Testing for MS Access (MS Access için Test)

Summary (Özet)

Genel SQL enjeksiyon bölümünde açıklandığı gibi, SQL enjeksiyonu güvenlik açıkları,
kullanıcı tarafından sağlanan girdi, yeterince kısıtlanmadan veya sterilize edilmeden bir SQL sorgusunun yapımı sırasında kullanıldığında ortaya çıkar. Bu güvenlik açığı sınıfı, bir saldırganın veritabanına bağlanmak için kullanılan kullanıcının ayrıcalıkları altında SQL kodunu
yürütmesine izin verir. Bu bölümde, Microsoft Access'in belirli özelliklerini kullanan ilgili SQL enjeksiyon teknikleri tartışılacaktır.

How To Test (Nasıl Test Edilir)

Fingerprinting (Parmak izi)

SQL destekli uygulamayı test ederken belirli veritabanı teknolojisini parmak izi atmak, potansiyel güvenlik açıklarını düzgün bir şekilde değerlendirmek için ilk adımdır. Ortak bir yaklaşım, veritabanı
istisnalarını tetiklemek için standart SQL enjeksiyonu saldırı kalıplarını (örneğin tek bir alıntı, çifte alıntı, ...) enjekte etmeyi içerir. Uygulamanın istisnaları özel sayfalarla ele almadığını varsayarsak, hata mesajlarını gözlemleyerek alt satır DBMS'nin parmak izi alınması mümkündür.

Kullanılan belirli web teknolojisine bağlı olarak, MS Access odaklı uygulamalar aşağıdaki hatalardan biriyle yanıt verecektir:

`Fatal error: Uncaught exception 'com_exception' with message Source: Microsoft JET Database Engine`

ya da

`Microsoft JET Database Engine error '80040e14'`

ya da

`Microsoft Office Access Database Engine`

Her durumda, MS Access veritabanını kullanarak bir uygulamayı test ettiğimize dair bir onayımız var.

Basic Testing (Temel Test)

Ne yazık ki, MS Access, aşağıdakiler de dahil olmak üzere, SQL enjeksiyon testi sırasında geleneksel olarak kullanılan tipik operatörleri desteklemez:

- Yorum yok karakterler
- Istiflenmiş sorgu yok
- LIMIT operatörü yok
- Operatörler de UYKU VEYA GÖSTERGE
- ve diğerleri

Bununla birlikte, bu işlevleri birden fazla operatörü birleştirerek veya alternatif teknikler kullanarak taklit etmek mümkündür. Belirtildiği gibi, karakterleri eklemenin püf noktasını kullanmak mümkün
değildir
`/*`, `--` ya da `#` Sorguyu kesmek için. Ancak, neyse ki bu sınırlamayı bir “roll” karakteri enjekte ederek atlayabiliriz. Bir null byte kullanmak `%00` Bir SQL sorgusu içinde, MS Access'in kalan tüm karakterleri görmezden gelmesiyle sonuçlanır. Bu, tüm dizelerin veri tabanı tarafından
kullanılan dahili temsilde NULL olarak sonlandırıldığını göz önünde bulundurarak açıklanabilir. Bahsetmeye değer: `null` Karakter bazen web sunucu seviyesinde ipleri kesebileceği için sorunlara neden olabilir. Ancak bu durumlarda başka bir karakter istihdam edebiliriz: `0x16` (`%16` URL kodlanmış formatta).

Aşağıdaki sorguyu göz önünde bulundurarak:

```
SELECT [username],[password] FROM users WHERE [username]='$myUsername' AND [password]='$myPassword'
```

Sorguyu aşağıdaki iki URL ile kesebiliriz:

- `http://www.example.com/page.asp?user=admin'%00&pass=foo`
- `http://www.example.com/page.app?user=admin'%16&pass=foo`

The (İngilizce) `LIMIT` Operatör MS Access'te uygulanmaz, ancak sonuç sayısını kullanarak sınırlamak mümkündür. `TOP` ya da `LAST` Bunun yerine operatörler.

```
http://www.example.com/page.app?id=2'+UNION+SELECT+TOP+3+name+FROM+appsTable%00
```

Her iki operatörü de birleştirerek, belirli sonuçları seçmek mümkündür. String ile kapsanması kullanılabilir `&` `(%26)` ve `+ (%2b)` Karakterler.

