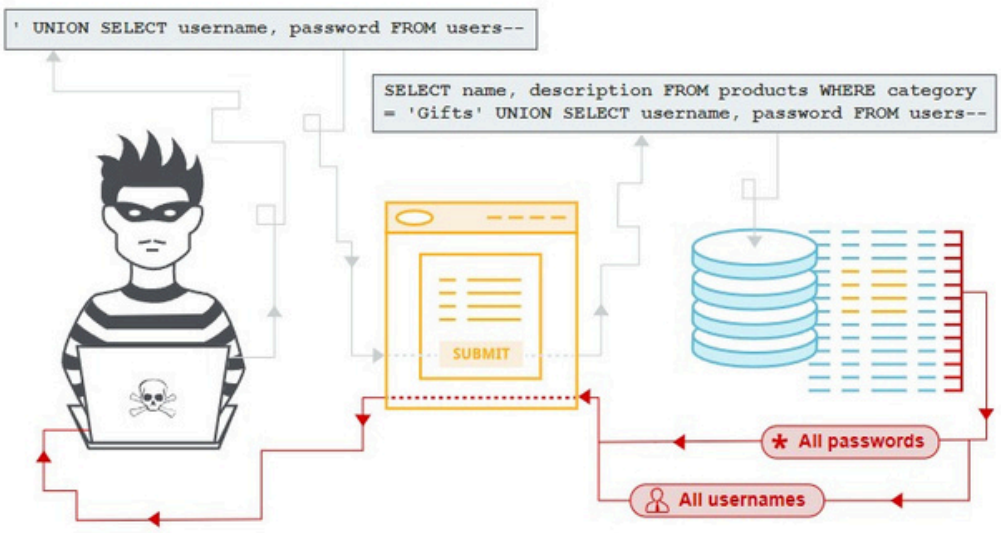
→**SQL İnjection :** Bir web uygulamasının SQL sorgularında kullanıcı tarafından sağlanan temizlenmiş verileri doğrudan görebilmeyi içeren kritik bir saldırı yöntemidir. Güvenlik açığı, görülmemesi gereken verilerin görüntülenmesine yol açar.

→**NoSQL İnjection:** Saldırganların kimlik doğrulama kontrollerini atlamasına verileri çalmasına ve uygulama üzerinde kontrol sağlamasına imkan veren bir açıktır. NoSQL veri tabanları ilişkisel oldukları için SQL veri tabanlarından bu şekilde ayırt edilir ve standart sorguları desteklemez kendine has sorgulama şekilleri vardır.

→ **OS command injection:** Saldırganın zafiyetli uygulamayı çalıştıran sunucuda gönlüne göre terminal, işletim sistemi komutları yürütebilme açığıdır.

Injection Türlerinin Nedenleri: Veri doğrulama (İnput validation) yapılmaması, sisteme girilen parametrelerin sorgulanmaması İnjection zafiyet kategorisinin ortaya çıkmasına neden olur.

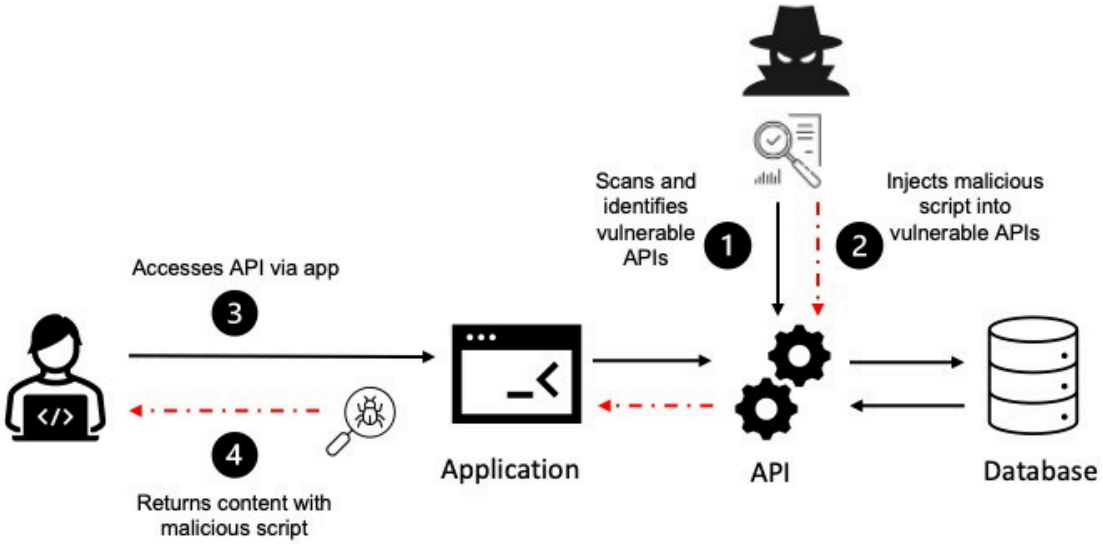
Injection Türlerini Önleme: Girilen parametreleri sorgulamak, veri doğrulama (input validation) yapmak enjeksiyonlara karşı önleme sağlayabilir.



