infinitum IT
Power of integrated Security

Remote Code Execution

infinitumitlabs

Remote Code Execution

Remote Code Execution (Uzak Kod Çalıştırma) (RCE), bir saldırganın bir bilgisayar veya sunucuda yer alan bir uygulama veya işletim sistemi üzerinde kötü niyetli betik (script) çalıştırmasına izin veren bir güvenlik açığıdır. Saldırgan, sistemi ele geçirmek ve kontrol etmek için yeterli olan geniş bir erişim sağlayabilir.

Bu güvenlik açığı genellikle uygulamadaki bir güvenlik açığından veya yanlış yapılandırılmış bir sunucudan kaynaklanır. RCE saldırıları, bir saldırganın hedef sistemde kötü amaçlı kod yüklemesine ve çalıştırmasına olanak tanır. Bu saldırgan, hedef sistemde çalışan herhangi bir işlemin kontrolünü ele geçirebilir, dosyaları okuyabilir, yazabilir veya silebilir, sistemi yeniden başlatabilir veya çökertebilir ve hedef sistemdeki diğer kaynakları kullanabilir.

RCE saldırısı, son yıllarda en yaygın kullanılan saldırı tekniklerinden biri haline geldi. Bu tür saldırılara karşı korunmak için uygulama ve sunucuların güncel tutulması, güvenlik açıklarına karşı düzenli olarak taranması, siber güvenlik önlemlerinin alınması ve kullanıcıların bilgilendirilmesi gerekmektedir. Ek olarak, uygulama geliştiricileri, kodlarını güvenli bir şekilde yazmak ve test etmek için özel çaba göstermelidir.

Remote Code Execution Nasıl Çalışır?

Remote Code Execution (Uzak Kod Çalıştırma) saldırıları, bir saldırganın hedef sisteme kötü amaçlı kod enjekte etmesi ve bu kodun çalıştırılmasını sağlaması yoluyla gerçekleştirilir. Bu kod, saldırganın hedef sistemi ele geçirmesine ve kontrol etmesine olanak tanıyacak şekilde tasarlanmıştır.

RCE saldırıları genellikle şu adımları içerir:

1. Zafiyet keşfi: Saldırgan, hedef sistemdeki bir uygulamanın veya sunucunun güvenlik açıklarını arar. Bu açıklar genellikle, bir uygulamanın girdilerini doğru bir şekilde doğrulamamasından veya yanlış yapılandırılmış bir sunucudan kaynaklanır.

2. Exploit yazma: Saldırgan, keşfettiği güvenlik açığına özel bir exploit yazarak, hedef sisteme kötü amaçlı kod enjekte eder. Bu kötü amaçlı kod, hedef sistemin belleğine yerleştirilir ve çalıştırılır.

3. Shell almaya çalışma: Exploit, saldırganın hedef sisteme erişmesine ve komutları yürütmesine olanak tanıyan bir kabuk veya oturum açar. Saldırgan, bu kabuk(shell) aracılığıyla hedef sistemi kontrol edebilir.

4. Kontrol: Saldırgan, hedef sistemi kontrol ederek sistemi istediği şekilde kullanabilir. Bu dosyaları okuyup yazmak, sistem kaynaklarını kullanmak, kullanıcı hesaplarına erişmek,

yeni kötü amaçlı yazılım yüklemek veya hedef sistemde diğer kötü amaçlı faaliyetler gerçekleştirmek gibi şeyleri içerebilir.

Remote Code Execution Türleri Nelerdir?

Remote Code Execution (Uzak Kod Çalıştırma) saldırıları, farklı şekillerde gerçekleştirilebilir. Aşağıda RCE saldırılarının bazı yaygın türleri açıklanmaktadır:

1. Command Injection: Bu tür saldırılar, bir uygulamanın girdilerini kötü niyetli komutlarla manipüle ederek gerçekleştirilir. Bu sayede saldırgan hedef sisteme komutlar gönderebilir ve istediği kodları çalıştırabilir.

2. File Inclusion: Bu tür saldırılar, bir uygulamanın web sayfalarında yer alan dinamik içeriklerdeki URL'leri manipüle ederek gerçekleştirilir. Bu sayede saldırgan hedef sisteme kötü amaçlı bir kod yükleyebilir ve çalıştırabilir.

3. Deserialization: Bu tür saldırılar, bir uygulamanın girdilerini seri hale getirerek veya tersine çevirerek gerçekleştirilir. Bu sayede saldırgan hedef sisteme kötü amaçlı bir seri nesne gönderebilir ve istediği kodları çalıştırabilir.

