Testing for ORM Injection (ORM Enjeksiyonu için Test)

Summary (Özet)

Nesne İlişkisel Haritalama (ORM) Enjeksiyon, bir ORM tarafından oluşturulan veri erişimi nesne modeline karşı SQL Injection kullanan bir saldırıdır. Bir test cihazının bakış açısından, bu saldırı bir SQL Injection saldırısı ile neredeyse aynıdır. Bununla birlikte, enjeksiyon güvenlik açığı ORM tabakası tarafından oluşturulan kodda bulunur.

Bir ORM aracı kullanmanın faydaları, ilişkisel bir veritabanına iletişim kurmak için hızlı bir nesne katmanının hızlı bir şekilde üretilmesini, bu nesneler için kod şablonlarını standartlaştırmayı ve
genellikle SQL Enjeksiyon saldırılarına karşı korunmak için bir dizi güvenli işlev sağlamasını içerir. ORM tarafından oluşturulan nesneler, bir veritabanında CRUD (Yaramak, Oku, Güncelleme, Sil) işlemlerini gerçekleştirmek için SQL veya bazı durumlarda SQL'in bir varyantı kullanabilir. Bununla birlikte, ORM tarafından oluşturulan nesneleri kullanan bir web uygulamasının, yöntemlerin sağlıksız giriş parametrelerini kabul edebilmesi durumunda SQL Injection saldırılarına karşı savunmasız olması mümkündür.

How to Test (Nasıl Test Edilir)

ORM katmanları, saldırı yüzeyini uzattıkları için güvenlik açıklarına eğilimli olabilir. Uygulamayı SQL sorgularıyla doğrudan hedeflemek yerine, kötü amaçlı SQL sorguları göndermek için ORM tabakasını kötüye kullanmaya odaklanacaksınız.

Identify the ORM Layer (ORM Katmanını Tespit Edin)

İstemleriniz ve arka uç sorguları arasında neler olduğunu etkili bir şekilde test etmek ve anlamak ve uygun test yapmakla ilgili her şeyde olduğu gibi, kullanılan teknolojiyi tanımlamak esastır. Bilgi toplama bölümünü takip ederek, eldeki uygulama tarafından kullanılan teknolojinin farkında olmalısınız. Bu listeyi kendi ORM'lerine eşleme dillerini işaretleyin.

Abusing the ORM Layer (ORM Tabakasını İstismar Etmek)

Kullanılmış olası ORM'yi belirledikten sonra, ayrıştırıcısının nasıl çalıştığını anlamak ve onu kötüye kullanma yöntemlerini incelemek veya belki de uygulamanın eski bir sürüm kullanıyorsa, kullanılan
kütüphaneyle ilgili CVE'leri tanımlamak esastır. Bazen, ORM katmanları düzgün bir

şekilde uygulanmaz ve böylece test cihazının ORM katmanı hakkında endişelenmeden normal SQL Enjeksiyonu yapmasına izin verir.

Weak ORM Implementation (Zayıf ORM Uygulaması)

SANS'tan alınan ORM tabakasının düzgün bir şekilde uygulanmadığı savunmasız bir senaryo:

```
List results = session.createQuery("from Orders as orders where orders.id = " + currentOrder.getId()).list();
List results = session.createSQLQuery("Select * from Books where author = " + book.getAuthor()).list();
```

Yukarıdakiler, geliştiricinin girdiyi bir ile değiştirmesini sağlayan konumsal parametreyi uygulamadı. `?`. . Bir örnek şöyle olurdu:

```
Query hqlQuery = session.createQuery("from Orders as orders where orders.id = ?");
List results = hqlQuery.setString(0, "123-ADB-567-QTWYTFDL").list(); // 0 is the first position, where it is dynamically replaced by the string set
```

Bu uygulama, doğrulama ve sanitasyonun ORM tabakası tarafından yapılmasını sağlar ve bunu atlamanın tek yolu ORM tabakasıyla ilgili bir sorunu belirlemek olacaktır.

Vulnerable ORM Layer (Savunmasız ORM Tabakası)

ORM katmanları kod, çoğu zaman üçüncü taraf kütüphanelerdir. Diğer kod parçaları gibi savunmasız olabilirler. Bir örnek, 2019'da savunmasız olduğu tespit edilen devamı ORM npm kütüphanesi olabilir. RIPS Tech tarafından yapılan bir başka araştırmada, Java tarafından kullanılan kış uykusunda bypass tespit edildi.

Blog makalelerine dayanarak, test cihazının sorunları tanımlamasına izin verebilecek bir hile sayfası aşağıdaki şekilde özetlenebilir:

DBMS	SQL Injection
MySQL	`abc\' INTO OUTFILE --`
PostgreSQL	`$$='$$=chr(61)||chr(0x27) and 1=pg_sleep(2)||version()'`
Oracle	`NVL(TO_CHAR(DBMS_XMLGEN.getxml('select 1 where 1337>1')),'1')!='1'`
MS SQL	`1<LEN(%C2%A0(select%C2%A0top%C2%A01%C2%A0name%C2%A0from%C2%A0users)`

Another example would include the Laravel Query-Builder, which was found to be vulnerable in 2019.

References(Referanslar)

- Wikipedia - ORM
- New Methods for Exploiting ORM Injections in Java Applications (HITB16)
- HITB2016 Slides - ORM Injections in Java Applications]

- Fixing SQL Injection: ORM is not enough
- PayloadAllTheThings - HQL Injection