A04:2021 – Insecure Design: 2021 yılında listeye girmiş bir zafiyettir. Eksik veya etkisiz kontrol tasarımı olarak ifade edebileceğimiz bir zafiyettir.

Insecure Design Nedenleri: Güvenlik gereksinimlerinin tasarım aşamasında göz önünde bulundurulmaması, zayıf oturum yönetimi, yetersiz hata yönetimi gibi nedenlerden bu açık ortaya çıkabilir.

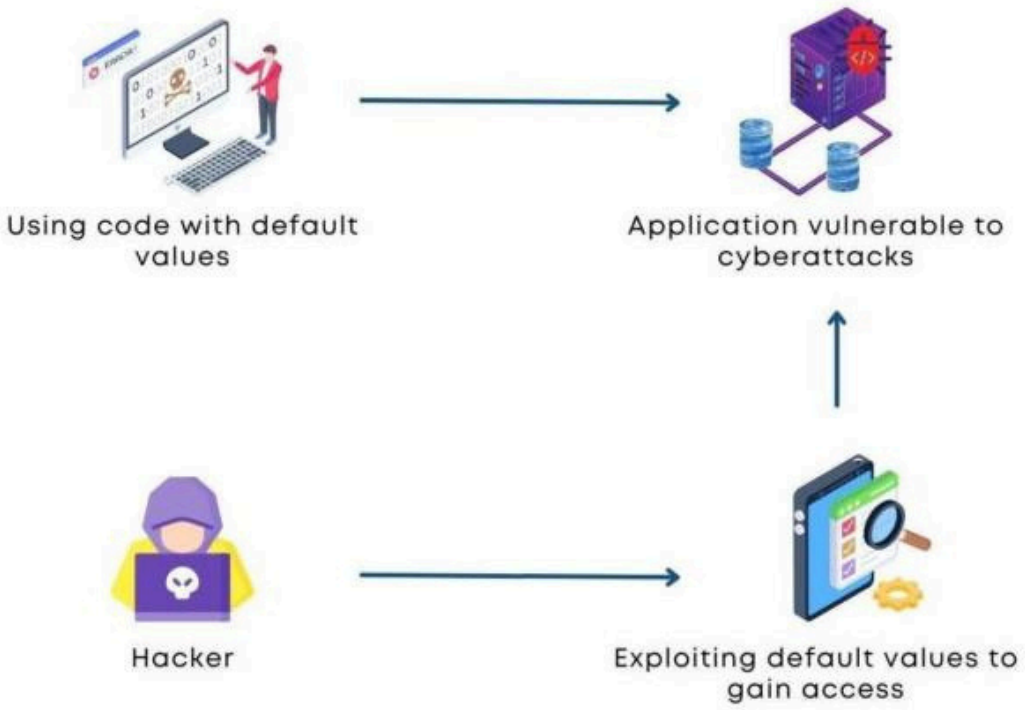
Insecure Design Önleme: Güvenlik gereksinimlerinin tasarım aşamasında göz önünde bulundurulması ve düzenlenmesi, güvenli mimari kalıpların kullanılması, konun bilirkişisi yani güvenlik uzmanlarıyla işbirliği yapılması bu açığı önlemeye yardımcı olabilir.



A05:2021 – Security Misconfiguration: Uygulamaların yanlış yapılandırılması veya güvenlik yamalarının yapılmaması gibi nedenlerle oluşan bir güvenlik açığıdır.

Security Misconfiguration Nedenleri: Kurulumdan sonra default admin hesaplarının aktif bırakılması, hata mesajlarında uygulama veya sunucu ile ilgili kritik bilgilerin verilmesi gibi düzgün yapılandırılmamış güvenlik gerekliliklerinin sonucu ortaya çıkan bir zafiyettir.

