Test Business Logic Data Validation (İş Mantığı Veri Doğrulamasını Test Edin)

Summary (Özet)

Uygulama, yalnızca mantıksal olarak geçerli verilerin ön uçta ve doğrudan bir sistem uygulamasının sunucu tarafına girebilmesini sağlamalıdır. Verileri yalnızca yerel olarak doğrulamak, uygulamaları
proxyler aracılığıyla veya diğer sistemlerle teslim alma yoluyla sunucu enjeksiyonlarına karşı savunmasız bırakabilir. Bu, daha zor olduğu ve çoğu durumda giriş noktasında doğrulanamadığı, ancak genellikle başka bir sistemi kontrol etmeyi gerektirdiği için Sınır Değer Analizi (BVA)
gerçekleştirmekten farklıdır.

Örneğin: Bir uygulama Sosyal Güvenlik Numaranızı isteyebilir. BVA'da uygulama, girilen veriler için formatları ve semantikleri (değer 9 hane uzunluğunda, negatif değil ve hepsi 0'lar değil) kontrol etmelidir, ancak mantık değerlendirmeleri de vardır. SSN'ler gruplandırılır ve kategorize edilir. Bu kişi bir ölüm dosyasında mı? Onlar ülkenin belli bir bölgesinden mi?

İş verilerinin onaylanmasıyla ilgili güvenlik açıkları, uygulamada spesifik ve sahtecilik talepleriyle ilgili güvenlik açıklarından farklı olmaları bakımından benzersizdir, çünkü iş mantığı iş akışını kırmanın aksine mantıksal veriler hakkında daha fazla endişe duyarlar.

Ön uç ve uygulamanın arka ucu, sahip olduğu, kullandığı ve geçtiği verilerin mantıksal olarak geçerli olduğunu doğrulamalı ve doğrulamalıdır. Kullanıcı bir uygulamaya geçerli veri sağlasa bile, iş mantığı, verilere veya koşullara bağlı olarak uygulamanın farklı davranmasını sağlayabilir.

Example 1 (Örnek 1)

Kullanıcıların halı sipariş etmelerini sağlayan çok katmanlı bir e-ticaret sitesini yönettiğinizi varsayalım. Kullanıcı halılarını seçer, boyuta girer, ödemeyi yapar ve ön uç uygulaması, girilen tüm bilgilerin doğru ve yasal olduğunu doğrulamıştır.

Ancak, arka plandaki iş mantığının iki yolu vardır, eğer halı stoktaysa, doğrudan deponuzdan gönderilir, ancak deponuzda stoktaysa, bir ortağın sistemine bir çağrı yapılır ve stokta bulunurlarsa, siparişi depolarından gönderir ve onlardan geri ödenir. Bir saldırgan geçerli bir borsa içi işlemi sürdürür ve ortağınıza stok dışı olarak gönderebilirse ne olur? Bir saldırgan ortasına girip ödeme yapmadan halı sipariş eden ortak depoya mesaj gönderebilirse ne olur?

Example 2 (Örnek 2)

Birçok kredi kartı sistemi artık hesap bakiyelerini her gece indiriyor, böylece müşteriler belirli bir değer altında tutarlar için daha hızlı bir şekilde kontrol edebiliyorlar. Tersi de doğrudur. Sabah kredi kartımı ödersem, akşamları mevcut krediyi kullanamayabilirim. Başka bir örnek, kredi kartımı çok hızlı bir şekilde birden fazla yerde kullanırsam, sistemlerin dün geceki verilere dayanarak kararlar almam mümkün olabilir.

Example 3 (Örnek 3)

Dağıtılmış Dolar Denkasyonu (DDo$): Bu, “The Pirate Bay” adlı web sitesinin kurucusu tarafından “The Pirate Bay” adlı hukuk bürosuna karşı önerilen bir kampanyaydı. Amaç, iş özelliklerinin tasarımındaki ve kredi transferinin geçerliliğindeki hatalardan yararlanmaktı.

Bu saldırı, hukuk firmasına 1 SEK (0.13 USD) çok az miktarda para göndererek gerçekleştirildi. Ödemelerin yönlendirildiği banka hesabında sadece 1000 ücretsiz transfer vardı, bundan sonra herhangi bir transfer hesap sahibi (2 SEK) ek ücreti var. İlk bin İnternet işleminden sonra, hukuk firmasına yapılan her 1 SEK bağışı aslında bunun yerine 1 SEK'e mal olacak.

Test Objectives (Test Hedefleri)

- Veri enjeksiyon noktalarını belirleyin.
- Tüm kontrollerin arka uçta gerçekleştiğini ve atlanamayacağını doğrulayın.
- Beklenen verilerin formatını kırmaya ve uygulamanın nasıl işlediğini analiz etme girişimi.

How To Test (Nasıl Test Edilir)

(Genel Test Yöntemi)

- Proje belgelerini gözden geçirin ve veri giriş noktalarını aramak için keşif testi kullanın veya sistemler veya yazılımlar arasındaki noktaları dağıtın.
- Bulunduktan sonra, uygulama / sisteme mantıksal olarak geçersiz kılınan verileri eklemeyi deneyin. Spesifik Test Yöntemi:
- Tek “geçerli” değerlerin kabul edilmesini sağlamak için uygulamada ön uç GUI Fonksiyonel Geçerli test yapın.
- Kesişen bir vekil kullanarak, maliyet ve kalite gibi değişkenlerin geçtiği yerleri arayan HTTP POST / GET'i gözlemleyin. Özellikle, enjeksiyon veya kurcalama noktaları olabilecek uygulama / sistemler arasında “el-off” arayın.
- Değişkenler bulunduktan sonra, alanı sosyal güvenlik numaraları veya var olmayan veya iş mantığına uymayan benzersiz tanımlayıcılar gibi mantıksal olarak “geçersiz” verilerle sorgulamaya başlar. Bu test, sunucunun düzgün çalıştığını ve mantıksal olarak geçersiz kılınan
  verileri kabul etmediğini doğrular.

Related Test Cases (İlgili Test Vakaları)

- Tüm Giriş Doğrulama testi vakaları.
- Hesap Numaralandırması ve Tahmini Kullanıcı Hesabı için Test.
- Bypass Oturum Yönetimi Şeması için Test.
- Maruz Kalan Oturum Değişkenleri için test.

Remediation (Düzeltme)

Uygulama / sistem, yalnızca “mantıksal olarak geçerli” verilerin başvurunun veya sistemin tüm giriş ve elden çıkış noktalarında kabul edilmesini ve verilerin sisteme girdikten sonra yalnızca güvenilir
olmadığından emin olmalıdır.

Tools (Araçlar)

- OWASP Zed Attack Proxy (ZAP)
- Burp Suite

References (Referanslar)

- OWASP Proactive Controls (C5) - Validate All Inputs
- OWASP Cheatsheet Series - Input_Validation_Cheat_Sheet