4. Server-Side Request Forgery (SSRF): Bu tür saldırılar, bir uygulamanın dış kaynaklarına yaptığı istekleri manipüle ederek gerçekleştirilir. Bu sayede saldırgan hedef sisteme kötü amaçlı bir istek gönderebilir ve istediği kodları çalıştırabilir.

5. Code Injection: Bu tür saldırılar, bir uygulamanın koduna kötü niyetli kod enjekte ederek gerçekleştirilir. Bu sayede saldırgan hedef sisteme istediği kodu yükleyebilir ve çalıştırabilir.

Bu tür saldırıların yaygın olmasının nedeni, birçok uygulamanın güvenlik açıklarına sahip olması veya doğru şekilde yapılandırılmamasıdır.

Remote Code Execution Güvenlik Açıkları Nelerdir?

Remote Code Execution (Uzak Kod Çalıştırma) güvenlik açıkları, bir uygulamanın girdilerinin doğru şekilde işlenmemesi veya doğru şekilde doğrulanmaması nedeniyle oluşur. Aşağıda, RCE güvenlik açıklarının bazı yaygın türleri açıklanmaktadır:

1. Injection vulnerabilities: Bu tür güvenlik açıkları bir uygulamanın girdilerinin doğru şekilde doğrulanmadan veya işlenmeden doğrudan yürütülmesi nedeniyle oluşur. Örneğin, bir kullanıcının arama sorgusu gibi bir girdiye kötü niyetli bir komut eklemesi ve uygulamanın bu komutu doğrudan çalıştırması durumunda saldırgan hedef sisteme kötü amaçlı kod enjekte edebilir.

2. File upload vulnerabilities: Bu tür güvenlik açıkları, bir uygulamanın dosya yükleme işlevselliğinin güvenli şekilde yapılandırılmaması nedeniyle oluşur. Örneğin, bir kullanıcının kötü amaçlı bir betik (script) içeren bir dosyayı yükleyerek hedef sisteme erişim sağlaması durumunda saldırgan bu betiği (scripti) hedef sistemde çalıştırabilir.

3. Deserialization vulnerabilities: Bu tür güvenlik açıkları, bir uygulamanın girdilerinin seri hale getirilerek veya tersine çevrilerek kullanılması nedeniyle oluşur. Örneğin, bir kullanıcının bir seri nesne içeren bir girdi göndermesi durumunda saldırgan bu seri nesneyi manipüle edebilir ve hedef sisteme kötü amaçlı kod enjekte edebilir.

4. Server-Side Request Forgery (SSRF) vulnerabilities: Bu tür güvenlik açıkları, bir uygulamanın dış kaynaklarına yapılan istekleri kontrol etmeyi veya sınırlandırmayı doğru şekilde yapılandırmaması nedeniyle oluşur. Örneğin, bir kullanıcının bir hedef sistem için kötü amaçlı bir URL göndermesi durumunda saldırgan bu URL aracılığıyla hedef sistemdeki kodları çalıştırabilir.

5. Code injection vulnerabilities: Bu tür güvenlik açıkları, bir uygulamanın koduna kötü niyetli kod enjekte edilmesi nedeniyle oluşur. Örneğin, bir kullanıcının bir girdiye bir SQL sorgusu veya JavaScript kodu eklemesi durumunda saldırgan bu kodu manipüle edebilir ve hedef sisteme kötü amaçlı kod enjekte edebilir.

Bu tür güvenlik açıklarının önlenmesi için uygulama geliştiricileri kullanıcı girdilerini doğru şekilde doğrulamak, sınırlamak ve işlemek için gereken tüm önlemleri almalıdırlar.

Remote Code Execution Örnekleri ve Saldırı Senaryoları

Remote Code Execution (Uzak Kod Çalıştırma) saldırıları, farklı senaryolarda gerçekleştirilebilir. Bu saldırıların birçoğu, web uygulamalarına yönelik gerçekleştirilen saldırılardır. Aşağıda, Remote Code Execution saldırılarına örnekler ve bu saldırıların nasıl gerçekleştirildiğine dair senaryolar bulunmaktadır:

1. Command Injection Saldırısı: Bu tür saldırılar, uygulamanın kullanıcı girdilerini doğru şekilde işlemediği durumlarda gerçekleşir. Saldırganlar, uygulamaya özel bir karakter dizisi girerek uygulamanın komut istemine saldırgan tarafından yazılan kodları yürütmesini sağlar. Bu şekilde saldırgan, uygulamanın sunucusunda kod yürütebilir. Örneğin, bir e-ticaret sitesindeki arama kutusuna "'; rm -rf /" gibi bir karakter dizisi yazılarak, sunucuda tüm dosyaların silinmesi sağlanabilir.