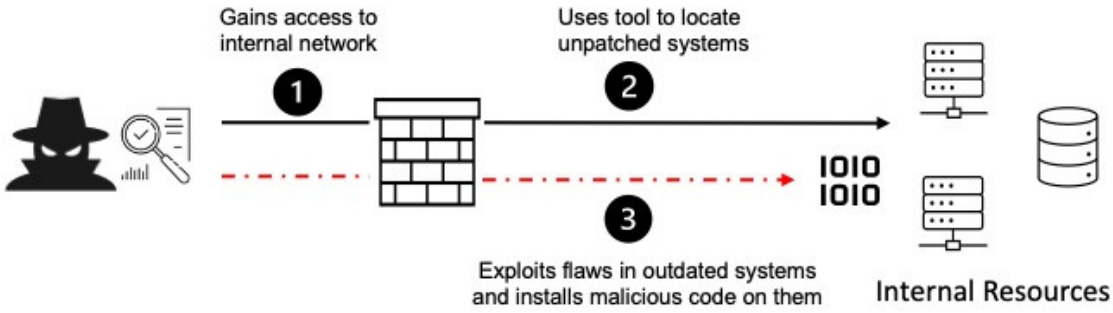
Security Misconfiguration Önleme: Güvenlik yapılandırmalarının düzenli bir şekilde gözden geçirilmesi, güvenlik güncellemelerinin zamanında yapılması, güvenlik araçları kullanılarak sistemin güvenliğinin düzenli bir şekilde test edilmesi bu açığın oluşmasını önlemeye yardımcı olabilir.



A06:2021 – Vulnerable and Outdated Components: Adından yola çıkarak anlaşılabilen, zafiyet barındıran uzunca güncellenmemiş uygulamaların veya yardımcı uygulamaların kullanılmasından kaynaklı ortaya çıkan bir zafiyet türüdür.

Vulnerable and Outdated Components Nedenleri: Sistemde kullanılan uygulamaların güncellemesinin vaktinde yapılmaması, uygulamaların aktif güvenlik açıklarının araştırılmadan kullanılması bu zafiyetin ortaya çıkmasına neden olabilir.

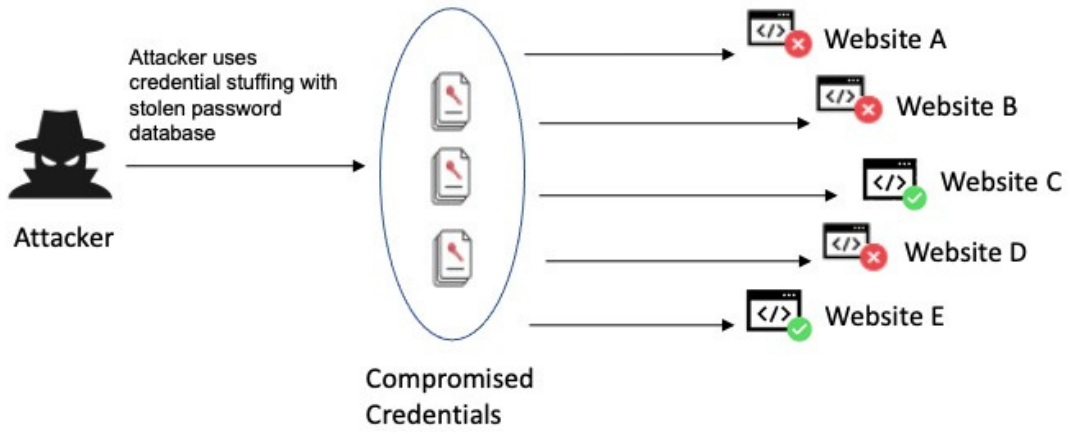
Vulnerable and Outdated Components Önleme: Kullanılan uygulamaların güncellemesini, güvenlik yamalarını vaktinde yapılması, uygulamaların güvenlik açıklarının aktif olarak takip edilmesi ve ona göre aksiyon alınması bu açığın ortaya çıkmaması için alınabilecek önlemlerden bazılarıdır.



A07:2021 – Identification and Authentication Failures: Yetkilendirme ve kimlik doğrulama esnasında yeterli güvenlik önlemi alınamaması veya alınmaması durumunda ortaya çıkan zafiyetlerdir.

Identification and Authentication Failures Nedenleri: Zayıf parolaların kullanılması, oturumların uzun süre açık bırakılması, kalması bu açığın başlıca nedenleridir.

Identification and Authentication Failures Önleme: Güçlü parola politikaları uygulamak, çok faktörlü kimlik doğrulama (two-factor authentication) kullanma, oturumları düzenli olarak yenileme, oturum süresi belirleme gibi önlemler bu açığın ortaya çıkmasını engelleyebilir.