SQL enjeksiyonunu test ederken kullanılabilecek başka birçok işlev de vardır, aşağıdakiler de dahil ancak bunlarla sınırlı değildir:

- ASC: Giriş olarak geçen bir karakterin ASCII değerini elde edin
- CHR: Giriş olarak geçen ASCII değerinin karakterini elde edin
- LEN: Parametre olarak geçen ipin uzunluğunu iade edin
- IIF: IF yapısıdır, örneğin aşağıdaki ifade `IIF(1=1, 'a', 'b')` Geri dönüş `a`
- MID: Bu işlev, örneğin aşağıdaki ifadeyi substring çıkarmanıza izin verir `mid('abc',1,1)` Geri dönüş `a`
- TOP: Bu işlev, sorgunun üstten geri dönmesi gereken maksimum sonuç sayısını belirtmenizi sağlar. Örneğin `TOP 1` Sadece 1 sıra geri dönecek.
- SON: Bu işlev, bir dizi sıranın sadece son sırasını seçmek için kullanılır. Örneğin aşağıdaki sorgu `SELECT last(*)` Kullanıcılardan gelen sonucun sadece son satırını döndürecek.

Bu operatörlerden bazıları kör SQL enjeksiyonlarından yararlanmak için gereklidir. Diğer gelişmiş operatörler için lütfen referanslardaki belgelere bakın.

Attributes Enumeration (Öznitelikler Numaralandırma)

Bir veritabanı tablosunun sütununu saymak için, ortak hataya dayalı bir teknik kullanmak mümkündür. Kısacası hata mesajlarını analiz ederek ve sorguyu farklı seçicilerle tekrarlayarak öznitelik adını elde edebiliriz. Örneğin, bir sütunun varlığını bildiğimizi varsayarsak, şu
sorgu ile kalan niteliklerin adını da alabiliriz:

```
' GROUP BY Id%00
```

Alınan hata mesajında bir sonraki sütunun adını gözlemlemek mümkün. Bu noktada, tüm niteliklerin adını alana kadar yöntemi yineleyebiliriz. İlk özelliğin adını bilmiyorsak, yine de hayali bir sütun adı
ekleyebilir ve hata mesajı içinde ilk özelliğin adını alabiliriz.

Obtaining Database Schema (Veritabanı Şema'nın Alınması)

MS Access'te varsayılan olarak çeşitli sistem tabloları, tablo adlarını ve sütunları elde etmek için potansiyel olarak kullanılabilecek vardır. Ne yazık ki, son MS Access veritabanı sürümlerinin varsayılan yapılandırmasında, bu tablolara erişilebilir değildir. Bununla birlikte, her zaman denemeye değer:

- MSys Objeleri
- MSysAC'lar
- MSysAccessXML'nin

Örneğin, bir sendika SQL enjeksiyonu güvenlik açığı varsa, aşağıdaki soruyu kullanabilirsiniz:

```
' UNION SELECT Name FROM MSysObjects WHERE Type = 1%00
```

Alternatif olarak, standart bir kelime listesi kullanarak veritabanı şemasını kabarmak her zaman mümkündür (örneğin. FuzzDb).

Bazı durumlarda, geliştiriciler veya sistem yöneticileri, gerçek de dahil olmak üzere bunun farkında değildir. `.mdb` Webroot uygulaması içindeki dosya, veritabanının tamamını indirmesine izin verebilir. Veri tabanı dosya adları aşağıdaki sorgu ile çıkarılabilir:

```
http://www.example.com/page.app?id=1'+UNION+SELECT+1+FROM+name.table%00
```

Nerede `name` İşte bu `.mdb` dosya adı ve `table` Geçerli bir veritabanı tablosudur. Şifre korumalı veritabanları durumunda, şifreyi kırmak için birden fazla yazılım kullanımı kullanılabilir. Lütfen referanslara bakın.

Blind SQL Injection Testing (Kör SQL Enjeksiyon Testi)

Kör SQL Enjeksiyonu güvenlik açıkları, gerçek yaşam uygulamalarını test ederken hiçbir şekilde en kolay çalıştırılabilir SQL enjeksiyonları değildir. MS Access'in son sürümlerinde, kabuk komutlarını yürütmek veya keyfi dosyaları okumak / yazmak da mümkün değildir.