2. File Inclusion Saldırısı: Bu saldırılar, web uygulamalarında kullanılan dosyaların doğru şekilde işlenmemesi durumunda gerçekleşir. Saldırganlar, uygulamanın dosya yolu girdilerine özel bir karakter dizisi girerek, uygulamanın sunucusunda kod yürütmesini sağlar. Bu şekilde saldırgan, sunucuda bulunan diğer dosyalara erişebilir. Örneğin, bir

blog uygulamasında yorumlar bölümünde bulunan dosya yoluna "../../../../../etc/passwd" gibi bir karakter dizisi yazılarak, sunucuda bulunan parolaların çalınması sağlanabilir.

3. Deserialization Saldırısı: Bu saldırılar, uygulamanın doğru şekilde seri hale getirilmemiş nesneleri işlememesi durumunda gerçekleşir. Saldırganlar, özel bir karakter dizisi içeren bir seri hale getirilmemiş nesne gönderirler. Uygulama bu nesneyi işlerken, saldırganın belirlediği kod yürütülür. Bu şekilde saldırgan, uygulamanın sunucusunda kod yürütebilir. Örneğin, bir web uygulamasında, bir kullanıcının profil bilgileri doğru şekilde seri hale getirilmediğinde, saldırgan özel bir karakter dizisi içeren bir nesne göndererek, sunucuda bulunan diğer dosyalara erişebilir.

4. RFI/LFI Saldırıları: Bu saldırılar, web uygulamalarındaki dosya yolu veya dosya adı girdilerini doğru şekilde işlememesi durumunda gerçekleşir. Saldırganlar, özel bir karakter dizisi içeren bir URL veya dosya yolu gönderirler. Uygulama bu URL veya dosya yolu girdisini işlediğinde, sunucuda bulunan diğer dosyalara erişebilirler. RFI (Remote File Inclusion) saldırılarında saldırgan, sunucuda bulunan bir uzak dosyayı uygulamaya dahil edebilir. LFI (Local File Inclusion) saldırılarında ise saldırgan, sunucuda bulunan bir yerel dosyayı uygulamaya dahil edebilir. Örneğin, bir web uygulamasında bir kullanıcının profil resmi olarak belirtilen URL saldırgan tarafından kontrol edilen bir URL olarak değiştirildiğinde saldırgan uygulamanın sunucusunda kod yürütebilir.

5. XXE (XML External Entity) Saldırıları: Bu saldırılar, XML verilerini işleyen uygulamaların, dış kaynaklı XML dosyalarını doğru şekilde işlememesi durumunda gerçekleşir. Saldırganlar, özel bir karakter dizisi içeren bir XML dosyası gönderirler. Bu XML dosyası, saldırganın belirlediği kodları yürütmek için sunucuda bulunan dosyalara erişebilir. Örneğin, bir web uygulamasında bir XML dosyası gönderildiğinde, bu dosyada bulunan özel bir karakter dizisi sunucuda bulunan diğer dosyalara erişmek için kullanılabilir.

Remote Code Execution Saldırılarından Nasıl Korunulur?

1. Güvenli kod yazma: Bir uygulamanın saldırılara karşı daha dayanıklı olmasını sağlar. Özellikle kullanıcı girdilerinin doğrudan yürütüldüğü veya başka bir sistemle etkileşime geçtiği durumlarda önemlidir. Uygulama geliştiricileri tarafından bilinmeli, uygulanmalı ve bu konuda bilinçlendirilmelidirler.

2. Kullanıcı girdilerinin doğru şekilde doğrulanması: Bir uygulamanın girdileri doğru şekilde doğrulanmazsa, saldırganlar bu girdileri kullanarak kod enjekte edebilirler. Kullanıcı girdileri doğru şekilde doğrulanarak, uygulamanın güvenliği artırılabilir.

3. Güvenli dosya yükleme işlevselliği: Bir uygulamanın dosya yükleme işlevselliği güvenli şekilde yapılandırılmalıdır. Bu dosyaların yalnızca belirli türlerde olmasını ve boyutlarının belirli bir sınırı aşmamasını sağlayarak yapılabilir.

4. Güvenli veri serielleştirme: Bir uygulamanın veri serielleştirme işlevselliği güvenli şekilde yapılandırılmalıdır. Bu verilerin yalnızca belirli bir türde olmasını ve güvenliğini sağlamak için doğru şekilde sınırlandırılmasını içerebilir.

