Test Integrity Checks (Bütünlük Kontrollerini Test Edin)

Summary (Özet)

Birçok uygulama, bazı girdileri gizli bırakarak durumun kullanıcısına bağlı olarak farklı alanları görüntülemek için tasarlanmıştır. Bununla birlikte, çoğu durumda, bir proxy kullanarak gizli alan değerlerini sunucuya göndermek mümkündür. Bu durumlarda sunucu tarafı kontrolleri, kullanıcı ve uygulamaya özel iş mantığına dayalı olarak uygun verilerin sunucuya izin verilmesini sağlamak için ilişkisel veya sunucu tarafı düzenlemeleri yapacak kadar akıllı olmalıdır.

Ek olarak, uygulama, iş mantığı işleme için düzenlenebilir olmayan kontrollere, açılır menülere veya gizli alanlara bağlı olmamalıdır, çünkü bu alanlar yalnızca tarayıcılar bağlamında düzenlenmez kalır.
Kullanıcılar, proxy editör araçlarını kullanarak değerlerini düzenleyebilir ve iş mantığını manipüle etmeye çalışabilirler. Uygulama, nicelik vb. gibi iş kurallarıyla ilgili değerleri açığa çıkarırsa,
kullanılabilir olmayan alanlar olarak, sunucu tarafında bir kopyasını muhafaza etmeli ve aynı şeyi iş mantığı işleme için kullanmalıdır. Son olarak, uygulama / sistem verileri bir yana, okuma, yazma ve
güncellemeyi önlemek için kayıt sistemleri güvence altına alınmalıdır.

İş mantığı bütünlüğü kontrol güvenlik açıkları, bu yanlış kullanım durumlarının uygulamaya özel olması ve kullanıcıların değişiklik yapabilmesi durumunda, yalnızca iş süreci mantığına göre belirli
zamanlarda belirli eserler yazabilmeleri veya güncelleyebilmesi veya düzenleyebilmesi gerekir.

Uygulama, ilişkisel düzenlemeleri kontrol etmek ve kullanıcıların doğrudan geçerli olmayan, güvenilir olmayan sunucuya bilgi göndermelerine izin vermeyecek kadar akıllı olmalıdır, çünkü düzenlenebilir olmayan kontrollerden veya kullanıcı ön uçtan itibaren teslim etmeye yetkili değildir. Ek olarak, kütükler gibi sistem eserleri izinsiz okuma, yazma ve kaldırmadan “korunmalıdır”.

Example 1 (Örnek 1)

Yönetici kullanıcısının yalnızca sistemdeki diğer kullanıcılar için şifreyi değiştirmesine izin veren bir ASP.NET uygulama GUI uygulaması hayal edin. Yönetici kullanıcı, kullanıcı adı ve şifreyi girmek için
kullanıcı adı ve şifre alanlarını görürken, diğer kullanıcılar her iki alanı da görmeyecek. Bununla birlikte, bir yönetici olmayan kullanıcı kullanıcı adı ve şifre alanında bir proxy aracılığıyla bilgi gönderirse, sunucuyu talebin bir yönetici kullanıcısından geldiğine ve diğer kullanıcıların şifresini değiştirdiğine inanmaya "düğmeyebilir".

Example 2 (Örnek 2)

Çoğu web uygulaması, kullanıcının durumlarını, doğum ayını vb. Hızlı bir şekilde seçmesini kolaylaştıran açılır listelere sahiptir. Bir Proje Yönetimi uygulamasının kullanıcıların giriş yapmalarına izin verdiğini varsayalım ve ayrıcalıklarına bağlı olarak onlara eriştikleri projelerin bir listesini sundu. Bir saldırgan, bir vekil aracılığıyla bilgiye erişememeleri ve göndermemeleri gereken başka bir projenin adını bulursa ne olur? Uygulama projeye erişim sağlayacak mı? Bir yetkilendirme iş mantığı kontrolünü atlamalarına rağmen erişime sahip olmamalıdırlar.