Kör SQL enjeksiyonları durumunda saldırgan sadece zaman farkları veya uygulama yanıtlarını değerlendirerek sorgunun sonucunu çıkarabilir. Okuyucunun, bu bölümün kalan kısmı MS Access spesifik ayrıntılarına odaklanacağından, kör SQL enjeksiyon saldırılarının arkasındaki teoriyi zaten bildiği varsayılmaktadır.

Aşağıdaki örnek kullanılır:

```
http://www.example.com/index.php?myId=[sql]
```

Kimlik parametresinin aşağıdaki sorguda kullanıldığı yer:

```
SELECT * FROM orders WHERE [id]=$myId
```

Hadi bunu düşünelim `myId` Kör SQL enjeksiyonuna karşı savunmasız parametre. Bir saldırgan olarak, sütunun içeriğini çıkarmak istiyoruz `username` Masanın içinde `users` Veritaban şemasını zaten açıkladığımızı varsayarsak.

10. sıraların kullanıcı adının ilk karakterini çıkarmak için kullanılabilecek tipik bir sorgu şunlardır:

```
http://www.example.com/index.php?id=IIF((select%20MID(LAST(username),1,1)%20from%20(select%20TOP%2010%20username%20from%20users)='a',0,'no')
```

İlk karakter ise `a`, sorgu geri dönecek `0` ya da aksi takdirde ip `no`. .

Bir kombinasyonunu kullanarak `IFF, MID, LAST` ve `TOP` İşlevler, özel olarak seçilen bir satırda kullanıcı adının ilk karakterini çıkarmak mümkündür. İç sorgu bir dizi rekoru döndürdüğünde ve sadece bir tane değil, doğrudan kullanmak mümkün değildir. Neyse ki, belirli bir dize çıkarmak için birden fazla işlevi birleştirebiliriz.

10. sıranın kullanıcı adını almak istediğimizi varsayalım. İlk olarak, aşağıdaki sorguyu kullanarak ilk on satırı seçmek için TOP işlevini kullanabiliriz:

```
SELECT TOP 10 username FROM users
```

Ardından bu alt kümesi kullanarak, son satırı SON fonksiyonu kullanarak ayıyı ayıklayabiliriz. Sadece bir satıra ve tam olarak dizemizi içeren sıraya sahip olduktan sonra, kullanıcı adının gerçek değerini çıkarmak için IFF, MID ve LAST işlevlerini kullanabiliriz. Örneğin, bir sayı veya bir ipi iade etmek için IFF kullanıyoruz. Bu numarayı kullanarak, uygulama hatası yanıtlarını gözlemleyerek gerçek bir yanıt alıp almadığımızı ayırt edebiliriz. Olduğu gibi `id` numeriktir, bir dize ile karşılaştırma, potansiyel olarak sızdırılabilecek bir SQL hatasıyla sonuçlanır `500 Internal Server Error pages`. . Aksi takdirde, bir standart `200 OK` Sayfa muhtemelen iade edilecektir.

Örneğin, aşağıdaki sorguya sahip olabiliriz:

```
http://www.example.com/index.php?
id='%20AND%201=0%20OR%20'a'=IIF((select%20MID(LAST(username),1,1)%20from%20(select%20TOP%2010%20username%20from%20users))='a','a','b')%00
```

İlk karakter ise, aksi takdirde yanlışsa, bu doğrudur.

Belirtildiği gibi, bu yöntem, satırer dizelerin değerini veritabanında yerine getirmesini sağlar:

1. Tüm yazdırılabilir değerleri deneyerek, bir kibrit bulana kadar
2. İpin uzunluğunu kullanarak ekerek `LEN` fonksiyon, ya da sadece tüm karakterleri bulduktan sonra durarak

Zaman tabanlı kör SQL enjeksiyonları da ağır sorguları istismar ederek mümkündür.

Referance (Referanslar)

- MS erişimi SQL enjeksiyon çit sayfası
- Erişim Yoluyla Erişim - Brett Moore
- SQL Enjeksiyonuna Erişim - Brett Moore
- MS Erişim: Fonksiyonlar
- Microsoft Erişim - Wikipedia