5. Dış kaynaklara yapılan isteklerin kontrol edilmesi: Uygulama geliştiricileri, dış kaynaklara yapılan istekleri kontrol etmelidir. Bu özellikle hedef sistemin özelliklerine erişmek için kullanılan diğer sistemlerle etkileşimli uygulamalarda önemlidir.

6. Kod enjekte edilmesini engelleme: Bir uygulama, kod enjekte edilmesini engellemek için gereken tüm önlemleri almalıdır. Bu özellikle kullanıcı girdilerinin doğrudan yürütüldüğü durumlarda önemlidir.

7. Güncellemeleri takip etmek: Uygulama geliştiricileri, uygulamanın kullanıldığı platformların güncellemelerini takip etmeliler ve gerektiğinde güncellemeleri uygulamalıdırlar.

8. Firewall kullanımı: Uygulamanın sunucusunda firewall kullanmak, gelen ve giden trafiği izleyerek saldırıları engellemeye yardımcı olabilir.

9. Saldırı tespit ve önleme yazılımları: Uygulamanın sunucusunda saldırı tespit etmek ve önlemek için özel yazılımlar kullanılabilir. Bu tür yazılımlar, uygulamanın sunucusuna yönelik saldırıları tespit edebilir ve saldırıların gerçekleşmesini engelleyebilir.

10. Yetkilendirme ve kimlik doğrulama: Uygulamanın kullanıcılarını doğru şekilde yetkilendirmesi ve kimlik doğrulaması yapması önemlidir. Bu, kullanıcıların yalnızca yetkileri çerçevesinde işlemler yapmalarını sağlar ve yetkisiz erişimleri engeller.

11. Güvenlik testleri: Uygulamanın güvenlik testleri yapılmalı ve olası zayıf noktalar belirlenmelidir. Bu testler, uygulama geliştiricilerinin potansiyel güvenlik açıklarını tespit etmelerine ve bunları gidermelerine yardımcı olur.

12. Güvenlik bilinci: Son olarak, uygulama geliştiricileri ve kullanıcılar, güvenlik konusunda bilinçli olmalıdırlar. Bu, uygulamanın güvenliği için gerekli olan adımların tam olarak uygulanmasını sağlar. Ayrıca, kullanıcılar da güvenli internet kullanımı konusunda bilinçli olmalıdırlar ve güvenli internet alışkanlıkları edinmelidirler.

Remote Code Execution Engellemek ve Tespit Etmek İçin Kullanılan Bazı Araçlar

Snyk

Snyk, açık kaynaklı yazılımların güvenliğini sağlamak için kullanılan bir bulut tabanlı güvenlik platformudur. Bu platform, uygulamanın kodunu tarar ve güvenlik açıklarını tespit eder. Ayrıca, uygulamanın kullanılan kütüphaneleri de tarar ve bunların güvenlik açıklarını tespit eder.

RCE zafiyeti için Snyk, uygulamanın kodunu tarar ve RCE saldırılarına yol açabilecek kod parçalarını tespit eder. Uygulamanın kullandığı açık kaynaklı kütüphanelerin de güvenlik açıklarını tarar ve bunları da tespit eder. Bu sayede, uygulamanın kullanılan kütüphanelerindeki güvenlik açıkları da kapatılabilir.

Snyk, uygulamanın güvenlik açıklarını tespit ettiğinde, uygulama geliştiricilerine raporlar sağlar. Bu raporlar, güvenlik açıklarının nerede olduğunu, nasıl sömürülebileceğini ve nasıl çözülebileceğini açıklar. Bunun yanı sıra uygulama geliştiricilerine güvenli yazılım geliştirme sürecinde rehberlik eder ve güvenli yazılım geliştirme konusunda öneriler sunar.

Checkmarx

Checkmarx, uygulama güvenliği testi yapmak için kullanılan bir bulut tabanlı bir yazılım analiz aracıdır. Uygulamanın kaynak kodunu tarar ve potansiyel güvenlik açıklarını tespit eder. Bu araç, statik kod analizi, dinamik testler, tarama sonrası analiz ve hata bulma gibi birçok özellik sunar.

Uygulamanın kodunu tarar ve RCE saldırılarına yol açabilecek kod parçalarını tespit eder. Checkmarx, RCE zafiyeti tespit ettiğinde, uygulamanın hangi bölümlerinin etkilendiğini, nasıl sömürülebileceğini ve nasıl çözülebileceğini raporlar.