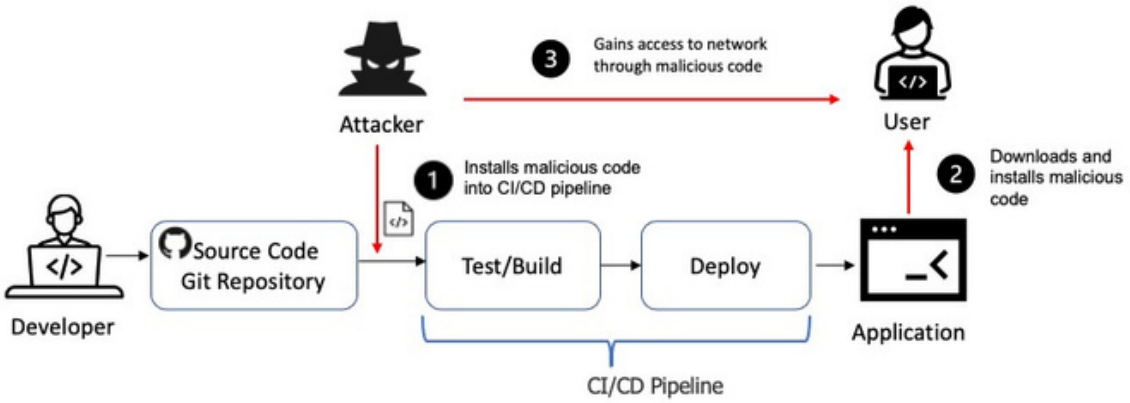


A08:2021 – Software and Data Integrity Failures: 2021 yılında listeye eklenen bir zafiyet türüdür. Bu zafiyetin ortaya çıkışı ise veri bütünlüğüne dair bir koruma sağlamayan kod ve altyapı ile ilgilidir.

Örnek Senaryo: onaylanmamış kullanıcılardan gelen güncelleme dosyalarını kabul eden bir uygulama olduğunu varsayalım. Saldırganlar uygulamaya yükledikleri zararlı dosyayı bir güncelleme dosyası gibi gösterip sisteme kimsenin anlamayacağı şekilde yükleyebilir**.**

Software and Data Integrity Failures Nedenleri: Güvenilmeyen kaynaklardan gelen verilerin doğrulama yapılmadan kullanılması, zararlı kodların sisteme bulaşması.

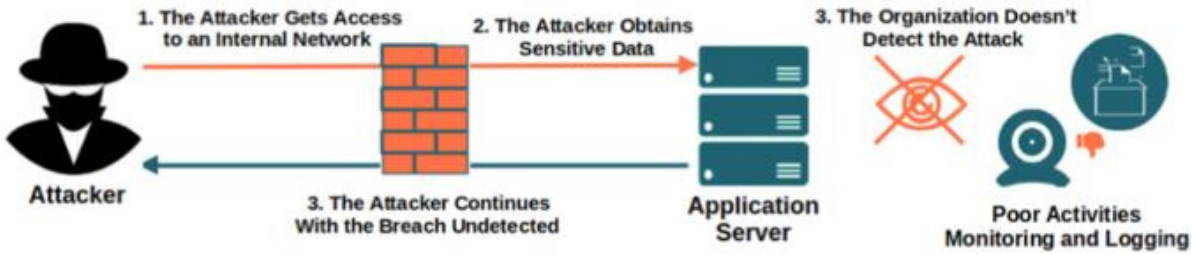
Software and Data Integrity Failures Önleme: Kod imzalama, şifreleme, güvenilir kaynakları kullanma, sisteme yüklenen kaynakların kontrolü gibi önlemler ile bu açığın önüne geçilebilir.



A09:2021 – Security Logging and Monitoring Failures: Güvenlik olaylarının izlenmesi ve kaydedilmesi işlemlerinin yetersizliği veya hatalı yapılandırılmaları sonucu ortaya çıkan bir zafiyet türüdür.

Security Logging and Monitoring Failures Nedenleri: Yetersiz günlük kayıt yapılandırmaları, olayların analiz edilmemesi, gibi nedenler bu zafiyet türünün ortaya çıkmasına yol açabilir.

Security Logging and Monitoring Failures Önleme: Detaylı ve düzenli günlük kayıt yapılandırmaları oluşturmak, oluşturulan bu günlükleri düzenli analiz etmek, SIEM (Security Information and Event Management) güvenlik sistemlerini kullanmak bu açığın ortaya çıkmasını önleyebilir.



A10:2021 – Server-Side Request Forgery (SSRF): Sunucuya gönderilen isteklerin herhangi bir onaydan geçmeden direkt sunucuya iletilmesi durumunda ortaya çıkan bir zafiyettir.

Örnek Senaryo: sunucuyu şaşırtıp localhost ve port vererek etc/passwd gibi çok kritik bilgilerin yer aldığı dosyalara erişim sağlanabilir.

Server-Side Request Forgery (SSRF) Nedenleri: Kullanıcı girdilerinin yeterince doğrulanmaması, güvenli url işlemi yapılmaması gibi nedenlerden bu zafiyet ortaya çıkabilir.

Server-Side Request Forgery (SSRF) Önleme: Güvenli url yapılandırılması, kullanıcı girdilerini kara listeye alarak sınırlandırmak, sadece güvenilir kaynaklardan gelen istekleri kabul etmek gibi önlemler bu açığın ortaya çıkmasının önüne geçme konusunda yardımcı olabilir.