Example 3 (Örnek 3)

Motorlu taşıt idare sisteminin, bir çalışanın başlangıçta her vatandaşa kimlik veya sürücü belgesi verdiklerinde belge ve bilgileri doğrulamasını gerektirdiğini varsayalım. Bu noktada iş süreci, başvuru tarafından gönderilen verilerin bütünlüğü kontrol edildiğinden yüksek düzeyde bütünlük seviyesine sahip veriler oluşturmuştur. Şimdi, uygulamanın İnternet'e taşındığını varsayalım, böylece çalışanlar tam hizmet için oturum açabilir veya vatandaşlar belirli bilgileri güncellemek için azaltılmış bir self servis başvurusu için oturum açabilirler. Bu noktada bir saldırgan, erişmemeleri gereken verileri eklemek veya güncellemek için yakalayıcı bir vekil kullanabilir ve vatandaşın evli olmadığını, ancak bir eşin adı için veri sağladığını belirterek verilerin bütünlüğünü yok edebilirler. Doğrulanmamış verilerin bu tür bir şekilde eklenmesi veya güncellenmesi veri bütünlüğünü yok eder ve iş süreci mantığına uyulsaydı önlenebilirdi.

Example 4 (Örnek 4)

Birçok sistem, denetim ve sorun giderme amacıyla giriş yapmayı içerir. Ancak, bu günlüklerdeki bilgiler ne kadar iyi / geçerlidir? Saldırganlar tarafından ya kasıtlı

olarak ya da yanlışlıkla
bütünlüklerini yok ettikleri için manipüle edilebilirler mi?

Test Objectives (Test Hedefleri)

- Verileri hareket eden, depolayan veya işleyen sistemin bileşenleri için proje belgelerini gözden geçirin.
- Hangi tür verilerin bileşen tarafından mantıksal olarak kabul edilebilir olduğunu ve sistemin hangi türlere karşı korumanız gerektiğini belirleyin.
- Bu verileri her bir bileşende kimin değiştirmesine veya okumasına izin verilmesi gerektiğini belirleyin.
- İş mantığı iş akışına göre izin verilmemesi gereken her bir bileşen tarafından kullanılan veri değerlerini ekleme, güncelleme veya silme girişimi.

How to Test (Nasıl Test Edilir)

Specific Testing Method 1 (Özel Test Yöntemi 1)

- Gizli alanlar arayan bir vekalet yakalama HTTP trafiğini kullanmak.
- Gizli bir alan bulunursa, bu alanların GUI uygulamasıyla nasıl karşılaştırıldığını görün ve iş sürecini atlatmaya ve erişmeniz amaçlanmayan değerleri manipüle etmeye çalışan farklı veri değerleri göndererek vekalet yoluyla bu değeri sorgulamaya başlayın.

Specific Testing Method 2 (Özel Test Yöntemi 2)

- Uygulamanın düzenlenemeyen alanlarına bilgi eklemek için bir yer arayan bir vekalet yakalama HTTP trafiğini kullanarak.
- Bu alanların GUI uygulamasıyla nasıl karşılaştırıldığına bakın ve iş sürecini atlatmaya ve erişmeniz amaçlanmayan değerleri manipüle etmeye çalışan farklı veri değerleri göndererek proxy aracılığıyla bu değeri sorgulamaya başlayın.

Specific Testing Method 3 (Özel Test Yöntemi 3)

- Uygulamanın veya sistemin, örneğin günlükler veya veritabanlar olarak etkilenebilecek bileşenlerini listeleyin.

- Tanımlanan her bileşen için, bilgilerini okumaya, düzenlemeye veya kaldırmaya çalışın. Örneğin, kayıt dosyaları tanımlanmalı ve Test Cihazları toplanan verileri / bilgileri manipüle etmeye çalışmalıdır.

Related Test Cases (İlgili Test Vakaları)

Tüm Giriş Doğrulama testi vakaları.

Remediation (Düzeltme)

Uygulama, verilerin ve eserlerin nasıl değiştirilebileceği ve okunabileceği ve verilerin bütünlüğünü sağlayan güvenilir kanallar aracılığıyla sıkı erişim kontrollerini takip etmelidir. Doğru kayıt yaptırmak, yetkisiz erişim veya değişikliğin gerçekleşmediğinden emin olmak için yerinde olmalıdır.

Tools (Araçlar)

- Various system/application tools such as editors and file manipulation tools.
- OWASP Zed Attack Proxy (ZAP)
- Burp Suite

References (Referanslar)

- Implementing Referential Integrity and Shared Business Logic in a RDB
- On Rules and Integrity Constraints in Database Systems
- Use referential integrity to enforce basic business rules in Oracle
- Maximizing Business Logic Reuse with Reactive Logic
- Tamper Evidence Logging