Checkmarx, uygulama geliştiricilerine güvenli kodlama uygulamaları konusunda rehberlik eder. Ayrıca, uygulama geliştiricilerine güvenli yazılım geliştirme sürecinde öneriler sunar ve uygulama güvenliği konusunda eğitim verir.

SonarQube

SonarQube, açık kaynak kodlu bir yazılım kalite kontrol platformudur. Uygulama güvenliği için kullanılan birçok aracı içinde barındırır. Uygulamanın güvenliği konusunda birçok güvenlik açığı tespit edebilir. Bunlar arasında, RCE zafiyeti de bulunmaktadır.

Bu araç uygulamanın kaynak kodunu tarar ve RCE saldırılarına neden olabilecek kod parçalarını tespit eder. Bu kod parçalarını tanımlayarak, uygulama geliştiricilerinin RCE zafiyetlerini gidermelerine yardımcı olur. Ayrıca uygulama geliştiricilerine güvenli yazılım geliştirme konusunda tavsiyeler sunar.

SonarQube, aynı zamanda statik kod analizi, dinamik testler, kod kapsamı raporları, test kapsamı raporları gibi birçok özellik sunar. Bu özellikler, uygulama geliştiricilerinin uygulamanın güvenliğini artırmalarına yardımcı olur.

OWASP ZAP

OWASP ZAP (Zed Attack Proxy), açık kaynak kodlu bir web uygulama güvenliği tarayıcısıdır. Uygulama güvenliği için birçok araç içinde barındırır. RCE zafiyeti tespiti ve önlenmesinde de etkilidir.

Web uygulamalarının RCE zafiyetlerini tespit etmek için çeşitli yöntemler kullanır. Web uygulamalarının URL'lerini, formlarını ve girdilerini analiz ederek, RCE zafiyetlerini tespit edebilir. Ayrıca web uygulamalarındaki SQL enjeksiyonu, XSS, CSRF, ve diğer birçok güvenlik açıklarını da tespit edebilir.

Otomatik olarak tarama yapabileceği gibi, manuel olarak da kullanılabilir. Uygulama geliştiricilerinin RCE zafiyetlerini tespit edip, uygulamanın güvenliğini artırmalarına yardımcı olur. Aynı zamanda uygulamanın güvenliğini test etmek ve güvenlik açıklarını gidermek için birçok özellik sunar.

OWASP ZAP, birçok platformda çalışabilir ve birçok dilleri destekler. Birçok eklenti ve entegrasyon seçeneği de sunar. OWASP ZAP, uygulama güvenliği için önemli bir araçtır ve uygulama geliştiricileri tarafından aktif olarak kullanılmaktadır.

Fortify

Fortify, uygulama güvenliği açısından kritik olan RCE zafiyetlerini tespit etmek ve önlemek için kullanılan bir statik kod analizi aracıdır. Fortify, uygulamanın kaynak kodunu tarar ve potansiyel güvenlik açıklarını tespit eder.

RCE zafiyetlerini tespit etmek için birçok teknik kullanır. Fortify, uygulamanın kodunda kullanılan girdileri analiz ederek, RCE zafiyetlerini tespit edebilir. Fortify, uygulamanın kaynak kodunda dinamik kod yürütülmesini sağlayan kod parçalarını da tespit edebilir. Ayrıca uygulamanın kaynak kodunda kullanılan harici kütüphaneleri de analiz eder ve potansiyel RCE zafiyetlerini tespit eder.

Uygulamanın kaynak kodunu tararken birçok farklı programlama dili için destek sunar. RCE zafiyetleri için özel olarak tasarlanmış birçok kural seti içerir. Bu kural setleri, uygulamanın kaynak kodunu tarayarak, potansiyel RCE zafiyetlerini tespit eder.

RCE zafiyetlerinin yanı sıra, diğer birçok güvenlik açığını da tespit edebilir. Tespit edilen güvenlik açıklarını önceliklendirir ve çözüm önerileri sunar. Bu çözüm önerileri, uygulama geliştiricilerinin hızlı bir şekilde güvenlik açıklarını gidermelerine yardımcı olur.

Fortify, uygulama geliştiricilerinin RCE zafiyetlerini tespit etmek ve önlemek için güçlü bir araçtır. Birçok farklı programlama dili için destek sunar ve potansiyel RCE zafiyetlerini tespit eder. Fortify, uygulamanın kaynak kodunu tararken, birçok farklı güvenlik açığını tespit eder ve çözüm önerileri sunar.

infinitum IT
Power of integrated Security

Remote Code Execution

infinitumitlabs